ANSYS LS-DYNA®
KEYWORD USER'S MANUAL

VOLUME III

Multiphysics Solvers

master@893a0ffce (11/06/26)
LS-DYNA Dev

ANSYS
PART OF SYNOPSYS®

Websites

https://lsdyna.ansys.com
https://support.ansys.com

Address

ANSYS, Inc.
Southpointe
2600 Ansys Drive
Canonsburg, PA 15317
http://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

Copyright and Trademark Information

Copyright © 2026 Synopsys, Inc. and ANSYS, Inc. This Synopsys product and all associated documentation are proprietary to ANSYS, Inc. and Synopsys, Inc. and may only be used pursuant to the terms and conditions of a written license agreement. All other use, reproduction, modification or distribution of the Synopsys product or the associated documentation is strictly prohibited.

ANSYS, Ansys Workbench, AUTODYN, CFX, FLUENT and any and all ANSYS, Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of ANSYS, Inc. or its subsidiaries located in the United States or other countries. ICEM CFD is a trademark used by ANSYS, Inc. under license. CFX is a trademark of Sony Corporation in Japan. All other brand, product, service and feature names or trademarks are the property of their respective owners. FLEXlm and FLEXnet are trademarks of Flexera Software LLC.

Disclaimer Notice

THIS ANSYS SOFTWARE PRODUCT AND PROGRAM DOCUMENTATION INCLUDE TRADE SECRETS AND ARE CONFIDENTIAL AND PROPRIETARY PRODUCTS OF ANSYS, INC., ITS SUBSIDIARIES, OR LICENSORS. The software products and documentation are furnished by ANSYS, Inc., its subsidiaries, or affiliates under a software license agreement that contains provisions concerning non-disclosure, copying, length and nature of use, compliance with exporting laws, warranties, disclaimers, limitations of liability, and remedies, and other provisions. The software products and documentation may be used, disclosed, transferred, or copied only in accordance with the terms and conditions of that software license agreement.

ANSYS, Inc. and ANSYS Europe, Ltd. are UL registered ISO 9001: 2015 companies.

U.S. Government Rights

For U.S. Government users, except as specifically granted by the ANSYS, Inc. software license agreement, the use, duplication, or disclosure by the United States Government is subject to restrictions stated in the ANSYS, Inc. software license agreement and FAR 12.212 (for non-DOD licenses).

Third-Party Software

See the legal information in the product help files for the complete Legal Notice for ANSYS proprietary software and third-party software. If you are unable to access the Legal Notice, contact ANSYS, Inc.

Published in the U.S.A.

TABLE OF CONTENTS

TABLE OF CONTENTS 0-4

INTRODUCTION 1-1

*BATTERY 2-1

*BATTERY_ECHEM_AGING 2-2

*BATTERY_ECHEM_CELL_GEOMETRY 2-8

*BATTERY_ECHEM_CONTROL_INTEGRATION 2-11

*BATTERY_ECHEM_CONTROL_SOLVER 2-16

*BATTERY_ECHEM_CONTROL_THERMAL 2-27

*BATTERY_ECHEM_GAS 2-30

*BATTERY_ECHEM_INITIAL 2-34

*BATTERY_ECHEM_MAT_ANODE 2-35

*BATTERY_ECHEM_MAT_CATHODE 2-44

*BATTERY_ECHEM_MAT_ELECTROLYTE 2-52

*BATTERY_ECHEM_MAT_THERMAL 2-57

*BATTERY_ECHEM_THERMAL_ABUSE 2-59

*BATTERY_ECHEM_VENT 2-67

*BATTERY_FIT_RANDLES_PARAMETERS 2-74

*CESE 3-1

*CESE_BOUNDARY_AXISYMMETRIC 3-3

*CESE_BOUNDARY_BLAST_LOAD 3-6

*CESE_BOUNDARY_CONJ_HEAT 3-9

*CESE_BOUNDARY_CYCLIC 3-12

*CESE_BOUNDARY_FSI 3-18

*CESE_BOUNDARY_NON_REFLECTIVE 3-21

*CESE_BOUNDARY_PRESCRIBED 3-24

*CESE_BOUNDARY_PRESCRIBED_VN 3-30

*CESE_BOUNDARY_REFLECTIVE 3-35

*CESE_BOUNDARY_SLIDING 3-38

*CESE_BOUNDARY_SOLID_WALL 3-41

*CESE_CHEMISTRY_D3PLOT 3-49

*CESE_CONTROL_LIMITER 3-50

*CESE_CONTROL_MESH_MOV 3-51

*CESE_CONTROL_SOLVER 3-52
*CESE_CONTROL_TIMESTEP 3-54
*CESE_DATABASE_ELOUT 3-55
*CESE_DATABASE_FLUXAVG 3-57
*CESE_DATABASE_FSIDRAG 3-59
*CESE_DATABASE_POINTOUT 3-60
*CESE_DATABASE_SSETDRAG 3-62
*CESE_DEFINE_NONINERTIAL 3-64
*CESE_DEFINE_POINT 3-66
*CESE_DRAG 3-67
*CESE_EOS_CAV_HOMOG_EQUILIB 3-68
*CESE_EOS_IDEAL_GAS 3-69
*CESE_EOS_INFLATOR1 3-70
*CESE_EOS_INFLATOR2 3-72
*CESE_FSI_EXCLUDE 3-75
*CESE_INITIAL 3-76
*CESE_INITIAL 3-77
*CESE_INITIAL_CHEMISTRY 3-78
*CESE_INITIAL_CHEMISTRY_ELEMENT 3-80
*CESE_INITIAL_CHEMISTRY_PART 3-82
*CESE_INITIAL_CHEMISTRY_SET 3-84
*CESE_MAT_000 3-86
*CESE_MAT_001(_GAS) 3-87
*CESE_MAT_002 3-88
*CESE_PART 3-90
*CESE_SURFACE_MECHSSID_D3PLOT 3-91
*CESE_SURFACE_MECHVARS_D3PLOT 3-92

*CHEMISTRY 4-1
*CHEMISTRY_BATTERY 4-3
*CHEMISTRY_COMPOSITION 4-4
*CHEMISTRY_CONTROL_0D 4-5
*CHEMISTRY_CONTROL_1D 4-7
*CHEMISTRY_CONTROL_CSP 4-9
*CHEMISTRY_CONTROL_FULL 4-10
*CHEMISTRY_CONTROL_INFLATOR 4-11
*CHEMISTRY_CONTROL_TBX 4-14

TABLE OF CONTENTS

*CHEMISTRY_CONTROL_ZND 4-15
*CHEMISTRY_DET_INITIATION 4-17
*CHEMISTRY_INFLATOR_PROPERTIES 4-19
*CHEMISTRY_MODEL 4-24
*CHEMISTRY_PATH 4-26

*DUALCESE 5-1
*DUALCESE_BLOCKMESH 5-11
*DUALCESE_BOUNDARY_AXISYMMETRIC 5-15
*DUALCESE_BOUNDARY_CYCLIC 5-17
*DUALCESE_BOUNDARY_FSI 5-21
*DUALCESE_BOUNDARY_NON_REFLECTIVE 5-24
*DUALCESE_BOUNDARY_PRESCRIBED 5-26
*DUALCESE_BOUNDARY_PRESCRIBED_HYBRID 5-31
*DUALCESE_BOUNDARY_PRESCRIBED_INLET_MACHNUM 5-37
*DUALCESE_BOUNDARY_PRESCRIBED_INLET_MFRATE 5-41
*DUALCESE_BOUNDARY_PRESCRIBED_INLET_PRESSURE 5-45
*DUALCESE_BOUNDARY_PRESCRIBED_OUTLET_EXMFRATE 5-49
*DUALCESE_BOUNDARY_PRESCRIBED_OUTLET_FARFIELD 5-53
*DUALCESE_BOUNDARY_PRESCRIBED_OUTLET_PRESSURE 5-57
*DUALCESE_BOUNDARY_PRESCRIBED_PHASE_CHANGE 5-61
*DUALCESE_BOUNDARY_PRESCRIBED_TWO-PHASE 5-67
*DUALCESE_BOUNDARY_PRESCRIBED_VN 5-72
*DUALCESE_BOUNDARY_REFLECTIVE 5-76
*DUALCESE_BOUNDARY_SLIDING 5-78
*DUALCESE_BOUNDARY_SOLID_WALL 5-80
*DUALCESE_CHEMISTRY_D3PLOT 5-85
*DUALCESE_CONTROL_LIMITER 5-86
*DUALCESE_CONTROL_MESH_MOV 5-87
*DUALCESE_CONTROL_SOLVER 5-88
*DUALCESE_CONTROL_TIMESTEP 5-90
*DUALCESE_DARCY-FORCHHEIMER_EQ 5-91
*DUALCESE_DATABASE_HISTORY_DRAG_LIFT 5-93
*DUALCESE_DATABASE_HISTORY_ELEMENT_SET 5-96
*DUALCESE_DATABASE_HISTORY_GLOBALS 5-97
*DUALCESE_DATABASE_HISTORY_NODE_SET 5-98
*DUALCESE_DATABASE_HISTORY_POINT_SET 5-99

*DUALCESE_DATABASE_HISTORY_SEGMENT_SET 5-100
*DUALCESE_DEM_COUPLING 5-102
*DUALCESE_D3PLOT 5-108
*DUALCESE_D3PLOT_FLUID_SSID 5-110
*DUALCESE_ELE2D 5-112
*DUALCESE_ELE3D 5-113
*DUALCESE_ELEMENTSET 5-115
*DUALCESE_ENSIGHT_DOMAIN 5-116
*DUALCESE_ENSIGHT_REGION 5-122
*DUALCESE_ENSIGHT_TIMESEQ 5-129
*DUALCESE_EOS_CAV_HOMOG_EQUILIB 5-132
*DUALCESE_EOS_COCHRAN_CHAN 5-133
*DUALCESE_EOS_COOLPROP 5-135
*DUALCESE_EOS_IDEAL_GAS 5-139
*DUALCESE_EOS_INFLATOR1 5-140
*DUALCESE_EOS_INFLATOR2 5-142
*DUALCESE_EOS_JWL 5-145
*DUALCESE_EOS_NASG 5-147
*DUALCESE_EOS_REFPROP 5-149
*DUALCESE_EOS_REFPROP_PATH 5-153
*DUALCESE_EOS_SET 5-154
*DUALCESE_EOS_STIFFENED_GAS 5-155
*DUALCESE_EOS_VAN_DER_WAALS_GENERALIZED 5-156
*DUALCESE_FSI_EXCLUDE 5-158
*DUALCESE_FSI_EXCLUDE_PARTSET 5-159
*DUALCESE_GRAVITY 5-160
*DUALCESE_INCLUDE_MODEL 5-161
*DUALCESE_INITIAL 5-162
*DUALCESE_INITIAL_SET 5-163
*DUALCESE_INITIAL_REGION 5-165
*DUALCESE_INITIAL_CHEMISTRY 5-167
*DUALCESE_INITIAL_CHEMISTRY_REGION 5-169
*DUALCESE_INITIAL_CHEMISTRY_SET 5-171
*DUALCESE_INITIAL_HYBRID 5-173
*DUALCESE_INITIAL_HYBRID_SET 5-175
*DUALCESE_INITIAL_PHASE_CHANGE 5-177
*DUALCESE_INITIAL_PHASE_CHANGE_SET 5-179

TABLE OF CONTENTS

*DUALCESE_INITIAL_TWO-PHASE 5-181
*DUALCESE_INITIAL_TWO-PHASE_SET 5-183
*DUALCESE_MAT_GAS 5-185
*DUALCESE_MAT_GAS_0 5-187
*DUALCESE_MAT_GAS_2 5-188
*DUALCESE_MAT_LIQUID 5-190
*DUALCESE_MESH_GEOMETRY 5-191
*DUALCESE_MESH_PART 5-193
*DUALCESE_MODEL 5-195
*DUALCESE_NODE2D 5-196
*DUALCESE_NODE3D 5-197
*DUALCESE_NODESET 5-198
*DUALCESE_PART 5-199
*DUALCESE_PART_MULTIPHASE 5-201
*DUALCESE_POINTSET 5-203
*DUALCESE_POINT_SOURCE 5-204
*DUALCESE_POINT_SOURCE_TWO-PHASE 5-207
*DUALCESE_POINT_SOURCE_STRUCTNODE_SET 5-211
*DUALCESE_POROUS_MODEL 5-212
*DUALCESE_POROUS_SPECIFY_JUMP 5-213
*DUALCESE_REACTION_RATE_IG 5-214
*DUALCESE_REACTION_RATE_IG_REDUCED 5-216
*DUALCESE_REACTION_RATE_P_DEPEND 5-217
*DUALCESE_SEGMENTSET 5-218
*DUALCESE_SOLVER_SELECTION 5-220
*DUALCESE_SOLVER_CAV_EQNS 5-222
*DUALCESE_SOLVER_CHEMICALLY_REACTING 5-223
*DUALCESE_SOLVER_EULER_EQNS 5-224
*DUALCESE_SOLVER_HYBRID_MULTIPHASE 5-225
*DUALCESE_SOLVER_NAVIER_STOKES 5-227
*DUALCESE_SOLVER_PHASE_CHANGE 5-228
*DUALCESE_SOLVER_TWO-PHASE_MULTIPHASE 5-229

*EM 6-1
*EM_2DAXI 6-5
*EM_BOUNDARY 6-6
*EM_BOUNDARY_PRESCRIBED 6-7

*EM_CIRCUIT 6-9
*EM_CIRCUIT_CONNECT 6-14
*EM_CIRCUIT_ROGO 6-15
*EM_CIRCUIT_SOURCE 6-17
*EM_CONTACT 6-21
*EM_CONTACT_RESISTANCE 6-23
*EM_CONTACT_SUBDOM 6-26
*EM_CONTROL 6-28
*EM_CONTROL_CONTACT 6-33
*EM_CONTROL_COUPLING 6-35
*EM_CONTROL_COUPLING_P2D 6-39
*EM_CONTROL_EP 6-44
*EM_CONTROL_EROSION 6-46
*EM_CONTROL_MAGNET 6-47
*EM_CONTROL_SOLUTION 6-48
*EM_CONTROL_SWITCH 6-51
*EM_CONTROL_SWITCH_CONTACT 6-52
*EM_CONTROL_TIMESTEP 6-53
*EM_DATABASE_CIRCUIT 6-56
*EM_DATABASE_CIRCUIT0D 6-58
*EM_DATABASE_CIRCUITSOURCE 6-59
*EM_DATABASE_ELOUT 6-60
*EM_DATABASE_FIELDLINE 6-61
*EM_DATABASE_GLOBALENERGY 6-64
*EM_DATABASE_NODOUT 6-65
*EM_DATABASE_PARTDATA 6-66
*EM_DATABASE_POINTOUT 6-67
*EM_DATABASE_ROGO 6-68
*EM_DATABASE_TIMESTEP 6-69
*EM_EOS_BURGESS 6-70
*EM_EOS_MEADON 6-74
*EM_EOS_PERMEABILITY 6-77
*EM_EOS_TABULATED1 6-81
*EM_EOS_TABULATED2 6-82
*EM_EP_CELLMODEL_DEFINE_FUNCTION 6-84
*EM_EP_CELLMODEL_FENTONKARMA 6-87
*EM_EP_CELLMODEL_FITZHUGHNAGUMO 6-90

TABLE OF CONTENTS

*EM_EP_CELLMODEL_STEP 6-92
*EM_EP_CELLMODEL_TENTUSSCHER 6-93
*EM_EP_CELLMODEL_TOMEK 6-102
*EM_EP_CELLMODEL_TOR_ORD 6-103
*EM_EP_CELLMODEL_TROVATO 6-104
*EM_EP_CELLMODEL_USERMAT 6-105
*EM_EP_CREATEFIBERORIENTATION 6-106
*EM_EP_ECG 6-108
*EM_EP_EIKONAL 6-109
*EM_EP_ISOCH 6-111
*EM_EP_LAPLACE_DIRICHLET 6-112
*EM_EP_PURKINJE_NETWORK 6-113
*EM_EP_TENTUSSCHER_STIMULUS 6-115
*EM_EP_TENTUSSCHER_STIMULUS2 6-116
*EM_EXTERNAL_CHARGE 6-117
*EM_EXTERNAL_ECVEL 6-118
*EM_EXTERNAL_FIELD 6-120
*EM_ISOPOTENTIAL 6-123
*EM_ISOPOTENTIAL_CONNECT 6-124
*EM_ISOPOTENTIAL_ROGO 6-130
*EM_MAT_001 6-131
*EM_MAT_002 6-135
*EM_MAT_003 6-139
*EM_MAT_004 6-143
*EM_MAT_005 6-146
*EM_MAT_006 6-151
*EM_MAT_007 6-153
*EM_OUTPUT 6-157
*EM_OUTPUT_BINARY 6-159
*EM_OUTPUT_FORCES 6-161
*EM_OUTPUT_VTK 6-163
*EM_P2D_BATMAC 6-164
*EM_P2D_MESHLESS 6-174
*EM_PERMANENT_MAGNET 6-179
*EM_POINT_SET 6-184
*EM_RANDLES 6-186
    *EM_RANDLES_BATMAC 6-189

*EM_RANDLES_EXOTHERMIC_REACTION 6-196
*EM_RANDLES_MESHLESS 6-197
*EM_RANDLES_TSHELL 6-203
*EM_RANDLES_SHORT 6-210
*EM_RANDLES_SOLID 6-213
*EM_ROTATION_AXIS 6-220
*EM_SOLVER_BEM 6-222
*EM_SOLVER_BEM_REASSEMBLY 6-224
*EM_SOLVER_BEMMAT 6-226
*EM_SOLVER_FEM 6-227
*EM_SOLVER_FEMBEM 6-229
*EM_SOLVER_FEMBEM_MONOLITHIC 6-230
*ICFD 7-1
*ICFD_BOUNDARY_CONJ_HEAT 7-6
*ICFD_BOUNDARY_CONVECTION_TEMP 7-7
*ICFD_BOUNDARY_FLUX_SPECIES 7-8
*ICFD_BOUNDARY_FLUX_TEMP 7-9
*ICFD_BOUNDARY_FREESLIP 7-10
*ICFD_BOUNDARY_FSI 7-11
*ICFD_BOUNDARY_FSI_EXCLUDE 7-12
*ICFD_BOUNDARY_FSI_FIXED 7-13
*ICFD_BOUNDARY_FSI_ONEWAY 7-14
*ICFD_BOUNDARY_FSWAVE 7-15
*ICFD_BOUNDARY_GROUND 7-18
*ICFD_BOUNDARY_IMMERSED_DYNAMIC 7-19
*ICFD_BOUNDARY_IMMERSED_STATIC 7-20
*ICFD_BOUNDARY_ISENTROPIC_NRBC 7-21
*ICFD_BOUNDARY_ISENTROPIC_NRBC0 7-24
*ICFD_BOUNDARY_NAVIERSLIP 7-26
*ICFD_BOUNDARY_NONSLIP 7-28
*ICFD_BOUNDARY_PERIODIC 7-29
*ICFD_BOUNDARY_PRESCRIBED_LEVELSET 7-31
*ICFD_BOUNDARY_PRESCRIBED_MOVEMESH 7-32
*ICFD_BOUNDARY_PRESCRIBED_PRE 7-34
*ICFD_BOUNDARY_PRESCRIBED_SPTRANSP_CONC 7-36
*ICFD_BOUNDARY_PRESCRIBED_TEMP 7-37

TABLE OF CONTENTS

*ICFD_BOUNDARY_PRESCRIBED_TURBULENCE 7-38
*ICFD_BOUNDARY_PRESCRIBED_VEL 7-41
*ICFD_BOUNDARY_PRESCRIBED_VISCOELASTIC 7-43
*ICFD_BOUNDARY_PRESCRIBED_WATERVAPOR 7-44
*ICFD_BOUNDARY_WEAKVEL 7-45
*ICFD_BOUNDARY_WINDKESSEL 7-46
*ICFD_CONTROL_ADAPT 7-50
*ICFD_CONTROL_ADAPT_SIZE 7-52
*ICFD_CONTROL_ADVECTION 7-54
*ICFD_CONTROL_BACKFLOW 7-56
*ICFD_CONTROL_CONJ 7-57
*ICFD_CONTROL_DEM_COUPLING 7-60
*ICFD_CONTROL_EMBEDSHELL 7-62
*ICFD_CONTROL_FSI 7-63
*ICFD_CONTROL_GAP 7-66
*ICFD_CONTROL_GENERAL 7-68
*ICFD_CONTROL_IMMERSED 7-70
*ICFD_CONTROL_IMMERSED_FSI 7-72
*ICFD_CONTROL_IMPOSED_MOVE 7-73
*ICFD_CONTROL_ISENTROPIC 7-78
*ICFD_CONTROL_LEVELSET 7-80
*ICFD_CONTROL_LOAD 7-82
*ICFD_CONTROL_MESH 7-84
*ICFD_CONTROL_MESH_MOV 7-86
*ICFD_CONTROL_MONOLITHIC 7-87
*ICFD_CONTROL_OUTPUT 7-88
*ICFD_CONTROL_OUTPUT_SUBDOM 7-91
*ICFD_CONTROL_OUTPUT_VAR 7-93
*ICFD_CONTROL_PARTITION 7-95
*ICFD_CONTROL_POROUS 7-96
*ICFD_CONTROL_RANS 7-97
*ICFD_CONTROL_STEADY 7-103
*ICFD_CONTROL_SURFMESH 7-104
*ICFD_CONTROL_TAVERAGE 7-105
*ICFD_CONTROL_TIME 7-106
*ICFD_CONTROL_TRANSIENT 7-111
*ICFD_CONTROL_TURBULENCE 7-112

*ICFD_CONTROL_TURB_SYNTHESIS....7-122
*ICFD_DATABASE_AVERAGE....7-123
*ICFD_DATABASE_BIN....7-124
*ICFD_DATABASE_DRAG....7-126
*ICFD_DATABASE_FLUX....7-129
*ICFD_DATABASE_FLUX_SURF....7-130
*ICFD_DATABASE_FORCE_DEM....7-131
*ICFD_DATABASE_GOA....7-132
*ICFD_DATABASE_HTC....7-133
*ICFD_DATABASE_NODEAVG....7-135
*ICFD_DATABASE_NODOUT....7-136
*ICFD_DATABASE_NTEMPOUT....7-137
*ICFD_DATABASE_POINTAVG....7-138
*ICFD_DATABASE_POINTOUT....7-139
*ICFD_DATABASE_RESIDUALS....7-141
*ICFD_DATABASE_SSOUT....7-142
*ICFD_DATABASE_SSOUT_EXCLUDE....7-143
*ICFD_DATABASE_TEMP....7-144
*ICFD_DATABASE_TIMESTEP....7-145
*ICFD_DATABASE_TPD....7-146
*ICFD_DATABASE_TWINBUILDER....7-148
*ICFD_DATABASE_UINDEX....7-150
*ICFD_DATABASE_VOL....7-151
*ICFD_DATABASE_WETNESS....7-152
*ICFD_DEFINE_GAPCLOSURE....7-153
*ICFD_DEFINE_GAPRESTRIC....7-156
*ICFD_DEFINE_HEATSOURCE....7-157
*ICFD_DEFINE_HEMOLYSIS_INDEX....7-159
*ICFD_DEFINE_POROUS_REGION....7-160
*ICFD_DEFINE_RESIDENCETIMESOURCE....7-163
*ICFD_DEFINE_SOURCE....7-166
*ICFD_DEFINE_SPTRANSPSOURCE....7-167
*ICFD_DEFINE_TRANSFORM....7-168
*ICFD_DEFINE_TURBSOURCE....7-169
*ICFD_DEFINE_POINT....7-171
*ICFD_DEFINE_NONINERTIAL....7-173
*ICFD_DEFINE_WAVE_DAMPING....7-175

TABLE OF CONTENTS

*ICFD_INITIAL 7-177
*ICFD_INITIAL_BIN 7-178
*ICFD_INITIAL_LEVELSET 7-180
*ICFD_INITIAL_SPTRANSP 7-182
*ICFD_INITIAL_TEMPNODE 7-183
*ICFD_INITIAL_TURBULENCE 7-184
*ICFD_INITIAL_VISCOELASTIC 7-185
*ICFD_INITIAL_WATERVAPOR 7-186
*ICFD_MAT 7-187
*ICFD_MODEL_NONNEWT 7-191
*ICFD_MODEL_POROUS 7-195
*ICFD_MODEL_SPECIES_TRANSPORT 7-204
*ICFD_MODEL_VISCOELASTIC 7-207
*ICFD_PART 7-208
*ICFD_PART_VOL 7-209
*ICFD_SECTION 7-211
*ICFD_SET_NODE_LIST 7-212
*ICFD_SOLVER_SPLIT 7-213
*ICFD_SOLVER_TOL_FSI 7-214
*ICFD_SOLVER_TOL_LSET 7-215
*ICFD_SOLVER_TOL_MMOV 7-216
*ICFD_SOLVER_TOL_MOM 7-217
*ICFD_SOLVER_TOL_MONOLITHIC 7-218
*ICFD_SOLVER_TOL_PRE 7-219
*ICFD_SOLVER_TOL_TEMP 7-220

*MESH 8-1
*MESH_BL 8-2
*MESH_BL_SYM 8-6
*MESH_EMBEDSHELL 8-7
*MESH_INTERF 8-8
*MESH_NODE 8-9
*MESH_SIZE 8-10
*MESH_SIZE_SHAPE 8-11
*MESH_SURFACE_ELEMENT 8-17
*MESH_SURFACE_NODE 8-19
*MESH_SURFACE_NULL 8-20

*MESH_VOLUME 8-21
*MESH_VOLUME_ELEMENT 8-22
*MESH_VOLUME_NODE 8-23
*MESH_VOLUME_PART 8-24

*STOCHASTIC 9-1
*STOCHASTIC_SPRAY_PARTICLES 9-2
*STOCHASTIC_TBX_PARTICLES 9-7

*LSO 10-1
*LSO_DOMAIN 10-2
*LSO_ID_SET 10-5
*LSO_POINT_SET 10-7
*LSO_TIME_SEQUENCE 10-9
*LSO_VARIABLE_GROUP 10-12

LS-DYNA MULTIPHYSICS USER'S MANUAL

INTRODUCTION

In this manual, there are five main solvers: two compressible flow solvers, an incompressible flow solver, an electromagnetism solver, and a battery electrochemistry solver. Each of them implements coupling with the structural solver in LS-DYNA.

The keywords covered in this manual fit into one of three categories. In the first category are the keyword cards that provide input to each of the multiphysics solvers that in turn couple with the structural solver. In the second category are keyword cards involving extensions to the basic solvers. Presently, the chemistry and stochastic particle solvers are the two solvers in this category, and they are used in conjunction with the *CESE compressible flow solver discussed below. In the third category are keyword cards for support facilities. A volume mesher that creates volume tetrahedral element meshes from bounding surface meshes is one of these tools. Another is a data output mechanism for a limited set of variables from some of the solvers in this manual. This mechanism is accessed through *LSO keyword cards.

The CESE solver is a compressible flow solver based upon the Conservation Element/Solution Element (CE/SE) method, originally proposed by Chang of the NASA Glenn Research Center. This method is a novel numerical framework for conservation laws. It has many non-traditional features, including a unified treatment of space and time, the introduction of separate conservation elements (CE) and solution elements (SE), and a novel shock capturing strategy without using a Riemann solver. This method has been used to solve many types of flow problems, such as detonation waves, shock/acoustic wave interaction, cavitating flows, supersonic liquid jets, and chemically reacting flows. In LS-DYNA, it has been extended to also solve fluid-structure interaction (FSI) problems. It does this with two approaches. The first approach solves the compressible flow equations on an Eulerian mesh while the structural mechanics is solved on a moving mesh that moves through the fixed CE/SE mesh. In the second approach (new with this version), the CE/SE mesh moves in a fashion such that its FSI boundary surface matches the corresponding FSI boundary surface of the moving structural mechanics mesh. This second approach is more accurate for FSI problems, especially with boundary layers flows. Another feature with the CESE moving mesh solver is conjugate heat transfer coupling with the solid thermal solver. The chemistry and stochastic particle solvers are two addon solvers that extend the CESE solver.

The dual CESE solver is another compressible flow solver that is also based upon the Conservation Element/Solution Element (CE/SE) method, but with improvements related to accuracy and robustness. This method follows a similar novel numerical

framework for conservation laws. In LS-DYNA, the dual CESE solver also include fluid-structure interaction (FSI) capabilities. It also does this with two approaches. The first approach solves the compressible flow equations on an Eulerian mesh while the structural mechanics is solved on a moving mesh that moves through the fixed dual CE/SE mesh. In the second approach. the dual CE/SE mesh moves in a fashion such that its FSI boundary surface matches the corresponding FSI boundary surface of the moving structural mechanics mesh. This second approach is more accurate for FSI problems, especially with boundary layers flows. One of the advances in these FSI computations with the dual CESE solver as compared with the older CESE solver is that each FSI approach (or fixed mesh Eulerian solver) may be employed in different subregions of the fluid mesh in the same problem. Unlike the *CESE solvers, the dual CESE solvers do not have conjugate heat transfer coupling with the solid thermal solver, nor coupling with the chemistry or stochastic particle solvers. It is not clear if these *CESE capabilities will be added to dual CESE solver. On the other hand, another advance capability available only with the dual CESE solvers is the availability of equations of state for pure and pseudo-pure fluids of industrial interest in the REFPROP and COOLPROP EOS libraries. These complex EOSes are generally expensive to evaluate, so a bi-cubic table look-up mechanism has been developed that greatly accelerates their use.

The third solver is the implicit computational flow solver (ICFD). This solver is fully coupled with the solid mechanics solver. This coupling permits robust FSI analysis via either an explicit technique when the FSI is weak, or using an implicit coupling when the FSI coupling is strong. In addition to being able to handle free surface flows, there is also a bi-phasic flow capability that involves modeling using a conservative Lagrangian interface tracking technique. Basic turbulence models are also supported. This solver uses a volume mesher that takes surface meshes bounding the fluid domain as input (*MESH keywords). In addition, during the time advancement of the flow, the solution is adaptively re-meshed as an automatic feature of the solver. Another important feature of the mesher is the ability to create boundary layer meshes. These anisotropic meshes become a crucial part of the model when shear stresses are to be calculated near fluid walls. The ICFD solver is also coupled to the solid thermal solver using a monolithic approach for conjugate heat transfer problems.

The fourth solver is an electromagnetics (EM) solver. This module solves the Maxwell equations in the Eddy current (induction-diffusion) approximation. This is suitable for cases where the propagation of electromagnetic waves in air (or vacuum) can be considered as instantaneous. Therefore, the wave propagation is not solved. The main applications are Magnetic Metal Forming, bending or welding, induced heating, ring expansions and so forth. The EM module allows the introduction of a source of electrical current into solid conductors and the computation of the associated magnetic field, electric field, as well as induced currents. The EM solver is coupled with the structural mechanics solver (the Lorentz forces are added to the mechanics equations of motion), and with the structural thermal solver (the ohmic heating is added to the thermal solver as an extra source of heat). The EM fields are solved using a Finite Element Method (FEM) for

the conductors and a Boundary Element Method (BEM) for the surrounding air/insulators. Thus no air mesh is necessary.

The fifth solver is a battery electrochemistry solver. At this time, the available capability involves a one-dimensional electrochemistry solver that is coupled to the structural mechanics and structural thermal solver in each structural element identified as being part of a battery cell. It solves these one-dimensional models implicitly so as to be able to run simulations for very long physical times that are typical of battery-structure interaction problems.

As stated above, the *CHEMISTRY and *STOCHASTIC cards are only used in the CESE solver at this time.

*BATTERY

The keyword *BATTERY provides input data for the electrochemistry solver:

*BATTERY_ECHEM_AGING

*BATTERY_ECHEM_CELL_GEOMETRY

*BATTERY_ECHEM_CONTROL_INTEGRATION

*BATTERY_ECHEM_CONTROL_SOLVER

*BATTERY_ECHEM_CONTROL_THERMAL

*BATTERY_ECHEM_GAS

*BATTERY_ECHEM_INITIAL

*BATTERY_ECHEM_MAT_ANODE

*BATTERY_ECHEM_MAT_CATHODE

*BATTERY_ECHEM_MAT_ELECTROLYTE

*BATTERY_ECHEM_THERMAL_ABUSE

*BATTERY_ECHEM_VENT

*BATTERY_FIT_RANDLES_PARAMETERS

*BATTERY_ECHEM_AGING

Purpose: Set the battery aging parameters that are used in the 10- or 15-equation P2D model. See *BATTERY_ECHEM_CONTROL_SOLVER. Aging only applies to the anode.

Card Summary:

Card 1. This card is required.

SEIMW	SEIRHO	SEIBRUG	SEIEPS	SEICO	SEITO	SEIP	

Card 2. This card is required.

SEII0	SEIRKA	SEICON	ECCO	DEC	ISTYPE	USEI	

Card 3. This card is optional. It must be included if Card 4 is included. See Remark 2.

AFI	EAT						

Card 4. This card is optional. See Remark 2.

HEC		HLED	HC2H4	HLI2CO3	HCO2		

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	SEIMW	SEIRHO	SEIBRUG	SEIEPS	SEICO	SEITO	SEIP	
Type	F	F	F	F	F	F	F	
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

VARIABLE	DESCRIPTION
SEIMW	Molecular weight of the solid electrolyte interphase or SEI (units: kg/mol)
SEIRHO	Density of the SEI (units: kg/m^3)
SEIBRUG	Symmetry coefficient, β, of the SEI
SEIEPS	Initial SEI porosity

VARIABLE	DESCRIPTION
SEIC0	Initial SEI concentration (units: mol/m^3)
SEIT0	Initial thickness of the SEI layer (units: m), δ_0
SEIP	SEI solid fraction exponent, p_{SEI}

Card 2	1	2	3	4	5	6	7	8
Variable	SEII0	SEIRKA	SEICON	ECC0	DEC	ISTYPE	USEI	
Type	F	F	F	F	F	I	F	
Default	0.0	0.0	0.0	0.0	0.0	0	0.0	

VARIABLE	DESCRIPTION
SEII0	Exchange current density, $i_{0,s}$ for the SEI reaction (units: A/m^2)
SEIRKA	Reaction rate constant for the SEI reaction (ignored if ISTYPE = 0)
SEICON	Ionic conductivity (units: S/m)
ECC0	Initial concentration of ethylene carbonate or EC, $c_{EC,0}$ (units: mol/m^3). This field is ignored if ISTYPE = 0.
DEC	Diffusion coefficient of EC (units: m^2/s)
ISTYPE	Formulation variation for SEI side reaction kinetics: EQ.0: SEII0 is used for the SEI side reaction. EQ.1: SEIRKA and ECCO are used for the SEI side reaction. This allows a feedback mechanism to occur if DEC is also defined.
USEI	Equilibrium (open-circuit) potential of the solvent reaction side reaction (units: V)

This card is optional. See Remark 2.

Card 3	1	2	3	4	5	6	7	8
Variable	AFI	EAT						
Type	F	F						
Default	0.0	0.0						

VARIABLE	DESCRIPTION
AFI	Frequency factor for the reaction, A_f (units: s^{-1})
EAT	Activation energy for the reaction, E_a (units: K)

This card is optional. See Remark 2.

Card 4	1	2	3	4	5	6	7	8
Variable	HEC		HLED	HC2H4	HLI2CO3	HCO2		
Type	F		F	F	F	F		
Default	0.0		0.0	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
HEC	Formation enthalpy of the EC, ΔH_{EC} (units: kJ/mol)
HLED	Formation enthalpy of the SEI layer, ΔH_{LED} (units: kJ/mol)
HC2H4	Formation enthalpy of ethylene, ΔH_{C2H4} (units: kJ/mol)
HLI2CO3	Formation enthalpy of lithium carbonate, ΔH_{LC} (units: kJ/mol)
HCO2	Formation enthalpy of CO_2, ΔH_{CO2} (units: kJ/mol)

Remarks:

1. **Aging model.** This aging model adds four equations to the P2D model. These equations only apply to nodes in the anode because the voltage of the anode is

in the range of where these aging effects can occur. The four unknowns solved for in these equations are:

j_s SEI side reaction flux (units: mol/(m^{2}s))

δ SEI layer thickness (units: m)

c_{SEI} SEI species concentration (units: mol/m^3)

$\varepsilon_{e,a}$ Electrolyte volume fraction adjusted due to changes in porosity

The first unknown, j_s, is determined from the SEI side reaction. The SEI side reaction is modeled as an irreversible Tafel-type process on the anode. We have implemente two models with ISTYPE controlling the model:

a) *ISTYPE = 0.* With this model, j_s is given by:

$$j_s = -i_{0,s} e^{-[\beta f_{RT}(\varphi_s - \varphi_e - U_{SEI} R_{SEI} i_{int})]}$$

where $i_{0,s}$ is the exchange current density (SEII0) for the side reaction, β is the symmetry (transfer) coefficient (SEIBRUG), f_{RT} is the derived Faraday ratio, U_{SEI} is the equilibrium (open-circuit) potential of the solvent reaction side rection (USEI), R_{SEI} is the initial SEI film resistence, φ_s is the solid phase potential, φ_e is the electrolyte potential, and i_{int} is the interfacial current density. $f_{RT} = F/R_g T$, where F is Faraday's constant, R_g is the universal gas constant, and T is the temperature. $R_{SEI} = \delta_0 / \kappa_{SEI}^{eff}$, where δ_0 is the initial SEI layer thickness (SEIT0), $\kappa_{SEI}^{eff} = \kappa_{SEI}(1. - \varepsilon_{SEI})^{p_{SEI}}$, κ_{SEI} is the conductivity of the solid SEI matrix (SEICON), ε_{SEI} is the SEI porosity, and p_{SEI} is the SEI solid fraction exponent (SEIP). $i_{int} = F(j_s + j_n)$, where j_n is the pore wall molar flux.

b) *ISTYPE = 1.* With this model, j_s is given by:

$$j_s = -k_{a,s} c_{EC,surf} e^{-[\beta f_{RT}(\varphi_s - \varphi_e - R_{SEI} i_{int})]}$$

where $k_{a,s}$ is the side reaction rate constant (SIERKA) and $c_{EC,surf}$ is the EC surface concentration. If DEC (D_{EC}), the EC diffusion coefficient, is not specified or zero, $c_{EC,surf}$ is set to $c_{EC,0}$, the initial EC concentration. If DEC is nonzero, a quasi-steady-state approximation is applied to account for the depletion of EC at the particle surface due to diffusion through through the growing SEI layer. Under this approximation, the EC concentration profile across the SEI film is linear, and the EC surface concentration is obtained algebraically:

$$c_{EC,surf} = \frac{c_{EC,0}}{1 + \dfrac{\delta k_{a,s} e^{-[\beta f_{RT}(\phi_s - \phi_e - U_{SEI} - R_{SEI} i_{int})]}}{D_{EC}(1 - \varepsilon_{SEI})^{p_{SEI}}}}$$

This model introduces a coupling between j_s and δ, capturing the physical feedback: as the SEI grows, EC must diffuse through a thicker film, reduciong $c_{EC,surf}$ and slowing the side reaction.

The SEI thickness. δ, grows through the side-reaction flux:

$$\frac{\partial \delta}{\partial t} = -\frac{M_{SEI}}{2\rho_{SEI}} j_s \ .$$

Here, M_{SEI} is the SEI molecular weight (SEIMW) and ρ_{SEI} is the SEI density (SEIRHO).

The SEI concentration, c_{SEI}, evolves according to:

$$\frac{\partial c_{SEI}}{\partial t} = -\frac{a_j}{2} j_s$$

where a_j is the specific interfacial area.

For the anode, the electrolyte volume fraction evolves due to volumetric changes associated with intercalation and the SEI side reaction:

$$\frac{\partial \varepsilon_{e,a}}{\partial t} = -\frac{M_{w,a}}{\rho_{s,a}} a_j (j_n + j_s) \ .$$

Here, $M_{w,a}$ is the partial molar volume of the anode (MWA on *BATTERY_ECHEM_MAT_ANODE), $\rho_{s,a}$ is the active material density (RHOEA on *BATTERY_ECHEM_MAT_ANODE), and j_n is is the pore wall molar flux. Note that by default the solve for the volume fraction serves as a diagnostic quantity and has no feedback on any other variable, including $\varepsilon_{e,a}$ remaining its initial value in other equations. Not including feedback is valid for short term or single discharges. For long-term aging simulations, feedback should be included and is enabled by setting IPORO to 1 on *BATTERY_ECHEM_CONTROL_SOLVER.

2. **Thermal effects.** Cards 3 and 4 only need to be defined if the aging heat flag (ITRA = 1) is enabled on *BATTERY_ECHEM_CONTROL_SOLVER. Enabling the aging heat flag causes the computation of two additional heat source terms beyond the six-equation models irreversible and reversible. These terms are the SEI formation heat and SEI decomposition heat. These terms are added to the total irreversible heat.

The SEI formation heat is given by:

$$\dot{q}_{SEI,form} = \int_0^{L_a} j_s \left(\Delta H_{net} + F(\varphi_s - \varphi_e) \right) dx$$

where L_a is the length of the anode, $\Delta H_{net} = 0.5(H_{EC} + H_{LI} - H_{LED} - H_{C2H4})$, F is Faraday's constant, φ_s is the solid phase potential, and φ_e is the electrolyte potential.

The SEI decompositon heat is given by:

$$\dot{q}_{\mathrm{SEI,decomp}} = \Delta H_{\mathrm{decomp}} A_{\mathrm{f}} \int_0^{L_{\mathrm{a}}} c_{\mathrm{SEI}} \mathrm{d}x \; e^{-E_a/RT}$$

where A_{f} is the frequency factor, E_a is the activation energy, $\Delta H_{\mathrm{decomp}} = H_{\mathrm{LED}} - H_{\mathrm{LC}} - H_{\mathrm{C2H4}} - H_{\mathrm{CO2}}$, R is the universal gas constant, T is temperature, and c_{SEI} is the SEI concentration (see Remark 1).

*BATTERY_ECHEM_CELL_GEOMETRY

Purpose: Set general-purpose geometry variables for a single cell battery model.

Card 1	1	2	3	4	5	6	7	8
Variable	CELLID	ALEN	SLEN	CLEN	CNLEN	CPLEN		
Type	I	F	F	F	F	F		
Default	none	none	none	none	none	none		

Card 2	1	2	3	4	5	6	7	8
Variable	NA	NS	NC	NRJ	QRAD			
Type	I	I	I	I	F			
Default	none	none	none	30	1.0			

VARIABLE	DESCRIPTION
CELLID	Optional cell ID. See Remark 6 of *BATTERY_ECHEM_CONTROL_SOLVER.
ALEN	Length of the negative electrode (anode), L_a (units: m). See Remark 3.
SLEN	Length of the separator, L_s (units: m). See Remark 3.
CLEN	Length of the positive electrode (cathode), L_c (units: m). See Remark 3.
CNLEN	Length of the negative current collector, L_{CC^-} (units: m)
CPLEN	Length of the positive current collector, L_{CC^+} (units: m)
NA	Number of elements in the anode, n_a
NS	Number of nodes in the separator, n_s. This includes the the two shared interface nodes.

VARIABLE	DESCRIPTION
NC	Number of elements in the cathode, n_c
NRJ	Number of radial nodes, n_{rj}, in the particle used when solving the solid-phase diffusion equation. This number excludes the center node. See Remark 2 and Figure 2-1.
QRAD	Radial mesh geometric grading, q_{rad}. See Remark 2 and Figure 2-1.

Remarks:

1. **Unit cell geometry and discretization.** The unit cell is a one-dimensional spatial domain that spans through three regions along the local through-thickness coordinate, x. The anode is the region from $0.0 \le x \le L_a$. It is discretized into n_a intervals with $n_a + 1$ nodes ($j = 1, \dots, n_a + 1$ where j is the node number). The separator is the region from $L_a < x < L_a + L_s$ that is discretized into $n_s - 1$ intervals with n_s nodes ($j = n_a + 1, \dots, n_a + n_s$). The cathode is the region from $L_a + L_s \le x \le L_a + L_s + L_c$ that is discretized into n_c intervals with $n_c + 1$ nodes ($j = n_a + n_s, \dots, n_a + n_s + n_c$). The total number of 1D mesh nodes is $n_{tot} = n_a + n_s + n_c$. Grid spacings are uniform within each region. Hence, $\Delta x_a = L_a/n_a$, $\Delta x_s = L_s/(n_s - 1)$, and $\Delta x_c = L_c/n_c$.

   At each node in the anode and cathode, a representative spherical electrode particle is also assumed to exist. The solid phase concentration equation is solved radially for each particle on each of those nodes. Remark 2 discusses the radial mesh for each particle.

   Note that the total length of the unit cell also includes the negative and positive current collectors, even though they are not part of the one-dimensional discretization:

$$L_{cell} = L_a + L_s + L_c + L_{CC^-} + L_{CC^+}$$

2. **Radial mesh geometric grading.** When $0. < q_{rad} \le 1.$, the radial mesh inside the spherical particle uses a geometric progression $\Delta r_{jj} = \Delta r_1 {q_{rad}}^{jj-1}$, with the first cell ($jj = 1$) being the largest and centermost and cells shrinking toward the surface ($jj = n_{rj}$). This grading concentrates nodes near the particle surface where concentration gradients are steepest during fast charge/discharge. The same ratio q_{rad} is applied to both anode and cathode solid particles, but the radii of the particles can differ (see RADA on *BATTERY_ECHEM_MAT_ANODE and RADC on *BATTERY_ECHEM_MAT_CATHODE). When used, setting a value of $q_{rad} \approx 0.85$ is a good starting point.

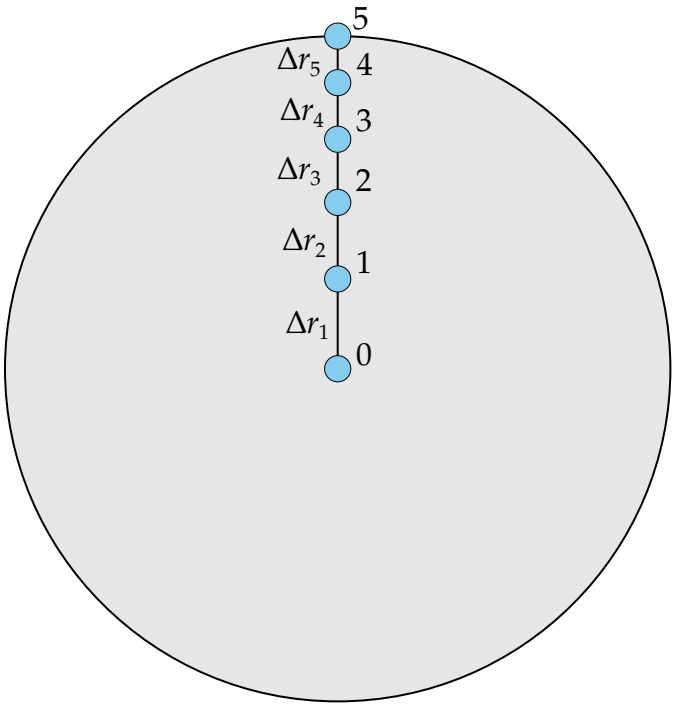


Figure 2-1. Example of the radial mesh inside a solid particle. Here, $n_{\text{rj}} = 5$ and $q_{\text{rad}} = 0.85$.

3. **Changes in the cell geometry as a degradation method.** The anode, cathode, and separator thicknesses can change as a result of thermal runaway, temperature, state of charge, or gas pressure. To include this degradation method for the anode, set ALEN to a negative value with |ALEN| referring to a *DEFINE_FUNCTION that returns the current effective length for the anode. This similarly applies to the cathode and separator by setting CLEN and SLEN to negative values to obtain their current effective lengths. The *DEFINE_FUNCTION supports the following arguments: f(a1, a2, a3, a4, a5, temp, soc_a, soc_c, pgas). Here, a*i* refers to the i^{th} reaction progress variable for thermal runaway (see *BATTERY_ECHEM_THERMAL_ABUSE), temp is the temperature, soc_a is the state of charge in the anode, soc_c is the state of charge in the cathode, and pgas is the gas pressure. Typical applications include separator melt/collapse, electrode swelling, or pressure-driven compression.

.

*BATTERY_ECHEM_CONTROL_INTEGRATION

Purpose: Set advanced parameters controlling the P2D solve.

Card 1	1	2	3	4	5	6	7	8
Variable	ISOLV	RCTOL	ACTOL	NITER	OMI	OMMAX	OMMIN	
Type	I	F	F	F	F	F	F	
Default	0	1.e-7	1.e-10	100	0.9	1.0	0.1	

This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	IELETIM	IRADTIM	VCLAMP	CCLAMP	ECLAMP	OCPEXT		
Type	I	I	F	F	F	F		
Default	0	0	0.02	1.e-6	2.	0.		

This card is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	MAXDIV	DTGR	DTAUT	DTMAX	DTEMP			
Type	I	F	F	F	F			
Default	10	1.1	0	optional	optional			

VARIABLE	DESCRIPTION
ISOLV	Band solver selection: EQ.0: Newman band solver. This solver is a custom block-tridiagonal algorithm with block pivoting (forward elimination + back-substitution, called per-node during assembly).

VARIABLE	DESCRIPTION
	EQ.1: LAPACK DGBTRF + DGBTRS. This solver is a general banded LU factorization with full partial pivoting across the entire band, followed by triangular solve. Assembly stores all blocks first, then solves the global system in one call. This method is more robust than ISOLV = 0 with a small overhead. EQ.2: LAPACK DGBSVX. This solver uses the same LU factorization as ISOLV = 1 but with row/column equilibration (rescales badly conditioned systems), reciprocal condition *r*, and iterative refinement of the solution.
RCTOL	Relative convergence tolerance for Newton scheme
ACTOL	Absolute convergence tolerance for Newton scheme
NITER	Maximum number of Newton iterations
OMI	Initial Newton damping factor, ω_{init}. See Remark 1.
OMMAX	Maximum Newton damping factor, ω_{max}. See Remark 1.
OMMIN	Minimum Newton damping factor, ω_{min}. See Remark 1.
IELETIM	Time integration numerical method for the electrolyte material balance equation: EQ.0: Backward Euler. This method provides maximum stability but less accuracy. EQ.1: Crank-Nicolson. This method is second-order accurate, but it can produce oscillatory solutions.
IRADTIM	Radial time integration method for the solid phase diffusion equation: EQ.0: TR-BDF2. This method is a two-stage scheme: a trapezoidal rule half, followed by a BDF2 step from with fixed sub-step ratio. It is L-stable (important for very stiff systems), second order, and self-starting but adds some computational cost. EQ.1: Standard BDF2. This method falls back to BDF1 (backward Euler) for the first step. It is an A-stable, second order integration method. EQ.2: Crank-Nicolson. This method is second-order

VARIABLE	DESCRIPTION
	accurate.
VCLAMP	Voltage clamping, V_{clamp}. See Remark 2.
CCLAMP	Stoichiometry clamp, c_{clamp}. See Remark 2.
ECLAMP	Overpotential clamp, η_{clamp}. See Remark 2.
OCPEXT	Open-circuit potential (OCP) linear extrapolation tolerance, ε_{ocp}. See Remark 2.
MAXDIV	Maximum number of time step divisions allowed before giving up. See Remarks 3 and 4. This field is ignored if DTAUTO = 1.
DTGR	Time step growth factor. For standalone mode, the time step can grow to DTMAX. For coupled analyses, DTGR only applies after halving has occurred and convergence has been achieved. See Remarks 3 and 4. This field is ignored if DTAUTO = 1.
DTAUTO	Time step adaptation (see Remarks 3 and 4): EQ.0: Time step for the P2D solve is allowed to grow (using DTGR) or be reduced in the case of nonconvergence. EQ.1: Fixed time step
DTMAX	Maximum time step. See Remarks 3 and 4. This field is ignored if DTAUTO = 1.
DTEMP	Maximum temperature change that can occur between two P2D time steps (see ITYPE = 1 in *BATTERY_ECHEM_CONTROL_THERMAL). This criterion is mostly useful when a thermal runaway model is activated (see *BATTERY_ECHEM_THERMAL_ABUSE). This field is ignored if DTAUTO = 1.

Remarks:

1. **Damping.** After each iteration, the correction $\delta\mathbf{u}$ from the band solver is applied with damping. Thus, the set of unknowns, $\mathbf{u}$, is only allowed to move by a certain factor between iterations, k:

$$\mathbf{u}^{k+1} = \mathbf{u}^{k} + \omega\delta\mathbf{u}$$

Here, ω is adapted dynamically between the minimum and maximum values provided by in the input: $\omega_{min} \leq \omega \leq \omega_{max}$. ω is reduced by half when clamping

occurs, and then increases by 20% when it no longer does. To modify the starting value of ω, set OMI.

2. **Clamping.** Clamping occurs when one of the following limits is reached after a Newton step:

   - *Electrolyte concentration, c_e.* c_e has a relative bound of $[c_e/100, c_e \times 100]$ and an absolute floor of $1.e^{-10}$.
   - *Potentials φ_s and φ_e.* The change in these potentials is limited by $\pm V_{clamp}$ (default 0.02 V) per iteration. To set V_{clamp}, use input parameter VCLAMP.
   - *Solid concentration, c_s.* c_s has a relative bound of $[c_s/100, c_s \times 100]$ and absolute physics-based bounds of $c_{clamp} c_{s,max} \le c_s \le (1. - c_{clamp}) c_{s,max}$. Here, c_{clamp} is set with input parameter CCLAMP, and $c_{s,max}$ is determined from input parameters for the anode and cathode (see *BATTERY_ECHEM_MAT_ANODE and *BATTERY_ECHEM_MAT_CATHODE). The absolute physics-based bounds prevent the stoichiometry $s = c_s/c_{s,max}$ from reaching the thermodynamic singularity at $s = 0$ or $s = 1$.
   - *Overpotential, η.* The change in η is limited by $\pm \eta_{clamp}$ (default 2.0 V) per iteration. To set η_{clamp}, use input parameter ECLAMP. This limit caps the exponential arguments and prevents floating-point overflow during early Newton iterations.
   - *OCP linear extrapolation.* Near full depletion ($s \to 0.$) or full saturation ($s \to 1.$), many OCP models exhibit steep gradients (polynomial fits) or outright singularities (Nernst logarithmic model). These extreme slopes amplify Newton-step corrections, producing large c_s overshoots that trigger the stoichiometric clamp, potential clamp, and ultimately substep halvings and convergence failures. It is possible to define ε_{ocp} as a small value limit that causes a linear interpolation of U_{ocp} around the $s < \varepsilon_{ocp}$ and $s > 1. - \varepsilon_{ocp}$ boundaries. This ε_{ocp}-induced linear extrapolation is simultaneously applied on the exchange current i_0. All the clamping methods mentioned in this bullet point are attempts to cause the simulation to run in a stable fashion a little bit longer near the boundaries, but it might be worth reconsidering the validity of a given U_{ocp} model at extreme ranges and provide an improved one.

3. **Time step for standalone battery models.** In standalone mode, *BATTERY_ECHEM_INITIAL provides the initial time step for each cycle. The default behavior is to allow the time step to grow (see DTGR) if several consecutive P2D solves have converged until an optional DTMAX has been reached. If the number of iterations required to converge rises above a certain threshold (NITER), the solver automatically divides the time step up to a limit of MAXDIV. Note that if DTAUTO = 1 the time step is fixed as the initial time step.

4. **Coupled battery models and subcycling.** Small time scales are often involved in the P2D resolution compared to the global cell operating condition changes. Thus, in coupled EM-P2D models, a larger time step can often be used for the EM analysis compared to the P2D solve. Our coupled implementation allows for subcycling in which the P2D model solves with a smaller time step but does not update the EM solver until it reaches the next time step of the EM solver. For coupled analyses, the initial time step, Δt_{sub}, for the P2D solve is provide by the EM solver and can be smaller than the EM time step. Δt_{EM}, if $n_{\text{sub}} > 1$:

$$\Delta t_{\text{sub}} = \frac{\Delta t_{\text{EM}}}{n_{\text{sub}}} \ .$$

n_{sub} is specified with NUMLS on *EM_CONTROL.

If a substep fails to converge within NITER Newton iterations, the P2D time step is halved and the substep is retried from the same state. If a substep converges but with a high iteration count (more than NITER/4), the step is accepted but is also halved as a precaution ("stall warning"). Both types of halving count toward a cumulative budget of MAXDIV divides per step. After halving and successfully converging, the time step is allowed to recover gradually to its original value or DTMAX (whichever is smaller) using growth factor DTGR so that a single stiff transient does not penalize the entire step. Note that DTMAX only applies when the time step is recovering and it is smaller than the initial time step from the EM solver. If DTAUTO = 1, the time step remains Δt_{sub} throughout the cycle, even if a substep fails to converge.

*BATTERY_ECHEM_CONTROL_SOLVER

Purpose: Set general purpose control variables for a battery electrochemistry simulation.

Card Summary:

Card 1. This card is required.

IMODEL	IDIM	NCYCLE		ITRA	IGAS	IP2D	IPORO

Card 2. Include NCYCLE instances of this card (one for each cycle) when IDIM = 1.

CMODE	CTYPE	CEND	TCUT	VCUT	CURR	TAPER	CELLID

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	IMODEL	IDIM	NCYCLE		ITRA	IGAS	IP2D	IPORO
Type	I	I	I		I	I	I	I
Default	none	none	1		0	0	0	0

VARIABLE	DESCRIPTION
IMODEL	Sets the battery model: EQ.1: 6-equation P2D model. ITRA and IGAS must be set to 0. EQ.2: 10- or 15-equation P2D model. With IGAS = 0, the model includes aging effects (10-equation model), while with IGAS = 1 or 2, the model includes both aging and gas generation effects (15-equation model). See Remark 1.
IDIM	Flag to select between a standalone model or coupled model: EQ.1: Standalone 1D model EQ.101: 1D models with EM-thermo-mechanical coupling. See Remark 4.
NCYCLE	Number of cycles to run. The default value is 1 cycle. The maximum value for this parameter is 100.

VARIABLE	DESCRIPTION
ITRA	Aging heat flag that is only available for IMODEL = 2 (see Remark 2 of this keyword and Remark 2 of *BATTERY_ECHEM_AGING): EQ.0: Off EQ.1: On
IGAS	Gas generation model flag that is only available for IMODEL = 2 (see Remark 1): EQ.0: Off EQ.1: On EQ.2: On and venting model activated
IP2D	Radial diffusion scheme: EQ.0: Superposition EQ.1: BDF finite differences
IPORO	Porosity feedback due to aging when IMODEL = 2 (see Remark 1 of *BATTERY_ECHEM_AGING): EQ.0: Off EQ.1: On

Cycle Card. Include NCYCLE instances of this card (one for each cycle) when IDIM = 1.

Card 2	1	2	3	4	5	6	7	8
Variable	CMODE	CTYPE	CEND	TCUT	VCUT	CURR	TAPER	CELLID
Type	I	I	I	F	F	F	F	I
Default	1	0	none	none	2.0	none	0.	optional

VARIABLE	DESCRIPTION
CMODE	Battery running mode flag (see Remark 5): EQ.0: Galvanostatic run (imposed current) EQ.1: Potentiostatic run (current chosen so voltage is held

VARIABLE	DESCRIPTION
	constant) EQ.2: CC-CV run (imposed current switching to constant voltage mode once VCUT is reached) EQ.3: Constant resistance run (current chosen so resistance is held constant)
CTYPE	Boundary condition type: EQ.0: Constant EQ.1: VCUT and CURR refer to load curve IDs.
CEND	Cause of battery cycle termination: EQ.1: Cycle run for a given time period. EQ.2: Cycle run until a given cut-off voltage.
TCUT	Total running time for the cycle (units: s)
VCUT	Cut-off voltage for the cycle or holding voltage depending on CMODE (units: V)
CURR	For CMODE = 0 or 2 (in CC mode), cycle operating current in the case of a constant current (CTYPE = 0) or a load curve ID giving the current as a function of time (CTYPE = 1). For CMODE = 1 and 3, CURR gives the initial current guess. (units: A/m^2)
TAPER	Taper current. Below this absolute current value, the cycle is terminated.
CELLID	Optional cell ID that links this particular cycle to a specific cell (see Remark 6)

Remarks:

1. **Battery models with aging and gas generation.** With IMODEL = 2, you can model a battery with aging effects or a battery with both aging and chemistry (gas generation) effects, depending on the value of IGAS. These two battery models include SEI tracking, thermal heat, and battery swelling. The 15-equation model additionally includes a gas generation model for modeling chemistry effects. See *BATTERY_ECHEM_AGING and *BATTERY_ECHEM_GAS. A venting model can be added to the gas generation model (see *BATTERY_ECHEM_VENT).

2. **ITRA.** Setting this field to 1 when IMODEL = 2 causes calculating aging-related heat due to SEI formation and decomposition. See Remark 2 of *BATTERY_ECHEM_AGING. In standalone models (IDIM = 1), it is consumed internally if ITYPE = 1 in *BATTERY_ECHEM_CONTROL_THERMAL. Otherwise it is just reported to an output file as part of the irreversible heat. For coupled simulations (IDIM = 101), it is returned to the thermal solver as a heat source if ITYPE = 0, but again consumed internally in the in the ITYPE = 1 case.

3. **P2D model**. The P2D model is a physics-based electrochemical model of lithium-ion batteries. It resolves what is happening inside the electrodes and electrolyte in space and time. The core idea is that the model describes the lithium-ion diffusion in the electrode, migrating from the particle centers to the particle surfaces, lithium-ion transfer to the electrolyte (via electrochemical reaction kinetics), followed by transport in the electrolyte through the separator and then diffusion again through the solid particles on the opposite electrode. It also must include charge conservation in both phases. Figure 2-2 provides a schematic of a unit cell.

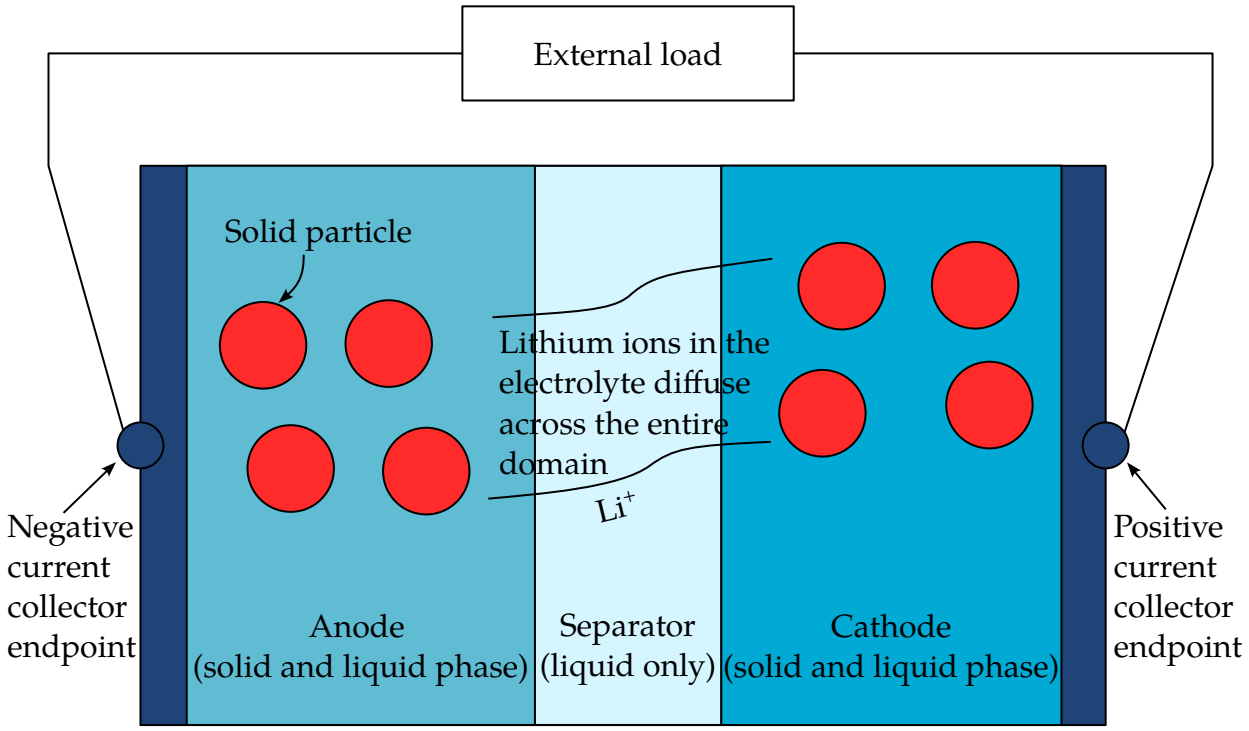


Figure 2-2. Schematic of a unit cell

Given an applied outside source current, i_{app}, (see Remark 5) and cell properties, the P2D model solves for the following six unknowns with additional unknowns in the case of aging and gas generation:

c_e Li^+ concentration in the electrolyte (units: mol/(m³))

φ_e Electrolyte potential (units: V)

i_2 Current density in the electrolyte (units: A/m^2)

φ_s Solid phase potential (units: V)

$c_{s,surf}$ Li^+ concentration on the surfaces of the solid electrode particles (units: mol/m^3)

j_n Pore wall molar flux between the solid and electrolyte (units: $mol/(m^2s)$)

See *BATTERY_ECHEM_AGING and *BATTERY_ECHEM_GAS for a discussion of the additional unknowns in the 10- and 15-equation models.

In the P2D model, each unit cell is discretized into a one-dimensional mesh along the local through-thickness coordinate x. The unit cell generally connects to the FEM mesh through the negative and positive current collect endpoints. The mesh points for the P2D mesh lie in the anode, cathode, and separater. See Figure 2-3. The anode, cathode, and separator all contain electrolyte. Each anode and cathode mesh point has a representative solid particle on which the diffusion of lithium through the particle is modeled radially; hence the pseudo-2D. Thus, in the anode and cathode all six unknowns are solved for, while only the unknowns pertaining to the electrolyte are solved for in the separator. For a detailed description of the cell discretization, see *BATTERY_ECHEM_CELL_GEOMETRY.

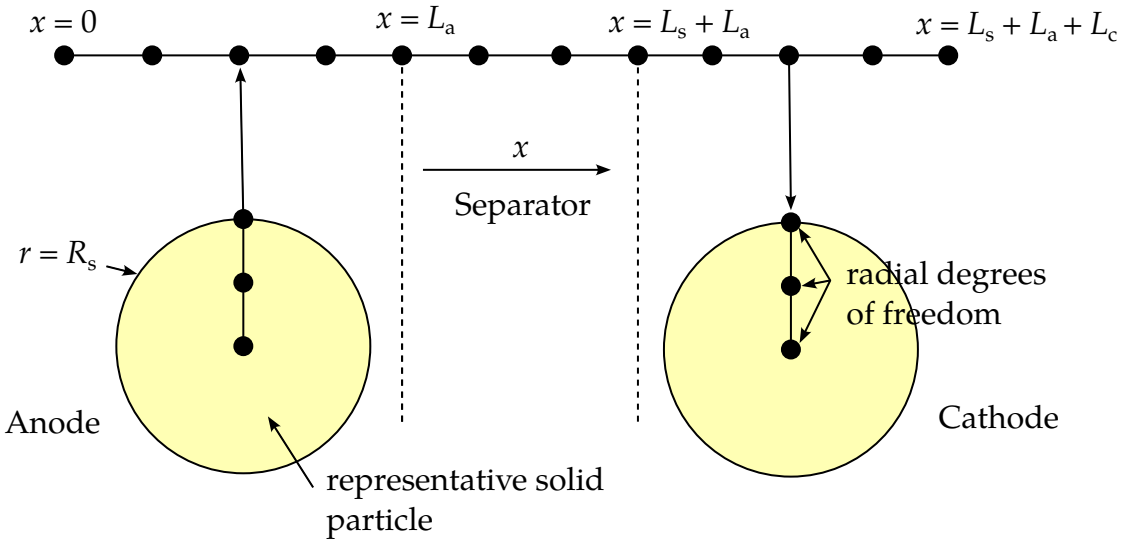


Figure 2-3. Visualization of the discretization

a) *Equation 1: electrolyte material balance.* The conservation of lithium ion concentration in the electrolyte pore space (liquid) is a transient parabolic partial differential equation with diffusion along x:

$$\varepsilon_{\mathrm{e},r}\frac{\partial c_\mathrm{e}}{\partial t}=\frac{\partial\left(D_e^{\mathrm{eff}}\frac{\partial c_\mathrm{e}}{\partial x}\right)}{\partial x}+\frac{1-t_+^0}{F}\frac{\partial i_2}{\partial x}$$

In the above, $\varepsilon_{\mathrm{e},r}$ is the electrolyte volume fraction in the relevant region r ($\varepsilon_{\mathrm{e,a}}$, $\varepsilon_{\mathrm{e,s}}$, and $\varepsilon_{\mathrm{e,c}}$ with a, s, and c standing for anode, seperator, and cathode), $D_\mathrm{e}^{\mathrm{eff}}=D_\mathrm{e}\varepsilon_{\mathrm{e},r}^p$ is the Bruggeman-corrected diffusion coefficient with D_e as the diffusion coefficient and p as the Bruggeman coefficent (see BRUGA on *BATTERY_ECHEM_MAT_ANODE, BRUGC on *BATTERY_ECHEM_MAT_CATHODE, and BRUGE on *BATTERY_ECHEM_MAT_ELECTROLYTE), t_+^0 is the cation transference number, and F is Faraday's constant. The electrolyte transport properties, meaning D_e and t_+^0, are functions of c_e and T that depend on the electrolyte model (see IOCPE on *BATTERY_ECHEM_MAT_ELECTROLYTE). The initial concentration c_e^0 is defined as LICS in *BATTERY_ECHEM_MAT_ELECTROLYTE.

The boundary conditions for this equation are zero flux, $\frac{\partial c_\mathrm{e}}{\partial x}=0$, at the endpoints of the unit cell mesh ($j=1$ and $j=n_{\mathrm{tot}}$). After each Newton iteration, a global lithium conservation constraint renormalizes c_e to preserve the total moles in the electrolyte.

b) *Equation 2: electrolyte potential.* This equation is a modified Ohm's law (steady, no transient effects) in the electrolyte liquid phase:

$$\frac{\partial\varphi_\mathrm{e}}{\partial x}=-\frac{i_2}{\kappa^{\mathrm{eff}}}+\mathrm{SF_T}\frac{1-t_+^0}{f_{\mathrm{RT}}}\left(\frac{1}{c_\mathrm{e}}+\frac{\partial\ln f_\pm}{\partial c_\mathrm{e}}\right)\frac{\partial c_e}{\partial x}$$

Here, κ^{eff} is the effective ionic conductivity of the electrolyte, meaning $\kappa^{\mathrm{eff}}=\kappa\varepsilon_{\mathrm{e},r}^p$. κ is a property of the electrolyte that is determined by the electrolyte model (see IOCPE on *BATTERY_ECHEM_MAT_ELECTROLYTE) and depends on c_e and T. $\mathrm{SF_T}$ is a thermodynamic factor specified with TFACT on *BATTERY_ECHEM_MAT_ELECTROLYTE; it typically is set to 2.0. $\frac{\partial\ln f_\pm}{\partial c_\mathrm{e}}$ is a function of c_e and T that is determined by the electrolyte model (see IOCPE on *BATTERY_ECHEM_MAT_ELECTROLYTE). f_{RT} is a derived Faraday ratio given by $F/(R_\mathrm{g}T)$. The boundary condition is $\varphi_\mathrm{e}=0$ at the cathode current collector interface ($j=n_{\mathrm{tot}}$). This is the reference ground.

c) *Equation 3: electrolyte current balance*. The conservation of charge (steady with no transient effects) in the electrolyte is given by:

$$\frac{\partial i_2}{\partial x}=a_\mathrm{s}Fj_\mathrm{n}$$

where $a_\mathrm{s}=3\varepsilon_{\mathrm{am},r}/R_{\mathrm{s},r}$, $R_{\mathrm{s},r}$ is the particle radius, and $\varepsilon_{\mathrm{am},r}$ is active material fraction. The active material is not an input parameter but a derived quantity that depends on the four input volume fractions:

$$\varepsilon_{\mathrm{am},r} = 1 - \varepsilon_{\mathrm{e},r} - \varepsilon_{\mathrm{p},r} - \varepsilon_{\mathrm{f},r} - \varepsilon_{\mathrm{g},r} \quad .$$

$\varepsilon_{\mathrm{e},r}$ is the electrolyte volume fraction, $\varepsilon_{\mathrm{p},r}$ is the polymer volume fraction, $\varepsilon_{\mathrm{f},r}$ is filler volume fraction, and $\varepsilon_{\mathrm{g},r}$ is the gas volume fraction. r stands for region and can be a for anode, s for separator, and c for cathode. These volume fractions are set on *BATTERY_ECHEM_MAT_ANODE, *BATTERY_ECHEM_MAT_ELECTROLYTE, and *BATTERY_ECHEM_MAT_CATHODE.

The boundary conditions are $i_2 = i_{\mathrm{app}}$ at the anode-separator interface ($j = n_{\mathrm{a}} + 1$), $i_2 = 0$ at the cathode current collector interface ($j = n_{\mathrm{tot}}$), and $j_{\mathrm{n}} = 0$ in the separator, making i_2 a constant in the separator. i_{app} is the current applied to the unit cell.

d) *Equation 4: Ohm's law in the solid phase.* The charge transport in the solid (electron-conducting) matrix is given by:

$$\frac{\partial \varphi_{\mathrm{s}}}{\partial x} = -\frac{i_{\mathrm{app}} - i_2}{\sigma_{\mathrm{s}}^{\mathrm{eff}}}$$

where $\sigma_{\mathrm{s}}^{\mathrm{eff}}$ is the effective solid conductivity. The boundary conditions are $i_2 = i_{\mathrm{app}}$ at the anode-separator interface ($j = n_{\mathrm{a}} + 1$), $i_2 = 0$ at the cathode current collector interface ($j = n_{\mathrm{tot}}$), and $\varphi_{\mathrm{s}} = 0$ in the separator.

e) *Equation 5: solid-phase diffusion*. This equation captures how lithium atoms move inside the active material particles of the electrodes (diffusion between the surface and interior). The P2D model assumes that each node in the anode and cathode along the x-axis contains a representative spherical electrode particle (see *BATTERY_ECHEM_CELL_GEOMETRY) of radius $R_{\mathrm{s},r}$ (r is the region with a for anode and c for cathode). The spherical particles are discretized radially based on the settings in *BATTERY_ECHEM_CELL_GEOMETRY. Fick's law in spherical coordinates is applied inside each of those particles:

$$\frac{\partial c_{\mathrm{s}}}{\partial t} = \frac{D_{\mathrm{s},r}}{r^2} \frac{\partial \left(r^2 \dfrac{\partial c_{\mathrm{s}}}{\partial r} \right)}{\partial r} \quad \text{for } 0 \le r \le R_{\mathrm{s},r}$$

In the above, $D_{\mathrm{s},r}$ is the solid diffusion coefficient with r standing for the region with a for anode and c for cathode. Thus, it can differ between the anode (DFSA for *BATTERY_ECHEM_MAT_ANODE) and cathode (DFSC on *BATTERY_ECHEM_MAT_CATHODE). Note that the discretization for the spherical particles is the same for both anode and cathode particles, but the radii can differ. The output from this equation is the surface concentration, $c_{\mathrm{s,surf}} = c_{\mathrm{s}}(R_{\mathrm{s},r}, j)$, at each j position.

The boundary conditions are:

$$\left.\frac{\partial c_s}{\partial t}\right|_{r=0} = 0$$
$$\left.\frac{\partial c_s}{\partial t}\right|_{r=R_{s,r}} = -\frac{j_n}{D_{s,r}}$$

f) *Equation 6: Butler-Volmer kinetics*. The Butler–Volmer equation is the key relation that connects electrochemistry to current flow in battery models like P2D. It simulates how fast lithium is inserted/extracted from the electrode as a function of voltage (overpotential) and illustrates the competition between forward (lithium insertion) and backward (lithium extraction) electrochemical reactions. This relation solves for the pore-wall flux, j_n.We have implemented four different formulations for the Butler-Volmer equation (see IBVTYPE on *BATTERY_ECHEM_MAT_ANODE and *BATTERY_ECHEM_MAT_CATHODE). Note that the anode and cathode can have different formulations.

In all the formulations, the key component is the overpotential, η, whose sign dictates the direction of the lithium transfer (for example, $\eta > 0$ means the forward direction dominates). η is expressed as:

$$\eta = \varphi_s - \varphi_e - U_{ocp}(c_{s,surf}) - Fj_nR_{film}$$

In the above, U_{ocp} is the open-circuit potential. It depends on the material model for the anode (see IOCPA on *BATTERY_ECHEM_MAT_ANODE) and electrode (see IOCPC on *BATTERY_ECHEM_MAT_CATHODE). R_{film} is the film resistance (see RANODE on *BATTERY_ECHEM_MAT_ANODE and RCATHDE on *BATTERY_ECHEM_MAT_CATHODE).

Another component of the Butler-Volmer equation is the exchange current density, i_0. The exchange current density expresses the reaction speed. We have implemented two versions, one for liquid electrolytes and one for polymer electrolytes. The version can be selected with IELETYPE on *BATTERY_ECHEM_MAT_ELECTROLYTE. However, this selection can be overridden individually for the anode and cathode with an exchange current density provided with a *DEFINE_FUNCTION. See I0FUNC on *BATTERY_ECHEM_MAT_ANODE and *BATTERY_ECHEM_MAT_CATHODE.

4. **EM-thermo-mechanical coupling (IDIM = 101).** When IDIM = 101, the P2D model directly couples with the EM resistive solver (set EMSOL = 3 on *EM_CONTROL). Through the EM solver, the P2D model can additionally couple to the thermal and/or mechanical solvers. By default, the P2D solver is automatically coupled with the thermal solver, but this can be turned off by setting IP2DTH = 1 on *EM_CONTROL_COUPLING_P2D. Coupling with the mechanical solver requires including *MAT_ADD_P2D_EXPANSION. To include swelling feedback to the mechanical solver, set ISWELL = 1 on *EM_CON-

TROL_COUPLING_P2D. Additionally, when using the BatMac model discussed below, the mechanical solver can influence shorts and cell geometry (see *EM_P2D_BATMAC for details). Thus, when IDIM = 101, the P2D model always couples to the EM solver but optionally can couple to the other solvers.

The EM resistive solver solves a Laplace equation for the scalar potential, φ, on the 3D finite element mesh:

$$\nabla \cdot (\sigma \nabla \varphi) = 0$$

In a battery simulation, for each P2D circuit, there are two degrees of freedom representing the scalar potentials at the positive current collector (φ_{CC^+}) and negative current collectors (φ_{CC^-}). A P2D circuit is generally a unit cell with an area multiplier, A_{icirc}, to represent many unit cells in parallel. These degrees of freedom are connected with a Robin boundary condition. This Robin boundary condition essentially adds a penalty to the stiffness matrix for these two degrees that is physically consistent with the inverse of the circuit's resistance and an additional term to the right-hand side vector that is physically consistent with a current.

For each circuit at each EM time step, the P2D model is solved twice (base and perturbed) to compute the effective resistance, R_{eff}, through finite difference perturbation:

$$R_{\text{eff}} = \frac{V_{\text{cell}}(i_{\text{app}} + \delta i) - V_{\text{cell}}(i_{\text{app}})}{\delta i}$$

In the above, i_{app} is the current density applied to the P2D circuit from the EM solver. This density often comes from the previous EM time step. δi is a small perturbation added to the current density. V_{cell} comes from the P2D solver and is the difference in the solid phase potential, φ_s, between the two ends of the unit cell. With R_{eff}, we can also express the current that enters the right-hand side as:

$$F = I_{\text{app}} - \frac{\varphi_{CC^+} - \varphi_{CC^-}}{R_{\text{eff}}}$$

where I_{app} is the current given as $I_{\text{app}} = i_{\text{app}} A_{\text{icirc}}$. Once an EM solve is performed, I_{app}, φ_{CC^+}, and φ_{CC^-} are updated for the next subsequent P2D solve.

We have two models for coupling the EM solver to the P2D circuit: meshless (*EM_P2D_MESHLESS) and BatMac (*EM_P2D_BATMAC). The meshless model has a single circuit, while BATMAC has a circuit per node in the finite element mesh. They also specify how A_{icirc} is calculated. For a more detailed description of each model, see their respective manual pages.

5. **Boundary conditions.** In the coupled EM-P2D approach, the applied outside current density, i_{app}, is always provided as input by the EM solver to drive the battery cell charge and discharge conditions (tabs connected to a

voltage/current source or resistance). When running the P2D solver in standalone mode, CMODE on Card 2 determines how the current density is provided. In galvanostatic mode (CMODE = 0), CURR directly supplies current density. Alternatively, you can prescribe the voltage with CMODE =1 (potentiostatic mode) or resistance with CMODE = 3. However, the P2D model requires an input current density, so running in potentiostatic or constant resistance mode triggers a secant search that increases the number of P2D solves. Lastly, selecting CMODE = 2 leads to a CC-CV mode which is a constant current followed by a constant voltage. This mode automates the two-phase charging protocol commonly used in practice. With this mode, a current density, CURR, is applied until cell voltage reaches VCUT. Then, the solver switches to the CV phase where the cell voltage is maintained at VCUT using the secant search approach with the current tapering out as the cell approaches equilibrium. The CV phase ends when the absolute current density drops below TAPER or TEND is reached.

6. **Cell ID.** The P2D model uses a cell ID to associate a set of electrochemical material properties with a particular simulation entity. The role of the cell ID differs between the standalone and EM-coupled paths:

   a) *Standalone mode (IDIM = 1).* In the standalone solver, the optional cell ID enables multi-configuration simulations within a single run. Each keyword (*BATTERY_ECHEM_CELL_GEOMETRY, *BATTERY_ECHEM_MAT_ANODE, *BATTERY_ECHEM_MAT_CATHODE, *BATTERY_ECHEM_MAT_ELECTROLYTE, and *BATTERY_ECHEM_MAT_THERMAL) carries a CELLID field on its first card. When multiple cards are defined with different CELLID values, the solver maintains independent per-cell electrochemical states. Cycling schedules (*BATTERY_ECHEM_CONTROL_SOLVER) reference a CELLID per cycle allowing for example, HPPC pulse profiles at different temperatures or C-rates against different cell parameterizations in a single simulation. At each cycle boundary where the CELLID changes, the outgoing cell's full state is saved, and the incoming cell's state is restored. If no cell ID is given, then the entire model will be considered as a single cell analysis (if multiple cycles are defined, they will be considered as being applied on the same single cell).

   b) *EM coupled mode (IDIM = 101).* Coupling to the EM solver has two different models: BATMAC (*EM_P2D_BATMAC) and meshless (*EM_P2D_MESHLESS). How cell ID is used depends on the chosen model. For BATMAC, the association between the finite element geometry and P2D electrochemistry is controlled entirely by *EM_P2D_BATMAC which has a cell ID (CELLID) and a part set ID (PSID) as input to associate the P2D model with specific solid elements. If the user leaves the cell ID fields blank in the *BATTERY keywords, then the P2D parameters are applied as default to all elements. For the meshless model, P2DMCID is an ID for the meshless

circuit that must match RDLID on *EM_ISOPOTENTIAL_CONNECT to connect the meshless circuit to the EM tabs. If CELLID is unused on the *BATTERY keywords, then this field can be any integer value as long as it matches RDLID. Otherwise, it must match the cell ID on the *BATTERY keywords to associate the correct P2D parameters with the circuit. The circuit is not associated directly with nodes in a mesh in this case, so PSID is only used for optional thermal feedback.

*BATTERY_ECHEM_CONTROL_THERMAL

Purpose: Set parameters for the thermal treatment in a P2D model.

Card 1	1	2	3	4	5	6	7	8
Variable		ITYPE					IARR	IHEAT
Type		I					I	I
Default		0					0	0

VARIABLE	DESCRIPTION
ITYPE	Flag for how temperature is determined (see Remark 1): EQ.0: Constant temperature in standalone models. For coupled models, the temperature remains constants during a P2D solve with macro thermal updates from the thermal solver. EQ.1: Variable temperature mode in standalone models. For coupled models, the heat sources from the P2D cause "micro" thermal updates during the P2D solve with all other heat changes coming from the thermal solver.
IARR	Arrhenius temperature corrections (see Remark 2): EQ.0: Off EQ.1: On
IHEAT	Heat generation calculation method for electrochemical heat generation (see Remark 5): EQ.0: Energy balance method EQ.1: Local evaluation method

Remarks:

1. **ITYPE.** The ITYPE field selects how temperature is handled within the P2D model:

   a) *ITYPE = 0*. For the standalone model, this setting causes the temperature to remain constant throught the P2D solve. Note that the temperature can be

provided as a function of time by setting T0 to a negative number on *BATTERY_ECHEM_MAT_THERMAL. The heating terms are calculated and reported but not used to change the temperature. For a coupled model in which thermal coupling is enabled, the cell heating terms are returned for subsequent coupling with the thermal solver. The thermal solver then takes the heating terms and determines the temperature.

b) *ITYPE = 1*. In both standalone and coupled models, a lumped temperature update is run at each subcycle n:

$$T^{n+1} = T^n + \frac{\mathrm{d}t}{C_\mathrm{p} m_\mathrm{t}}\left[h(T_\mathrm{amb} - T^n) + \varepsilon_\mathrm{rad}\sigma_\mathrm{SB}(T_\mathrm{amb}^4 - (T^n)^4 + \dot{q}_\mathrm{cell}\right]$$

In the above, C_p, h, T_amb, and ε_rad are quantities provided by *BATTERY_ECHEM_MAT_THERMAL and σ_SB is the Stefan Boltzmann constant. m_t can be provided *BATTERY_ECHEM_MAT_THERMAL or calculated using electrode densities, volume fractions, and layer thicknesses. $\dot{q}_\mathrm{cell}$ is the sum of all the heat generation terms (see Remark 3).

In coupled models, this method allows for a more granular "micro" thermal update during the P2D solve. Since this temperature update already consumes the cell heat sources, they are returned as 0. to the EM solver so that subsequent "macro" coupling with the thermal solver does not count those contributions twice. With this approach, we recommend setting h and ε_rad to zero and letting the thermal "macro" solve handle the heat exchange with the external surfaces of the cell. After completing the P2D solve, the final T^{n+1} "micro" temperature update is returned for each circuit and overwrites the local temperature for the subsequent thermal "macro" solve.

2. **Arrhenius temperature corrections.** When the IARRH = 1, the reaction rate constants for the anode and cathode (see RATEA on *BATTERY_ECHEM_MAT_ANODE and RATEC on *BATTERY_ECHEM_MAT_CATHODE), solid diffusion coefficients for the anode and cathode (see DFSA on *BATTERY_ECHEM_MAT_ANODE and DFSC on *BATTERY_ECHEM_MAT_CATHODE), film resistances for the anode and cathode (see RANODE on *BATTERY_ECHEM_MAT_ANODE and RCATHDE on *BATTERY_ECHEM_MAT_CATHODE), and electrolyte transport properties D_e and κ (see *BATTERY_ECHEM_MAT_ELECTROLYTE) are corrected for temperature:

$$X(T) = X_\mathrm{ref} e^{\frac{E_X/R_\mathrm{g}(T - T_\mathrm{ref})}{T T_\mathrm{ref}}}$$

where E_X/R_g (in K) is the activation energy divided by the gas constant and X is the quantity being corrected.

3. **Heat generation sources from the P2D cell.** The total cell heat source, $\dot{q}_\mathrm{cell}$, with all model options turned on is given by:

$$\dot{q}_{\text{cell}} = \dot{q}_{\text{ec}} + \dot{q}_{\text{SEI,form}} + \dot{q}_{\text{SEI,decomp}} + \dot{q}_{\text{gas}} + \dot{q}_{\text{abuse}} - \dot{q}_{\text{vent}} - \dot{q}_{\text{boil}} - \dot{q}_{\text{ej}}$$

4. $\dot{q}_{\text{ec}}$ is the electrochemical heat generation term that is calculated for all versions of the P2D model. The way it is calculated depends on IHEAT (see Remark 5). $\dot{q}_{\text{SEI,form}}$ and $\dot{q}_{\text{SEI,decomp}}$ are the SEI formation and decomposition heat calculated when aging is enabled with the 10-equation or 15-equation P2D models as long as ITRA = 1 on *BATTERY_ECHEM_CONTROL_SOLVER. See Remark 2 of *BATTERY_ECHEM_AGING. $\dot{q}_{\text{gas}}$ is the heat from gas generation calculated when using the 15-equation model (see Remark 4 of *BATTERY_ECHEM_GAS). $\dot{q}_{\text{abuse}}$ is the heat generated from the decomposition reactions in the thermal runaway model (see *BATTERY_ECHEM_THERMAL_ABUSE). $\dot{q}_{\text{vent}}$, $\dot{q}_{\text{boil}}$, and $\dot{q}_{\text{ej}}$ are heat sinks from the burst disk model (see Remark 6 of *BATTERY_ECHEM_VENT).

5. **IHEAT.** IHEAT selects between two equivalent but algebraically distinct formulations for computing the electrochemical heat generation based on Rao and Newman [1997]. IHEAT = 0 uses an enthalpy-based method. However, at low C-rates, two of the electrochemical heat generation terms (ohmic heating and irreversible heating due to overpotential dissipation) are close in magnitude but opposite in sign, so the sum of the terms can be sensitive to discretization errors. In this case, IHEAT = 1 may give a more robust solution as this method evaluates each source of heat dissipation locally and sums the contributions.

*BATTERY_ECHEM_GAS

Purpose: Set the battery gas generation parameters that are used by the 15-equation P2D model. See *BATTERY_ECHEM_CONTROL_SOLVER. This model only applies to the anode.

Card 1	1	2	3	4	5	6	7	8
Variable	IC2H4	I02	ICO2	IH2O	IH2			
Type	F	F	F	F	F			
Default	1.E-5	1.E-5	1.E-5	1.E-5	1.E-5			

Card 2	1	2	3	4	5	6	7	8
Variable	AG1	AG2	EG1	EG2				
Type	F	F	F	F				
Default	3.426E26	5.028E-6	2.50e5	2.51e5				

This card is optional. See Remark 3.

Card 3	1	2	3	4	5	6	7	8
Variable	HC2H4		HCO2	HH2O	HLIC6		HLIOH	
Type	F		F	F	F		F	
Default	0.		0.	0.	0.		0.	

VARIABLE	DESCRIPTION
IC2H4	Initial C_2H_4 concentration, c_{0,C_2H_4} (units: mol/m^3)
I02	Initial O_2 concentration, c_{0,O_2} (units: mol/m^3)
ICO2	Initial CO_2 concentration, c_{0,CO_2} (units: mol/m^3)

VARIABLE	DESCRIPTION
IH20	Initial H_2O concentration, c_{0,H_2O} (units: mol/m^3)
IH2	Initial H_2 concentration, c_{0,H_2} (units: mol/m^3)
AG1	Frequency factor for ethylene oxidation, A_{g1} (units: s^{-1})
AG2	Frequency factor for lithium hydration, A_{g2} (units: s^{-1})
EG1	Activation factor for ethylene oxidation, E_{ag1}(units: J/mol)
EG2	Activation factor for lithium hydration, E_{ag2}(units: J/mol)
HC2H4	Formation enthalpy of C_2H_4, $\Delta H_{C_2H_4}$ (units: kJ/mol)
HCO2	Formation enthalpy of CO_2, ΔH_{CO_2} (units: kJ/mol)
HH20	Formation enthalpy of H_2O, ΔH_{H_2O} (units: kJ/mol)
HLIC6	Formation enthalpy of LiC_6, ΔH_{LiC_6} (units: kJ/mol)
HLIOH	Formation enthalpy of LiOH, ΔH_{LiOH} (units: kJ/mol)

Remarks:

1. **Gas generation model.** This gas generation model adds five mass balance equations to the P2D model. These equations only apply to nodes in the anode. The additional unknowns solved for in these equations are:

   $c_{C_2H_4}$ Ethylene (C_2H_4) concentration (units: mol/m^3)

   c_{O_2} Oxygen (O_2) concentration (units: mol/m^3)

   c_{H_2O} Water (H_2O) concentration (units: mol/m^3)

   c_{CO_2} Carbon dioxide (CO_2) concentration (units: mol/m^3)

   c_{H_2} Hydrogen (H_2) concentration (units: mol/m^3)

   The mass balance for C_2H_4 is given by:

$$\frac{\partial c_{C_2H_4}}{\partial t} = -\frac{a_j}{2} j_s - a_j (r_d + r_{g1})$$

   where a_j is the specific interfacial area, j_s is the SEI side reaction flux (see *BATTERY_ECHEM_AGING), r_d is the SEI decomposition rate (see Remark 2), and

r_{g1} is the first gas generation reaction rate (see Remark 2). This equation reflects the ethylene being consumed by the SEI side reaction, SEI decomposition and first gas generation reaction.

The mass balance for O_2 is given by:

$$\frac{\partial c_{O_2}}{\partial t} = -\frac{a_j}{2}\left(r_d + r_{g1}\right) \ .$$

This equation reflects the oxygen being consumed by the SEI decomposition and first gas generation reaction.

The mass balance for H_2O is given by:

$$\frac{\partial c_{H_2O}}{\partial t} = 2a_j\left(r_{g1} - \frac{r_{g2}}{2}\right) \ .$$

where r_{g2} is a reaction rate discussed in Remark 2. Water is produced by the first gas generation reaction and consumed by the second gas generation reaction.

The mass balance for CO_2 is given by:

$$\frac{\partial c_{CO_2}}{\partial t} = 2a_j\left(r_{g1} - \frac{r_d}{2}\right)$$

Carbon dioxide is produced by the first gase generation reaction and consumed by the second gas generation reaction.

The mass balance for H_2 is given by:

$$\frac{\partial c_{H_2}}{\partial t} = a_j r_{g2}$$

Hydrogen is produced by the second gas generation reaction.

2. **Reaction rates**. The gas-phase evolution is driven by three Arrhenius-type reaction rates. The first is the decomposition rate given by:

$$r_d = A_f c_{SEI} e^{-E_a/(RT)}$$

where A_f is the frequency factor for the SEI decompostion set on Card 3 of *BATTERY_ECHEM_AGING, c_{SEI} is the SEI species concentration, E_a is the activation for the SEI decompostion set on Card 3 of *BATTERY_ECHEM_AGING, R is the universal gas constant, and T is temperature. The second is the first gas generation rate given by:

$$r_{g1} = A_{g1}\left(c_{C_2H_2} + c_{CO_2}\right)e^{-E_{ag1}/(RT)}$$

where A_{g1} is the frequency factor for ethylene oxidation and E_{ag1} is the activation factor for ethylene oxidation. The third is the second gas generation rate given by:

$$r_{g2} = A_{g2}\left(c_s + c_{H_2H_2}\right)e^{-E_{ag2}/(RT)}$$

where A_{g2} is the frequency factor for lithium hydrationn, c_s is the solid lithium concentration, and E_{ag2} is the activation factor for lithium hydration.

3. **Formation enthalpies.** An alternative to defining the enthalpies directly in this keyword is to refer to a Chemkin-format file using *CHEMISTRY_BATTERY. *CHEMISTRY_BATTERY provides the file names for the battery chemistry input file, the corresponding thermodynamics data file, and the transport properties file. Note that nonzero values provided on Card 3 will override data provided by *CHEMISTRY_BATTERY.

4. **Heat generation.** The two gas generation reactions release (or absorb) heat proportional to the net formation enthalpy change. The heat coming from gas generation is expressed as:

$$\dot{q}_{gas} = \Delta H_1 \int_0^{L_a} r_{g1}\mathrm{d}x + \Delta H_2 \int_0^{L_a} r_{g2}\mathrm{d}x \ .$$

ΔH_1 is the enthalpy from the first gas reaction:

$$C_2H_4 + O_2 \rightarrow CO_2 + H_2O$$
$$\Delta H_1 = \Delta H_{CO_2} + \Delta H_{H_2O} - \Delta H_{C_2H_4}$$

and ΔH_2 is the enthalpy from the second gas reaction:

$$H_2O + LiC_6 \rightarrow H_2 + LiOH$$
$$\Delta H_2 = \Delta H_{LiOH} - \Delta H_{H_2O} - \Delta H_{LiC_6}$$

In the above integral L_a is the length of the anode (see *BATTERY_ECHEM_CELL_GEOMETRY).

Note that for standalone battery models, this heat is only used if ITYPE = 1 on *CONTROL_ECHEM_THERMAL. It is output if ITYPE = 0 but not used. For coupled models, this heat is used regardless of ITYPE.

*BATTERY_ECHEM_INITIAL

Purpose: Provide the initial time step size for each cycle in standalone battery models (IDIM = 1 on *BATTERY_ECHEM_CONTROL_SOLVER). In an EM-P2D coupled analysis (IDIM = 101 on *BATTERY_ECHEM_CONTROL_SOLVER), this keyword is ignored, and the EM solver determines the initial time step size. See Remark 3 of *BATTERY_ECHEM_CONTROL_INTEGRATION.

Card 1	1	2	3	4	5	6	7	8
Variable	DT0							
Type	F							
Default	none							

VARIABLE	DESCRIPTION
DT0	Initial time step size for each cycle

*BATTERY_ECHEM_MAT_ANODE

Purpose: Set the battery material parameters for the anode side electrode.

Card Summary:

Card 1. This card is required.

CELLID	IOCPA	CAPTA	SOCA	RADA	RATEA	RANODE	IATYPE

Card 2. This card is required.

RHOEA	RHOFA	RHOCCA	DFSA	CONDA	MWA		

Card 3. This card is required.

VFEA	VFPA	VFFA	VFGA	BRUGA	B1STAR	LPORE	

Card 4. This card is optional.

IBVTYPE	ALFAA	ALFAC	IOFUNC				

Card 5. This card is optional.

EARK	EARDS	EARF					

Card 6. This card is optional.

S0A	S100A	PMV					

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	CELLID	IOCPA	CAPTA	SOCA	RADA	RATEA	RANODE	IATYPE
Type	I	I	F	F	F	F	F	I
Default	optional	none	none	none	none	none	0.0	0

VARIABLE	DESCRIPTION
CELLID	Optional cell ID. See Remark 6 of *BATTERY_ECHEM_CONTROL_SOLVER.
IOCPA	Material type for the open-circuit potential in the anode electrode (see Remark 5): LT.0: Absolute value points to a *DEFINE_FUNCTION ID. EQ.1: Sony carbon (petroleum coke). See Doyle et al [1993]. EQ.2: MCMB 2510. See Cai and White [2009]. EQ.3: MCMB 2528 EQ.4: KS_6 graphite. See Doyle et al [1996]. EQ.5: LiC_6 graphite. See Safari and Delacourt [2011]. EQ.7: Metal hybrid. See Paxton and Newman [1997]. EQ.10: Nernst ideal intercalation. See Newman and Balsara [2021].
CAPTA	Coulombic capacity of the anode material, $c_{sp,a}$ (units: Ah/kg)
SOCA	Initial lithium stoichiometric coefficient of the anode side active material. For example, Li_xC_6 ($0 < x < 0.7$).
RADA	Radius of spherical particles, $R_{s,a}$, in the anode side active material (units: m)
RATEA	Reaction rate constant, k_a, for the anode electrode (units: $m^{2.5}mol^{-0.5}s^{-1}$)
RANODE	Film resistance for the anode electrode, R_{film} (units: $\Omega \cdot m^2$). This field can be left as 0. See Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER.
IATYPE	Anode type: EQ.0: Porous intercalation anode EQ.1: Lithium metal foil (see Remark 6)

Card 2	1	2	3	4	5	6	7	8
Variable	RHOEA	RHOFA	RHOCCA	DFSA	CONDA	MWA	CMREF	
Type	F	F	F	F	F	F	F	
Default	none	optional	optional	none	none	none	0.0	

VARIABLE	DESCRIPTION
RHOEA	Density of the active material (electrode particles), $\rho_{s,a}$. Units: kg/m^3.
RHOFA	Density of the inert filler $\rho_{f,a}$ in the anode (units: kg/m^3). This is used for the cell mass calculation if ITYPE = 1 on *BATTERY_ECHEM_CONTROL_THERMAL unless CMASS on *BATTERY_ECHEM_MAT_THERMAL is nonzero.
RHOCCA	Density of the positive current collector, ρ_{CC^+} (units: kg/m^3). This is used for the cell mass calculation if ITYPE = 1 on *BATTERY_ECHEM_CONTROL_THERMAL unless CMASS on *BATTERY_ECHEM_MAT_THERMAL is nonzero.
DFSA	Diffusion coefficient, $D_{s,a}$, of lithium ions in the anode electrode material (units: m^2/s)
CONDA	Effective electronic conductivity, $\sigma_{s,a}$, of the anode porous electrode (units: S/m)
MWA	Molecular weight, $M_{w,a}$, of the anode electrode (units: kg/mol). This field is only used in the 10- and 15-equation models for the porosity evolution equation. See Remark 1 of *BATTERY_ECHEM_AGING.
CMREF	Reference concentration value (units: mol/m^3). This value is required when IATYPE = 1.

Card 3	1	2	3	4	5	6	7	8
Variable	VFEA	VFPA	VFFA	VFGA	BRUGA	B1STAR	LPORE	
Type	F	F	F	F	F	F	F	
Default	none	0.	0.	none	1.5	0.	0.	

VARIABLE	DESCRIPTION
VFEA	Volume fraction of electrolyte in the anode electrode, $\varepsilon_{e,a}$
VFPA	Volume fraction of the polymer phase in the anode electrode, $\varepsilon_{p,a}$
VFFA	Volume fraction of the inert filler in the anode electrode, $\varepsilon_{f,a}$
VFGA	Volume fraction of the gas in the anode electrode, $\varepsilon_{g,a}$. This value is only used in the 15-equation model.
BRUGA	Bruggeman coefficient, p, in the anode
B1STAR	Thiele closure coefficient, B_1^* (see Remark 1): EQ.0.0: The Thiele-modulus correction is not used.
LPORE	Optional pore length scale, l_γ (units: m). It is used when B1STAR defined. See Remark 1. EQ.0.0: $l_\gamma = \varepsilon_{e,a}/a_s$

This card is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	IBVTYPE	ALFAA	ALFAC	IOFUNC				
Type	I	F	F	I				
Default	0	0.5	0.5	0				

VARIABLE	DESCRIPTION
IBVTYPE	Butler–Volmer equation formulation (see Remark 2): EQ.0: Symmetric EQ.1: Asymmetric EQ.2: Linearized EQ.3: Inverse formulation
ALFAA/ ALFAC	Butler–Volmer equation alpha parameters α_a and α_c. See Remark 2.
I0FUNC	ID of a *DEFINE_FUNCTION that replaces the built-in exchange current density in the anode (see IELTYPE on *BATTERY_ECHEM_MAT_ELECTROLYTE). This field is optional. See Remark 3.

This card is optional.

Card 5	1	2	3	4	5	6	7	8
Variable	EARK	EARDS	EARF					
Type	F	F	F					
Default	0.	0.	0.					

VARIABLE	DESCRIPTION
EARK	Activation energy for the reaction rate constant (k_a), $E_{a,k_a}/R_g$ (units: K). See Remark 7.
EARDS	Activation energy for the diffusion coefficient of lithium ions ($D_{s,a}$), $E_{a,D_{s,a}}/R_g$ (units: K). See Remark 7.
EARF	Activation energy for film resistance (R_{film}), $E_{a,R_{film}}/R_g$ (units: K). See Remark 7.

This card is optional.

Card 6	1	2	3	4	5	6	7	8
Variable	S0A	S100A	PMV					
Type	F	F	F					
Default	0.	0.	0.					

VARIABLE	DESCRIPTION
S0A/S100A	Stoichiometric coefficient at 0% and 100% state of charge.These fields are not used by the solve, but they are useful for tracking the evolution of the state of charge.
PMV	Partial molar volume, Ω_a (units: m^3/mol). This field is only used for swelling; see ISWELL on *EM_CONTROL_COUPLING_P2D.

Remarks:

1. **Thiele-modulus correction.** Defining a nonzero value for B1STAR activates a pore-scale Thiele-modulus correction (Lenne et al [2024]) that scales the effective diffusivity, D_e^{eff}, and the conductivity, κ^{eff}, in the anode (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER). This correction factor accounts for sub-pore concentration non-uniformity at high C-rates. It is given by:

$$1 + B_1^* \varphi_t^2 \ ,$$

where φ_t is the local Thiele modulus value:

$$\varphi_t^2 = \frac{l_\gamma^2 a_s (1 - t_+^0)|j_n|}{F c_e D_e} \ .$$

In the above, l_γ is the characteristic pore length, $a_s = 3\varepsilon_{am,a}/R_{s,a}$ (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER), t_+^0 is the cation transference number (see *BATTERY_ECHEM_MAT_ELECTROLYTE), j_n is the pore wall molar flux between the solid and electrolyte (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER), F is Faraday's constant, c_e is the electrolyte potential (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER), and D_e is the diffusion coefficient (see *BATTERY_ECHEM_MAT_ELECTROLYTE).

2. **Butler–Volmer kinetics.** The Butler–Volmer equation is the key relation that connects electrochemistry to current flow. It simulates how fast lithium is

inserted/extracted from the electrode as a function of voltage (overpotential) and illustrates the competition between forward (lithium insertion) and backward (lithium extraction) electrochemical reactions. This relation solves for the pore-wall flux, j_n. Note that the anode and cathode can have different formulations of this relation.

We have implemented four different formulations for the Butler-Volmer equation specified with IBVTYPE:

a) *Symmetric (IBVTYPE = 0).*

$$j_n = \frac{i_0}{F}\frac{e^{-0.5 f_{RT}\eta} - e^{0.5 f_{RT}\eta}}{1 + \frac{c_0}{c_e}e^{-0.5 f_{RT}\eta}}$$

b) *Asymmetric (IBVTYPE = 1).*

$$j_n = \frac{i_0}{F}\frac{e^{-\alpha_a f_{RT}\eta} - e^{\alpha_c f_{RT}\eta}}{1 + \frac{c_0}{c_e}e^{-\alpha_c f_{RT}\eta}}$$

c) *Linearized simple model (IBVTYPE = 2).*

$$j_n = \frac{i_0}{F}(\alpha_a + \alpha_c) f_{RT}\eta$$

d) *Inverse (asinh) formulation (IBVTYPE = 3).*

$$\eta_{inv} = \frac{2}{(\alpha_a + \alpha_c) f_{RT}} \operatorname{asinh}\frac{j_n}{2 i_0}$$

In these models, α_a and α_c are the anodic and cathodic charge transfer coefficients input with ALFAA and ALFAC, η is the overpotential (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER), f_{RT} is a derived Faraday ratio given by $F/(R_g T)$, i_0 is the exchange current density (see IELETYPE on *BATTERY_ECHEM_MAT_ELECTROLYTE and I0FUNC), and c_0 is a regularization parameter such that

$$c_0 = \begin{cases} 80 & \text{for } T \le T_{ref} \\ 1 & \text{for } T > T_{ref} \end{cases}$$

T_{ref} is hard-coded to 298.15 K.

Among the pore wall flux solving approaches, the linearized model is a simplified approach that converges easily but is only valid for $|f_{RT}\eta| \ll 1$. The key advantage of the inverse formulation is that the derivative $\frac{\partial \eta_{inv}}{\partial j_n}$ is bounded and decays for large $|j_n|$, yielding a well-conditioned Jacobian even at high overpotentials.

3. **I0FUNC.** If I0FUNC is nonzero, then it refers to a *DEFINE_FUNCTION that replaces the built-in exchange current density formula in the anode. The

function needs to return the exchange current as well as the derivatives with respect to the electrolyte and solid surface concentration. The function has six arguments: f(ce, cs, csm, temp, rk, step). Here, ce refers to the Li^+ concentration in the electrolyte (c_e), cs refers to the Li^+ concentration on the surfaces of the solid electrode particles ($c_{s,surf}$), csm refers to the maximum solid lithium concentration in the anode ($c_{s,max}$), temp is the temperature, and rk is the reaction rate constant (k_a). The step argument determines what is being returned by the function:

a) *step* = *1*. The function returns i_0 (units: A/m^2).

b) *step* = *2*. The function returns $\frac{\partial i_0}{\partial c_e}$ (units: $A \cdot m/mol$).

c) *step* = *3*. The function returns $\frac{\partial i_0}{\partial c_s}$ (units: $A \cdot m/mol$).

Thus, the *DEFINE_FUNCTION must include three if conditions for the different values of step.

4. **Maximum solid phase lithium concentration.** The maximum solid phase lithium concentration, $c_{s,max}$, for the anode is given by:

$$c_{s,max} = \frac{3600}{F} c_{sp,a} \rho_{s,a}$$

where $c_{sp,a}$ is input parameter CAPTA and $\rho_{s,a}$ is input parameter RHOEA.

5. **Open-circuit potential models.** IOCPA selects the model for the open-circuit potential, U_{ocp}, for the anode. This potential is used to find η in the Butler-Volmer relation (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER). U_{ocp} relates the equilibrium electrode voltage to the stochiometry, s, where $s = c_s/c_{s,max}$. It depends on s and temperature (T). These models must provide U_{ocp}, $\frac{\partial U_{ocp}}{\partial s}$, and $\frac{\partial U_{ocp}}{\partial T}$.

IOCPA = 1, 2, 3, 4, 5, and 7 are material-specific correlations fitted to experimental data. These fits are valid over specific stoichiometry ranges. Extrapolation outside those ranges may produce unphysical values.

IOCPA = 10 is the Nernst ideal intercalation (Newman and Balsara [2021]). It is a thermodynamic model for generic testing:

$$U_{ocp} = U_0 + \frac{RT_{ref}}{F} \ln\left(\frac{1-s}{s}\right)$$

$$\frac{dU_{ocp}}{ds} = -\frac{RT_{ref}}{Fs(1-s)}$$

T_{ref} is hard-coded to 298.15 K.

To specify the open-circuit potential with a *DEFINE_FUNCTION, set IOCPA to a negative number where |IOCPA| is a *DEFINE_FUNCTION ID. The function needs to return U_{ocp}, $\frac{\partial U_{ocp}}{\partial s}$, and $\frac{\partial U_{\mathrm{ocp}}}{\partial T}$. It has four arguments: f(sto, csm, temp, step). Here, sto refers to s, csm means $c_{\mathrm{s,max}}$, and temp is the temperature. The step argument determines what is being returned by the function:

a) *step* = *1*. The function returns U_{ocp}.

b) *step* = *2*. The function returns $\frac{\partial U_{\mathrm{ocp}}}{\partial s}$.

c) *step* = *3*. The function returns $\frac{\partial U_{\mathrm{ocp}}}{\partial T}$.

6. **Solid-state batteries.** Solid-state batteries typically pair a lithium metal anode (invoked with IATYPE = 1) with a solid (polymer or ceramic) electrolyte and a standard intercalation cathode. The lithium metal anode differs from a porous graphite/silicon electrode in three fundamental ways:

   a) *No porosity*. The foil is a dense, non-porous surface. The anode region in the P2D domain effectively collapses to a single boundary layer.

   b) *No solid diffusion*. There is no intercalation host lattice. Hence, no radial problem based on Fick's law to solve on the anode: $c_{\mathrm{s}} = c_{\mathrm{s,max}}$.

   c) *Zero open-circuit potential*. By definition, the potential of pure lithium metal is reference, meaning $U_{\mathrm{ocp}} = 0$ and $\frac{\partial U_{\mathrm{ocp}}}{\partial T} = 0$. Thus, the IOCPA is ignored.

   These simplifications remove the solid-concentration unknown from the anode nodes and reduce the exchange current density to a function of electrolyte concentration only. The exchange current density becomes for a liquid electrolyte (IELTYPE = 0 on *BATTERY_ECHEM_MAT_ELECTROLYTE):

$$i_0 = F k_{\mathrm{a}} c_{\mathrm{ref}} \sqrt{c_{\mathrm{e}}}$$

   and for a polymer electrolyte (IELTYPE = 1 on *BATTERY_ECHEM_MAT_ELECTROLYTE):

$$i_0 = F k_{\mathrm{a}} c_{\mathrm{ref}} \sqrt{c_{\mathrm{e}}} \sqrt{\frac{c_{\mathrm{e,max}} - c_{\mathrm{e}}}{c_{\mathrm{e,max}}}}$$

   In the above, c_{ref} is a reference metal concentration specified with CMREF.

7. **Arrhenius temperature corrections.** Activation energies EARK, EARDS, and EARF are only used when IARR = 1 in *BATTERY_ECHEM_CONTROL_THERMAL.

*BATTERY_ECHEM_MAT_CATHODE

Purpose: Set the battery material parameters for the positive electrode (cathode).

Card Summary:

Card 1. This card is required.

CELLID	IOCPC	CAPTC	SOCC	RADC	RATEC	RCATHDE	

Card 2. This card is required.

RHOEA	RHOFA	RHOCCA	DFSA	CONDA			

Card 3. This card is required.

VFEA	VFPA	VFFA	VFGA	BRUGA	B1STAR	LPORE	

Card 4. This card is optional.

IBVTYPE	ALFAA	ALFAC	IOFUNC				

Card 5. This card is optional.

EARK	EARDS	EARF					

Card 6. This card is optional.

S0A	S100A	PMV					

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	CELLID	IOCPC	CAPTC	SOCC	RADC	RATEC	RCATHDE	
Type	I	I	F	F	F	F	F	
Default	optional	none	none	none	none	none	0.0	

VARIABLE	DESCRIPTION
CELLID	Optional cell ID. See Remark 6 of *BATTERY_ECHEM_CONTROL_SOLVER.
IOCPC	Material type for the open-circuit potential in the cathode electrode (see Remark 5): LT.0: Absolute value points to a *DEFINE_FUNCTION ID. EQ.1: Mn_2O_4 (lower plateau) ($1.1 < s < 1.99$ where s is the stoichiometry). See Doyle et al [1996]. EQ.2: Mn_2O_4 (upper plateau) ($0.17 < s < 0.99$). See Doyle et al [1996]. EQ.3: Cobalt dioxide 1, Li_yCoO_2 ($0.0 < s < 0.99$) EQ.4: Cobalt dioxide 2, Li_yCoO_2 ($0.0 < s < 0.99$). See Doyle [1995]. EQ.5: Mn_2O_4 ($0.17 < s < 0.99$). See Cai and White [2009]. EQ.6: NMC-111. See Stewart el al [2008]. EQ.7: MC-811 EQ.8: $Ni_{0.8}Co_{0.2}O_2$ EQ.9: $LiFePO_4$. See Safari and Delacouty [2011]. EQ.10: Nernst ideal intercalation. See Newman and Balsara [2021]. EQ.12: $NiO(OH)/Ni(OH)_2$. See Albertus el al [2009]. EQ.13: $LiCoO_2$. See Cai and White [2009].
CAPTC	Coulombic capacity of the cathode material, $c_{sp,c}$ (units: Ah/kg)
SOCC	Initial lithium stoichiometric coefficient for the cathode side active material. For example, Li_yWO_3 ($0 < y < 0.67$).
RADC	Radius of spherical particles, $R_{s,c}$, in the cathode side active material (units: m)
RATEC	Reaction rate constant, k_c, for the cathode electrode (units: $m^{2.5}mol^{-0.5}s^{-1}$)
RCATHDE	Film resistance for the cathode electrode, R_{film} (units: $\Omega \cdot m^2$). This field can be left as 0. See Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER.

Card 2	1	2	3	4	5	6	7	8
Variable	RHOEC	RHOFC	RHOCCC	DFSC	CONDC			
Type	F	F	F	F	F			
Default	none	optional	optional	none	none			

VARIABLE	DESCRIPTION
RHOEC	Density of active material (electrode particles), $\rho_{s,c}$. Units: kg/m^3.
RHOFC	Density of the inert filler $\rho_{f,c}$ in the cathode (units: kg/m^3). This is used for the cell mass calculation if ITYPE = 1 on *BATTERY_ECHEM_CONTROL_THERMAL unless CMASS on *BATTERY_ECHEM_MAT_THERMAL is nonzero.
RHOCCC	Density of the negative current collector, ρ_{CC^-} (units: kg/m^3). This is used for the cell mass calculation if ITYPE = 1 on *BATTERY_ECHEM_CONTROL_THERMAL unless CMASS on *BATTERY_ECHEM_MAT_THERMAL is nonzero.
DFSC	Diffusion coefficient, $D_{s,c}$, of lithium ions in the cathode electrode material (units: m^2/s)
CONDC	Effective electronic conductivity, $\sigma_{s,c}$, of the cathode porous electrode (units: S/m)

Card 3	1	2	3	4	5	6	7	8
Variable	VFEC	VFPC	VFFC	VFGC	BRUGC	B1STAR	LPORE	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	1.5	0.	0.	

VARIABLE	DESCRIPTION
VFEC	Volume fraction of electrolyte in the cathode electrode, $\varepsilon_{e,c}$
VFPC	Volume fraction of the polymer phase in the cathode electrode, $\varepsilon_{p,c}$

VARIABLE	DESCRIPTION
VFFC	Volume fraction of the inert filler in the cathode electrode, $\varepsilon_{c,a}$
VFGC	Volume fraction of the gas in the cathode electrode, $\varepsilon_{g,c}$. This value is only used in the 15-equation model.
BRUGC	Bruggeman coefficient, p, in the cathode
B1STAR	Thiele closure coefficient, B_1^* (see Remark 1): EQ.0.0: The Thiele-modulus correction is not used.
LPORE	Optional pore length scale, l_γ (units: m). It is used when B1STAR defined. See Remark 1. EQ.0.0: $l_\gamma = \varepsilon_{e,c}/a_s$

This card is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	IBVTYPE	ALFAA	ALFAC	I0FUNC				
Type	I	F	F	I				
Default	0	0.5	0.5	0				

VARIABLE	DESCRIPTION
IBVTYPE	Butler–Volmer equation formulation (see Remark 2): EQ.0: Symmetric EQ.1: Asymmetric EQ.2: Linearized EQ.3: Inverse formulation
ALFAA/ ALFAC	Butler–Volmer equation alpha parameters α_a and α_c.
I0FUNC	ID of a *DEFINE_FUNCTION that replaces the built-in exchange current density in the cathode (see IELTYPE on *BATTERY_ECHEM_MAT_ELECTROLYTE). This field is optional. See Remark 3.

This card is optional.

Card 5	1	2	3	4	5	6	7	8
Variable	ECRK	ECRDS	ECRF					
Type	F	F	F					
Default	0.	0.	0.					

VARIABLE	DESCRIPTION
ECRK	Activation energy for the reaction rate constant (k_c), $E_{a,k_c}/R_g$ (units: K). See Remark 6.
ECRDS	Activation energy for the diffusion coefficient of lithium ions ($D_{s,c}$), $E_{a,D_{s,c}}/R_g$ (units: K). See Remark 6.
ECRF	Activation energy for film resistance (R_{film}), $E_{a,R_{film}}/R_g$ (units: K). See Remark 6.

This card is optional.

Card 6	1	2	3	4	5	6	7	8
Variable	S0C	S100C	PMV					
Type	F	F	F					
Default	0.	0.	0.					

VARIABLE	DESCRIPTION
S0C/S100C	Stoichiometric coefficient at 0% and 100% state of charge.These fields are not used by the solve, but they are useful for tracking the evolution of the state of charge.
PMV	Partial molar volume, Ω_a (units: m^3/mol). This field is only used for swelling; see ISWELL on *EM_CONTROL_COUPLING_P2D.

Remarks:

1. **Thiele-modulus correction.** Defining a nonzero value for B1STAR activates a pore-scale Thiele-modulus correction (Lenne et al [2024]) that scales the effective diffusivity, D_e^{eff}, and the conductivity, κ^{eff}, in the cathode (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER). This correction factor accounts for sub-pore concentration non-uniformity at high C-rates. It is given by:

$$1 + B_1^* \varphi_t^2 \ ,$$

where φ_t is the local Thiele modulus value:

$$\varphi_t^2 = \frac{l_\gamma^2 a_s (1 - t_+^0)|j_n|}{F c_e D_e} \ .$$

In the above, l_γ is the characteristic pore length, $a_s = 3\varepsilon_{am,c}/R_{s,c}$ (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER), t_+^0 is the cation transference number (see *BATTERY_ECHEM_MAT_ELECTROLYTE), j_n is the pore wall molar flux between the solid and electrolyte (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER), F is Faraday's constant, c_e is the electrolyte potential (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER), and D_e is the diffusion coefficient (see *BATTERY_ECHEM_MAT_ELECTROLYTE).

2. **Butler–Volmer kinetics.** The Butler–Volmer equation is the key relation that connects electrochemistry to current flow. It simulates how fast lithium is inserted/extracted from the electrode as a function of voltage (overpotential) and illustrates the competition between forward (lithium insertion) and backward (lithium extraction) electrochemical reactions. This relation solves for the pore-wall flux, j_n. Note that the anode and cathode can have different formulations of this relation.

   We have implemented four different formulations for the Butler-Volmer equation specified with IBVTYPE:

   a) *Symmetric (IBVTYPE = 0).*

$$j_n = \frac{i_0}{F} \frac{e^{-0.5 f_{RT} \eta} - e^{0.5 f_{RT} \eta}}{1 + \frac{c_0}{c_e} e^{-0.5 f_{RT} \eta}}$$

   b) *Asymmetric (IBVTYPE = 1).*

$$j_n = \frac{i_0}{F} \frac{e^{-\alpha_a f_{RT} \eta} - e^{\alpha_c f_{RT} \eta}}{1 + \frac{c_0}{c_e} e^{-\alpha_c f_{RT} \eta}}$$

   c) *Linearized simple model (IBVTYPE = 2).*

$$j_n = \frac{i_0}{F} (\alpha_a + \alpha_c) f_{RT} \eta$$

d) *Inverse (asinh) formulation (IBVTYPE = 3).*

$$\eta_{\text{inv}} = \frac{2}{(\alpha_a + \alpha_c) f_{RT}} \operatorname{asinh} \frac{j_n}{2 i_0}$$

In these models, α_a and α_c are the anodic and cathodic charge transfer coefficients input with ALFAA and ALFAC, η is the overpotential (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER), f_{RT} is a derived Faraday ratio given by $F/(RT)$, i_0 is the exchange current density (see IELETYPE on *BATTERY_ECHEM_MAT_ELECTROLYTE and I0FUNC), and c_0 is a regularization parameter such that

$$c_0 = \begin{cases} 80 & \text{for } T \le T_{\text{ref}} \\ 1 & \text{for } T > T_{\text{ref}} \end{cases}$$

T_{ref} is hard-coded to 298.15 K.

Among the pore wall flux solving approaches, the linearized model is a simplified approach that converges easily but is only valid for $|f_{RT}\eta| \ll 1$. The key advantage of the inverse formulation is that the derivative $\frac{\partial \eta_{\text{inv}}}{\partial j_n}$ is bounded and decays for large $|j_n|$, yielding a well-conditioned Jacobian even at high overpotentials.

3. **I0FUNC.** If I0FUNC is nonzero, then it refers to a *DEFINE_FUNCTION that replaces the built-in exchange current density formula in the cathode. The function needs to return the exchange current as well as the derivatives with respect to the electrolyte and solid surface concentration. The function has six arguments: f(ce, cs, csm, temp, rk, step). Here, ce refers to the Li^+ concentration in the electrolyte (c_e), cs refers to the Li^+ concentration on the surfaces of the solid electrode particles ($c_{s,surf}$), csm refers to the maximum solid lithium concentration in the cathode ($c_{s,max}$), temp is the temperature, and rk is the reaction rate constant (k_c). The step argument determines what is being returned by the function:

   a) *step = 1*. The function returns i_0 (units: A/m^2).

   b) *step = 2*. The function returns $\frac{\partial i_0}{\partial c_e}$ (units: $A \cdot m/mol$).

   c) *step = 3*. The function returns $\frac{\partial i_0}{\partial c_s}$ (units: $A \cdot m/mol$).

   Thus, the *DEFINE_FUNCTION must include three if conditions for the different values of step.

4. **Maximum solid phase lithium concentration.** The maximum solid phase lithium concentration, $c_{s,max}$, for the cathode is given by:

$$c_{s,max} = \frac{3600}{F} c_{sp,c} \rho_{s,c}$$

where $c_{sp,c}$ is input parameter CAPTC and $\rho_{s,c}$ is input parameter RHOEC.

5. **Open-circuit potential models.** IOCPC selects the model for the open-circuit potential, U_{ocp}, for the cathode. This potential is used to find η in the Butler-Volmer relation (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER). U_{ocp} relates the equilibrium electrode voltage to the stochiometry, s, where $s = c_s/c_{s,max}$. It depends on s and temperature (T). These models must provide U_{ocp}, $\frac{\partial U_{ocp}}{\partial s}$, and $\frac{\partial U_{ocp}}{\partial T}$.

   IOCPC = 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, and 13 are material-specific correlations fitted to experimental data. These fits are valid over specific stoichiometry ranges. Extrapolation outside those ranges may produce unphysical values.

   IOCPC = 10 is the Nernst ideal intercalation (Newman and Balsara [2021]). It is a thermodynamic model for generic testing:

$$U_{ocp} = U_0 + \frac{RT_{ref}}{F} \ln\left(\frac{1-s}{s}\right)$$
$$\frac{dU_{ocp}}{ds} = -\frac{RT_{ref}}{Fs(1-s)}$$

   T_{ref} is hard-coded to 298.15 K.

   To specify the open-circuit potential with a *DEFINE_FUNCTION, set IOCPC to a negative number where |IOCPA| is a *DEFINE_FUNCTION ID. The function needs to return U_{ocp}, $\frac{\partial U_{ocp}}{\partial s}$, and $\frac{\partial U_{ocp}}{\partial T}$. It has four arguments: f(sto, csm, temp, step). Here, sto refers to s, csm means $c_{s,max}$, and temp is the temperature. The step argument determines what is being returned by the function:

   a) *step* = *1*. The function returns U_{ocp}.

   b) *step* = *2*. The function returns $\frac{\partial U_{ocp}}{\partial s}$.

   c) *step* = *3*. The function returns $\frac{\partial U_{ocp}}{\partial T}$.

6. **Arrhenius temperature corrections.** Activation energies ECRK, ECRDS, and ECRF are only used when IARR = 1 in *BATTERY_ECHEM_CONTROL_THERMAL.

*BATTERY_ECHEM_MAT_ELECTROLYTE

Purpose: Set the battery material variables for the electrolyte and separator.

Card 1	1	2	3	4	5	6	7	8
Variable	CELLID	IOCPE	IELTYPE	RHOEL	RHOP	RHOS		
Type	I	I	I	F	F	F		
Default	optional	none	0	none	none	none		

Card 2	1	2	3	4	5	6	7	8
Variable	VFES	VFPS	VFGS	LICE	BRUGE	CLMAX		
Type	F	F	F	F	F	F		
Default	none	none	none	none	1.5	0.		

This card is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	EADE	EAKA	TFACT					
Type	F	F	F					
Default	0.	0.	2.					

VARIABLE	DESCRIPTION
CELLID	Optional cell ID. See Remark 6 of *BATTERY_ECHEM_CONTROL_SOLVER.

VARIABLE	DESCRIPTION
IOCPE	Electrolyte transport model: LT.0: Absolute value points to a *DEFINE_FUNCTION ID. See Remark 1. EQ.1: $LiPF_6$ in EC:DMC (1:1). See Doyle [1995]. EQ.2: $LiPF_6$ in EC:DMC (2:1). See Doyle et al [1996]. EQ.3: $LiPF_6$ in EC:DMC (1:2). See Doyle et al [1996]. EQ.4: $LiPF_6$ in PC EQ.5: $LiClO_4$ in PC. See Fuller et al [1994]. EQ.10: Simplified model EQ.13: 30% KOH in H_2O (NiMH system). See Paxton and Newman [1997]. Here, EC is ethylene carbonate, DMC is dimethyl carbonate, and PC is propylene carbonate.
IELTYPE	Type of electrolyte (see Remark 2): EQ.0: Liquid electrolyte EQ.1: Polymer electrolyte
RHOEL	Density of the electrolyte, ρ_e (units: kg/m^3)
RHOP	Density of the polymer phase, ρ_p (units: kg/m^3)
RHOS	Density of the separator material, ρ_s (units: kg/m^3)
VFES	Volume fraction of electrolyte in the separator, $\varepsilon_{e,s}$
VFPS	Volume fraction of the polymer phase in the separator, $\varepsilon_{p,s}$
VFGS	Volume fraction of the gas in the separator, $\varepsilon_{g,s}$. This value is only used in the 15-equation model.
LICE	Initial concentration of electroly Li^+, $c_{e,0}$ (units: mol/m^3)
BRUGE	Bruggeman coefficient in the separator, p
CLMAX	Maximum electrolyte salt concentration, $c_{e,max}$ (units: mol/m^3). This field is required when IELTYPE = 1 (see Remark 2).
EADE	Activation energy for electrolyte diffusion (D_e), $E_{a,D_e}/R_g$ (units: K). See Remark 3.

VARIABLE	DESCRIPTION
EAKA	Activation energy for electrolyte conduction (κ), $E_{a,\kappa}/R_g$ (units: K). See Remark 3.
TFACT	Thermodynamic factor, SF_T. See Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER.

Remarks:

1. **Electrolyte transport model.** IOCPE selects the correlations for the four electrolyte transport properties: D_e, κ, t_+^0, and $\frac{\partial \ln f_\pm}{\partial c_e}$. See Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER. These properties are all functions of c_e (Li^+ concentration in the electrolyte) and T (temperature). Their derivatives with respect to c_e are also required for the Jacobian, which is important for solving the coupled equation system.

   For IOCPE = 1, 2, 3, 4, 5, and 13, material-specific fits from the literature are used to determine these quantities.

   Alternatively, setting IOCPE to 10 invokes a simplified model with no concentration dependence:

$$\begin{aligned} D_e &= 2.5e^{-10}\ \mathrm{m^2 s^{-1}} \\ \kappa &= 1.0\ \mathrm{S/m} \\ t_+^0 &= 0.4 \\ \frac{\partial \ln f_\pm}{\partial c_e} &= 0. \end{aligned}$$

   All derivatives with respect to c_e are zero, eliminating the off-diagonal Jacobian coupling between c_e and the potential equations through the transport properties. This substantially reduces stiffness and is recommended for initial testing of new cell configurations.

   Lastly, you can specify these properties and their derivatives using a *DEFINE_FUNCTION by setting IOCPE to a negative value. |IOCPE| refers to the ID of the *DEFINE_FUNCTION. The function needs to return the four transport properties along with their derivatives with repsect to c_e. The function has three arguments: f(eps, ce, step). Here, eps refers to the volume fraction of the electrolyte in the specific region ($\varepsilon_{e,r}$ with r referrring to the region) and ce is c_e. The step argument determines what is being returned by the function:

   a) *step = 1*. The function returns D_e.

   b) *step = 2*. The function returns $\frac{\partial D_e}{\partial c_e}$.

c) *step* = 3. The function returns κ

d) *step* = 4. The function returns $\frac{\partial \kappa}{\partial c_e}$.

e) *step* = 5. The function returns t_+^0.

f) *step* = 6. The function returns $\frac{\partial t_+^0}{\partial c_e}$.

g) *step* = 7. The function returns $\frac{\partial \ln f_\pm}{\partial c_e}$.

h) *step* = 8. The function returns $\frac{\partial^2 \ln f_\pm}{\partial c_e^2}$.

Thus, the *DEFINE_FUNCTION must include eight if conditions for the different values of step.

2. **Exchange current density.** The Butler-Volmer relation (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER, Remark 2 of *BATTERY_ECHEM_MAT_ANODE, and Remark 2 of *BATTERY_ECHEM_MAT_CATHODE) depends on the exchange current density, i_0. IELTYPE specifies the model for the exchange current density in both the anode and cathode. However, the model can be overridden individually for the anode and cathode by specifying I0FUNC on their respective material cards. For instance, specifying I0FUNC as as a nonzero value for *BATTERY_ECHEM_MAT_ANODE but not for *BATTERY_ECHEM_MAT_CATHODE causes a *DEFINE_FUNCTION to be used for the anode and the built-in model dictated by IELTYPE to be used for the cathode.

For a liquid electrolyte (IELTYPE = 0), the following model is used for i_0:

$$i_0 = Fk_r\sqrt{c_e}\,\sqrt{c_{s,max} - c_{s,surf}}\sqrt{c_{s,surf}}$$

where k_r is the reaction rate constant in region r with r as either a for anode or c for cathode (RATEA on *BATTERY_ECHEM_MAT_ANODE and RATEC on *BATTERY_ECHEM_MAT_CATHODE), c_e is the Li^+ concentration in the electrolyte (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER), $c_{s,surf}$ is the Li^+ concentration on the surfaces of the solid electrode particles (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER), and $c_{s,max}$ maximum solid lithium concentration in the domain (see Remark 4 of *BATTERY_ECHEM_MAT_ANODE and Remark 4 of *BATTERY_ECHEM_MAT_CATHODE.

For a polymer electrolyte (IELTYPE = 1), the following model is used for i_0:

$$i_0 = Fk_r\sqrt{c_e}\,\sqrt{c_{s,max} - c_{s,surf}}\sqrt{c_{s,surf}}\sqrt{\frac{c_{e,max} - c_e}{c_{e,max}}}$$

where $c_{e,max}$ is given by CLMAX in the input.

3. **Arrhenius temperature corrections.** The activation energies on the electrolyte transport properties D_e and κ are only used when IARR = 1 in *BATTERY_ECHEM_CONTROL_THERMAL.

*BATTERY_ECHEM_MAT_THERMAL

Purpose: Provide per cell thermal material properties and boundary conditions for a lumped thermal solve (ITYPE = 1 in *BATTERY_ECHEM_CONTROL_THERMAL). For a standalone P2D model with constant temperature (ITYPE = 0), this keyword sets the initial temperature or how the temperature evolves with time.

Card 1	1	2	3	4	5	6	7	8
Variable	CELLID	CP	HTC	T0	TAMB	CMASS	EMISS	
Type	I	F	F	F	F	F	F	
Default	optional	0.	0.	0.	0.	0.	0.	

VARIABLE	DESCRIPTION
CELLID	Optional cell ID. See Remark 6 of *BATTERY_ECHEM_CONTROL_SOLVER.
CP	Cell heat capacity, C_p (units:J/(kg·K))
HTC	Heat transfer coefficient, h (units: W/(m²·K). See Remark 1.
T0	Initial temperature, T_0 (units: K). See Remark 3.
TAMB	Ambient temperature, T_{amb}. (units: K). See Remark 3.
CMASS	Mass per area of the cell, m_t (units: kg/m²). See Remark 2.
EMISS	Surface emission coefficient ε_{rad}. See Remark 1.

Remarks:

1. **External boundary conditions.** In the coupled EM-P2D mode, it is better to let the thermal solver handle convection and radiation with the exterior. To do this, leave HTC and EMISS as 0.

2. **Cell mass per area.** When CMASS= 0, the cell mass per unit area is autocomputed from electrode densities, volume fractions, and layer thicknesses. It may be a realistic representation of the jellyroll mass but might miss some material mass present in the cell that is critical for heating analysis.

3. **Prescribed temperature.** If ITYPE = 0 in *BATTERY_ECHEM_CONTROL_THERMAL for a standalone model and a negative value is entered for T0, then |T0| is a load curve ID (*DEFINE_CURVE) that gives how the temperature evolves as a function of time. This temperature is used for all temperature-dependent quantities. For ITYPE = 1, the ambient temperature can evolve with time by setting TAMB to a negative number where |TAMB| is a load curve ID.

*BATTERY_ECHEM_THERMAL_ABUSE

Purpose: Turn on thermal runaway models. This feature can be used for the standalone model with ITYPE = 1 on *BATTERY_ECHEM_CONTROL_THERMAL or for either value of ITYPE when coupled with the thermal solver.

Card Summary:

Card 1. This card is required.

IMODEL	IFEEDBK	TSEI0	ITSEIR	SF5	IGASGEN	IKIN	

Card 2. Include one instance of this card for IMODEL = 1, four instances of this card for IMODEL = 4, and five instances of this card for IMODEL = 5.

A*i*	EAR*i*	N*i*	HW*i*	TON*i*	M*i*	P*i*	ALFA0*i*

Card 3. Include this card if IFEEDBK = 1.

PKAP	PDSA	PDSC	PIOA	PIOC	PRFA	PRFC	PSIG

Card 4. Include this card if IFEEDBK = 1.

PEPS1	PEPS2						

Card 5. If IGASGEN = 1, include one instance of this card for IMODEL = 1, four instances of this card for IMODEL = 4, and five instances of this card for IMODEL = 5.

OMC2H4*i*	OMO2*i*	OMH20*i*	OMCO2*i*	OMH2*i*			

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	IMODEL	IFEEDBK	TSEI0	ITSEIR	SF5	IGASGEN	IKIN	
Type	I	I	F	I	F	I	I	
Default	0	0	0.	0	0.	0	0	

VARIABLE	DESCRIPTION
MODEL	Flag to select thermal runaway model (see Remark 3): EQ.0: Off EQ.1: One-equation model EQ.4: Four-equation model EQ.5: Five-equation model
IFEEDBK	Feedback model: EQ.0: Off EQ.1: On. See Remark 6.
TSEI0	Initial SEI reference thickness, z_0. If nonzero, an optional attenuation multiplier term is included. See Remarks 3 and 4.
ITSEIR	SEI thickness coupling (see Remark 4): EQ.0: Tunneling regime EQ.1: Diffusion regime EQ.2: Auto regime
SF5	Optional scaling coefficient on reaction 5 (for IMODEL = 5 only), SF_5
IGASGEN	Thermal abuse gas generation mole tracking model: EQ.0: Off EQ.1: On. See Remark 7.
IKIN	Flag to determine if reaction 5 is modified (for IMODEL = 5 only): EQ.0: Unmodified formulation EQ.1: Modified based on Coman et al [2016]

Model Parameters Card. Include one instance of this card for IMODEL = 1, four instances of this card for IMODEL = 4, and five instances of this card for IMODEL = 5.

Card 2	1	2	3	4	5	6	7	8
Variable	A*i*	EAR*i*	N*i*	HW*i*	TON*i*	M*i*	P*i*	ALFA0*i*
Type	F	F	F	F	F	F	F	F
Default	0.	0.	1.0	0.	0.	0.	0.	1.0

VARIABLE	DESCRIPTION
A*i*	Frequency factor (units: s^{-1}) for the i^{th} equation, A_i. See Remarks 3 and 5. GE.0.0: Frequency factor for the i^{th} equation LT.0.0: |A*i*| refers to a *DEFINE_FUNCTION. See Remark 5.
EAR*i*	Activation energy (units: K) for the i^{th} equation, $\frac{E_{a_i}}{R_g}$.
N*i*	Reaction order for the i^{th} equation, n_i
HW*i*	Enthalpy times mass loading for the i^{th} equation, HW_i (units: J/m^3)
M*i*	Second reaction order for the i^{th} equation for the i^{th} equation, m_i
P*i*	Third reaction order for the i^{th} equation for the i^{th} equation, p_i
ALFA0*i*	Initial α_i

First Feedback Card. Include this card if FEEDBK = 1.

Card 3	1	2	3	4	5	6	7	8
Variable	PKAP	PDSA	PDSC	PIOA	PIOC	PRFA	PRFC	PSIG
Type	F	F	F	F	F	F	F	F
Default	0.	0.	0.	0.	0.	0.	0.	0.

Second Feedback Card. Include this card if FEEDBK = 1.

Card 4	1	2	3	4	5	6	7	8
Variable	PEPS1	PEPS2						
Type	F	F						
Default	0.	0.						

VARIABLE	DESCRIPTION
PKAP, …, PEPS2	Degradation parameters to be defined if IFEEDBK = 1. See Remark 6.

Gas Generation Cards. IF IGASGEN = 1, include one instance of this card for IMODEL = 1, four instances of this card for IMODEL = 4, and five instances of this card for IMODEL = 5.

Card 5	1	2	3	4	5	6	7	8
Variable	OMC2H4*i*	OMO2*i*	OMH2O*i*	OMCO2*i*	OMH2*i*			
Type	F	F	F	F	F			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
OMC2H4*i*, …, OMH2*i*	Gas yield values for the different species for the i^{th} reaction (units: mol/kg). These field are needed for tracking generated moles when IGASGEN = 1. See Remark 7.

Remarks:

1. **Thermal runaway and ITYPE = 0.** When ITYPE = 0 on *BATTERY_ECHEM_CONTROL_THERMAL with thermal coupling enabled, the temperature is constant during P2D subcycling with the temperature evolution handled by the thermal solver. In this case, the thermal runaway behavior may be missed if this keyword is used. Thus, we recommend using *LOAD_HEAT_EXOTHERMIC_REACTION instead for modeling thermal runaway to allow the thermal solver to handle the thermal runaway directly as the thermal solver drives the temperature change. When using *LOAD_HEAT_EXOTHERMIC_REACTION, we recommend using the nonlinear thermal solver as it is more capable of capturing the sharp temperature growth that can occur during thermal runaway without resorting to an unreasonably small thermal time step.

2. **Thermal runaway and ITYPE = 1.** When ITYPE = 1, the thermal runaway heat is used in the lumped temperature of the P2D solve to change the temperature. In coupled models, this is part of the micro thermal update.

3. **Thermal runaway modeling.** IMODEL = 0 invokes the one-equation thermal runaway model (MacNeil and Dahn [2001]) where a single reaction model tracks the decomposition of a reactive component (typically SEI) through a reaction progress variable ($0 \leq \alpha_1 \leq 1$) and the heat generated. The four-equation model (Kim et al [2007]), also called NREL's 4-equation model, is an extension of the the 1-equation model that tracks four reactions with their own respective variables and the heat generated from these reactions. IMODEL = 5 additionally includes a fifth reaction that was originally intended for modeling the electromagnetic heat as the state of charge is consumed.

   For all the models, the first or first through fourth reaction equations are given by:

$$\frac{d\alpha_i}{dt} = -A_i \alpha_i^{n_i} e^{-\frac{E_{a_i}}{R_g T}} (1-\alpha_i)^{m_i} (-\ln \alpha_i)^{p_i}$$

   where $i = 1$ for the 1-equation model and $i = 1, \ldots, 4$ for the four- and five-equation models. In the above, A_i, E_{a_i}/R_g, n_i, m_i, and p_i are input parameters. $\alpha_i = 1$ is the pristine state, and it decays toward 0 as the decomposition proceeds.

Note that for the four-equation and five-equation models, the equation for $\frac{\mathrm{d}\alpha_2}{\mathrm{d}t}$ can be modified to include multiplication of an optional attenuation term, g_{SEI}, on the right-hand side due to the growing SEI layer. This is activated by setting TSEI0 to a nonzero value (see Remark 4).

The five-equation model includes a fifth reaction progress variable with a similar format for the reaction rate to the other reaction equations but with an optional cross-coupling term. The original motivation is to include the electromagnetic heat as the state of charge is consumed. The version of this equation depends on IKIN. For IKIN = 0:

$$\frac{\mathrm{d}\alpha_5}{\mathrm{d}t} = -A_5{\alpha_5}^{n_5} e^{-\frac{E_{\mathrm{a}_5}}{R_{\mathrm{g}}T}} (1-\alpha_5)^{m_5} (-\ln\alpha_5)^{p_5} - \mathrm{SF}_5 \left(\frac{\mathrm{d}\alpha_2}{\mathrm{d}t} - \frac{\mathrm{d}\alpha_3}{\mathrm{d}t}\right) \alpha_5 \,,$$

while for IKIN = 1:

$$\frac{\mathrm{d}\alpha_5}{\mathrm{d}t} = -A_5{\alpha_2}^{n_5} e^{-\frac{E_{\mathrm{a}_5}}{R_{\mathrm{g}}T}} (1-\alpha_3)^{m_5} (-\ln\alpha_5)^{p_5} - \mathrm{SF}_5 \left(\frac{\mathrm{d}\alpha_2}{\mathrm{d}t} - \frac{\mathrm{d}\alpha_3}{\mathrm{d}t}\right) \alpha_5$$

The abuse heat terms are then expressed as:

$$\dot{q}_{\mathrm{abuse},i} = \mathrm{HW}_i \, A_i \alpha_i^{n_i} e^{-\frac{E_{\mathrm{a}_i}}{R_{\mathrm{g}}T}} (1-\alpha_i)^{m_i} (-\ln\alpha_i)^{p_i} \, L_{\mathrm{cell}}$$

Here, HW_i is an input parameter, and L_{cell} is the length of the cell.

4. **SEI growth.** If the optional attenuation term, g_{SEI}, due to the growing SEI layer is enabled as a multiplier for $\frac{\mathrm{d}\alpha_2}{\mathrm{d}t}$ by setting TSEI0 to a nonzero value (see Remark 3), the attenuation term is determined by ITSEIR:

$$g_{\mathrm{SEI}} = \begin{cases} e^{-\frac{z}{z_0}} & \text{for ITSEIR} = 0 \text{ (tunneling)} \\ \dfrac{z}{z_0} & \text{for ITSEIR} = 1 \text{ (diffusion)} \\ \min\left(e^{-\frac{z}{z_0}}, \dfrac{z}{z_0}\right) & \text{for ITSEIR} = 2 \text{ (automatic)} \end{cases}$$

In the above, z is the SEI thickness and z_0 is the initial thickness (TSEI0).

The SEI thickness evolves as:

$$\frac{\mathrm{d}z}{\mathrm{d}t} = -\frac{\mathrm{d}\alpha_2}{\mathrm{d}t}$$

5. **General form.** If Ai is set to a negative value for any of the equations, then the absolute value refers to a *DEFINE_FUNCTION allowing full control of the thermal runaway behavior for that α_i. In this case, the solver solves the following equation with the returned value from the *DEFINE_FUNCTION:

$$\frac{d\alpha_i}{dt} = [\text{returned value}] \times e^{-\frac{E_{a_i}}{R_g T}}$$

The *DEFINE_FUNCTION accepts the following values as arguments: f(a1, a2, a3, a4, a5, temp, r1, r2, r3, r4, r5, volt, curr, soc_a, soc_c, pgas). Here, a*i* is the i^{th} reaction progress variable, temp is the temperature, r*i* is the i^{th} reaction rate, volt is the voltage, curr is the current, soc_a is the state of charge in the anode, soc_c is the state of charge in the cathode, and pgas is the gas pressure.

6. **Degradation.** When IFEEDBK = 1 on Card 1, the abuse reaction progress variables α_j feed back into the P2D transport and kinetics through multiplicative scaling using an exponent:

$$X = X_{\text{ref}} \alpha_j^{\,p}$$

For the 1-equation model, $\alpha_j = \alpha_1$ and α_1^p is applied on the eleven scaled quantities, but for the four- or five-equation model, the different α_j are applied on different quantities as shown in the table below:

Property	Linked α_j	Exponent p input parameter
κ	4	PKAP
D_{e}	4	PKAP
$D_{\text{s,a}}$	2	PDSA
$D_{\text{s,c}}$	3	PDSC
$i_{0,\text{a}}$	1	PIOA
$i_{0,\text{c}}$	3	PIOC
$R_{\text{f,a}}$	1	PRFA
$R_{\text{f,c}}$	3	PRFC
σ	3	PSIG
$\varepsilon_{\text{s,a}}$	1	PEPSA
$\varepsilon_{\text{s,c}}$	3	PEPSC

7. **Gas generation from thermal abuse.** When IGASGEN = 1, gas generation is tracked from thermal abuse for five species gas species: C_2H_4, O_2, H_2O, CO_2, and H_2. Card 5 provides the stoichiometric yield matrix, ω_{ij}, where j is the species and i is the reaction. The gas generation rate per species is:

$$\dot{n}_j = \sum_{i=1}^{N_{\text{r}}} \omega_{ij} \frac{\text{d}\alpha_i}{\text{d}t} m_{\text{t}}$$

In the above, m_t mass per area of the cell (see Remark 2 of *BATTERY_ECHEM_-MAT_THERMAL) and N_r is the number thermal abuse reactions specified with IMODEL.

*BATTERY_ECHEM_VENT

Purpose: Set the vent parameters to allow for tracking gas venting out of a battery. The burst disk vent model simulates the irreversible opening of a cell safety vent when the internal gas pressure exceeds a burst threshold, and the subsequent release of gas, electrolyte vapor, and solid particles. This model is supported for all P2D equation variants.

Card 1	1	2	3	4	5	6	7	8
Variable	IVENT	PBURST	KVENT	PAMB	VPOR			
Type	I	F	F	F	F			
Default	0	none	none	1.013E5	0.			

This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	AVENT	GAMMA	MWG	MEJ	CPEJ	SFEJ		
Type	F	F	F	F	F	F		
Default	0.	1.4	0.029	0.	830.	0.		

This card is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	ABOIL	EABOIL	HVAP	MEL	MWEL	QLMIN		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.084	0.		

This card is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	CPC2H4	CPO2	CPH2O	CPCO2	CPH2			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

VARIABLE	DESCRIPTION
IVENT	Flag for turning on the vent model: EQ.0: Off EQ.1: On
PBURST	Burst pressure, P_{burst} (units: Pa)
KVENT	Venting coefficient, K_{vent} (units: mol/(s·Pa·m)): GE.0.0: Constant venting coefficient LT.0.0: |KVENT| refers to a load curve ID for the curve giving K_{vent} as a function of time.
PAMB	Ambient pressure, P_{atm} (units: Pa)
VPOR	Effective pore volume per unit electrode area, V_{pore} (units: m)
AVENT	Vent electrode area ratio, A_{vent}
GAMMA	Heat capacity ratio, γ
MWG	Mean molar mass of venting gas mixture M_{wg} (units:kg/mol)
MEJ	Mass per unit area ejected during venting, M_{ej} (units: kg/m^2)
CPEJ	Heat capacity of ejected solid fragments, $C_{p_{ej}}$ (units: J/(kg·K))
SFEJ	Scale factor on ejected mass. See Remark 2. GT.0.0: Constant scale factor, SF_{ef} LT.0.0: |SFEJ| refers to a load curve ID giving the scale factor, SF_{ej}, as a function of time.
ABOIL	Frequency factor for boiling reaction, A_{boil} (units: s^{-1}). See Remark 3.

VARIABLE	DESCRIPTION
EABOIL	Activation energy for boiling reaction, $E_{a_{boil}}/R_g$ (units: K). See Remark 3.
HVAP	Latent heat of vaporization, h_{vap} (units: J/kg).
MEL	Initial mass per unit area of electrolyte that can boil, m_e (units: kg/m^2). See Remark 3.
MWEL	Electrolyte molar mass M_{we} (units: kg/mol). See Remark 3.
QLMIN	Electrolyte depletion threshold. Venting stops if this criterion is reached.
CPC2H4, …, CPH2	Molar heat capacities of different species participating in venting model (units: J/(mol·K)). See Remark 5.

Remarks:

1. **Vent flow rate.** This model assumes that the cell has a safety vent (burst disk) that opens irreversibly when the total effective gas pressure, $P_{eff} = P_{gas} + P_{evap}$, exceeds the burst pressure, P_{burst}. P_{gas} is the gas pressure discussed in Remark 2, and P_{evap} is the evaporation pressure discussed in Remark 3. Once the vent is open, gas escapes at a rate proportional to the gauge pressure:

$$\dot{n}_{vent} = K_{vent}(P_{gas} - P_{atm})$$

where P_{atm} is the ambient or atmospheric pressure and K_{vent} is a venting coefficient.

When the vent area ratio, A_{vent}, is nonzero, the linear rate is capped by an isentropic choked-flow limit. The maximum mass flux through the orifice at sonic conditions is:

$$\dot{n}_{chk} = \frac{\dot{m}_{chk}}{M_{wg}} \quad ,$$

where

$$\dot{m}_{chk} = \rho_g a A_{vent} \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

$$a = \sqrt{\gamma \frac{R}{M_{wg}} T}$$

$$\rho_g = \frac{P_{eff} M_{wg}}{R_g T}$$

In the above, ρ_g is the gas density, a is the speed of sound, γ is the heat-capacity ratio, M_{wg} is the mean molar mass of the gas mixture, and A_{vent} is the vent-to-electrode dimensionless area ratio.

Each gas species (C_2H_4, O_2, CO_2, H_2O, and H_2) vents in proportion to its partial pressure fraction:

$$\dot{n}_i = \frac{P_i}{P_{gas}} \dot{n}_{vent} \ .$$

Those vented moles are removed after each P2D solve. For the 15-equation model, the concentrations are reduced uniformly to

$$\Delta n_i = \dot{n}_i \Delta t$$
$$c_{gas,i} = c_{gas,i} - \frac{\Delta n_i}{L_a}$$

and for the 6 and 10 equation models, the headspace accumulator is decremented directly as

$$n_i = n_i - \Delta n_i$$

2. **Gas pressure.** How the gas pressure, P_{gas}, is calculated depends on the equation count of the P2D model and the IGASGEN flag on *BATTERY_ECHEM_THERMAL_ABUSE. Four different configurations exist:

   a) *15-equation model with IGASGEN = 0.* Gas species concentrations, $c_{gas,i}$ with $i = 1, \ldots, 5$, are tracked as PDE unknowns. Electrochemical gas-generation reactions (r_d, r_{g1}, r_{g2}) provide the source terms. See Remark 1 of *BATTERY_ECHEM_GAS for details. The total gas pressure is computed from the porosity-weighted volume average and the integrated gas species concentrations over the anode, cathode, and separator:

   $$P_{gas} = R_g T \sum_{i=1}^{5} \bar{c}_i$$

   b) *15-equation model with IGASGEN = 1.* This is the same as the previous one, except that after each step, the generated moles from the N_r thermal abuse reactions are distributed uniformly to the anode gas concentrations based on the reaction progress variables, α_i (see Remarks 3 and 7 on *BATTERY_ECHEM_THERMAL_ABUSE):

   $$c_{gas,j,a} = c_{gas,j,a} + \frac{1}{L_a} \sum_{i=1}^{N_r} \omega_{ij} \Delta\alpha_i m_t$$

   c) *6- and 10-equation models with IGASGEN = 1.* No gas PDE exists. Instead, a lumped headspace accumulator, n_j (unis: mol/m^2) tracks the moles of each species j generated by abuse reactions (see Remark 7 on *BATTERY_ECHEM_THERMAL_ABUSE):

$$n_j^{n+1} = n_j^n + \sum_{i=1}^{N_r} \omega_{ij} \Delta\alpha_i m_t$$

The total gas pressure is computed by the ideal gas law:

$$P_{gas} = R_g T \sum_{j=1}^{5} \frac{n_j}{V_{pore}}$$

V_{pore} is the effective pore volume per unit electrode area. If V_{pore} is not supplied, it is derived automatically from the cell geometry as:

$$V_{pore} = L_a \varepsilon_{g,a} + L_s \varepsilon_{g,s} + L_c \varepsilon_{g,c}$$

In the above, L_a, L_s, and L_c are the lengths of the anode, separator, and cathode (see *BATTERY_ECHEM_CELL_GEOMETRY), and $\varepsilon_{g,a}$, $\varepsilon_{g,s}$, and $\varepsilon_{g,c}$ are the gas volume fractions for the anode, separator, and cathode.

d) *6- and 10-equation models with IGASGEN = 0.* No abuse gas generation is tracked, $P_{gas} = 0$. The only source of internal gas pressure is the electrolyte vapor P_{evap} from the boiling model (see Remark 3). If boiling is also disabled, $P_{eff} = 0$, and venting does not occur.

3. **Electrolyte boiling.** When ABOIL is greater than zero, an Arrhenius-driven electrolyte vaporization model is activated. A liquid fraction, $0 \le q_e \le 1$, evolves as:

$$\frac{dq_e}{dt} = -q_e A_{boil} e^{-\frac{E_{a_{boil}}}{R_g T}}$$

The electrolyte vapor also contributes to a partial pressure:

$$P_{evap} = \frac{(1 - q_e) m_e R_g T}{M_{we} V_{pore}}$$

This pressure is added to the gas pressure for the burst disk check.

4. **Solid ejecta.** During thermal runaway, solid material (active particles, binder, current-collector fragments, …) can be optionally ejected alongside the gas. Two mechanisms are available with SFEJ taking priority:

a) *Dynamic ejecta (SFEJ).* When SFEJ is defined, the instantaneous solid mass flow rate is proportional to the gas mass flow using SF_{ej}:

$$\dot{m}_{ej} = SF_{ej} \dot{m}_{gas} \quad .$$

b) *Mass fraction ejecta (MEJ).* When MEJ > 0.0 and SFEJ is not defined, the total ejecta mass is released proportionally to the cumulative gas vent fraction:

$$m_{\mathrm{ej}}(t) = M_{\mathrm{ej}} \frac{m_{\mathrm{g}}(t)}{m_{\mathrm{t}}(0.)}$$

5. **Specifying molar heat capacities.** An alternative to defining the molar heat capacities with this keyword is to refer to a CHEMKIN type format file. *CHEMISTRY_BATTERY can be included in the input deck to provide the file names for the battery chemistry input file, the corresponding thermodynamics data file, and the transport properties file. See *CHEMISTRY_BATTERY for further details. Note that nonzero values defined with *BATTERY_ECHEM_VENT overwrite data provided by *CHEMISTRY_BATTERY. The advantage of using the CHEMKIN file is that the molar heat capacities can vary as a function of temperature rather than being a constant value.

6. **Mass and energy balance.** From a mass balance perspective, we are venting gas and ejecting solid fragments (optionally) so that the effective thermal mass of the cell in the lumped thermal model decreases:

$$m_{\mathrm{t}}(t) = m_{\mathrm{t}}(0.) - m_{\mathrm{g}}(t) - m_{\mathrm{ej}}(t)$$

The total vented mass per electrode area via gas release is:

$$m_{\mathrm{g}} = M_{\mathrm{wg}} \sum_{i=1}^{5} N_i$$

where N_i is the vented moles per species over the P2D steps. From an energy balance perspective, energy will be lost due to gas venting and particle ejecting but also due to electrolyte evaporation if the boiling model is active.

The venting gas heat sink contribution is:

$$\dot{q}_{\mathrm{vent}} = \sum_{i=1}^{5} \dot{n}_i C_{\mathrm{p},i} T$$

where $C_{p,i}$ is in the molar heat capacity of species i at constant pressure. The heat capacity can be provided as constant values in the fourth line or, alternatively, read from the CHEMKIN file in *CHEMISTRY_BATTERY (see Remark 5). In the latter case, the heat capacity follows a polynomial and is updated at runtime as a function of temperature. If no input is provided, default values are used.

The evaporation ($A_{\mathrm{boil}} > 0.$) contributes to the heat sink by:

$$\dot{q}_{\mathrm{boil}} = m_{\mathrm{e}} h_{\mathrm{vap}} \frac{(q_{\mathrm{e}}^{n} - q_{\mathrm{e}}^{n+1})}{\Delta t}$$

where h_{vap} is the latent heat of vaporization.

Finally, the solid mass ejection contributes to the heat sink during venting for each species i as:

$$\dot{q}_{\mathrm{ej}} = \frac{\Delta m_{\mathrm{ej}}}{\Delta t} C_{\mathrm{p_{ej}}} T$$

*BATTERY_FIT_RANDLES_PARAMETERS

Purpose: Derive equivalent-circuit-model (Randles) parameters from battery test data and automatically generates a parameterized Randles circuit keyword file for subsequent battery simulations. The fitting process can be performed using either internally generated virtual tests leveraging *BATTERY_ECHEM_... or externally supplied test data. The generated output contains temperature-dependent OCV-SOC curves, resistance and capacitance tables, optional SOC correction, and complete *EM_RANDLES keyword definitions.

Card Summary:

Card 1. This card is required.

ISOURCE	IMETH	IOCV	ISOCS				IVERBOSE

Card 2. This card is required.

Q	CGCELLID	ACELLID	CCELLID	ECELLID	THCELLID		

Card 3. This card is required.

VCUTL	CUTH	TAUMIN	TAUMAX				

Card 4. This card is required.

RDLMOD	RDLID	LCOFF	CQ	SOCINIT	TREF		

Card 5. This card is required.

NTEMP	REFT						

Card 6. This card is required. Specify NTEMP > 0 temperature levels, putting 8 on each card. Include as many cards as necessary.

T1	T2	T3	T4	T5	T6	T7	T8

Card 7. This card is required.

NRATE	REFRT						

Card 8. This card is required. Specify NRATE > 0 C-rates putting 8 on each card. Include as many cards as necessary

C1	C2	C3	C4	C5	C6	C7	C8

Card 9. Include NTEMP × NRATE instances of this card if ISOURCE = 1.

FILENAME

Card 10. This card is required.

NHPPC	DSOC	DIPLSE	CHPLSE	TPULSE	TRPLSE	TREST	

Card 11. Include NTEMP instances of this card if ISOURCE = 1.

FILENAME

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	ISOURCE	IMETH	IOCV	ISOCS				IVERBOSE
Type	I	I	I	I				I
Default	0	0	0	0				0

VARIABLE	DESCRIPTION
ISOURCE	Source of data used for parameter identification: EQ.0: Virtual tests performed internally. This data source requires a fully defined *BATTERY_ECHEM_... model (see Remarks 1 and 5). EQ.1: Hardware or simulation test data read from files
IMETH	Method for Randles circuit parameter identification: EQ.0: Method according to Bateau-Meyer et al [2012]. This method outputs a first-order Randles circuit. EQ.1: Method according to Hu et al [2013]. This method outputs a second-order Randles circuit.
IOCV	Determines how the curve specifying OCV(SOC) is derived: EQ.0: Based on the discharge test at the reference C-rate (see REFRT on Card 7). It uses the discharge test with the lowest specified C-rate by default.

VARIABLE	DESCRIPTION
	EQ.1: Averaged from the discharge test at the reference C-rate (see REFRT on Card 7) and the charge test whose C-rate is closest to it. It is recommended to use the identical C-rate and only flip the sign. IOCV = 0 is used if no charge test exists.
ISOCS	SOC correction handling at higher C-rates: EQ.0: No correction parameters identified. EQ.1: SOC-shift calculation according to Bateau-Meyer [2018] based on a multi-rate capacity test EQ.2: Derive capacity coefficient based on a multi-rate capacity test to use a capacity fade table instead of an SOC shift
IVERBOSE	Level of output passed to the terminal, messag files, and d3hsp files: EQ.0: Basic process information and fatal errors EQ.1: Same as 0 but also includes warnings EQ.2: Same as 1 but also includes advanced process information

Card 2	1	2	3	4	5	6	7	8
Variable	Q	CGCELLID	ACELLID	CCELLID	ECELLID	THCELLID		
Type	F	I	I	I	I	I		
Default	0.0	0	0	0	0	0		

VARIABLE	DESCRIPTION
Q	Nominal capacity, Q: GT.0.0: Constant nominal capacity (units: Ah). LE.0.0: For ISOURCE = 0, Q is calculated based on the provided *BATTERY model. For ISOURCE = 1, Q is obtained from Coulomb counting during the discharge test at the reference C-rate (see REFRT on Card 7) and additionally averaged with the result from the paired charge test when IOCV = 1.

VARIABLE	DESCRIPTION
CGCELLID	CELLID of the cell geometry from *BATTERY_ECHEM_CELL_GEOMETRY (ISOURCE = 0 only). See Remark 6.
ACELLID	CELLID of the anode material from *BATTERY_ECHEM_MAT_ANODE (ISOURCE = 0 only). See Remark 6.
CCELLID	CELLID of the cathode material from *BATTERY_ECHEM_MAT_CATHODE (ISOURCE = 0 only). See Remark 6.
ECELLID	CELLID of the electrolyte material from *BATTERY_ECHEM_MAT_ELECTROLYTE (ISOURCE = 0 only). See Remark 6.
THCELLID	CELLID of the thermal material from *BATTERY_ECHEM_MAT_THERMAL (ISOURCE = 0 only). See Remark 6.

Card 3	1	2	3	4	5	6	7	8
Variable	VCUTL	VCUTH	TAUMIN	TAUMAX				
Type	F	F	F	F				
Default	2.5	4.2	10.0	3600.0				

VARIABLE	DESCRIPTION
VCUTL	Lower terminal voltage cut-off (V) for virtual test (ISOURCE = 0 only)
VCUTH	Upper terminal voltage cut-off (V) for virtual test (ISOURCE = 0 only)
TAUMIN	Minimum of the SOC-shift relaxation time constant τ search range (only used if ISOCS = 1)
TAUMAX	Maximum of the SOC-shift relaxation time constant τ search range (only used if ISOCS = 1)

Card 4	1	2	3	4	5	6	7	8
Variable	RDLMOD	RDLID	LCOFF	CQ	SOCI	TREF		
Type	I	I	I	F	F	F		
Default	0	0	100	1./36.	100.0	0.0		

VARIABLE	DESCRIPTION
RDLMOD	Randles model keyword to output: EQ.0: *EM_RANDLES_MESHLESS EQ.1: *EM_RANDLES_BATMAC EQ.2: *EM_RANDLES_SOLID EQ.3: *EM_RANDLES_TSHELL
RDLID	Randles model RDLID in the output keyword
LCOFF	Offset to the base LCID for the output curves and tables
CQ	SOC conversion factor (units: %/s) output to the keyword deck for the derived Randles circuit
SOCI	Initial SOC (in %) output to the keyword deck for the derived Randles circuit
TREF	Randles model TREF in the output keyword. It defaults to the temperature on Card 6 with index REFT.

Card 5	1	2	3	4	5	6	7	8
Variable	NTEMP	REFT						
Type	I	I						
Default	1	1						

VARIABLE	DESCRIPTION
NTEMP	Number of temperature levels
REFT	Index of the temperature level used as reference when calculating Q, OCV(SOC,T), and the SOC correction parameters

Temperature Level Cards. Specify NTEMP temperature levels putting 8 on each card. Include as many cards as necessary but at least one containing one temperature level.

Card 6	1	2	3	4	5	6	7	8
Variable	T1	T2	T3	T4	T5	T6	T7	T8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
T*i*	Temperature levels (K) to be considered in the virtual test matrix (ISOURCE = 0) or present in the test files provided (ISOURCE = 1).

Card 7	1	2	3	4	5	6	7	8
Variable	NRATE	REFRT						
Type	I	I						
Default	1	1						

VARIABLE	DESCRIPTION
NRATE	Number of C-rates
REFRT	Index of the C-rate used as reference when calculating Q, OCV(SOC,T) and SOC correction parameters.

C-rate Cards. Specify NRATE C-rates, putting 8 on each card. Include as many cards as necessary but at least one containing one C-rate.

Card 8	1	2	3	4	5	6	7	8
Variable	C1	C2	C3	C4	C5	C6	C7	C8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
C*i*	C-rates to be considered in the virtual test matrix (ISOURCE = 0) or present in the test files provided (ISOURCE = 1). A positive C-rate corresponds to a discharge test and a negative C-rate to a charge test.

Capacity (Dis-)Charge Test File Cards. Include NTEMP × NRATE instances of this card if ISOURCE = 1.

Card 9	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	none							

VARIABLE	DESCRIPTION
FILENAME	File name of the capacity (dis-)charge test file to be included (see Remarks 2, 3 and 4)

Card 10	1	2	3	4	5	6	7	8
Variable	NHPPC	DSOC	DIPLSE	CHPLSE	TPLSE	TRPLSE	TREST	
Type	I	F	F	F	F	F	F	
Default	1	10.0	1.0	0.75	10.0	40.0	3600.0	

VARIABLE	DESCRIPTION
NHPPC	Number of HPPC pulse pairs
DSOC	Delta SOC (%) to be removed after each HPPC pulse cycle
DIPLSE	C-rate of the discharge pulse
CHPLSE	C-rate of the charge pulse
TPLSE	Pulse duration (units: s)
TRPLSE	Rest time (units: s) between pulses
TREST	Rest time (units: s) after the SOC removal step

HPPC Test File Cards. Include NTEMP instances of this card if ISOURCE = 1.

Card 11	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	none							

VARIABLE	DESCRIPTION
FILENAME	File name of the HPCC test file to be included (see Remarks 2, 3, and 4).

Remarks:

1. **Electrochemistry model definition.** For virtual testing (ISOURCE = 0), a single cell *BATTERY model must be present and referenced on Card 2. For robust virtual fitting, provide S0A, S100A, S0C, S100C and define initial SOCA and SOCC as the fully charged stoichiometric state (SOCA = S100A and SOCC = S100C), using a physically consistent stoichiometry window, on *BATTERY_ECHEM_MAT_ANODE and *BATTERY_ECHEM_MAT_CATHODE. *BATTERY_ECHEM_CONTROL_SOLVER must be in the input deck, but NCYCLE should be left as its default and Card 2 of that keyword should also be left as blank or default. NCYCLE and Card 2 of that keyword are populated as needed (see Remark 5 for details) to perform the virtual tests. Optionally, *BATTERY_ECHEM_CONTROL_INTEGRATION can be provided to control the solution.

2. **File name length and path limitations.** File names are currently limited to 80 characters and one line. Paths are currently not supported; the file must reside in the LS-DYNA run directory.

3. **File order.** Each file's associated temperature level and C-rate are determined solely via the file's position in the input list; the parser does not extract temperature or C-rate information from the file. Therefore, capacity test files must be provided in C-rate major and temperature suborder. This means all temperatures (T1, T2, etc.) for C1, followed by all temperatures (T1, T2, etc.) for C2, etc., while HPPC files must be provided solely in the same sequence as the temperature levels are defined (T1, T2, etc.).

4. **File format.** There is a single, strict file format supported by the parser which follows the file format output by the *BATTERY solver in ba_P2DCell_mesos*.dat. Every file must contain at some point before the first line of numerical data, a line reading exactly (case-insensitive, whitespace/comma/tab delimited):

   ```
   time-(1)   voltage-(2)   current-(3)
   ```

   Thus, comment lines in the files are allowed before that mandatory header. The parser will only parse from the required column header onwards. The parser allows files to contain more than the required columns but will ignore them. Thus, you can use ba_P2DCell_mesos*.dat together with ISOURCE = 1 without any file modifications.

5. **Virtual testing.** When ISOURCE = 0, this keyword creates a virtual test matrix where the number of tests, n, is NRATE × NTEMP capacity tests + NTEMP HPPC tests. Card 2 of *BATTERY_ECHEM_CONTROL_SOLVER is internally populated with the required cycles and an individual CELLID for each test is created. For each of the n cell models, the provided *BATTERY model is cloned. Virtual tests run automatically in sequential order, and result files are written for each individual test.

6. **CELLID.** For ISOURCE = 0, the CELLIDs of the various battery components referenced on Card 2 can be distinct. The CELLID on *BATTERY_ECHEM_CONTROL_SOLVER in the input deck is ignored. If a CELLID is not found, then the solver looks for a CELLID = 0 and uses that for the component.

*CESE

The keyword *CESE provides input data for the Conservation Element/Solution Element (CESE) compressible fluid solver:

CESE_BOUNDARY_AXISYMMETRIC_*{OPTION}

CESE_BOUNDARY_BLAST_LOAD}*

CESE_BOUNDARY_CONJ_HEAT_*{OPTION}

CESE_BOUNDARY_CYCLIC_*{OPTION}

CESE_BOUNDARY_FSI_*{OPTION}

CESE_BOUNDARY_NON_REFLECTIVE_*{OPTION}

CESE_BOUNDARY_PRESCRIBED_*{OPTION}

CESE_BOUNDARY_REFLECTIVE_*{OPTION}

CESE_BOUNDARY_SLIDING_*{OPTION}

CESE_BOUNDARY_SOLID_WALL_*{OPTION1}_{OPTION2}

*CESE_CHEMISTRY_D3PLOT

*CESE_CONTROL_LIMITER

*CESE_CONTROL_MESH_MOV

*CESE_CONTROL_SOLVER

*CESE_CONTROL_TIMESTEP

*CESE_DATABASE_ELOUT

*CESE_DATABASE_FLUXAVG

*CESE_DATABASE_FSIDRAG

*CESE_DATABASE_POINTOUT

*CESE_DATABASE_SSETDRAG

*CESE_DEFINE_NONINERTIAL

*CESE_DEFINE_POINT

*CESE_DRAG

*CESE_EOS_CAV_HOMOG_EQUILIB_

*CESE_EOS_IDEAL_GAS

*CESE_EOS_INFLATOR1

*CESE_EOS_INFLATOR2

*CESE_FSI_EXCLUDE

*CESE_INITIAL

CESE_INITIAL_*{OPTION}

*CESE_INITIAL_CHEMISTRY

*CESE_INITIAL_CHEMISTRY_ELEMENT

*CESE_INITIAL_CHEMISTRY_PART

*CESE_INITIAL_CHEMISTRY_SET

*CESE_MAT_000

*CESE_MAT_001 (*CESE_MAT_GAS)

*CESE_MAT_002

*CESE_PART

*CESE_SURFACE_MECHSSID_D3PLOT

*CESE_SURFACE_MECHVARS_D3PLOT

Note that when performing a chemistry calculation with the CESE solver, initialization should only be done with the *CESE_INITIAL_CHEMISTRY_… cards, not the *CESE_-INITIAL… cards.

CESE_BOUNDARY_AXISYMMETRIC_*OPTION

Available options are:

MSURF

MSURF_SET

SET

SEGMENT

Purpose: Define an axisymmetric boundary condition on the axisymmetric axis for the 2D axisymmetric CESE compressible flow solver.

The MSURF and MSURF_SET options are used when the CESE mesh has been created using *MESH cards. The SET and SEGMENT cards are used when *ELEMENT_SOLID cards are used to define the CESE mesh.

Card Summary:

Card 1a. This card is included if the keyword option is MSURF. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

MSURFID							

Card 1b. This card is included if the keyword option is MSURF_SET. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

MSURF_S							

Card 1c. This card is included if the keyword option is SET. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

SSID							

Card 1d. This card is included if the keyword option is SEGMENT. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

N1	N2	N3	N4				

Data Card Definitions:

Surface Part Card. Card 1 format used when the MSURF keyword option is active. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSURFID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSURFID	Mesh surface part ID referenced in *MESH_SURFACE_ELEMENT cards.

Surface Part Set Card. Card 1 format used when the MSURF_SET keyword option is active. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1b	1	2	3	4	5	6	7	8
Variable	MSURF_S							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSURF_S	Identifier of a set of mesh surface part IDs created with a *LSO_-ID_SET card, where each mesh surface part ID in the set is referenced in *MESH_SURFACE_ELEMENT cards.

Set Card. Card 1 format used when the SET keyword option is active. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1c	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
SSID	Segment set ID

Segment Cards. Card 1 format used when SEGMENT keyword option is active. Include an additional card for each corresponding pair of segments. This input ends at the next keyword ("*") card.

Card 1d	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
N1, N2, …	Node IDs defining a segment

Remarks:

1. **Restrictions.** This boundary condition can only be used on the axisymmetric axis for the 2D axisymmetric CESE fluid solver.

CESE_BOUNDARY_BLAST_LOAD_*OPTION

Available options include:

MSURF

MSURF_SET

SET

SEGMENT

Purpose: For the CESE compressible flow solver, set boundary values for velocity, density, and pressure from a blast wave defined by a *LOAD_BLAST_ENHANCED card. Boundary values are applied at the centroid of elements connected with this boundary. *OPTION* = SET and *OPTION* = SEGMENT are for user defined meshes whereas *OPTION* = MSURF and *OPTION* = MSURF_SET are associated with the automatic volume mesher (See *MESH keywords). In other words, the MSURF and MSURF_SET options are used when the CESE mesh has been created using *MESH cards. The SET and SEGMENT cards are used when *ELEMENT_SOLID cards are used to define the CESE mesh.

Card Summary:

Card 1a. This card is included if the keyword option is MSURF.

BID	MSURFID						

Card 1b. This card is included if the keyword option is MSURF_SET.

BID	MSURF_S						

Card 1c. This card is included if the keyword option is SET.

BID	SSID						

Card 1d. This card is included if the keyword option is SEGMENT.

BID	N1	N2	N3	N4			

Data Card Definitions:

Surface Part Card. Card 1 format used when the MSURF keyword option is active.

Card 1a	1	2	3	4	5	6	7	8
Variable	BID	MSURFID						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
BID	Blast source ID (see *LOAD_BLAST_ENHANCED)
MSURFID	A mesh surface part ID referenced in *MESH_SURFACE_ELEMENT cards

Surface Part Set Card. Card 1 format used when the MSURF_SET keyword option is active.

Card 1b	1	2	3	4	5	6	7	8
Variable	BID	MSURF_S						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
BID	Blast source ID (see *LOAD_BLAST_ENHANCED)
MSURF_S	Identifier of a set of mesh surface part IDs created with a *LSO_ID_SET card, where each mesh surface part ID in the set is referenced in *MESH_SURFACE_ELEMENT cards.

Set Card. Card 1 format used when the SET keyword option is active.

Card 1c	1	2	3	4	5	6	7	8
Variable	BID	SSID						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
BID	Blast source ID (see *LOAD_BLAST_ENHANCED)
SSID	Segment set ID

Segment Card. Card 1 for SEGMENT keyword option is active.

Card 1d	1	2	3	4	5	6	7	8
Variable	BID	N1	N2	N3	N4			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
BID	Blast source ID (see *LOAD_BLAST_ENHANCED)
N1, N2, …	Node ID's defining a segment

CESE_BOUNDARY_CONJ_HEAT_*OPTION

Available options are:

MSURF

MSURF_SET

SET

SEGMENT

Purpose: Define a conjugate heat transfer interface condition for CESE compressible flows. This condition identifies those boundary faces of the CESE mesh that are in contact with non-moving structural parts and through which heat flows. This is only possible when the structural thermal solver is also being used in the structural parts.

The MSURF and MSURF_SET options are used when the CESE mesh has been created using *MESH cards. The SET and SEGMENT cards are used when *ELEMENT_SOLID cards are used to define the CESE mesh.

Card Summary:

Card 1a. This card is included when the keyword option is MSURF. Include as many cards as necessary. This input ends at the next keyword ("*") card.

MSURFID							

Card 1b. This card is included when the keyword option is MSURF_SET. Include as many cards as necessary. This input ends at the next keyword ("*") card.

MSURF_S							

Card 1c. This card is included when the keyword option is SET. Include as many cards as necessary. This input ends at the next keyword ("*") card.

SSID							

Card 1d. This card is included when the keyword option is SEGMENT. Include as many cards as necessary. This input ends at the next keyword ("*") card.

N1	N2	N3	N4				

Data Card Definitions:

Surface Part Card. Card 1 used when the MSURF keyword option is active. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSURFID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSURFID	Mesh surface part ID referenced in *MESH_SURFACE_ELEMENT cards.

Surface Part Set Card. Card 1 used when the MSURF_SET keyword option is active. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1b	1	2	3	4	5	6	7	8
Variable	MSURF_S							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSURF_S	Identifier of a set of mesh surface part IDs created with an *LSO_-ID_SET card, where each mesh surface part ID in the set is referenced in *MESH_SURFACE_ELEMENT cards.

Set Card. Card 1 used when the SET keyword option is active. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1c	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
SSID	Segment set ID

Segment Cards. Card 1 used when SEGMENT keyword option is active. Include an additional card for each corresponding pair of segments. This input ends at the next keyword ("*") card.

Card 1d	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
N1, N2, …	Node IDs defining a segment

Remarks:

1. **Restrictions.** This boundary condition should only be imposed on a CESE mesh boundary that is in contact with non-moving structural parts. An Eulerian CESE solver is required as is use of the structural thermal solver.

CESE_BOUNDARY_CYCLIC_*OPTION

Available options are:

MSURF

MSURF_SET

SET

SEGMENT

Purpose: Define a cyclic (periodic) boundary condition for CESE compressible flows. This cyclic boundary condition can be used on periodic boundary surfaces.

The MSURF and MSURF_SET options are used when the CESE mesh has been created using *MESH cards. The SET and SEGMENT cards are used when *ELEMENT_SOLID cards are used to define the CESE mesh.

Card Summary:

Include as many sets of the following cards as needed. This input ends at the next keyword ("*") card.

Card 1a. This card is included when the keyword option is set to MSURF.

MSURFID1	MSURFID2	CYCTYP					

Card 1b. This card is included when the keyword option is set to MSURF_SET.

MSRF_S1	MSRF_S2	CYCTYP					

Card 1c. This card is included when the keyword option is set to SET.

SSID1	SSID2	CYCTYP					

Card 1d. This card is included when the keyword option is set to SEGMENT.

ND1	ND2	ND3	ND4	NP1	NP2	NP3	NP4

Card 2a. This card is included for the MSURF, MSURF_SET, and SET options when CYCTYP = 1.

AXISX1	AXISY1	AXISZ1	DIRX	DIRY	DIRZ	ROTANG	

Card 2b. This card is included for the MSURF, MSURF_SET, and SET options when CYCTYP = 2.

TRANSX	TRANSY	TRANSZ					

Data Card Definitions:

Surface Part Card. Card 1 format when the MSURF keyword option is active

Card 1a	1	2	3	4	5	6	7	8
Variable	MSURFID1	MSURFID2	CYCTYP					
Type	I	I	I					
Default	none	none	0					
Remarks	2	2	1					

VARIABLE	DESCRIPTION
MSURFID1, MSURFID2	Mesh surface part IDs referenced in *MESH_SURFACE_ELEMENT cards.
CYCTYP	Relationship between the two cyclic boundary condition surfaces: EQ.0: None assumed (default) EQ.1: The first surface is rotated about an axis to match the second surface. EQ.2: The faces of the first surface are translated in a given direction to obtain the corresponding faces on the second surface.

Surface Part Set Card. Card 1 format when the MSURF_SET keyword option is active

Card 1b	1	2	3	4	5	6	7	8
Variable	MSRF_S1	MSRF_S2	CYCTYP					
Type	I	I	I					
Default	none	none	0					
Remarks	3	3	1					

VARIABLE	DESCRIPTION
MSRF_S1, MSRF_S2	Identifiers of two sets of mesh surface part IDs, each created with a *LSO_ID_SET card, where each mesh surface part ID in each set is referenced in *MESH_SURFACE_ELEMENT cards.
CYCTYP	Relationship between the two cyclic boundary condition surfaces: EQ.0: None assumed (default) EQ.1: The first surface is rotated about an axis to match the second surface. EQ.2: The faces of the first surface are translated in a given direction to obtain the corresponding faces on the second surface.

Set Card. Card 1 format when the SET keyword option is active

Card 1c	1	2	3	4	5	6	7	8
Variable	SSID1	SSID2	CYCTYP					
Type	I	I	I					
Default	none	none	0					
Remarks	4	4	1					

VARIABLE	DESCRIPTION
SSID1, SSID2	A pair of segment set IDs.
CYCTYP	Relationship between the two cyclic boundary condition surfaces: EQ.0: None assumed (default) EQ.1: The first surface is rotated about an axis to match the second surface. EQ.2: The faces of the first surface are translated in a given direction to obtain the corresponding faces on the second surface.

Segment Card. Card 1 format when SEGMENT keyword option is active. Include an additional card for each corresponding pair of segments. This input ends at the next keyword ("*") card.

Card 1d	1	2	3	4	5	6	7	8
Variable	ND1	ND2	ND3	ND4	NP1	NP2	NP3	NP4
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
ND*i*, NP*i*	Node IDs defining a pair of segments: ND1, ND2, ND3, ND4 define the first segment, while NP1, NP2, NP3, NP4 define the second segment. This pair of segments must match either through a geometric translation or rotation.

Rotation Case Card. Additional card for the MSURF, MSURF_SET, and SET options when CYCTYP = 1.

Card 2a	1	2	3	4	5	6	7	8
Variable	AXISX1	AXISY1	AXISZ1	DIRX	DIRY	DIRZ	ROTANG	
Type	F	F	F	F	F	F	F	
Default	0.0	0.0	0.0	none	none	none	none	

VARIABLE	DESCRIPTION
AXIS[Z,Y,Z]1	A point on the axis of rotation
DIR[X,Y,Z]	The direction which together with AXIS[X,Y,Z]1 defines the axis of rotation
ROTANG	The angle of rotation (in degrees) that transforms the centroid of each face on the first surface to the centroid of the corresponding face on the second surface

Translation Case Card. Additional card for the MSURF, MSURF_SET, and SET options when CYCTYP = 2.

Card 2b	1	2	3	4	5	6	7	8
Variable	TRANSX	TRANSY	TRANSZ					
Type	F	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
TRANS[X,Y,Z]	The translation direction that enables the identification of the segment in the second surface that matches a segment in the first surface

Remarks:

1. **CYCTYP.** For the MSURF, MSURF_SET, or SET options with CYCTYP = 0, the code examines the geometry of two faces of the two surfaces in order to determine if the surfaces are approximately parallel (CYCTYP = 2) or related through a rotation (CYCTYP = 1). The geometric parameters required are then computed.

2. **MSURF.** For the MSURF option, the number of mesh surface elements must be the same in each mesh surface part. The mesh surface elements in each mesh surface part are then internally ordered to match pairwise between the two mesh surface parts.

3. **MSURF_SET.** For the MSURF_SET option, the number of mesh surface elements must be the same in each mesh surface part set. The mesh surface elements in each mesh surface part set are then internally ordered to match pairwise between the two mesh surface part sets.

4. **SET.** For the SET option, the same number of segments must be in each set. The segments in each set are then internally ordered to match pairwise between the two sets.

CESE_BOUNDARY_FSI_*OPTION

Available options are:

MSURF

MSURF_SET

SET

SEGMENT

Purpose: Define an FSI boundary condition for the moving mesh CESE compressible flow solver. This card must not be combined with the immersed-boundary method CESE solver; doing so will result in an error termination condition.

This boundary condition must be applied on a surface of the CESE computational domain that is co-located with surfaces of the outside boundary of the structural mechanics' mesh. The nodes of the two meshes will generally not be shared.

The MSURF and MSURF_SET options are used when the CESE mesh has been created using *MESH cards. The SET and SEGMENT cards are used when *ELEMENT_SOLID cards are used to define the CESE mesh.

Card Summary:

Card 1a. This card is included if the keyword option is MSURF. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

MSURFID							

Card 1b. This card is included if the keyword option is MSURF_SET. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

MSURF_S							

Card 1c. This card is included if the keyword option is SET. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

SSID							

Card 1d. This card is included if the keyword option is SEGMENT. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

N1	N2	N3	N4				

Data Card Definitions:

Surface Part Card. Card 1 format used when the MSURF keyword option is active. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSURFID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSURFID	Mesh surface part ID referenced in *MESH_SURFACE_ELEMENT cards

Surface Part Set Card. Card 1 format used when the MSURF_SET keyword option is active. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1b	1	2	3	4	5	6	7	8
Variable	MSURF_S							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSURF_S	Identifier of a set of mesh surface part IDs created with a *LSO_-ID_SET card, where each mesh surface part ID in the set is referenced in *MESH_SURFACE_ELEMENT cards.

Set Card. Card 1 format used when the SET keyword option is active. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1c	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
SSID	Segment set ID

Segment Cards. Card 1 format used when SEGMENT keyword option is active. Include an additional card for each corresponding pair of segments. This input ends at the next keyword ("*") card.

Card 1d	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
N1, …	Node IDs defining a segment

Remarks:

1. **Conjugate Heat Transfer Problems.** This boundary condition card is also needed for conjugate heat transfer problems with the moving mesh CESE solver.

CESE_BOUNDARY_NON_REFLECTIVE_*OPTION

Available options are:

MSURF

MSURF_SET

SET

SEGMENT

Purpose: Define a passive boundary condition for CESE compressible flows. This non-reflective boundary condition (NBC) provides an artificial computational boundary for an open boundary that is passive.

Use the MSURF and MSURF_SET options when you use the *MESH cards to generate the CESE mesh. Use the SET and SEGMENT cards when *ELEMENT_SOLID cards define the CESE mesh.

Card Summary:

Card 1a. Include this card for the MSURF keyword option. Include as many cards as necessary. This input ends at the next keyword ("*") card.

MSURFID							

Card 1b. Include this card for the MSURF_SET keyword option. Include as many cards as necessary. This input ends at the next keyword ("*") card.

MSURF_S							

Card 1c. Include this card for the SET keyword option. Include as many cards as necessary. This input ends at the next keyword ("*") card.

SSID							

Card 1d. Include this card for the SEGMENT keyword option. Include an additional card for each corresponding pair of segments. This input ends at the next keyword ("*") card.

N1	N2	N3	N4				

Data Card Definitions:

Surface Part Card. Card 1 used when the MSURF keyword option is active. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSURFID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSURFID	Mesh surface part ID referenced by *MESH_SURFACE_ELEMENT cards

Surface Part Set Card. Card 1 used when the MSURF_SET keyword option is active. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1b	1	2	3	4	5	6	7	8
Variable	MSURF_S							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSURF_S	ID of a set of mesh surface parts created with an *LSO_ID_SET card, where each mesh surface part ID in the set is referenced by *MESH_SURFACE_ELEMENT cards.

Set Card. Card 1 used when the SET keyword option is active. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1c	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
SSID	Segment set ID

Segment Cards. Card 1 used when the SEGMENT keyword option is active. Include an additional card for each corresponding pair of segments. This input ends at the next keyword ("*") card.

Card 1d	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
N1, N2, …	Node IDs defining a segment

Remarks:

1. **Normal application of this boundary condition.** We recommend imposing this boundary condition on an open surface far from the main disturbed flow (the further away, the better), meaning an almost uniform flow on the boundary surface.

2. **Default boundary condition for CESE.** LS-DYNA automatically assigns the non-reflective boundary condition to any boundary segment not already assigned a boundary condition by any of the *CESE_BOUNDARY_… cards.

CESE_BOUNDARY_PRESCRIBED_*OPTION

Available options include:

MSURF

MSURF_SET

SET

SEGMENT

Purpose: For the CESE compressible flow solver, set boundary values for velocity, density, pressure, and temperature. Boundary values are applied at the centroid of elements connected with this boundary. Use keyword options SET or SEGMENT for user-defined meshes, meaning mesh specified with *ELEMENT_SOLID card. Use MSURF or MSURF_SET for meshes created with the automatic volume mesher, meaning meshes generated with the *MESH cards.

Card Summary:

A set of data cards for this keyword consists of 3 of the following cards:

1. Card 1 specifies the object to which the boundary condition is applied. Its format depends on the keyword option.
2. Card 2 reads in load curve IDs.
3. Card 3 reads in scale factors.

For each boundary condition to be specified include one set of cards. This input ends at the next keyword ("*") card.

Card 1a. Include this card for the MSURF keyword option.

MSURFID	IDCOMP						

Card 1b. Include this card for the MSURF_SET keyword option.

MSURF_S	IDCOMP						

Card 1c. Include this card for the SET keyword option.

SSID	IDCOMP						

Card 1d. Include this card for the SEGMENT keyword option.

N1	N1	N3	N4	IDCOMP			

Card 2. This card is required.

LC_U	LC_V	LC_W	LC_RHO	LC_P	LC_T		

Card 3. This card is required.

SF_U	SF_V	SF_W	SF_RHO	SF_P	SF_T		

Data Card Definitions:

Surface Part Card. Card 1 format for the MSURF keyword option

Card 1a	1	2	3	4	5	6	7	8
Variable	MSURFID	IDCOMP						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
MSURFID	Mesh surface part ID referenced in *MESH_SURFACE_ELEMENT cards
IDCOMP	For inflow boundaries in problems involving chemical reacting flows, this ID references a *CHEMISTRY_COMPOSITION card that gives the chemical mixture of the fluid entering the domain

Surface Part Set Card. Card 1 format for the MSURF_SET keyword option

Card 1b	1	2	3	4	5	6	7	8
Variable	MSURF_S	IDCOMP						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
MSURF_S	ID of a set of mesh surface parts created with a *LSO_ID_SET card, where each mesh surface part ID in the set is referenced in *MESH_SURFACE_ELEMENT cards.
IDCOMP	For inflow boundaries in problems involving chemical reacting flows, this ID references a *CHEMISTRY_COMPOSITION card that gives the chemical mixture of the fluid entering the domain

Set Card. Card 1 format for the SET keyword option

Card 1c	1	2	3	4	5	6	7	8
Variable	SSID	IDCOMP						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
SSID	Segment set ID
IDCOMP	For inflow boundaries in problems involving chemical reacting flows, this ID references a *CHEMISTRY_COMPOSITION card that gives the chemical mixture of the fluid entering the domain

Segment Card. Card 1 for the SEGMENT keyword option

Card 1d	1	2	3	4	5	6	7	8
Variable	N1	N1	N3	N4	IDCOMP			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
N1, N2, …	Node IDs defining a segment

VARIABLE	DESCRIPTION
IDCOMP	For inflow boundaries in problems involving chemical reacting flows, this ID references a *CHEMISTRY_COMPOSITION card that gives the chemical mixture of the fluid entering the domain

Load Curve Card.

Card 2	1	2	3	4	5	6	7	8
Variable	LC_U	LC_V	LC_W	LC_RHO	LC_P	LC_T		
Type	I	I	I	I	I	I		
Remarks	1	1	1	1	1	1		

VARIABLE	DESCRIPTION
LC_U	Load curve ID to describe the x-component of the velocity as a function of time; see *DEFINE_CURVE. GT.0: Load curve ID EQ.0: Constant value of SF_U EQ.-1: Solver computes the boundary value for x-component of velocity
LC_V	Load curve ID to describe the y-component of the velocity as a function of time GT.0: Load curve ID EQ.0: Constant value of SF_V EQ.-1: Solver computes the boundary value for y-component of velocity
LC_W	Load curve ID to describe the z-component of the velocity as a function of time GT.0: Load curve ID EQ.0: Constant value of SF_W EQ.-1: Solver computes the boundary value for x-component of velocity

VARIABLE	DESCRIPTION
LC_RHO	Load curve ID to describe the density as a function of time GT.0: Load curve ID EQ.0: Constant value of SF_RHO EQ.-1: Solver computes the boundary value for density
LC_P	Load curve ID to describe the pressure as a function of time GT.0: Load curve ID EQ.0: Constant value of SF_P EQ.-1: Solver computes the boundary value for pressure
LC_T	Load curve ID to describe the temperature as a function of time GT.0: Load curve ID EQ.0: Constant value of SF_T EQ.-1: Solver computes the boundary value for temperature

Scale Factor Card.

Card 3	1	2	3	4	5	6	7	8
Variable	SF_U	SF_V	SF_W	SF_RHO	SF_P	SF_T		
Type	F	F	F	F	F	F		
Default	1.0	1.0	1.0	1.0	1.0	1.0		

VARIABLE	DESCRIPTION
SF_U	Scale factor for LC_U
SF_V	Scale factor for LC_V
SF_W	Scale factor for LC_W
SF_RHO	Scale factor for LC_RHO
SF_P	Scale factor for LC_P
SF_T	Scale factor for LC_T

Remarks:

1. **Well-posed boundaries.** On each centroid or set of centroids, the variables (v_x, v_y, v_z, ρ, P, T) with provided values must be consistent and make the model well-posed (that is, be such that the solution of the model exists, is unique, and is physical).

CESE_BOUNDARY_PRESCRIBED_VN_*OPTION

Available options include:

MSURF

MSURF_SET

SET

SEGMENT

Purpose: For the CESE compressible flow solver, set boundary values for velocity, density, pressure, and temperature. Boundary values are applied at the centroid of elements connected with this boundary. Use keyword options SET or SEGMENT for user-defined meshes, that is, meshes specified with *ELEMENT_SOLID cards. Use MSURF or MSURF_SET for meshes generated with the automatic volume mesher, meaning created with *MESH cards.

Card Summary:

A set of data cards for this keyword consists of 3 of the following cards:

1. Card 1 specifies the object to which the boundary condition is applied. Its format depends on the keyword option.
2. Card 2 reads in load curve IDs.
3. Card 3 reads in scale factors.

For each boundary condition, include one set of cards. This input ends at the next keyword ("*") card.

Card 1a. Include this card for the MSURF keyword option.

MSURFID	IDCOMP						

Card 1b. Include this card for the MSURF_SET keyword option.

MSURF_S	IDCOMP						

Card 1c. Include this card for the SET keyword option.

SSID	IDCOMP						

Card 1d. Include this card for the SEGMENT keyword option.

N1	N1	N3	N4	IDCOMP			

Card 2. This card is required.

LC_VN			LC_RHO	LC_P	LC_T		

Card 3. This card is required.

SF_VN			SF_RHO	SF_P	SF_T		

Data Card Definitions:

Surface Part Card. Card 1 format for the MSURF keyword option

Card 1a	1	2	3	4	5	6	7	8
Variable	MSURFID	IDCOMP						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
MSURFID	A mesh surface part ID referenced in *MESH_SURFACE_ELEMENT cards
IDCOMP	For inflow boundaries in problems involving chemical reacting flows, this ID references a *CHEMISTRY_COMPOSITION card that gives the chemical mixture of the fluid entering the domain.

.

Surface Part Set Card. Card 1 format for the MSURF_SET keyword option

Card 1b	1	2	3	4	5	6	7	8
Variable	MSURF_S	IDCOMP						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
MSURF_S	ID of a set of mesh surface parts created with a *LSO_ID_SET card, where each mesh surface part ID in the set is referenced in *MESH_SURFACE_ELEMENT cards.
IDCOMP	For inflow boundaries in problems involving chemical reacting flows, this ID references a *CHEMISTRY_COMPOSITION card that gives the chemical mixture of the fluid entering the domain.

Set Card. Card 1 format for the SET keyword option

Card 1c	1	2	3	4	5	6	7	8
Variable	SSID	IDCOMP						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
SSID	Segment set ID
IDCOMP	For inflow boundaries in problems involving chemical reacting flows, this ID references a *CHEMISTRY_COMPOSITION card that gives the chemical mixture of the fluid entering the domain.

Segment Card. Card 1 for the SEGMENT keyword option

Card 1d	1	2	3	4	5	6	7	8
Variable	N1	N1	N3	N4	IDCOMP			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
N1, N2, …	Node IDs defining a segment

VARIABLE	DESCRIPTION
IDCOMP	For inflow boundaries in problems involving chemical reacting flows, this ID references a *CHEMISTRY_COMPOSITION card that gives the chemical mixture of the fluid entering the domain.

Load Curve Card.

Card 2	1	2	3	4	5	6	7	8
Variable	LC_VN			LC_RHO	LC_P	LC_T		
Type	I			I	I	I		
Remarks	1			1	1	1		

VARIABLE	DESCRIPTION
LC_VN	Load curve ID to describe the normal velocity as a function of time; see *DEFINE_CURVE. GT.0: Load curve ID EQ.0: Constant value of SF_VN EQ.-1: Solver determines the boundary value for the normal velocity
LC_RHO	Load curve ID to describe the density as a function of time GT.0: Load curve ID EQ.0: Constant value of SF_RHO EQ.-1: Solver determines the boundary value for the density
LC_P	Load curve ID to describe the pressure as a function of time GT.0: Load curve ID EQ.0: Constant value of SF_P EQ.-1: Solver determines the boundary value for the pressure
LC_T	Load curve ID to describe the temperature as a function of time GT.0: Load curve ID EQ.0: Constant value of SF_T

VARIABLE	DESCRIPTION
	EQ.-1: Solver determines the boundary value for the temperature

Scale Factor Card.

Card 3	1	2	3	4	5	6	7	8
Variable	SF_VN			SF_RHO	SF_P	SF_T		
Type	F			F	F	F		
Default	1.0			1.0	1.0	1.0		

VARIABLE	DESCRIPTION
SF_VN	Scale factor for LC_VN
SF_RHO	Scale factor for LC_RHO
SF_P	Scale factor for LC_P
SF_T	Scale factor for LC_T

Remarks:

1. **Well-posed boundary conditions.** On each centroid or set of centroids, the variables (V_N, ρ, P, T) with provided values must be consistent and make the model well-posed (that is, be such that the solution of the model exists, is unique and physical).

CESE_BOUNDARY_REFLECTIVE_*OPTION

Available options are:

MSURF

MSURF_SET

SET

SEGMENT

Purpose: Define a reflective boundary condition for the CESE compressible flow solver. You can apply this boundary condition on a symmetrical surface or a solid wall of the computational domain. Use the MSURF or MSURF_SET keyword options when generating the CESE mesh with the *MESH cards. Use the SET or SEGMENT keyword options cards when *ELEMENT_SOLID cards specify CESE mesh.

Card Summary:

Card 1a. Include this card for the MSURF keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

MSURFID							

Card 1b. Include this card for the MSURF_SET keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

MSURF_S							

Card 1c. Include this card for the SET keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

SSID							

Card 1d. Include this card for the SEGMENT keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

N1	N2	N3	N4				

Data Card Definitions:

Surface Part Card. Card 1 format for the MSURF keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSURFID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSURFID	Mesh surface part ID referenced in *MESH_SURFACE_ELEMENT cards.

Surface Part Set Card. Card 1 format for the MSURF_SET keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1b	1	2	3	4	5	6	7	8
Variable	MSURF_S							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSURF_S	ID of a set of mesh surface parts created with a *LSO_ID_SET card, where each mesh surface part ID in the set is referenced in *MESH_SURFACE_ELEMENT cards.

Set Card. Card 1 format for the SET keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1c	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
SSID	Segment set ID

Segment Cards. Card 1 format for the SEGMENT keyword option. Include an additional card for each corresponding pair of segments. This input ends at the next keyword ("*") card.

Card 1d	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
N1, N2, …	Node IDs defining a segment

Remarks:

1. **Effect of the boundary condition.** This boundary condition has the same effect as a solid-wall boundary condition for inviscid flows.

CESE_BOUNDARY_SLIDING_*OPTION

Available options are:

MSURF

MSURF_SET

SET

SEGMENT

Purpose: Allows the nodes of a fluid surface to translate in the main direction of the mesh movement. This feature is useful in piston-type applications.

Use the MSURF or MSURF_SET keyword options when generating the CESE mesh with *MESH cards. Use the SET and SEGMENT keywords when *ELEMENT_SOLID cards specify the CESE mesh.

Card Summary:

Card 1a. Include this card for the MSURF keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

MSURFID							

Card 1b. Include this card for the MSURF_SET keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

MSURF_S							

Card 1c. Include this card for the SET keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

SSID							

Card 1d. Include this card for the SEGMENT keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

N1	N2	N3	N4				

Data Card Definitions:

Surface Part Card. Card 1 format for the MSURF keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSURFID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSURFID	Mesh surface part ID referenced in *MESH_SURFACE_ELEMENT cards.

Surface Part Set Card. Card 1 format used when the MSURF_SET keyword option is active. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1b	1	2	3	4	5	6	7	8
Variable	MSURF_S							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSURF_S	ID of a set of mesh surface parts created with a *LSO_ID_SET card, where each mesh surface part ID in the set is referenced in *MESH_SURFACE_ELEMENT cards.

Set Card. Card 1 format used when the SET keyword option is active. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1c	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
SSID	Segment set ID

Segment Cards. Card 1 format used when SEGMENT keyword option is active. Include an additional card for each corresponding pair of segments. This input ends at the next keyword ("*") card.

Card 1d	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
N1, N2, …	Node IDs defining a segment

CESE_BOUNDARY_SOLID_WALL_*OPTION1_{OPTION2}

For *OPTION1*, the choices are:

MSURF

MSURF_SET

SET

SEGMENT

For *OPTION2*, the choices are:

<BLANK>

ROTAT

Purpose: Define a solid wall boundary condition for this CESE compressible flow solver. This boundary condition models a solid boundary, that is, the physical boundary for the flow field. This boundary condition is a slip boundary condition for inviscid flows and a no-slip boundary condition for viscous flows.

Use the MSURF or MSURF_SET options when generating the CESE mesh through the *MESH cards. Use the SET or SEGMENT keyword options cards when *ELEMENT_-SOLID cards define the CESE mesh. Use the ROTAT keyword option for a rotating solid boundary condition.

Card Summary:

Card Sets. The following sequence of cards comprises a *single set*. LS-DYNA continues reading *CESE_BOUNDARY_SOLID_WALL card sets until the next keyword ("*") card.

Card 1a. Include this card when *OPTION1* is MSURF and *OPTION2* is not set (<BLANK>).

MSURFID	LCID	VX	VY	VZ			

Card 1b. Include this card when *OPTION1* is MSURF and *OPTION2* is ROTAT.

MSURFID	LCID	VX	VY	VZ	NX	NY	NZ

Card 1c. Include this card when *OPTION1* is MSURF_SET and *OPTION2* is not set (<BLANK>).

MSURF_S	LCID	VX	VY	VZ			

Card 1d. Include this card when *OPTION1* is MSURF_SET and *OPTION2* is ROTAT.

MSURF_S	LCID	VX	VY	VZ	NX	NY	NZ

Card 1e. Include this card when *OPTION1* is SET and *OPTION2* is not set (<BLANK>).

SSID	LCID	VX	VY	VZ			

Card 1f. Include this card when *OPTION1* is SET and *OPTION2* is ROTAT.

SSID	LCID	VX	VY	VZ	NX	NY	NZ

Card 1g. Include this card when *OPTION1* is SEGMENT and *OPTION2* is not set (<BLANK>).

N1	N2	N3	N4	LCID	VX	VY	VZ

Card 1h.1. Include this card when *OPTION1* is SEGMENT and *OPTION2* is ROTAT.

N1	N2	N3	N4	LCID	VX	VY	VZ

Card 1h.2. Include this card when *OPTION1* is SEGMENT and *OPTION2* is ROTAT.

N1	N2	N3					

Data Card Definitions:

Surface Part Card (Non-Rotating Boundary). Card 1 format used when the MSURF keyword option is active and the ROTAT keyword option is inactive.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSURFID	LCID	VX	VY	VZ			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			
Remarks		2	2	2	2			

VARIABLE	DESCRIPTION
MSURFID	Mesh surface part ID referenced in *MESH_SURFACE_ELEMENT cards.
LCID	Load curve ID to define this solid wall boundary movement
VX, VY, VZ	Velocity vector of the solid wall: LCID.EQ.0: Vector defined by (VX, VY, VZ). LCID.NE.0: Vector defined by both the load curve and (VX, VY, VZ).

Surface Part Card (Rotating Boundary). Card 1 format used when the MSURF keyword option is active and the ROTAT keyword option is active.

Card 1b	1	2	3	4	5	6	7	8
Variable	MSURFID	LCID	VX	VY	VZ	NX	NY	NZ
Type	I	I	F	F	F	F	F	F
Default	none	none	0.0	0.0	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
MSURFID	Mesh surface part ID referenced in *MESH_SURFACE_ELEMENT cards.
LCID	Load curve ID giving the rotating speed frequency (HZ). LCID must be specified.
VX, VY, VZ	x, y, and z-coordinates of a point on the rotating axis
NX, NY, NZ	Unit vector of the rotating axis (not used for the 2D case)

Surface Part Set Card (Non-Rotating Boundary). Card 1 format used when the MSURF_SET keyword option is active and the ROTAT keyword option is not active.

Card 1c	1	2	3	4	5	6	7	8
Variable	MSURF_S	LCID	VX	VY	VZ			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			
Remarks		2	2	2	2			

VARIABLE	DESCRIPTION
MSURF_S	ID of a set of mesh surface part IDs created with a *LSO_ID_SET card, where each mesh surface part ID in the set is referenced in *MESH_SURFACE_ELEMENT cards.
LCID	Load curve ID to define this solid wall boundary movement
VX, VY, VZ	Velocity vector of the solid wall: LCID.EQ.0: Vector defined by (VX, VY, VZ). LCID.NE.0: Vector defined by both the load curve and (VX, VY, VZ).

Surface Part Set Card (Rotating Boundary). Card 1 format used when the MSURF_SET keyword option is active and the ROTAT keyword option is active.

Card 1d	1	2	3	4	5	6	7	8
Variable	MSURF_S	LCID	VX	VY	VZ	NX	NY	NZ
Type	I	I	F	F	F	F	F	F
Default	none	0	0.0	0.0	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
MSURF_S	ID of a set of mesh surface part IDs created with a *LSO_ID_SET card, where each mesh surface part ID in the set is referenced in *MESH_SURFACE_ELEMENT cards.
LCID	Load curve ID giving the rotating speed frequency (HZ). LCID must be specified.
VX, VY, VZ	x, y, and z-coordinates of a point on the rotating axis
NX, NY, NZ	Unit vector of the rotating axis (not used for the 2D case)

Segment Set Card (Non-Rotating Boundary). Card 1 format used when the SET keyword option is active and the ROTAT keyword option is not active.

Card 1e	1	2	3	4	5	6	7	8
Variable	SSID	LCID	VX	VY	VZ			
Type	I	I	F	F	F			
Default	none	none	0.0	0.0	0.0			
Remarks		2	2	2	2			

VARIABLE	DESCRIPTION
SSID	Segment set ID
LCID	Load curve ID to define this solid wall boundary movement
VX, VY, VZ	Velocity vector of the solid wall: LCID.EQ.0: Vector defined by (VX, VY, VZ). LCID.NE.0: Vector defined by both the load curve and (VX, VY, VZ).

Segment Set Card (Rotating Boundary). Card 1 format used when the SET keyword option is active and the ROTAT keyword option is active.

Card 1f	1	2	3	4	5	6	7	8
Variable	SSID	LCID	VX	VY	VZ	NX	NY	NZ
Type	I	I	F	F	F	F	F	F
Default	none	none	0.0	0.0	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
SSID	Segment set ID
LCID	Load curve ID giving the rotating speed frequency (HZ). LCID must be specified.
VX, VY, VZ	x, y, and z-coordinates of a point on the rotating axis
NX, NY, NZ	Unit vector of the rotating axis (not used for the 2D case)

Segment Card (Non-Rotating Boundary). Card 1 format used when the SEGMENT keyword option is active and the ROTAT keyword option is not active.

Card 1g	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4	LCID	VX	VY	VZ
Type	I	I	I	I	I	F	F	F
Default	none	none	none	none	0	0.0	0.0	0.0
Remarks					2	2	2	2

VARIABLE	DESCRIPTION
N1, N2, N3, N4	Node IDs defining a segment
LCID	Load curve ID to define this solid wall boundary movement

VARIABLE	DESCRIPTION
VX, VY, VZ	Velocity vector of the solid wall: LCID.EQ.0: Vector defined by (VX, VY, VZ). LCID.NE.0: Vector defined by both the load curve and (VX, VY, VZ).

Segment Card (Rotating Boundary). Card 1 format used when the SEGMENT keyword option is active and the ROTAT keyword option is active.

Card 1h.1	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4	LCID	VX	VY	VZ
Type	I	I	I	I	I	F	F	F
Default	none	none	none	none	0	0.0	0.0	0.0

Rotating Axis Card for Segment. Additional card for the "Segment Card (Rotating Boundary)" case that is read when the ROTAT keyword option is used.

Card 1h.2	1	2	3	4	5	6	7	8
Variable	NX	NY	NZ					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
N1, N2, N3, N4	Node IDs defining a segment
LCID	Load curve ID giving the rotating speed frequency (HZ). LCID must be specified.
VX, VY, VZ	x, y, and z-coordinates of a point on the rotating axis
NX, NY, NZ	Unit vector of the rotating axis (not used for the 2D case)

Remarks:

1. **Boundary movement.** In this solid-wall condition, the boundary movement can only be in the tangential direction of the wall and should not affect the fluid domain size and mesh during the calculation. Otherwise, you should use an FSI or moving mesh solver. Also, this moving solid boundary condition only affects viscous flows (no-slip BC).

2. **Regular solid wall boundary condition.** If LCID = 0 and VX = VY = VZ = 0.0 (default), this boundary condition is a regular solid wall boundary condition.

*CESE_CHEMISTRY_D3PLOT

Purpose: Cause mass fractions of the listed chemical species to be added to the CESE d3plot output. This is only used when chemistry is being solved with the CESE solver.

Card 1	1	2	3	4	5	6	7	8
Variable	MODELID							
Type	I							
Default	none							

Species Cards. Include one card for each species to be included in the d3plot database. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	SPECIES							
Type	A							

VARIABLE	DESCRIPTION
MODELID	Identifier of a Chemkin-compatible chemistry model.
SPECIES	Name of a chemical species that is defined in the chemistry model identified by MODELID (see *CHEMISTRY_MODEL).

*CESE_CONTROL_LIMITER

Purpose: Sets some stability parameters used in the CESE scheme for this CESE compressible flow solver.

Card 1	1	2	3	4	5	6	7	8
Variable	IDLMT	ALFA	BETA	EPSR				
Type	I	F	F	F				
Default	0	0.0	0.0	0.0				
Remarks		1	2	3				

VARIABLE	DESCRIPTION
IDLMT	Set the stability limiter option (See CESE theory manual): EQ.0: limiter format 1 (Re-weighting). EQ.1: limiter format 2 (Relaxing).
ALFA	Re-weighting coefficient (See CESE theory manual)
BETA	Numerical viscosity control coefficient (See CESE theory manual)
EPSR	Stability control coefficient (See CESE theory manual)

Remarks:

1. $\alpha \geq 0$; larger values give more stability, but less accuracy. Usually $\alpha = 2.0$ or 4.0 will be enough for normal shock problems.

2. $0 \leq \beta \leq 1$; larger values give more stability. For problems with shock waves, $\beta = 1.0$ is recommended.

3. $\varepsilon \geq 0$; larger values give more stability, but less accuracy.

*CESE_CONTROL_MESH_MOV

Purpose: For moving mesh CESE, this keyword is used to choose the type of algorithm to be used for calculating mesh movement.

Card 1	1	2	3	4	5	6	7	8
Variable	MMSH	LIM_ITER	RELTOL					
Type	I	I	F					
Default	1	100	1.0e-3					

VARIABLE	DESCRIPTION
MMSH	Mesh motion selector: EQ.1: mesh moves using an implicit ball-vertex spring method. EQ.9: the IDW scheme is used to move the mesh.
LIM_ITER	Maximum number of linear solver iterations for the ball-vertex linear system.
RELTOL	Relative tolerance to use as a stopping criterion for the iterative linear solver (conjugate gradient solver with diagonal scaling preconditioner).

.

*CESE_CONTROL_SOLVER

Purpose: Set general purpose control variables for the CESE compressible flow solver.

Card 1	1	2	3	4	5	6	7	8
Variable	ICESE	IFLOW	IGEOM	IFRAME	MIXID	IDC	ISNAN	
Type	I	I	I	I	I	F	I	
Default	0	0	none	0	none	0.25	0	
Remarks			1			2		

VARIABLE	DESCRIPTION
ICESE	Sets the framework of the CESE solver. EQ.0: Fixed Eulerian EQ.100: Moving Mesh FSI EQ.200: Immersed boundary FSI
IFLOW	Sets the compressible flow types: EQ.0: Viscous flows (laminar) EQ.1: Inviscid flows
IGEOM	Sets the geometric dimension: EQ.2: Two-dimensional (2D) problem EQ.3: Three-dimensional (3D) problem EQ.101: 2D axisymmetric
IFRAME	Choose the frame of reference: EQ.0: Usual non-moving reference frame (default) EQ.1000: Non-inertial rotating reference frame
MIXID	Chemistry model ID that defines the chemical species to include in the mixing model (see *CHEMISTRY_MODEL). The species information is given through the model's card specifying the Chemkin-compatible input.

VARIABLE	DESCRIPTION
IDC	Contact interaction detection coefficient (for FSI and conjugate heat transfer problems).
ISNAN	Flag to check for a NaN in the CESE solver solution arrays at the completion of each time step. This option can be useful for debugging purposes. There is a cost overhead when this option is active. EQ.0: No checking EQ.1: Checking is active.

Remarks:

1. **2D Solvers and Mesh Geometry.** For the 2D (IGEOM = 2) or 2D axisymmetric (IGEOM = 101) solver, the mesh should only be distributed in the x-y plane with the boundary conditions given only at the x-y domain boundaries. Otherwise, a warning message will be given, and the 3D solver will be triggered instead.

   The 2D axisymmetric case will work only if the 2D mesh and corresponding boundary conditions are properly defined, with the x and y coordinates corresponding to the radial and axial directions, respectively.

2. **Contact Interaction Detection Coefficient.** IDC is the same type of variable that is input on the *ICFD_CONTROL_FSI card. For an explanation, see Remark 1 for the *ICFD_CONTROL_FSI card.

3. **CESE Solver and Restarts.** Restarts are supported for the CESE solver. However, restarts do not currently work for simulations with conjugate heat transfer.

*CESE_CONTROL_TIMESTEP

Purpose: Sets the time-step control parameters for the CESE compressible flow solver.

Card 1	1	2	3	4	5	6	7	8
Variable	IDDT	CFL	DTINT					
Type	I	F	F					
Default	0	0.9	10^{-3}					

VARIABLE	DESCRIPTION
IDDT	Sets the time step option: EQ.0: fixed time step size (DTINT, meaning the given initial time step size) NE.0: the time step size will be calculated based on the given CFL-number and the flow solution at the previous time step.
CFL	CFL number (Courant–Friedrichs–Lewy condition) ($0.0 < CFL \leq 1.0$)
DTINT	Initial time step size

*CESE_DATABASE_ELOUT

Purpose: Enable the output of CESE data on elements. If more than one element set is defined, then several output files will be generated.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	OUTLV	DTOUT						
Type	I	F						
Default	0	0.						

Element Sets Card. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	ELSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
OUTLV	Determines if the output file should be created: EQ.0: No output file is generated. EQ.1: The output file is generated.
DTOUT	Time interval to print the output. If DTOUT is equal to 0.0, then the CESE timestep will be used.
ELSID	Solid elements set ID

Remarks:

1. **Database File Name.** The file name for this database is cese_elout.dat.

2. **Restrictions.** The *CESE_DATABASE… cards (pointout/elout) are restricted to CFD-only problems. They do not give correct results for chemically reacting flow where *CESE couples with *CHEMISTRY.

*CESE_DATABASE_FLUXAVG

Purpose: Enable the output of CESE data on segment sets. If more than one segment set is defined, then several output files will be generated.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	OUTLV	DTOUT						
Type	I	F						
Default	0	0.						

Segment Sets Card. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
OUTLV	Determines if the output file should be created: EQ.0: No output file is generated. EQ.1: The output file giving the average fluxes is generated.
DTOUT	Time interval to print the output. If DTOUT is equal to 0.0, then the CESE time step will be used.
SSID	Segment set ID

Remarks:

1. **Database File Name.** The file name for this database is cese_fluxavg.dat.

2. **Restrictions.** The *CESE_DATABASE… cards (pointout/elout/…) are restricted to CFD-only problems. They do not give correct results for chemically reacting flow where *CESE couples with *CHEMISTRY.

*CESE_DATABASE_FSIDRAG

Purpose: Enable the output of the total fluid pressure force applied to solid parts in FSI problems at every time step.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	OUTLV							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
OUTLV	Determines if the output file should be created: EQ.0: No output file is generated. EQ.1: The output file giving the pressure forces is generated.

Remarks:

1. **Database File Names.** The file names for this database are cese_dragsol.dat, cese_dragshell.dat, cese_dragsol2D.dat and cese_dragbeam.dat, depending on what kind of solid is used.

*CESE_DATABASE_POINTOUT

Purpose: Enable the output of CESE data at points.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID	DTOUT	PSTYPE	VX	VY	VZ		
Type	I	F	I	F	F	F		
Default	0	0.	0	0.	0.	0.		

Point Specification Card. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	PID	X	Y	Z				
Type	I	F	F	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
PSID	Point set ID
DTOUT	Time interval to print the output. If DTOUT is equal to 0.0, then the CESE timestep will be used.
PSTYPE	Point set type: EQ.0: Fixed points EQ.1: Tracer points using prescribed velocity EQ.2: Tracer points using fluid velocity
VX, VY, VZ	Constant velocities to be used when PSTYPE = 1
PID	Point ID

VARIABLE	DESCRIPTION
X, Y, Z	Initial coordinates of the point

Remarks:

1. **Database File Name.** The file name for this database is cese_pointout.dat.

2. **Restrictions.** The *CESE_DATABASE… cards (pointout/elout) are restricted to CFD-only problems. They do not give correct results for chemically reacting flow where *CESE couples with *CHEMISTRY.

*CESE_DATABASE_SSETDRAG

Purpose: Enable the output of CESE drag forces on segment sets. If more than one segment set is specified, then several output files will be generated.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	OUTLV	DTOUT						
Type	I	F						
Default	0	0.						

Segment Sets Card. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
OUTLV	Determines if the output file should be created: EQ.0: No output file is generated. EQ.1: The output file giving the average fluxes is generated.
DTOUT	Time interval to print the output. If DTOUT is equal to 0.0, then the CESE timestep will be used.
SSID	Segment set ID

Remarks:

1. **Database File Name.** The file name for this database is cese_ssetdrag.dat.

2. **Friction Drag Results and Mesh Size.** For the friction drag to give consistent results, special care must be given to the mesh close to the solid wall boundary (good capturing of the boundary layer behavior). We recommend a very fine structured mesh.

*CESE_DEFINE_NONINERTIAL

Purpose: Define the CESE problem domain as a non-inertial rotating frame that rotates at a constant rate. This is used in rotating problems such as spinning cylinders, wind turbines and turbo machinery.

Card 1	1	2	3	4	5	6	7	8
Variable	FREQ	LCID	PID	Nx	Ny	Nz		
Type	F	I	I	F	F	F		
Default	none	0	none	none	none	none		

Card 2	1	2	3	4	5	6	7	8
Variable	L	R	RELV					
Type	F	F	I					
Default	none	none	0					

VARIABLE	DESCRIPTION
FREQ	Frequency of rotation.
LCID	Load curve ID for scaling factor of FREQ.
PID	Starting point ID for the reference frame (See *CESE_DEFINE_-POINT).
Nx, Ny, Nz	Rotating axis direction.
L	Length of rotating frame.
R	Radius of rotating frame.

VARIABLE	DESCRIPTION
RELV	Velocity display mode: EQ.0: Relative velocity, only the non-rotating components of the velocity are output. EQ.1: Absolute velocity is output.

*CESE_DEFINE_POINT

Purpose: Define points to be used by the CESE solver.

Point Cards. Include one card for each point. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	NID	X	Y	Z				
Type	I	F	F	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
NID	Identifier for this point.
X, Y, Z	Coordinates of the point.

*CESE_DRAG

Purpose: Provide the far-field (or free-stream) fluid pressure.

Card 1	1	2	3	4	5	6	7	8
Variable	PRESS							
Type	F							

VARIABLE	DESCRIPTION
PRESS	Value of the free-stream fluid pressure (in units used by the current problem).

*CESE_EOS_CAV_HOMOG_EQUILIB

Purpose: Define the coefficients in the equation of state (EOS) for the homogeneous equilibrium cavitation model.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	ρ_{vap}	ρ_{liq}	a_{vap}	a_{liq}	μ_{vap}	μ_{liq}	P_{SatVap}
Type	I	F	F	F	F	F	F	F
Default	none	0.8	880.0	334.0	1386.0	1.435e-5	1.586e-4	1.2e+4

VARIABLE	DESCRIPTION
EOSID	Equation of state identifier
ρ_{vap}	density of the saturated vapor
ρ_{liq}	density of the saturated liquid
a_{vap}	sound speed of the saturated vapor
a_{liq}	sound speed of the saturated liquid
μ_{vap}	dynamic viscosity of the vapor
μ_{liq}	dynamic viscosity of the liquid
P_{SatVap}	pressure of the saturated vapor

Remarks:

1. Once a cavitation EOS is used, the cavitation flow solver will be triggered.

2. In this homogeneous equilibrium cavitation model, a barotropic equation of state is used. This model can be used in small scale & high speed cavitation flows, and it is not good for large-scale, low-speed cavitation calculations.

*CESE_EOS_IDEAL_GAS

Purpose: Define the coefficients Cv and Cp in the equation of state for an ideal gas in the CESE fluid solver.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	Cv	Cp					
Type	I	F	F					
Default	none	717.5	1004.5					

VARIABLE	DESCRIPTION
EOSID	Equation of state identifier
Cv	Specific heat at constant volume
Cp	Specific heat at constant pressure

Remarks:

1. **Units.** As with other solvers in LS-DYNA, the user is responsible for unit consistency. For example, if a user wants to use dimensionless variables, Cv and Cp should also be replaced by the corresponding dimensionless ones.

*CESE_EOS_INFLATOR1

Purpose: To define an EOS using Cp and Cv thermodynamic expansions for an inflator gas mixture with a single temperature range.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID							
Type	I							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	Cp0	Cp1	Cp2	Cp3	Cp4			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

Card 3	1	2	3	4	5	6	7	8
Variable	Cv0	Cv1	Cv2	Cv3	Cv4			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

VARIABLE	DESCRIPTION
EOSID	Equation of state identifier for the CESE solver.
Cp0, …, Cp4	Coefficients of temperature-dependent specific heat at constant pressure $C_p(T) = C_{p0} + C_{p1}\,T + C_{p2}\,T^2 + C_{p3}\,T^3 + C_{p4}\,T^4$

VARIABLE	DESCRIPTION
Cv0, …, Cv4	Coefficients of temperature-dependent specific heat at constant volume $C_v(T) = C_{v0} + C_{v1}\,T + C_{v2}\,T^2 + C_{v3}\,T^3 + C_{v4}\,T^4$

Remark:

1. These coefficient expansions for the specific heats over the entire temperature range are generated by the 0-D inflator model solver. See *CHEMISTRY_CONTROL_-INFLATOR and *CHEMISTRY_INFLATOR_PROPERTIES for details related to running that solver.

*CESE_EOS_INFLATOR2

Purpose: To define an EOS using Cp and Cv thermodynamic expansions for an inflator gas mixture with two temperature ranges, one below 1000 degrees Kelvin, and the other above 1000 degrees Kelvin.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID							
Type	I							
Default	none							

Card for the expansion of Specific Heat at Constant Pressure. Valid for T < 1000 ^{0}K

Card 2	1	2	3	4	5	6	7	8
Variable	Cp1_0	Cp1_1	Cp1_2	Cp1_3	Cp1_4			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

Card for the expansion of Specific Heat at Constant Pressure. Valid for T > 1000 ^{0}K.

Card 3	1	2	3	4	5	6	7	8
Variable	Cp2_0	Cp2_1	Cp2_2	Cp2_3	Cp2_4			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

Card for the expansion of Specific Heat at Constant Volume. Valid for T < 1000 ^{0}K

Card 4	1	2	3	4	5	6	7	8
Variable	Cv1_0	Cv1_1	Cv1_2	Cv1_3	Cv1_4			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

Card for the expansion of Specific Heat at Constant Volume. Valid for T > 1000 ^{0}K.

Card 5	1	2	3	4	5	6	7	8
Variable	Cv2_0	Cv2_1	Cv2_2	Cv2_3	Cv2_4			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

VARIABLE	DESCRIPTION
EOSID	Equation of state identifier for the CESE solver.
Cp1_0, …, Cp1_4	Coefficients of temperature-dependent specific heat at constant pressure valid for T < 1000 ^{0}K. $C_{p1}(T) = C_{p1_0} + C_{p1_1}\,T + C_{p1_2}\,T^2 + C_{p1_3}\,T^3 + C_{p1_4}\,T^4$
Cp2_0, …, Cp2_4	Coefficients of temperature-dependent specific heat at constant pressure valid for T > 1000 ^{0}K. $C_{p2}(T) = C_{p2_0} + C_{p2_1}\,T + C_{p2_2}\,T^2 + C_{p2_3}\,T^3 + C_{p2_4}\,T^4$
Cv1_0, …, Cv1_4	Coefficients of temperature-dependent specific heat at constant volume valid for T < 1000 ^{0}K. $C_{v1}(T) = C_{v1_0} + C_{v1_1}\,T + C_{v1_2}\,T^2 + C_{v1_3}\,T^3 + C_{v1_4}\,T^4$
Cv2_0, …, Cv2_4	Coefficients of temperature-dependent specific heat at constant volume valid for T > 1000 ^{0}K. $C_{v2}(T) = C_{v2_0} + C_{v2_1}\,T + C_{v2_2}\,T^2 + C_{v2_3}\,T^3 + C_{v2_4}\,T^4$

Remark:

2. These coefficient expansions for the specific heats over two temperature ranges are generated by the 0-D inflator model solver. See *CHEMISTRY_CONTROL_INFLATOR and *CHEMISTRY_INFLATOR_PROPERTIES for details related to running that solver.

*CESE_FSI_EXCLUDE

Purpose: Provide a list of mechanics solver parts that are not involve in the CESE FSI calculation. This is intended to be used as an efficiency measure for parts that will not involve significant FSI interactions with the CESE compressible fluid solver..

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID1	PID2	PID3	PID4	PID5	PID6	PID7	PID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
PIDn	IDs of mechanics parts that will be excluded from the FSI interaction calculation with the CESE solver.

*CESE_INITIAL

Purpose: Specify constant initial conditions (ICs) for flow variables at the centroid of each fluid element.

Card 1	1	2	3	4	5	6	7	8
Variable	U	V	W	RH	P	T		
Type	F	F	F	F	F	F		
Default	0	0.0	0.0	1.225	0.0	0.0		

VARIABLE	DESCRIPTION
U, V, W	x-, y-, z-velocity components, respectively
RHO	Density, ρ
P	Pressure, P
T	Temperature, T

Remarks:

1. **Required Input.** Usually, only two of ρ, P, and T need to be specified (besides the velocity). If all three are given, only ρ and P will be used.

2. **Applicable Elements.** These initial conditions will be applied only in those elements that have not been assigned a value by *CESE_INITIAL_*OPTION* cards for individual elements or sets of elements.

CESE_INITIAL_*OPTION

Available options include:

SET

ELEMENT

Purpose: Specify initial conditions for the flow variables at the centroid of each element in a set of elements or at the centroid of a single element.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	EID/ESID	U	V	W	RHO	P	T	
Type	I	F	F	F	F	F	F	
Default	none	0.0	0.0	0.0	1.225	0.0	0.0	
Remarks					1	1	1	

VARIABLE	DESCRIPTION
EID/ESID	Solid element ID (EID) or solid element set ID (ESID)
U, V, W	*x*-, *y*-, *z*-velocity components, respectively
RHO	Density, ρ
P	Pressure, P
T	Temperature, T

Remarks:

1. **Required Input.** Usually, only two of ρ, P, and T need to be specified (along with the velocity). If all three are given, only ρ and P will be used.

2. **Initial Condition Specification Priority.** The priority of this card is higher than *CESE_INITIAL, meaning that if an element is assigned an initial value by this card, *CESE_INITIAL will no longer apply to that element.

*CESE_INITIAL_CHEMISTRY

Purpose: Initializes the chemistry and fluid state in every element of the CESE mesh that has not already been initialized by one of the other *CESE_INITIAL_CHEMISTRY cards. This is only used when chemistry is being solved with the CESE solver.

Card 1	1	2	3	4	5	6	7	8
Variable	CHEMID	COMPID						
Type	I	I						
Default	none	none						

Card 2	1	2	3	4	5	6	7	8
Variable	UIC	VIC	WIC	RHOIC	PIC	TIC	HIC	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
CHEMID	Identifier of chemistry control card to use
COMPID	Identifier of chemical composition to use
UIC	*X*-component of the fluid velocity
VIC	*Y*-component of the fluid velocity
WIC	*Z*-component of the fluid velocity
RHOIC	Initial fluid density
PIC	Initial fluid pressure
TIC	Initial fluid temperature

VARIABLE	DESCRIPTION
HIC	Initial fluid enthalpy. However, when CHEMID refers to a ZND 1-step reaction card, this is the progressive variable (degree of combustion).

*CESE_INITIAL_CHEMISTRY_ELEMENT

Purpose: Initialize the chemistry and fluid state in every element of the list of CESE elements. This keyword is only used when chemistry is being solved with the CESE solver.

Card 1	1	2	3	4	5	6	7	8
Variable	CHEMID	COMPID						
Type	I	I						
Default	none	none						

Card 2	1	2	3	4	5	6	7	8
Variable	UIC	VIC	WIC	RHOIC	PIC	TIC	HIC	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	none	none	none	

Element List Card. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 3	1	2	3	4	5	6	7	8
Variable	ELE1	ELE2	ELE3	ELE4	ELE5	ELE6	ELE7	ELE8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
CHEMID	Identifier of chemistry control card to use (see *CHEMISTRY_CONTROL._...)
COMPID	Identifier of chemical composition to use (see *CHEMISTRY_COMPOSITION)
UIC	*X*-component of the fluid velocity

VARIABLE	DESCRIPTION
VIC	*Y*-component of the fluid velocity
WIC	*Z*-component of the fluid velocity
RHOIC	Initial fluid density
PIC	Initial fluid pressure
TIC	Initial fluid temperature
HIC	Initial fluid enthalpy. However, when CHEMID refers to a ZND 1-step reaction card (see *CHEMISTRY_CONTROL_ZND), this is the progressive variable (degree of combustion).
ELE*i*	User element numbers to initialize

*CESE_INITIAL_CHEMISTRY_PART

Purpose: Initialize the chemistry and fluid state in every element of the specified CESE part that has not already been initialized by *CESE_INITIAL_CHEMISTRY_ELEMENT or *CESE_INITIAL_CHEMISTRY_SET cards. This keyword is only used when chemistry is being solved with the CESE solver.

Card 1	1	2	3	4	5	6	7	8
Variable	PARTID	CHEMID	COMPID					
Type	I	I	I					
Default	none	none	none					

Card 2	1	2	3	4	5	6	7	8
Variable	UIC	VIC	WIC	RHOIC	PIC	TIC	HIC	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
PARTID	Identifier of the CESE part on which to initialize (see *CESE_PART)
CHEMID	Identifier of the chemistry control card to use (see *CHEMISTRY_-CONTROL._...)
COMPID	Identifier of the chemical composition to use (see *CHEMISTRY_-COMPOSITION)
UIC	*X*-component of the fluid velocity
VIC	*Y*-component of the fluid velocity
WIC	*Z*-component of the fluid velocity
RHOIC	Initial fluid density

VARIABLE	DESCRIPTION
PIC	Initial fluid pressure
TIC	Initial fluid temperature
HIC	Initial fluid enthalpy. However, when CHEMID refers to a ZND 1-step reaction card (see *CHEMISTRY_CONTROL_ZND), this is the progressive variable (degree of combustion).

*CESE_INITIAL_CHEMISTRY_SET

Purpose: Initializes the chemistry and fluid state in every element of the specified element set in the CESE mesh that has not already been initialized by *CESE_INITIAL_CHEMISTRY_ELEMENT cards. This is only used when chemistry is being solved with the CESE solver.

Card 1	1	2	3	4	5	6	7	8
Variable	SETID	CHEMID	COMPID					
Type	I	I	I					
Default	none	none	none					

Card 2	1	2	3	4	5	6	7	8
Variable	UIC	VIC	WIC	RHOIC	PIC	TIC	HIC	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
SETID	Identifier of the CESE element set to initialize
CHEMID	Identifier of chemistry control card to use (see *CHEMISTRY_CONTROL._...)
COMPID	Identifier of chemical composition to use (see *CHEMISTRY_COMPOSITION)
UIC	*X*-component of the fluid velocity
VIC	*Y*-component of the fluid velocity
WIC	*Z*-component of the fluid velocity
RHOIC	Initial fluid density

VARIABLE	DESCRIPTION
PIC	Initial fluid pressure
TIC	Initial fluid temperature
HIC	Initial fluid enthalpy. However, when CHEMID refers to a ZND 1-step reaction card (see *CHEMISTRY_CONTROL_ZND), this is the progressive variable (degree of combustion).

*CESE_MAT_000

Purpose: Define the fluid (gas) properties in a viscous flow for the CESE solver.

Material Definition Cards. Include one card for each instance of this material type. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	MU	K					
Type	I	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
MID	Material identifier
MU	Fluid dynamic viscosity. For Air at 15 °C, MU = 1.81×10^{-5} $^{kg}/_{ms}$
K	Thermal conductivity of the fluid

Remarks:

1. The viscosity is only used viscous flows, so for inviscid flows, it is not necessary to define it. The thermal conductivity is only used to calculate the heat transfer between the structure and the thermal solver when coupling is activated.

2. As with other solvers in LS-DYNA, the user is responsible for unit consistency. For example, if dimensionless variables are used, MU should be replaced by the corresponding dimensionless one.

*CESE_MAT_001(_GAS)

Purpose: Define the fluid (gas) properties in a viscous flow for the CESE solver.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	C1	C2	PRND				
Type	I	F	F	F				
Default	none	1.458E-6	110.4	0.72				

VARIABLE	DESCRIPTION
MID	Material identifier
C1, C2	Two coefficients in the Sutherland's formula for viscosity, i.e., $$\mu = \frac{C_1 T^{\frac{3}{2}}}{T + C_2}$$ where C_1 and C_2 are constants for a given gas. For example, for air at moderate temperatures, $$C_1 = 1.458 \times 10^{-6}\,\mathrm{kg/msK^{1/2}}, \quad C_2 = 110.4\,\mathrm{K}$$
PRND	The Prandtl Number (used to determine the coefficient of thermal conductivity). It is approximately constant for most gases. For air at standard conditions PRND = 0.72.

Remarks:

1. C1 and C2 are only used to calculate the viscosity in viscous flows, so for inviscid flows, this material card is not needed. The Prandtl number is used to extract the thermal conductivity, which is used when thermal coupling with the structure is activated.

2. As with other solvers in LS-DYNA, the user is responsible for unit consistency. For example, if dimensionless variables are used, C_1 and C_2 should be replaced by the corresponding dimensionless ones.

*CESE_MAT_002

Purpose: Define the fluid (gas) properties in a viscous flow for the CESE solver.

Material Definition Cards. Include one card for each instance of this material type. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	MU0	SMU	K0	SK	T0		
Type	I	F	F	F	F	F		
Default	none	1.716E-5	111.	0.0241	194.0	273.0		

VARIABLE	DESCRIPTION
MID	Material identifier
MU0 / SMU	Two coefficients appearing in the equation derived by combining Sutherland's formula with the power law for dilute gases: $$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0}\right)^{3/2} \frac{T_0 + S_\mu}{T + S_\mu} \ .$$ μ_0 is a reference value, and S_μ is an effective temperature called the Sutherland constant, which is characteristic of the gas. For air at moderate temperatures, $$\mu_0 = 1.716 \times 10^{-5}\,\mathrm{Ns/m^2}\,, \qquad S_\mu = 111\ \mathrm{K}$$
K0/SK	Two coefficients appearing in the equation derived by combining Sutherland's formula with the power law for dilute gases: $$\frac{k}{k_0} = \left(\frac{T}{T_0}\right)^{3/2} \frac{T_0 + S_k}{T + S_k} \ .$$ Here k is the thermal conductivity, k_0 is a reference value, and S_k is the Sutherland constant, which is characteristic of the gas. For air at moderate temperatures, $$k_0 = 0.0241\ \mathrm{W/m}\,, \qquad S_k = 194\ \mathrm{K}$$
T0	Reference temperature, T_0. The default value (273.0) is for air, in degrees K.

Remarks:

1. **Fields that Depend on Problem Physics.** The viscosity is only used for viscous flow. Therefore, for inviscid flows, it is not necessary to define it. The thermal conductivity is only used to calculate the heat transfer between the structure and the thermal solver when coupling is activated.

2. **Unit Consistency.** As with other solvers in LS-DYNA, the user is responsible for unit consistency. For example, if dimensionless variables are used, MU should be replaced by the corresponding dimensionless one.

*CESE_PART

Purpose: Define CESE solver parts, meaning connect CESE material and EOS information.

Part Cards. Include one card for each CESE part. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	MID	EOSID					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
PID	Part identifier (must be different from any PID on a *PART card)
MID	Material identifier defined by a *CESE_MAT_… card
EOSID	Equation of state identifier defined by a *CESE_EOS_… card

Remarks:

1. **Material for inviscid flows.** Since material coefficients are only used in viscous flows, the MID can be left blank for inviscid flows.

*CESE_SURFACE_MECHSSID_D3PLOT

Purpose: Identify the surfaces to be used in generating surface D3PLOT output for the CESE solver. These surfaces must be on the outside of volume element parts that are in contact with the CESE fluid mesh. The variables in question are part of the CESE FSI solution process or of the CESE conjugate heat transfer solver.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	SurfaceLabel						
Type	I	A						
Default	none	none						

VARIABLE	DESCRIPTION
SSID	Mechanics solver segment set ID that is in contact with the fluid CESE mesh.
SurfaceLabel	Name to use in d3plot output to identify the SSID for the LSPP user.

*CESE_SURFACE_MECHVARS_D3PLOT

Purpose: List of variables to output on the surfaces designated by the segment set IDs given in the *CESE_SURFACE_MECHSSID_D3PLOT cards. Most of the allowed variables are defined only on the fluid-structure interface, and so the segment set IDs defining a portion of the fluid-structure interface must involve only segments (element faces) that are on the outside of volume element parts that are in contact with the CESE fluid mesh.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	Output Quantity							
Type	A							
Default	none							

VARIABLE	DESCRIPTION
Output Quantity	Descriptive phrase for the mechanics surface variable to output for the LSPP user. Output will be done on all SSIDs selected by the *CESE_SURFACE_MECHSSID_D3PLOT cards in the problem. Supported variables include: FLUID FSI FORCE FLUID FSI PRESSURE INTERFACE TEMPERATURE SOLID INTERFACE HEAT FLUX FLUID INTERFACE HEAT FLUX INTERFACE HEAT FLUX RATE SOLID INTERFACE DISPLACEMENT SOLID INTERFACE VELOCITY SOLID INTERFACE ACCELERATION Force, displacement, velocity, and acceleration are output as vector quantities. The rest of the variables are scalar quantities. The fluxes are in the normal direction to the fluid/structure interface, with the heat fluxes relative to the normal pointing into the structure.

*CHEMISTRY

The keyword *CHEMISTRY is used to access chemistry databases that include Chemkin-based descriptions of a chemical model, as well as to select a method of solving the model. The keyword cards in this section are defined in alphabetical order:

*CHEMISTRY_BATTERY†

*CHEMISTRY_COMPOSITION

*CHEMISTRY_CONTROL_0D

*CHEMISTRY_CONTROL_1D†

*CHEMISTRY_CONTROL_CSP

*CHEMISTRY_CONTROL_FULL

*CHEMISTRY_CONTROL_INFLATOR†

*CHEMISTRY_CONTROL_TBX

*CHEMISTRY_CONTROL_ZND†

*CHEMISTRY_DET_INITIATION†

*CHEMISTRY_INFLATOR_PROPERTIES†

*CHEMISTRY_MODEL

*CHEMISTRY_PATH

†: Card may be used only once in a given model

An additional option "_TITLE" may be appended to all *CHEMISTRY keywords. If this option is used, then an 80 character string is read as a title from the first card of that keyword's input. At present, LS-DYNA does not make use of the title. Inclusion of titles gives greater clarity to input decks.

The *CHEMISTRY_BATTERY card is only intended to be used in battery electrochemistry models that involve *BATTERY_... cards. Note that none of the comments below apply for this particular case.

*CHEMISTRY

In order to use one of the chemistry solvers, the input must include at least one *CHEMISTRY_MODEL card. For each spatial region containing a different chemical composition, at least one *CHEMISTRY_COMPOSITION card is required.

The *CHEMISTRY_CONTROL_0D card is intended to be used in a standalone fashion to verify the validity of a given chemistry model. This model includes the total number of species and all elementary reactions with their Arrhenius rate parameters. For instance, this solver could be used to check the induction time of the model.

The *CHEMISTRY_BLAST_INITIATION, *CHEMISTRY_CONTROL_1D, *CHEMISTRY_DET_INITIATION, and *CHEMISTRY_CONTROL_ZND cards are intended to provide a one-dimensional initialization to a 2D or 3D chemically-reacting flow.

In order to perform a full, general purpose chemistry calculation in 2D or 3D, the *CHEMISTRY_CONTROL_FULL card should be used.

The *CHEMISTRY_CONTROL_CSP card is an option for reducing the number of species and reactions that are used in a general purpose chemistry calculation. Other reduction mechanisms are planned for the future.

An airbag inflator model is available with *CHEMISTRY_CONTROL_INFLATOR along with *CHEMISTRY_INFLATOR_PROPERTIES and a chemistry model that is referenced via three chemical compositions. This involves zero-dimensional modeling, with pyrotechnic inflator, and cold and hot flow hybrid inflator options.

The *CHEMISTRY_CONTROL_TBX card is intended for use only in a stochastic particle model, where the *STOCHASTIC_TBX_PARTICLES card is used.

*CHEMISTRY_BATTERY

Purpose: Identify the files that define a Chemkin chemistry model for use with the battery electrochemistry solver (see *BATTERY).

Battery Reaction and Species Input File Card (Chemkin-compatible)

Card 1	1	2	3	4	5	6	7	8
Variable	FILE1							
Type	A							

Thermodynamics Database File Card

Card 2	1	2	3	4	5	6	7	8
Variable	FILE2							
Type	A							

Transport Properties Database File Card

Card 3	1	2	3	4	5	6	7	8
Variable	FILE3							
Type	A							

VARIABLE	DESCRIPTION
FILE1	Name of the file containing the Chemkin-compatible input
FILE2	Name of the file containing the chemistry thermodynamics database
FILE3	Name of the file containing the chemistry transport properties database

*CHEMISTRY_COMPOSITION

Purpose: Provide a general way to specify a chemical composition via a list of species mole numbers in the context of a Chemkin database model.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	MODELID						
Type	I	I						
Default	none	none						

Species List Card. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	MOLFR	SPECIES						
Type	F	A						
Default	none	none						

VARIABLE	DESCRIPTION
ID	A unique identifier among all chemistry compositions
MODELID	Identifier of a Chemkin-compatible chemistry model (see *CHEMISTRY_MODEL)
MOLFR	The number of moles corresponding to the species named in the SPECIES field. But, if used with a *STOCHASTIC_TBX_PARTICLES card, it is the molar concentration of the species (in units of moles/[length]^3, where [length] is the user's length unit).
SPECIES	The Chemkin-compatible name of a chemical species that is defined in the chemistry model identified by MODELID (see *CHEMISTRY_MODEL)

*CHEMISTRY_CONTROL_0D

Purpose: Performs a zero-dimensional isotropic chemistry calculation that operates standalone (does not call the CESE solver). This is for ISOBARIC or ISOCHORIC cases.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	COMPID	SOLTYP	PLOTDT	CSP_SEL			
Type	I	I	I	F	I			
Default	none	none	none	1.0e-6	0			
Remarks					1			

Card 2	1	2	3	4	5	6	7	8
Variable	DT	TLIMIT	TIC	PIC	RIC	EIC		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

CSP Parameters Card. Include cards for each chemical species in the following format when CSP_SEL.GT.0. This input ends at the next keyword ("*") card.

Card 3	1	2	3	4	5	6	7	8
Variable	AMPL	YCUT						
Type	F	F						
Default	none	none						

VARIABLE	DESCRIPTION
ID	Identifier for this 0D computation.

VARIABLE	DESCRIPTION
COMPID	Chemical composition identifier of composition to use.
SOLTYP	Type of 0D calculation: EQ.1: Isochoric EQ.2: Isobaric
PLOTDT	Simulation time interval for output both to the screen and to the isocom.csv file. This file can be loaded into LS-PREPOST for curve plotting using the x-y plot facility.
CSP_SEL	CSP solver option: EQ.0: Do not use the CSP solver, and ignore the AMPL and YCUT parameters (default). GT.0: Use the CSP solver, with the AMPL and YCUT parameters.
DT	Initial time step
TLIMIT	Time limit for the simulation
TIC	Initial temperature
PIC	Initial pressure
RIC	Initial density
EIC	Initial internal energy
AMPL	Relative accuracy for the mass fraction of a chemical species in the Chemkin input file.
YCUT	Absolute accuracy for the mass fraction of a chemical species in the Chemkin input file.

Remarks:

1. If CSP_SEL.GT.0, then instead of using the full chemistry solver, the computational singular perturbation (CSP) method solver is used.

*CHEMISTRY_CONTROL_1D

Purpose: Loads a previously-computed one-dimensional detonation. It is then available for use in the CESE solver for initializing a computation. In the product regions, this card overrides the initialization of the *CESE_INITIAL_CHEMISTRY_… cards.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	XYZD	DETDIR	CSP_SEL				
Type	I	F	I	I				
Default	none	none	none	0				
Remarks				1				

One-Dimensional Solution LSDA Input File Card.

Card 2	1	2	3	4	5	6	7	8
Variable	FILE							
Type	A							

CSP Parameters Card Include cards for each chemical species in the following format when CSP_SEL > 0. This input ends at the next keyword ("*") card.

Card 3	1	2	3	4	5	6	7	8
Variable	AMPL	YCUT						
Type	F	F						
Default	none	none						

VARIABLE	DESCRIPTION
ID	Identifier for this one-dimensional detonation solution.
XYZD	Position of the detonation front in the DETDIR direction.

VARIABLE	DESCRIPTION
DETDIR	Detonation propagation direction EQ.1: x EQ.2: y EQ.3: z
CSP_SEL	CSP solver option: EQ.0: Do not use the CSP solver, and ignore the AMPL and YCUT parameters (default). GT.0: Use the CSP solver, with the AMPL and YCUT parameters.
FILE	Name of the LSDA file containing the one-dimensional solution.
AMPL	Relative accuracy for the mass fraction of a chemical species in the chemkin input file.
YCUT	Absolute accuracy for the mass fraction of a chemical species in the chemkin input file.

Remarks:

1. If CSP_SEL > 0, then instead of using the full chemistry solver, the computational singular perturbation (CSP) method solver is used.

*CHEMISTRY_CONTROL_CSP

Purpose: Computes reduced chemistry for a specified Chemkin chemistry model using the Computational Singular Perturbation (CSP) method. This card can be used for general-purpose chemical reaction calculations.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	IERROPT						
Type	I	I						
Default	none	none						

CSP Parameters Card. Include cards for each chemical species in the following format as indicated by the value of IERROPT. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	AMPL	YCUT						
Type	F	F						
Default	none	none						

VARIABLE	DESCRIPTION
ID	Identifier for this computational singular perturbation solver.
IERROPT	Selector: EQ.0: AMPL and YCUT values for all chemical species are required. EQ.1: One CSP Parameter Card should be provided, and it will be used for all species.
AMPL	Relative accuracy for the mass fraction of a chemical species in the Chemkin input file.
YCUT	Absolute accuracy for the mass fraction of a chemical species in the Chemkin input file.

*CHEMISTRY_CONTROL_FULL

Purpose: Computes the full chemistry specified by a Chemkin chemistry model. This card can be used for general-purpose chemical reaction calculations.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	ERRLIM	RHOMIN	TMIN				
Type	I	F	F	F				
Default	none	none	0.0	0.0				

VARIABLE	DESCRIPTION
ID	Identifier for this full chemistry calculation
ERRLIM	Error tolerance for the full chemistry calculation
RHOMIN	Minimum fluid density above which chemical reactions are computed
TMIN	Minimum temperature above which chemical reactions are computed

*CHEMISTRY_CONTROL_INFLATOR

Purpose: Provide the required properties of an inflator model for airbag inflation.

Card 1	1	2	3	4	5	6	7	8
Variable	MODEL	OUT_TYPE	TRUNTIM	DELT	PTIME			
Type	I	I	F	F	F			
Remarks	1	2,4						

Inflator Output Database File (an ASCII file) Card.

Card 2	1	2	3	4	5	6	7	8
Variable	FILE							
Type	A							

Densities for Condensed Species. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 3	1	2	3	4	5	6	7	8
Variable	DENSITY	Species Name						
Type	F	A						
Default	none	none						
Remark		3						

VARIABLE	DESCRIPTION
MODEL	Type of inflator model to compute. EQ.1: Pyrotechnic model EQ.2: Hybrid model with cold flow option in the gas chamber EQ.3: Hybrid model with heat flow in the gas chamber EQ.4: Hybrid model with heat flow in one additional gas chamber EQ.5: Hybrid model with heat flow in two additional gas chambers
OUT_TYPE	Selects the output file format that will be used in an airbag simulation. EQ.0: Screen output calibration output (see Remark 4) EQ.1: CESE compressible flow solver (default) EQ.2: ALE solver EQ.3: CPM solver (with 2nd-order expansion of C_p) EQ.4: CPM solver (with 4th-order expansion of C_p)
TRUNTIM	Total run time.
DELT	Delta(t) to use in the model calculation.
PTIME	Time interval for output of time history data to FILE.
FILE	Name of the ASCII file in which to write the time history data and other data output by the inflator simulation.
DENSITY	Density of a condensed-phase species present in the inflator.
Species Name	Chemkin-compatible name of a condensed-phase species.

Remarks:

1. If MODEL = 3, the solution of an elementary reaction system is required for the finite-rate chemistry in the gas chamber. For MODEL = 4 and 5, the condensed phase is computed only in the combustion chamber.

2. Output file includes all of the necessary thermodynamics variables and load curves for the species mass flow rate, temperature, and density curve. This will make it possible to generate the velocity curve which is required by each solver that carries out an airbag simulation.

3. At least one of these cards will be input if condensed-phase species are present during the propellant combustion. In this case, the user must specify each condensed-phase density. This density is then used to compute the volume fractions in both the combustion and gas chamber, where the energy equations are needed.

4. If OUT_TYPE = 0, the propellant information will be displayed on the screen, including total mass, remaining mass percentage, and mass burning rate versus time, and the calibration data will be saved in the output file, including the time versus pressure, temperature, total mass flow rate, and individual species mass fractions for all chambers. With this option, the user can quickly see the effect of changing the parameters on the first three *CHEMISTRY_INFLATOR_PROPERTIES cards.

*CHEMISTRY_CONTROL_TBX

Purpose: Specify a chemistry solver for use in conjunction with stochastic TBX particles. This is intended only for modeling the second phase of an explosion where the explosive has embedded metal (aluminum) particles that are too large to have burned in the first phase of the explosion.

This chemistry card points to a *CHEMISTRY_MODEL card (via IDCHEM) with its associated *CHEMISTRY_COMPOSITION cards to set up the initial conditions. That is, it establishes the spatial distribution of the species in the model.

It is assumed that there is no chemical reaction rate information in the chemistry model files. This is done since a special chemical reaction mechanism is implemented for TBX modeling. If particles other than solid aluminum particles are embedded in the explosive, then another burn model has to be implemented.

Surface Part Card. Card 1 format used when the PART keyword option is active**.**

Card 1	1	2	3	4	5	6	7	8
Variable	IDCHEM	USEPAR						
Type	I	I						
Default	none	1						

VARIABLE	DESCRIPTION
IDCHEM	Identifier for this chemistry solver.
USEPAR	Coupling flag indicating if a *STOCHASTIC_TBX_PARTICLES card is provided for this model: EQ.1: uses a *STOCHASTIC_TBX_PARTICLES card (default). EQ.0: does not use such a card.

*CHEMISTRY_CONTROL_ZND

Purpose: Compute the one-dimensional reduced chemistry of a ZND model. It is then used in the initialization of the chemistry part of the CESE solver. When this card is used, the *CESE_INITIAL_CHEMISTRY… cards must specify the progressive variable (degree of combustion) in the HIC field.

Card 1	1	2	3	4	5	6	7	8
Variable	ID							
Type	I							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	F	EPLUS	Q0	GAM	XYZD	DETDIR		
Type	F	F	F	F	F	I		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
ID	Identifier for this full chemistry calculation
F	Overdriven factor
EPLUS	EPLUS parameter of the ZND model
Q0	Q0 parameter of the ZND model
GAM	GAM parameter of the ZND model
XYZD	Position of the detonation front in the DETDIR direction
DETDIR	Detonation propagation direction: EQ.1: X-direction EQ.2: Y-direction

VARIABLE	DESCRIPTION
	EQ.3: Z-direction

*CHEMISTRY_DET_INITIATION

Purpose: Performs a one-dimensional detonation calculation based upon a chemical composition and initial conditions. It is then available for use immediately in the CESE solver for initializing a computation, or it can be subsequently used by the *CHEMISTRY_CONTROL_1D card in a later run. In the product regions, this card overrides the initialization of the *CESE_INITIAL_CHEMISTRY… cards.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	COMPID	NMESH	DLEN	CFL	TLIMIT	XYZD	DETDIR
Type	I	I	I	F	F	F	F	I
Default	none	none	none	none	none	none	none	none

LSDA Output File Card.

Card 2	1	2	3	4	5	6	7	8
Variable	FILE							
Type	A							

VARIABLE	DESCRIPTION
ID	Identifier for this one-dimensional detonation computation.
COMPID	Chemical composition identifier of composition to use.
NMESH	Number of equal-width elements in the one-dimensional domain.
DLEN	Length of the one-dimensional domain.
CFL	Time-step limiting factor.
TLIMIT	Time limit for the simulation
XYZD	Position of the detonation front in the DETDIR direction.
DETDIR	Detonation propagation direction (1 => X; 2 => Y; 3 => Z)

VARIABLE	DESCRIPTION
FILE	Name of the LSDA file in which to write the one-dimensional solution.

*CHEMISTRY_INFLATOR_PROPERTIES

Purpose: Provide the required properties of an inflator model.

Card 1	1	2	3	4	5	6	7	8
Variable	COMP_ID	PDIA	PHEIGHT	PMASS	TOTMASS			
Type	I	F	F	F	F			
Remarks	1	2	2					

Card 2	1	2	3	4	5	6	7	8
Variable	TFLAME	PINDEX	A0	TDELAY	RISETIME			
Type	F	F	F	F	F			
Default	none	none	none	none	None			

Combustion Chamber Parameter Card.

Card 3	1	2	3	4	5	6	7	8
Variable	COMP1ID	VOL1	AREA1	CD1	P1	T1	DELP1	DELTI
Type	I	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Gas Plenum Parameter Card.

Card 4	1	2	3	4	5	6	7	8
Variable	COMP2ID	VOL2	AREA2	CD2	P2	T2	DELP2	DELT2
Type	I	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Tank Parameter Card.

Card 5	1	2	3	4	5	6	7	8
Variable	COMP3ID	VOL3	P3	T3				
Type	I	F	F	F				
Default	none	none	none	none				

Gas Chamber 1 (Optional, see Remark 3) Card.

Card 6	1	2	3	4	5	6	7	8
Variable	COMP4ID	VOL4	AREA4	CD4	P4	T4	DELP4	DELT4
Type	I	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Gas Chamber 2 (Optional, see Remark 3) Card.

Card 7	1	2	3	4	5	6	7	8
Variable	COMP5ID	VOL5	AREA5	CD5	P5	T5	DELP5	DELT5
Type	I	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
COMP_ID	Chemical composition identifier of the composition for the steady-state propellant combustion (see Remark 1).
PDIA	Propellant diameter (see Remark 2).
PHEIGHT	Propellant height (see Remark 2).
PMASS	Individual cylinder (or sphere) propellant mass.
TOTMASS	Total propellant mass.
TFLAME	Adiabatic flame (combustion) temperature.
PINDEX	Power of the pressure in rate of burn model.
A0	Steady-state constant.
TDELAY	Ignition time delay.
RISETIME	Rise time.
COMP1ID	Chemical composition identifier of composition to use in the combustion chamber.
VOL1	Volume of the combustion chamber.
AREA1	Area of the combustion chamber.
CD1	Discharge coefficient of the combustion chamber.
P1	Pressure in the combustion chamber.

VARIABLE	DESCRIPTION
T1	Temperature in the combustion chamber.
DELP1	Rupture pressure in the combustion chamber.
DELT1	Elapsed time for breaking the burst disk between the chambers
COMP2ID	Chemical composition identifier of composition to use in the gas plenum.
VOL2	Volume of the gas plenum.
AREA2	Area of the gas plenum.
CD2	Discharge coefficient of the gas plenum.
P2	Pressure in the gas plenum.
T2	Temperature in the gas plenum.
DELP2	Rupture pressure in the gas plenum.
DELT2	Elapsed time for breaking the burst disk between the chambers
COMP3ID	Chemical composition identifier of composition to use in the tank.
VOL3	Volume of the tank.
P3	Pressure in the tank.
T3	Temperature in the tank.
COMP4ID	Chemical composition identifier of composition to use in the additional (second) gas chamber.
VOL4	Volume of the second gas chamber.
P4	Pressure in the second gas chamber.
T4	Temperature in the second gas chamber.
DELP4	Rupture pressure in the second gas chamber.
DELT4	Elapsed time for breaking the burst disk between the first and second gas chambers
COMP5ID	Chemical composition identifier of composition to use in the additional (third) gas chamber.

VOL5	Volume of the third gas chamber.
P5	Pressure in the third gas chamber.
T5	Temperature in the third gas chamber.
DELP5	Rupture pressure in the third gas chamber.
DELT5	Elapsed time for breaking the burst disk between the second and third gas chambers

Remarks:

1. The propellant composition can be obtained by running a chemical equilibrium program such as NASA CEA, the CHEETAH code, or the PEP code. LSTC provides a modified version of the PEP code along with documentation for users; it is available upon request.

2. A spherical shape for the propellant particles can be chosen if an identical value for the diameter and height is given.

3. To simulate a 4 or 5 chamber inflator, an additional chamber card can be used. In these cases of the inflator models, the condensed phase species are limited to the combustion chamber only if involved in the propellant combustion.

*CHEMISTRY_MODEL

Purpose: Identifies the files that define a Chemkin chemistry model.

Card 1	1	2	3	4	5	6	7	8
Variable	MODELID	JACSEL	ERRLIM					
Type	I	I	F					
Default	none	1	1.0e-3					

Chemkin Input File Card.

Card 2	1	2	3	4	5	6	7	8
Variable	FILE1							
Type	A							

Thermodynamics Database File Card.

Card 3	1	2	3	4	5	6	7	8
Variable	FILE2							
Type	A							

Transport Properties Database File Card.

Card 4	1	2	3	4	5	6	7	8
Variable	FILE3							
Type	A							

VARIABLE	DESCRIPTION
MODELID	Identifier for this Chemkin-based chemistry model

VARIABLE	DESCRIPTION
JACSEL	Selects the form of the Jacobian matrix for use in the source term: EQ.1: Fully implicit (default) EQ.2: Simplified implicit
ERRLIM	Allowed error in element balance in a chemical reaction
FILE1	Name of the file containing the Chemkin-compatible input
FILE2	Name of the file containing the chemistry thermodynamics database
FILE3	Name of the file containing the chemistry transport properties database

*CHEMISTRY_PATH

Purpose: To specify one or more search paths to look for chemistry database files.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	DIR							
Type	A							

VARIABLE	DESCRIPTION
DIR	Directory path to add to the search set.

*DUALCESE

The keyword *DUALCESE provides input data for the dual Conservation Element/Solution Element (dual CESE) compressible fluid solver:

*DUALCESE_BLOCKMESH

DUALCESE_BOUNDARY_AXISYMMETRIC_*{OPTION}

DUALCESE_BOUNDARY_CYCLIC_*{OPTION}

DUALCESE_BOUNDARY_FSI_*{OPTION}

DUALCESE_BOUNDARY_NON_REFLECTIVE_*{OPTION}

DUALCESE_BOUNDARY_PRESCRIBED_*{OPTION}

DUALCESE_BOUNDARY_PRESCRIBED_HYBRID_*{OPTION}

DUALCESE_BOUNDARY_PRESCRIBED_INLET_MACHNUM_*{OPTION}

DUALCESE_BOUNDARY_PRESCRIBED_INLET_MFRATE_*{OPTION}

DUALCESE_BOUNDARY_PRESCRIBED_INLET_PRESSURE_*{OPTION}

DUALCESE_BOUNDARY_PRESCRIBED_OUTLET_EXMFRATE_*{OPTION}

DUALCESE_BOUNDARY_PRESCRIBED_OUTLET_FARFIELD_*{OPTION}

DUALCESE_BOUNDARY_PRESCRIBED_OUTLET_PRESSURE_*{OPTION}

DUALCESE_BOUNDARY_PRESCRIBED_PHASE_CHANGE_*{OPTION}

DUALCESE_BOUNDARY_PRESCRIBED_TWO-PHASE_*{OPTION}

DUALCESE_BOUNDARY_REFLECTIVE_*{OPTION}

DUALCESE_BOUNDARY_SLIDING_*{OPTION}

DUALCESE_BOUNDARY_SOLID_WALL_*{OPTION1}_{OPTION2}

*DUALCESE_CHEMISTRY_D3PLOT

*DUALCESE_CONTROL_LIMITER

*DUALCESE_CONTROL_MESH_MOV

*DUALCESE_CONTROL_SOLVER (deprecated as of R14)

*DUALCESE

*DUALCESE_CONTROL_TIMESTEP

*DUALCESE_DARCY-FORCHHEIMER_EQ

*DUALCESE_DATABASE_HISTORY_DRAG_LIFT

*DUALCESE_DATABASE_HISTORY_ELEMENT_SET

*DUALCESE_DATABASE_HISTORY_GLOBALS

*DUALCESE_DATABASE_HISTORY_NODE_SET

*DUALCESE_DATABASE_HISTORY_POINT_SET

*DUALCESE_DATABASE_HISTORY_SEGMENT_SET

*DUALCESE_DEM_COUPLING

*DUALCESE_D3PLOT

*DUALCESE_D3PLOT_FLUID_SSID

*DUALCESE_ELE2D

*DUALCESE_ELE3D

*DUALCESE_ELEMENTSET

*DUALCESE_ENSIGHT_DOMAIN

*DUALCESE_ENSIGHT_REGION

*DUALCESE_ENSIGHT_TIMESEQ

*DUALCESE_EOS_CAV_HOMOG_EQUILIB

*DUALCESE_EOS_COCHRAN_CHAN

*DUALCESE_EOS_COOLPROP (currently not working)

*DUALCESE_EOS_IDEAL_GAS

*DUALCESE_EOS_INFLATOR1

*DUALCESE_EOS_INFLATOR2

*DUALCESE_EOS_JWL

*DUALCESE_EOS_NASG

*DUALCESE_EOS_REFPROP (available on request)

*DUALCESE_EOS_REFPROP_PATH

*DUALCESE_EOS_SET (deprecated as of R14)

*DUALCESE_EOS_STIFFENED_GAS

*DUALCESE_EOS_VAN_DER_WAALS_GENERALIZED

*DUALCESE_FSI_EXCLUDE

*DUALCESE_FSI_EXCLUDE_PARTSET

*DUALCESE_GRAVITY

*DUALCESE_INCLUDE_MODEL

*DUALCESE_INITIAL

DUALCESE_INITIAL_*{OPTION}

*DUALCESE_INITIAL_REGION

*DUALCESE_INITIAL_CHEMISTRY

*DUALCESE_INITIAL_CHEMISTRY_REGION

*DUALCESE_INITIAL_CHEMISTRY_SET

*DUALCESE_INITIAL_HYBRID

*DUALCESE_INITIAL_HYBRID_SET

*DUALCESE_INITIAL_PHASE_CHANGE

*DUALCESE_INITIAL_PHASE_CHANGE_SET

*DUALCESE_INITIAL_TWO-PHASE

*DUALCESE_INITIAL_TWO-PHASE_SET

*DUALCESE_MAT_GAS

*DUALCESE_MAT_GAS_0

*DUALCESE_MAT_GAS_2

*DUALCESE_MAT_LIQUID

*DUALCESE_MESH_GEOMETRY

*DUALCESE_MESH_PART

*DUALCESE_MODEL

*DUALCESE_NODE2D

*DUALCESE_NODE3D

*DUALCESE_NODESET

*DUALCESE_PART (deprecated as of R14)

*DUALCESE_PART_MULTIPHASE (deprecated as of R14)

*DUALCESE_POINT_SOURCE

*DUALCESE_POINT_SOURCE_STRUCTNODE

*DUALCESE_POINT_SOURCE_TWO-PHASE

*DUALCESE_POINTSET

*DUALCESE_POROUS_MODEL

*DUALCESE_POROUS_SPECIFY_JUMP

*DUALCESE_REACTION_RATE_IG

*DUALCESE_REACTION_RATE_IG_REDUCED

*DUALCESE_REACTION_RATE_P_DEPEND

*DUALCESE_SEGMENTSET

*DUALCESE_SOLVER_SELECTION

*DUALCESE_SOLVER_CAV_EQNS

*DUALCESE_SOLVER_CHEMICALLY_REACTING

*DUALCESE_SOLVER_EULER_EQNS

*DUALCESE_SOLVER_HYBRID_MULTIPHASE

*DUALCESE_SOLVER_NAVIER_STOKES

*DUALCESE_SOLVER_PHASE_CHANGE

*DUALCESE_SOLVER_TWO-PHASE_MULTIPHASE

An additional keyword option TITLE may be appended to the *DUALCESE keywords. If this option is used, then an addition line is read for the DUALCESE card in 80a format

which can be used to describe that particular DUALCESE card. At present, the title serves no purpose other than to perhaps lend clarity to input decks.

Dual CESE Keyword Deck Structure:

The structure of the keyword setup for the *DUALCESE solvers is different from the way most keyword input is handled in the LS-DYNA input phase. This stems from the fact that there can be several *DUALCESE models in the same problem. In order to deal with this, each such model is restricted to be specified with one file hierarchy that starts with the keyword file designated with the *DUALCESE_MODEL card.

That keyword file can include any number of other keyword files with the *DUALCESE_-INCLUDE_MODEL card, and each of those files can in turn include other keyword files, again with the *DUALCESE_INCLUDE_MODEL card. Standard *INCLUDE cards are not allowed. In fact, in each file in the file hierarchy of a *DUALCESE_MODEL card, only *DUALCESE cards may be used. The only exception to this is when the fluid mesh is defined using *MESH cards; a fatal error will be encountered when using other non-*DUALCESE keywords. Any required non-*DUALCESE keyword cards should be defined in some other place in the keyword input (outside the scope of the *DUALCESE_MODEL keyword card).

The mesh for each dual CESE model must be defined within the keyword input file hierarchy for that model. If a mesh created with *MESH cards is used, the *MESH cards for that mesh must all be defined within the scope of the *DUALCESE_MODEL card.

With R14, we introduce an updated, preferred way of specifying the dual CESE problem. The required or recommended keywords to use are:

*DUALCESE_MESH_GEOMETRY

*DUALCESE_MESH_PART

*DUALCESE_SOLVER_SELECTION

along with one of the following specific solver cards:

*DUALCESE_SOLVER_CAV_EQNS

*DUALCESE_SOLVER_CHEMICALLY_REACTING (added in R17)

*DUALCESE_SOLVER_EULER_EQNS

*DUALCESE_SOLVER_HYBRID_MULTIPHASE

*DUALCESE_SOLVER_NAVIER_STOKES

*DUALCESE_SOLVER_PHASE_CHANGE (added in R15)

*DUALCESE_SOLVER_TWO-PHASE_MULTIPHASE

The following keywords are deprecated as of R14:

*DUALCESE_CONTROL_SOLVER

*DUALCESE_EOS_SET

*DUALCESE_PART

*DUALCESE_PART_MULTIPHASE

We also added time history cards with R14 that cause certain output to the binout file:

*DUALCESE_DATABASE_HISTORY_ELEMENT_SET

*DUALCESE_DATABASE_HISTORY_GLOBALS

*DUALCESE_DATABASE_HISTORY_NODE_SET

*DUALCESE_DATABASE_HISTORY_POINT_SET

*DUALCESE_DATABASE_HISTORY_SEGMENT_SET

Another time history-type card is new with R15; it causes drag and lift-related quantities to be output to the binout file from the immersed boundary method (IBM) FSI-type dual CESE solvers.

*DUALCESE_DATABASE_HISTORY_DRAG_LIFT

We support two external equation-of-state (EOS) libraries, REFPROP and COOLPROP, for dual CESE. Including these libraries can be complex, so we will clarify how to do so here. Each of them is accessed through a shared library that has to be loaded at runtime with *MODULE_LOAD, like in the following:

```
*MODULE_LOAD
UserA                  DUALCESE REFPROP
<path to the installed REFPROP shared library>
```

Note that this *MODULE_LOAD card must not be given inside a keyword file in the file hierarchy of a *DUALCESE_MODEL card. As discussed above, this is the case for all non-*DUALCESE keyword cards. Note also that since *MODULE is not available in the Windows version of LS-DYNA, this capability cannot be used in that version.

REFPROP and COOLPROP Libraries:

The REFPROP v10.0 version library and its directory of data sets are provided by ANSYS. Please contact ACE support staff for the dual CESE solver to obtain these files.

At this time, COOLPROP is not functioning correctly and should not be used.

Multiphase Capabilities:

The multiphase capabilities added for R13 with modifications in R14 all have FSI capabilities, including with structural element failure. This includes the 'hybrid' multiphase, 'two-phase' multiphase, and the cavitation solvers. New with R15 is a 'phase-change' multiphase solver. These keywords are involved with these capabilities:

*DUALCESE_BOUNDARY_PRESCRIBED_HYBRID

*DUALCESE_BOUNDARY_PRESCRIBED_PHASE_CHANGE

*DUALCESE_BOUNDARY_PRESCRIBED_TWO-PHASE

*DUALCESE_EOS_CAV_HOMOG_EQUILIB

*DUALCESE_EOS_COCHRAN_CHAN

*DUALCESE_EOS_JWL

*DUALCESE_EOS_NASG

*DUALCESE_EOS_SET (deprecated as of R14; use one of the *DUALCESE_-SOLVER_... cards instead)

*DUALCESE_EOS_STIFFENED_GAS

*DUALCESE_EOS_VAN_DER_WAALS_GENERALIZED

*DUALCESE_INITIAL_HYBRID

*DUALCESE_INITIAL_HYBRID_SET

*DUALCESE_INITIAL_PHASE_CHANGE

*DUALCESE_INITIAL_PHASE_CHANGE_SET

*DUALCESE_INITIAL_TWO-PHASE

*DUALCESE_INITIAL_TWO-PHASE_SET

*DUALCESE_PART_MULTIPHASE (deprecated as of R14; use *DUALCESE_-MESH_PART instead)

*DUALCESE_REACTION_RATE_IG

*DUALCESE_REACTION_RATE_IG_REDUCED

*DUALCESE_REACTION_RATE_P_DEPEND

*DUALCESE_SOLVER_HYBRID_MULTIPHASE (added in R14)

*DUALCESE_SOLVER_PHASE_CHANGE (added in R15)

*DUALCESE_SOLVER_TWO-PHASE_MULTIPHASE (added in R14)

Chemically Reacting Flow Solver:

With R17, the chemically reacting flow solver from the CESE solver has been brought into the dual CESE solver framework. Because of the superior numerical accuracy of the dual CESE solver, the chemically reacting flow solver's accuracy is improved. The solver options ported from the CESE version include 3D, 2D, and 2D axisymmetric versions, with and without the immersed-boundary method (IBM) FSI capability. At this time, the only chemistry solver that is available with this dual CESE chemically reacting flow solver is that specified by *CHEMISTRY_CONTROL_FULL.

Additional capabilities added with the dual CESE version include:

1. Coupling of dual CESE point sources
2. Coupling with dual CESE porous media
3. binout output of species mass fractions
4. Initialization based upon *DEFINE_REGION
5. Restarts
6. Output of species mass fractions to the output file in the Ansys Ensight™ Gold format (see the section on the Ansys EnSight results file below)

Flow Inlet and Exit Boundary Conditions:

The following prescribed inlet and prescribed outlet boundary condition cards are new with R14:

DUALCESE_BOUNDARY_PRESCRIBED_INLET_MACHNUM_*{OPTION}

DUALCESE_BOUNDARY_PRESCRIBED_INLET_MFRATE_*{OPTION}

DUALCESE_BOUNDARY_PRESCRIBED_INLET_PRESSURE_*{OPTION}

DUALCESE_BOUNDARY_PRESCRIBED_OUTLET_EXMFRATE_*{OPTION}

DUALCESE_BOUNDARY_PRESCRIBED_OUTLET_FARFIELD_*{OPTION}

DUALCESE_BOUNDARY_PRESCRIBED_OUTLET_PRESSURE_*{OPTION}

Output Mechanism for the Ansys EnSight Post-Processor:

Output of most dual CESE results to files in the Ansys Ensight Gold format is available with R17. When the *DUALCESE_ENSIGHT_... cards are used, output to d3plot of the dual CESE results is disabled.

This output mechanism gives more flexibility in optimizing output for post-processing. In addition, it provides support for dual CESE meshes containing pyramid meshes and those generated by a future solver based upon adaptive meshing.

This new capability is an alpha-level feature for which additional features are in development.

The following *DUALCESE_ENSIGHT_... cards are available in R17:

*DUALCESE_ENSIGHT_TIMESEQ

*DUALCESE_ENSIGHT_DOMAIN

*DUALCESE_ENSIGHT_REGION

As already noted, when *DUALCESE_ENSIGHT_... cards occur in an input deck, the d3plot output from the dual CESE solvers is disabled. Output to d3plot from all other LS-DYNA solvers remains active. Because the Ansys EnSight post-processor permits multiple datasets to be loaded in a given session from one or more simulations, including d3plot as well as native Ansys EnSight data sets, it is possible to perform FSI problem visualization in Ansys EnSight using dual CESE data in the Gold format and structural solver data in the d3plot file format as usual. The Ansys EnSight post-processor loads each data set in succession. For instance, the structural d3plot data could be loaded first, followed by the dual CESE Gold format files.

Notably, output to the Gold format files of dual CESE element-centered data is an option, as well as the standard output at the mesh nodes. Also, a wide range of time history outputs is available. For instance, values on a node or element set can be output, meaning output for specified variables at each node in the node set or at each element in an element set. Each of these is output as a separate time history. Also, Average, minimum, maximum, or sum over a node or element set, giving a single value at each given time, can be output.

Comparison to CESE:

The capabilities implemented in the dual CESE solvers are most of what is available in the *CESE solvers. The coupling with the *STOCHASTIC_PARTICLE solver is missing, along with some conjugate heat transfer capabilities. In the dual CESE solver, conjugate heat transfer is only available for the immersed boundary method FSI single-fluid solvers in versions R17 and later.

Some capabilities are only in the newer dual CESE solvers, in particular, the multiphase solvers, the many added EOSes, the binout-based time history capabilities, and the point-source-type fluid injection methods implemented in the single-phase CFD solvers and the two-phase multiphase solvers. Also, drag, lift, and related variables are only computed for the dual CESE IBM FSI solvers (starting with R15).

Restarts:

Dump/restart capabilities became available for all dual CESE solvers with the R17 release.

*DUALCESE_BLOCKMESH

Purpose: Create a single block rectangular mesh for use with the dual CESE solver. When this approach to creating a mesh is in use, the usual cards that define a dual CESE mesh and its boundary for boundary conditions should not be included in the input. In other words, when *DUALCESE_BLOCKMESH is used, any of the following cards will cause an error to be reported: *DUALCESE_NODE2D, *DUALCESE_NODE3D, *DUALCESE_ELE2D, *DUALCESE_ELE3D, and *DUALCESE_SEGMENTSET.

Card Summary:

Card 1. This card is required.

PID							

Card 2. Include a set of Cards 2 and 2.1 to specify the one-dimensional mesh one each axis. More than one instantiation of Card 2.1 can be in a given set. Include as many sets as needed. Note that only one set is allowed per direction. See Remark 3.

AXTYPE	SEGID_L	SEGID_H	COORD_L	SCALE			

Card 2.1. Include as many cards as required of this card in a set with Card 2 to define one or more contiguous one-dimensional meshes to cover each axis. See Remark 3.

NELE	ALEN	RATIO	CRATIO	WID_A	WID_B		

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
PID	Dual CESE part ID for all elements in the block mesh. See *DUALCESE_MESH_PART.

Define mesh parameters and boundary IDs of a mesh axis. Include as many cards as needed to define the one-dimensional mesh on each axis.

Card 2	1	2	3	4	5	6	7	8
Variable	AXTYPE	SEGID_L	SEGID_H	COORD_L	SCALE			
Type	A	F	F	F	F			
Default	none	1	2	0.0	1.0			
Remarks	1							

VARIABLE	DESCRIPTION
AXTYPE	Axis to define, meaning one of 'X_AXIS', 'Y_AXIS', or 'Z_AXIS'. One of each is required for a 3D mesh, and the 'X_AXIS' and 'Y_AXIS' are required for a 2D mesh.
SEGID_L	ID to use for the boundary segment set to be created at the low-index boundary
SEGID_H	ID to use for the boundary segment set to be created at the high-index boundary
COORD_L	Minimum value of the coordinate on the AXTYPE axis
SCALE	Factor to use to scale all positions on the AXTYPE axis

Parameters for creating a one-dimensional mesh block. Include as many cards as needed to define an overall one-dimensional mesh. These cards create a sequence of one-dimensional meshes that when glued together will create the net one-dimensional mesh for the active AXTYPE. The algorithm that is implemented for this takes any three of the six parameters to define the given mesh block; at least three parameters must be specified. But there are several sets of three parameters that will not suffice to create a mesh. These illegal combinations are noted in Remark 2.

Card 2.1	1	2	3	4	5	6	7	8
Variable	NELE	ALEN	RATIO	CRATIO	WID_A	WID_B		
Type	I	I	F	F	F	F		
Default	none	none	none	none	none	none		
Remarks	2	2	2	2	2	2		

VARIABLE	DESCRIPTION
NELE	Number of elements to put into the current mesh block
ALEN	Length of the given mesh block
RATIO	Total growth ratio, meaning the ratio by which the length of the last element in the block changes from the first element
CRATIO	Element-to-element growth ratio
WID_A	Width of the initial element in the block
WID_B	Width of the last element in the block

Remarks:

1. **AXTYPE uniqueness.** Only one instance of each AXTYPE name may appear in defining the block mesh.

2. **Invalid combinations of sets of three parameters to create a 1D-mesh.** The following are invalid combinations of parameters for Card 2.1:

   a) ALEN, RATIO, WID_B

   b) RATIO, WID_A, WID_B

c) NELE, RATIO, CRATIO

3. **Example input.** The following is sample input works for this keyword:

```
*DUALCESE_BLOCKMESH
$      PID
      1000
$ First set of Cards 2 and 2.1
$ Card 2 for the X-axis
$   AXTYPE   SEGID_L   SEGID_H   COORD_L     SCALE
    X_AXIS      2000      2001      20.0       1.0
$ First Card 2.1 for the X-axis set
$     NELE      ALEN     RATIO    CRATIO     WID_A     WID_B
        20      10.0       1.5
$ Second Card 2.1 for the X-axis set
$     NELE      ALEN     RATIO    CRATIO     WID_A     WID_B
                10.0       0.9                0.5
$ Third Card 2.1 for the X-axis set
$     NELE      ALEN     RATIO    CRATIO     WID_A     WID_B
        15                                     0.3      0.75
$ Card 2 for the Y-axis
$   AXTYPE   SEGID_L   SEGID_H   COORD_L     SCALE
    Y_AXIS      2002      2003      -5.0       1.0
$ First Card 2.1 for this set
$     NELE      ALEN     RATIO    CRATIO     WID_A     WID_B
        20      10.0       1.0
$ Second Card 2.1 for this set
$     NELE      ALEN     RATIO    CRATIO     WID_A     WID_B
        20      10.0      0.75
$ Card 2 for Z-axis
$   AXTYPE   SEGID_L   SEGID_H   COORD_L     SCALE
    Z_AXIS      2004      2005       0.0       1.0
$ First Card 2.1 for this set
$     NELE      ALEN     RATIO    CRATIO     WID_A     WID_B
                15.0                           0.5      0.95
```

DUALCESE_BOUNDARY_AXISYMMETRIC_*OPTION

Available options are:

MSURF

SEGMENT_SET

Purpose: Define an axisymmetric boundary condition on the axisymmetric axis for the 2D axisymmetric dual CESE compressible flow solver.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Summary:

Card 1a. Include this card for the MSURF keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

MSPID							

Card 1b. Include this card for the SEGMENT_SET keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

SSID							

Data Card Definitions:

Surface Part Card. Card 1 used when the MSURF keyword option is active. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT

Set Card. Card 1 used when the SEGMENT_SET keyword option is active. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH

Remarks:

This boundary condition can only be used on the axisymmetric axis for the 2D axisymmetric dual CESE fluid solver.

DUALCESE_BOUNDARY_CYCLIC_*OPTION

Available options are:

MSURF

SEGMENT_SET

Purpose: Define a cyclic (periodic) boundary condition for dual CESE compressible flows. This cyclic boundary condition can be used on periodic boundary surfaces.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Summary:

Card Sets. The following sequence of cards comprises a *single set*. LS-DYNA will continue reading these cards sets until the next keyword ("*") card is encountered.

Card 1a. Include this card if the MSURF keyword option is used.

MSPID1	MSPID2	CYCTYP					

Card 1b. Include this card if the SEGMENT_SET keyword option is used.

SSID1	SSID2	CYCTYP					

Card 2a. Include this card when CYCTYP = 1.

AXISX1	AXISY1	AXISZ1	DIRX	DIRY	DIRZ	ROTANG	

Card 2b. Include this card when CYCTYP = 2.

TRANSX	TRANSY	TRANSZ					

Data Card Definitions:

Surface Part Card. Card 1 format used when the MSURF keyword option is active.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID1	MSPID2	CYCTYP					
Type	I	I	I					
Default	none	none	0					
Remarks			1, 2					

VARIABLE	DESCRIPTION
MSPID1, MSPID2	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT
CYCTYP	Relationship between the two cyclic boundary condition surfaces: EQ.0: Relationship determined by LS-DYNA (default) EQ.1: The first surface is rotated about an axis to match the second surface. EQ.2: The faces of the first surface are translated in a given direction to obtain the corresponding faces on the second surface.

Set Card. Card 1 format used when the SEGMENT_SET keyword option is active.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID1	SSID2	CYCTYP					
Type	I	I	I					
Default	none	none	0					
Remarks			1, 3					

VARIABLE	DESCRIPTION
SSID1, SSID2	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH
CYCTYP	Relationship between the two cyclic boundary condition surfaces: EQ.0: Relationship determined by LS-DYNA (default) EQ.1: The first surface is rotated about an axis to match the second surface. EQ.2: The faces of the first surface are translated in a given direction to obtain the corresponding faces on the second surface.

Rotation Case Card. Additional card when CYCTYP = 1.

Card 2a	1	2	3	4	5	6	7	8
Variable	AXISX1	AXISY1	AXISZ1	DIRX	DIRY	DIRZ	ROTANG	
Type	F	F	F	F	F	F	F	
Default	0.0	0.0	0.0	none	none	none	none	

VARIABLE	DESCRIPTION
AXIS[X,Y,Z]1	A point on the axis of rotation for the transformation between the surfaces
DIR[X,Y,Z]	The direction which together with AXIS[X,Y,Z]1 defines the axis of rotation for the transformation between the surfaces
ROTANG	The angle of rotation (in degrees) that transforms the centroid of each face on the first surface to the centroid of the corresponding face on the second surface

Translation Case Card. Additional card when CYCTYP = 2.

Card 2b	1	2	3	4	5	6	7	8
Variable	TRANSX	TRANSY	TRANSZ					
Type	F	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
TRANS[X,Y,Z]	The translation direction that enables the identification of the segment in the second surface that matches a segment in the first surface

Remarks:

1. **Unspecified relationship between surfaces.** For the case CYCTYP = 0, LS-DYNA examines the geometry of two faces of the two surfaces in order to determine if the surfaces are approximately parallel (CYCTYP = 2) or related through a rotation (CYCTYP = 1). The geometric parameters required are then computed.

2. **MSURF.** For the MSURF option, each mesh surface part must contain the same number of mesh surface elements. The mesh surface elements in each mesh surface part are internally ordered for pairwise matching between the two mesh surface parts.

3. **SEGMENT_SET.** For the SEGMENT_SET option, each segment set must contain the same number of segments. The segments in each set are internally ordered for pairwise matching between the two sets.

DUALCESE_BOUNDARY_FSI_*OPTION

Available options are:

MSURF

SEGMENT_SET

Purpose: Define an FSI boundary condition for the moving mesh dual CESE compressible flow solver. This keyword must not be combined with the dual CESE immersed-boundary method FSI solver in the same dual CESE part on the same dual CESE mesh. Doing so will result in an error termination condition.

This boundary condition must be applied on a surface of the dual CESE computational domain that is co-located with surfaces of the outside boundary of the structural mesh. The nodes of the two meshes will generally not be shared.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Summary:

Card 1a. Include this card for the MSURF keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

MSPID	REF_P						

Card 1b. Include this card for the SEGMENT_SET keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

SSID	REF_P						

Data Card Definitions:

Surface Part Card. Card 1 used when the MSURF keyword option is active. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID	REF_P						
Type	I	F						
Default	none	0.0						

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT
REF_P	Ambient/reference pressure of the fluid domain on the side opposite this structural interface to the fluid simulation domain. This ambient pressure only needs to be specified in the case where the FSI structural part(s) connected with this FSI interface are not immersed in the dual CESE mesh. This reference pressure defaults to 0.0 since moving mesh FSI calculations most often involve structures surrounded by the dual CESE mesh, and there is no need for a reference pressure in that case.

Set Card. Card 1 used when the SEGMENT_SET keyword option is active. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID	REF_P						
Type	I	F						
Default	none	0.0						

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH

VARIABLE	DESCRIPTION
REF_P	Ambient/reference pressure of the fluid domain on the side opposite this structural interface to the fluid simulation domain. This ambient pressure only needs to be specified in the case where the FSI structural part(s) connected with this FSI interface are not immersed in the dual CESE mesh. This reference pressure defaults to 0.0 since moving mesh FSI calculations most often involve structures surrounded by the dual CESE mesh, and there is no need for a reference pressure in that case..

Remarks:

This boundary condition card is also needed for conjugate heat transfer problems with the moving mesh dual CESE solver. However, the conjugate heat transfer capability is *not* implemented in the dual CESE solver.

DUALCESE_BOUNDARY_NON_REFLECTIVE_*OPTION

Available options are:

MSURF

SEGMENT_SET

Purpose: Define a passive boundary condition for dual CESE compressible flows. This non-reflective boundary condition provides an artificial computational boundary for an open boundary that is passive.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Summary:

Card 1a. Include this card when the MSURF keyword option is used. Include as many cards as necessary. This input ends at the next keyword ("*") card.

MSPID							

Card 1b. Include this card when the SEGMENT_SET keyword option is used. Include as many cards as necessary. This input ends at the next keyword ("*") card.

SSID							

Data Card Definitions:

Surface Part Card. Card 1 used when the MSURF keyword option is active. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT

Set Card. Card 1 used when the SET keyword option is active. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH

Remarks:

1. **Boundary surface flow.** This boundary condition is usually imposed on an open surface that is far from the main disturbed flow (the further away, the better), meaning the flow on the boundary surface should be almost uniform.

2. **Default boundary condition**. If any boundary segment has not been assigned a boundary condition by any of the *DUALCESE_BOUNDARY_… cards, then it will automatically be assigned this non-reflective boundary condition.

DUALCESE_BOUNDARY_PRESCRIBED_*OPTION

Available options include:

MSURF

SEGMENT_SET

Purpose: For the dual CESE compressible flow solver, set boundary values for velocity, density, pressure, and temperature. Boundary values are applied at the centroid of elements connected with this boundary.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Sets:

A set of data cards for this keyword consists of 3 of the following cards:

1. Card 1 specifies the object to which the boundary condition is applied. Its format depends on the keyword option.
2. Card 2 reads in load curve IDs.
3. Card 3 reads in scale factors.

For each boundary condition to be specified include one set of cards. This input ends at the next keyword ("*") card.

Card Summary:

Card 1a. Include this card if the MSURF keyword option is used.

MSPID	IDCOMP	DIRX	DIRY	DIRZ			

Card 1b. Include this card if the SEGMENT_SET keyword option is used.

SSID	IDCOMP	DIRX	DIRY	DIRZ			

Card 2. This card is required.

LC_U	LC_V	LC_W	LC_RHO	LC_P	LC_T		

Card 3. This card is required.

SF_U	SF_V	SF_W	SF_RHO	SF_P	SF_T		

Data Card Definitions:

Surface Part Card. Card 1 format used when the MSURF keyword option is active.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID	IDCOMP	DIRX	DIRY	DIRZ			
Type	I	I	F	F	F			
Default	none	none	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT
IDCOMP	For inflow boundaries in problems involving chemical reacting flows, the chemical mixture of the fluid entering the domain is defined with a *CHEMISTRY_COMPOSITION card with this ID [Not yet available].
DIRX, DIRY, DIRZ	If this vector is non-zero, then it is used as the prescribed flow direction.

Set Card. Card 1 format used when the SEGMENT_SET keyword option is active.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID	IDCOMP	DIRX	DIRY	DIRZ			
Type	I	I	F	F	F			
Default	none	none	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH
IDCOMP	For inflow boundaries in problems involving chemical reacting flows, the chemical mixture of the fluid entering the domain is defined with a *CHEMISTRY_COMPOSITION card with this ID [Not yet available].
DIRX, DIRY, DIRZ	If this vector is non-zero, then it is used as the prescribed flow direction.

Load Curve Card.

Card 2	1	2	3	4	5	6	7	8
Variable	LC_U	LC_V	LC_W	LC_RHO	LC_P	LC_T		
Type	I	I	I	I	I	I		
Remarks	1	1	1	1	1	1		

VARIABLE	DESCRIPTION
LC_U	Load curve ID (see *DEFINE_CURVE) to describe the x-component of the velocity as a function of time or function ID (see *DEFINE_FUNCTION) to give the x-component of the velocity as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: x-component of velocity is a constant with value SF_U. EQ.-1: x-component of velocity is computed by the solver.
LC_V	Load curve ID to describe the y-component of the velocity as a function of time or function ID to give the y-component of the velocity as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: y-component of velocity is a constant with value SF_V. EQ.-1: y-component of velocity is computed by the solver.
LC_W	Load curve ID to describe the z-component of the velocity as a function of time or function ID to give the z-component of the

VARIABLE	DESCRIPTION
	velocity as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: z-component of velocity is a constant with value SF_W. EQ.-1: z-component of velocity is computed by the solver.
LC_RHO	Load curve ID to describe the density as a function of time or function ID to give the density as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: Density is a constant with value SF_RHO. EQ.-1: Density is computed by the solver.
LC_P	Load curve ID to describe the pressure as a function of time or function ID to give the pressure as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: Pressure is a constant with value SF_P. EQ.-1: Pressure is computed by the solver.
LC_T	Load curve ID to describe the temperature as a function of time or function ID to give the temperature as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: Temperature is a constant with value SF_T. EQ.-1: Temperature is computed by the solver.

Scale Factor Card.

Card 3	1	2	3	4	5	6	7	8
Variable	SF_U	SF_V	SF_W	SF_RHO	SF_P	SF_T		
Type	F	F	F	F	F	F		
Default	1.0	1.0	1.0	1.0	1.0	1.0		

VARIABLE	DESCRIPTION
SF_U	Scale factor for LC_U

VARIABLE	DESCRIPTION
SF_V	Scale factor for LC_V
SF_W	Scale factor for LC_W
SF_RHO	Scale factor for LC_RHO
SF_P	Scale factor for LC_P
SF_T	Scale factor for LC_T

Remarks:

1. **Consistent boundary values.** On each centroid or set of centroids, the variables (v_x, v_y, v_z, ρ, P, T) that are given values must be consistent and make the model well-posed (meaning be such that the solution of the model exists, is unique, and is physical).

DUALCESE_BOUNDARY_PRESCRIBED_HYBRID_*OPTION

Available options include:

MSURF

SEGMENT_SET

Purpose: For the dual CESE compressible flow solver, set boundary values for velocity, density, pressure, temperature, and other values in the hybrid multiphase model. Boundary values are applied at the centroid of elements connected with this boundary.

Use the MSURF option when the dual CESE mesh has been created using *MESH cards. Use the SEGMENT_SET option when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Sets:

A set of data cards for this keyword consists of 3 of the following cards:

1. Card 1 specifies the object to which the boundary condition is applied. Its format depends on the keyword option.
2. Cards 2 and 3 provide load curve IDs.
3. Cards 4 and 5 provide scale factors.

For each boundary condition to be specified, include one set of cards. This input ends at the next keyword ("*") card.

Card Summary:

Card 1a. Include this card if the keyword option is set to MSURF.

MSPID		DIRX	DIRY	DIRZ			

Card 1b. Include this card if the keyword option is set to SSID.

SSID		DIRX	DIRY	DIRZ			

Card 2. This card is required.

LC_Z1	LC_RA	LC_U	LC_V	LC_W	LC_D1	LC_DA	LC_DB

Card 3. This card is required.

LC_P	LC_T						

Card 4. This card is required.

SF_Z1	SF_RA	SF_U	SF_V	SF_W	SF_D1	SF_DA	SF_DB

Card 5. This card is required.

SF_P	SF_T						

Data Card Definitions:

Surface Part Set Card. Card 1 format used when the MSURF keyword option is active.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID		DIRX	DIRY	DIRZ			
Type	I		F	F	F			
Default	none		0.0	0.0	0.0			

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT
DIRX, DIRY, DIRZ	If this vector is nonzero, then it is used as the prescribed flow direction.

Segment Set Card. Card 1 format used for the SEGMENT_SET keyword option.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID		DIRX	DIRY	DIRZ			
Type	I		F	F	F			
Default	none		0.0	0.0	0.0			

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH
DIRX, DIRY, DIRZ	If this vector is nonzero, then it is used as the prescribed flow direction.

Load Curve Card. See Remark 1.

Card 2	1	2	3	4	5	6	7	8
Variable	LC_Z1	LC_RA	LC_U	LC_V	LC_W	LC_D1	LC_DA	LC_DB
Type	I	I	I	I	I	I		

VARIABLE	DESCRIPTION
LC_Z1	Load curve ID or function ID to describe the volume fraction of material 1 as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The volume fraction is a constant with value SF_Z1. EQ.-1: The volume fraction is computed by the solver.
LC_RA	Load curve or function ID to describe the mass fraction of reactant (material α) with respect to the explosive mixture (material 2) as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The mass fraction is a constant with value SF_RA. EQ.-1: The mass fraction is computed by the solver.
LC_U	Load curve or defined function ID to describe the x-component of the velocity as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The x-component of velocity is a constant with value SF_U. EQ.-1: The x-component of velocity is computed by the solver.

VARIABLE	DESCRIPTION
LC_V	Load curve or defined function ID to describe the y-component of the velocity as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The y-component of velocity is a constant with value SF_V. EQ.-1: The y-component of velocity is computed by the solver.
LC_W	Load curve or defined function ID to describe the z-component of the velocity as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The z-component of velocity is a constant with value SF_W. EQ.-1: The z-component of velocity is computed by the solver.
LC_D1	Load curve or defined function ID to describe the density of the first multiphase material as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The density of the first multiphase material is a constant with value SF_D1. EQ.-1: The density of the first multiphase material is computed by the solver.
LC_DA	Load curve or defined function ID to describe the density of the reactant (material α) as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The density of the reactant is a constant with value SF_DA. EQ.-1: The density of the reactant is computed by the solver.
LC_DB	Load curve or defined function ID to describe the density of the product (material β) as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The density of the product is a constant with value SF_DB.

VARIABLE	DESCRIPTION
	EQ.-1: The density of the product is computed by the solver.

Load Curve Card 2. See Remark 1.

Card 3	1	2	3	4	5	6	7	8
Variable	LC_P	LC_T						
Type	I	I						

VARIABLE	DESCRIPTION
LC_P	Load curve or defined function ID to describe the pressure as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The pressure is a constant with value SF_P. EQ.-1: The pressure is computed by the solver.
LC_T	Load curve or defined function ID to describe the temperature as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The temperature is a constant with value SF_T. EQ.-1: The temperature is computed by the solver.

Scale Factor Card.

Card 4	1	2	3	4	5	6	7	8
Variable	SF_Z1	SF_RA	SF_U	SF_V	SF_W	SF_D1	SF_DA	SF_DB
Type	F	F	F	F	F	F	F	F
Default	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

VARIABLE	DESCRIPTION
SF_Z	Scale factor for LC_Z1
SF_RA	Scale factor for LC_RA
SF_U	Scale factor for LC_U
SF_V	Scale factor for LC_V
SF_W	Scale factor for LC_W
SF_D1	Scale factor for LC_D1
SF_DA	Scale factor for LC_DA
SF_DB	Scale factor for LC_DB

Scale Factor Card 2.

Card 5	1	2	3	4	5	6	7	8
Variable	SF_P	SF_T						
Type	F	F						
Default	1.0	1.0						

VARIABLE	DESCRIPTION
SF_P	Scale factor for LC_P
SF_T	Scale factor for LC_T

Remark:

1. **Consistent boundary values.** On each centroid or set of element centroids, the variables ($v_x, v_y, v_z, \rho, P, T, \ldots$) that are given values must be consistent and make the model well-posed (meaning be such that the solution of the model exists, is unique, and is physical).

DUALCESE_BOUNDARY_PRESCRIBED_INLET_MACHNUM_*OPTION

Available options include:

MSURF

SEGMENT_SET

Purpose: For the dual CESE compressible flow solver, set inlet boundary values using total pressure, total temperature, and Mach number. Specifying an arbitrary inlet flow direction is an option. Boundary values are applied at the centroid of elements connected with this boundary.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Sets:

A set of data cards for this keyword consists of 3 of the following cards:

1. Card 1 specifies the object to which the boundary condition is applied. Its format depends on the keyword option.
2. Card 2 reads in load curve IDs.
3. Card 3 reads in scale factors.

For each boundary condition to be specified, include one set of cards. This input ends at the next keyword ("*") card.

Card Summary:

Card 1a. Include this card if the MSURF keyword option is used.

MSPID	IWALL	DIR_X	DIR_Y	DIR_Z			

Card 1b. Include this card if the SEGMENT_SET keyword option is used.

SSID	IWALL	DIR_X	DIR_Y	DIR_Z			

Card 2. This card is required.

LC_TOTP	LC_TOTT	LC_MACH					

Card 3. This card is required.

SF_TOTP	SF_TOTT	SF_MACH					

Data Card Definitions:

Surface Part Card. Include this card when the MSURF keyword option is active.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID	IWALL	DIRX	DIRY	DIRZ			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT
IWALL	Artificial wall flag: EQ.0: Do not treat as an artificial wall if there is flow recirculation at this inlet boundary. EQ.1: Do treat this inlet boundary as an artificial wall whenever flow recirculation exists there.
DIRX, DIRY, DIRZ	If this vector is nonzero, it is used as the prescribed flow direction.

Set Card. Include this card when the SEGMENT_SET keyword option is active.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID	IWALL	DIRX	DIRX	DIRY			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH
IWALL	Artificial wall flag: EQ.0: Do not treat as an artificial wall if there is flow recirculation at this inlet boundary. EQ.1: Do treat this inlet boundary as an artificial wall whenever flow recirculation exists there.
DIRX, DIRY, DIRZ	If this vector is nonzero, it is used as the prescribed flow direction.

Load Curve Card.

Card 2	1	2	3	4	5	6	7	8
Variable	LC_TOTP	LC_TOTT	LC_MACH					
Type	I	I	I					
Remarks	1	1	1					

VARIABLE	DESCRIPTION
LC_TOTP	Load curve ID (see *DEFINE_CURVE) to describe the total pressure as a function of time or function ID (see *DEFINE_FUNCTION) to give the total pressure as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: Total pressure is a constant with value SF_TOTP.
LC_TOTT	Load curve ID (see *DEFINE_CURVE) to describe the total temperature as a function of time or function ID (see *DEFINE_FUNCTION) to give the total temperature as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: Total temperature is a constant with value SF_TOTT.
LC_MACH	Load curve ID to describe the Mach number as a function of time or function ID to give the Mach number as a function of position,

VARIABLE	DESCRIPTION
	velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time).
	EQ.0: Mach number is a constant with value SF_MACH.

Scale Factor Card.

Card 3	1	2	3	4	5	6	7	8
Variable	SF_TOTP	SF_TOTT	SF_MACH					
Type	F	F	F					
Default	1.0	1.0	1.0					

VARIABLE	DESCRIPTION
SF_TOTP	Scale factor for LC_TOTP
SF_TOTT	Scale factor for LC_TOTT
SF_MACH	Scale factor for LC_MACH

Remarks:

1. **Consistent boundary values.** On each centroid or set of centroids, the variables (P_{total}, T_{total}, Mach number) that are given values must be consistent and make the model well-posed (meaning be such that the solution of the model exists, is unique, and is physical).

DUALCESE_BOUNDARY_PRESCRIBED_INLET_MFRATE_*OPTION

Available options include:

MSURF

SEGMENT_SET

Purpose: For the dual CESE compressible flow solver, set inlet boundary values using total pressure, total temperature, and mass flow rate. Specifying an arbitrary inlet flow direction is an option. Boundary values are applied at the centroid of elements connected with this boundary.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Sets:

A set of data cards for this keyword consists of 3 of the following cards:

1. Card 1 specifies the object to which the boundary condition is applied. Its format depends on the keyword option.
2. Card 2 reads in load curve IDs.
3. Card 3 reads in scale factors.

For each boundary condition to be specified, include one set of cards. This input ends at the next keyword ("*") card.

Card Summary:

Card 1a. Include this card if the MSURF keyword option is used.

MSPID	IWALL	DIR_X	DIR_Y	DIR_Z			

Card 1b. Include this card if the SEGMENT_SET keyword option is used.

SSID	IWALL	DIR_X	DIR_Y	DIR_Z			

Card 2. This card is required.

LC_TOTP	LC_TOTT	LC_MFRT					

Card 3. This card is required.

SF_TOTP	SF_TOTT	SF_MFRT					

Data Card Definitions:

Surface Part Card. Include this card when the MSURF keyword option is active.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID	IWALL	DIRX	DIRY	DIRZ			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT
IWALL	Artificial wall flag: EQ.0: Do not treat as an artificial wall if there is flow recirculation at this inlet boundary. EQ.1: Do treat this inlet boundary as an artificial wall whenever flow recirculation exists there.
DIRX, DIRY, DIRZ	If this vector is nonzero, it is used as the prescribed flow direction.

Set Card. Include this card when the SEGMENT_SET keyword option is active.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID	IWALL	DIRX	DIRX	DIRY			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH
IWALL	Artificial wall flag: EQ.0: Do not treat as an artificial wall if there is flow recirculation at this inlet boundary. EQ.1: Do treat this inlet boundary as an artificial wall whenever flow recirculation exists there.
DIRX, DIRY, DIRZ	If this vector is nonzero, it is used as the prescribed flow direction.

Load Curve Card.

Card 2	1	2	3	4	5	6	7	8
Variable	LC_TOTP	LC_TOTT	LC_MFRT					
Type	I	I	I					
Remarks	1	1	1					

VARIABLE	DESCRIPTION
LC_TOTP	Load curve ID (see *DEFINE_CURVE) to describe the total pressure as a function of time or function ID (see *DEFINE_FUNCTION) to give the total pressure as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: Total pressure is a constant with value SF_TOTP.
LC_TOTT	Load curve ID (see *DEFINE_CURVE) to describe the total temperature as a function of time or function ID (see *DEFINE_FUNCTION) to give the total temperature as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: Total temperature is a constant with value SF_TOTT.
LC_MFRT	Load curve ID to describe the mass flow rate as a function of time or function ID to give the mass flow rate as a function of position,

VARIABLE	DESCRIPTION
	velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time).
	EQ.0: Mass flow rate is a constant with value SF_MFRT.

Scale Factor Card.

Card 3	1	2	3	4	5	6	7	8
Variable	SF_TOTP	SF_TOTT	SF_MFRT					
Type	F	F	F					
Default	1.0	1.0	1.0					

VARIABLE	DESCRIPTION
SF_TOTP	Scale factor for LC_TOTP
SF_TOTT	Scale factor for LC_TOTT
SF_MFRT	Scale factor for LC_MFRT

Remarks:

1. **Consistent boundary values.** On each centroid or set of centroids, the variables (P_{total}, T_{total}, mass flow rate) that are given values must be consistent and make the model well-posed (meaning be such that the solution of the model exists, is unique, and is physical).

DUALCESE_BOUNDARY_PRESCRIBED_INLET_PRESSURE_*OPTION

Available options include:

MSURF

SEGMENT_SET

Purpose: For the dual CESE compressible flow solver, set inlet boundary values using total pressure, total temperature, and static pressure. Specifying an arbitrary inlet flow direction is an option. Boundary values are applied at the centroid of elements connected with this boundary.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Sets:

A set of data cards for this keyword consists of 3 of the following cards:

1. Card 1 specifies the object to which the boundary condition is applied. Its format depends on the keyword option.
2. Card 2 reads in load curve IDs.
3. Card 3 reads in scale factors.

For each boundary condition to be specified, include one set of cards. This input ends at the next keyword ("*") card.

Card Summary:

Card 1a. Include this card if the MSURF keyword option is used.

MSPID	IWALL	DIR_X	DIR_Y	DIR_Z			

Card 1b. Include this card if the SEGMENT_SET keyword option is used.

SSID	IWALL	DIR_X	DIR_Y	DIR_Z			

Card 2. This card is required.

LC_TOTP	LC_TOTT	LC_PSTAT					

Card 3. This card is required.

SF_TOTP	SF_TOTT	SF_PSTAT					

Data Card Definitions:

Surface Part Card. Include this card when the MSURF keyword option is active.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID	IWALL	DIRX	DIRY	DIRZ			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT
IWALL	Artificial wall flag: EQ.0: Do not treat as an artificial wall if there is flow recirculation at this inlet boundary. EQ.1: Do treat this inlet boundary as an artificial wall whenever flow recirculation exists there.
DIRX, DIRY, DIRZ	If this vector is nonzero, it is used as the prescribed flow direction.

Set Card. Include this card when the SEGMENT_SET keyword option is active.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID	IWALL	DIRX	DIRX	DIRY			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH
IWALL	Artificial wall flag: EQ.0: Do not treat as an artificial wall if there is flow recirculation at this inlet boundary. EQ.1: Do treat this inlet boundary as an artificial wall whenever there is flow recirculation there.
DIRX, DIRY, DIRZ	If this vector is nonzero, it is used as the prescribed flow direction.

Load Curve Card.

Card 2	1	2	3	4	5	6	7	8
Variable	LC_TOTP	LC_TOTT	LC_PSTAT					
Type	I	I	I					
Remarks	1	1	1					

VARIABLE	DESCRIPTION
LC_TOTP	Load curve ID (see *DEFINE_CURVE) to describe the total pressure as a function of time or function ID (see *DEFINE_FUNCTION) to give the total pressure as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: Total pressure is a constant with value SF_TOTP.
LC_TOTT	Load curve ID (see *DEFINE_CURVE) to describe the total temperature as a function of time or function ID (see *DEFINE_FUNCTION) to give the total temperature as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: Total temperature is a constant with value SF_TOTT.
LC_PSTAT	Load curve ID to describe the static pressure as a function of time or function ID to give the static pressure as a function of position,

VARIABLE	DESCRIPTION
	velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time).
	EQ.0: Static pressure is a constant with value SF_PSTAT.

Scale Factor Card.

Card 3	1	2	3	4	5	6	7	8
Variable	SF_TOTP	SF_TOTT	SF_PSTAT					
Type	F	F	F					
Default	1.0	1.0	1.0					

VARIABLE	DESCRIPTION
SF_TOTP	Scale factor for LC_TOTP
SF_TOTT	Scale factor for LC_TOTT
SF_PSTAT	Scale factor for LC_PSTAT

Remarks:

1. **Consistent boundary values.** On each centroid or set of centroids, the variables (P_{total}, T_{total}, P_{static}) that are given values must be consistent and make the model well-posed (meaning be such that the solution of the model exists, is unique, and is physical).

DUALCESE_BOUNDARY_PRESCRIBED_OUTLET_EXMFRATE_*OPTION

Available options include:

MSURF

SEGMENT_SET

Purpose: For the dual CESE compressible flow solver, set outlet boundary values using reference pressure, reference temperature, and exit mass flow rate. Specifying an arbitrary outlet flow direction is an option. Boundary values are applied at the centroid of elements connected with this boundary.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Sets:

A set of data cards for this keyword consists of 3 of the following cards:

1. Card 1 specifies the object to which the boundary condition is applied. Its format depends on the keyword option.
2. Card 2 reads in load curve IDs.
3. Card 3 reads in scale factors.

For each boundary condition to be specified include one set of cards. This input ends at the next keyword ("*") card.

Card Summary:

Card 1a. Include this card if the MSURF keyword option is used.

MSPID	IWALL	DIR_X	DIR_Y	DIR_Z			

Card 1b. Include this card if the SEGMENT_SET keyword option is used.

SSID	IWALL	DIR_X	DIR_Y	DIR_Z			

Card 2. This card is required.

LC_PREF	LC_TREF	LC_MFRT					

Card 3. This card is required.

SF_PREF	SF_TREF	SF_MFRT					

Data Card Definitions:

Surface Part Card. Include this card when the MSURF keyword option is active.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID	IWALL	DIRX	DIRY	DIRZ			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT
IWALL	Artificial wall flag: EQ.0: Do not treat as an artificial wall if there is flow recirculation at this outlet boundary. EQ.1: Do treat this outlet boundary as an artificial wall whenever flow recirculation exists there.
DIRX, DIRY, DIRZ	If this vector is nonzero, it is used as the prescribed flow direction.

Set Card. Include this card when the SEGMENT_SET keyword option is active.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID	IWALL	DIRX	DIRX	DIRY			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH
IWALL	Artificial wall flag: EQ.0: Do not treat as an artificial wall if there is flow recirculation at this outlet boundary. EQ.1: Do treat this outlet boundary as an artificial wall whenever flow recirculation exists there.
DIRX, DIRY, DIRZ	If this vector is nonzero, it is used as the prescribed flow direction.

Load Curve Card.

Card 2	1	2	3	4	5	6	7	8
Variable	LC_PREF	LC_TREF	LC_MFRT					
Type	I	I	I					
Remarks	1	1	1					

VARIABLE	DESCRIPTION
LC_PREF	Load curve ID (see *DEFINE_CURVE) to describe the reference pressure as a function of time or function ID (see *DEFINE_FUNCTION) to give the reference pressure as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: Reference pressure is a constant with value SF_PREF.
LC_TREF	Load curve ID (see *DEFINE_CURVE) to describe the reference temperature as a function of time or function ID (see *DEFINE_FUNCTION) to give the reference temperature as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: Reference temperature is a constant with value SF_TREF.
LC_MFRT	Load curve ID to describe the exit mass flow rate as a function of time or function ID to give the exit mass flow rate as a function of

VARIABLE	DESCRIPTION
	position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time).
	EQ.0: exit mass flow rate is a constant with value SF_MFRT.

Scale Factor Card.

Card 3	1	2	3	4	5	6	7	8
Variable	SF_PREF	SF_TREF	SF_MFRT					
Type	F	F	F					
Default	1.0	1.0	1.0					

VARIABLE	DESCRIPTION
SF_PREF	Scale factor for LC_PREF
SF_TREF	Scale factor for LC_TREF
SF_MFRT	Scale factor for LC_MFRT

Remarks:

1. **Consistent boundary values.** On each centroid or set of centroids, the variables (P_{ref}, T_{ref}, Exit Mass Flow Rate) that are given values must be consistent and make the model well-posed (meaning be such that the solution of the model exists, is unique, and is physical).

DUALCESE_BOUNDARY_PRESCRIBED_OUTLET_FARFIELD_*OPTION

Available options include:

MSURF

SEGMENT_SET

Purpose: For the dual CESE compressible flow solver, set outlet boundary values using static pressure, static temperature, and Mach number. Specifying an arbitrary outlet flow direction is an option. Boundary values are applied at the centroid of elements connected with this boundary.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Sets:

A set of data cards for this keyword consists of 3 of the following cards:

1. Card 1 specifies the object to which the boundary condition is applied. Its format depends on the keyword option.
2. Card 2 reads in load curve IDs.
3. Card 3 reads in scale factors.

For each boundary condition to be specified, include one set of cards. This input ends at the next keyword ("*") card.

Card Summary:

Card 1a. Include this card if the MSURF keyword option is used.

MSPID	IWALL	DIR_X	DIR_Y	DIR_Z			

Card 1b. Include this card if the SEGMENT_SET keyword option is used.

SSID	IWALL	DIR_X	DIR_Y	DIR_Z			

Card 2. This card is required.

LC_PSTAT	LC_TSTAT	LC_MACH					

Card 3. This card is required.

SF_PSTAT	SF_TSTAT	SF_MACH					

Data Card Definitions:

Surface Part Card. Include this card when the MSURF keyword option is active.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID	IWALL	DIRX	DIRY	DIRZ			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT
IWALL	Artificial wall flag: EQ.0: Do not treat as an artificial wall if there is flow recirculation at this outlet boundary. EQ.1: Do treat this outlet boundary as an artificial wall whenever flow recirculation exists there.
DIRX, DIRY, DIRZ	If this vector is nonzero, it is used as the prescribed flow direction.

Set Card. Include this card when the SEGMENT_SET keyword option is active.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID	IWALL	DIRX	DIRX	DIRY			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH
IWALL	Artificial wall flag: EQ.0: Do not treat as an artificial wall if there is flow recirculation at this outlet boundary. EQ.1: Do treat this outlet boundary as an artificial wall whenever flow recirculation exists there.
DIRX, DIRY, DIRZ	If this vector is nonzero, it is used as the prescribed flow direction.

Load Curve Card.

Card 2	1	2	3	4	5	6	7	8
Variable	LC_PSTAT	LC_TSTAT	LC_MACH					
Type	I	I	I					
Remarks	1	1	1					

VARIABLE	DESCRIPTION
LC_PSTAT	Load curve ID (see *DEFINE_CURVE) to describe the static pressure as a function of time or function ID (see *DEFINE_FUNCTION) to give the static pressure as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: Static pressure is a constant with value SF_PSTAT.
LC_TSTAT	Load curve ID (see *DEFINE_CURVE) to describe the static temperature as a function of time or function ID (see *DEFINE_FUNCTION) to give the static temperature as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: Static temperature is a constant with value SF_TSTAT.
LC_MACH	Load curve ID to describe the Mach number as a function of time or function ID to give the Mach number as a function of position,

VARIABLE	DESCRIPTION
	velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time).
	EQ.0: Mach number is a constant with value SF_MACH.

Scale Factor Card.

Card 3	1	2	3	4	5	6	7	8
Variable	SF_PSTAT	SF_TSTAT	SF_MACH					
Type	F	F	F					
Default	1.0	1.0	1.0					

VARIABLE	DESCRIPTION
SF_PSTAT	Scale factor for LC_PSTAT
SF_TSTAT	Scale factor for LC_TSTAT
SF_MACH	Scale factor for LC_MACH

Remarks:

1. **Consistent boundary values.** On each centroid or set of centroids, the variables (P_{static}, T_{static}, Mach number) that are given values must be consistent and make the model well-posed (meaning be such that the solution of the model exists, is unique, and is physical).

DUALCESE_BOUNDARY_PRESCRIBED_OUTLET_PRESSURE_*OPTION

Available options include:

MSURF

SEGMENT_SET

Purpose: For the dual CESE compressible flow solver, set outlet boundary values using static pressure. Specifying an arbitrary outlet flow direction is an option. Boundary values are applied at the centroid of elements connected with this boundary.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Sets:

A set of data cards for this keyword consists of 3 of the following cards:

1. Card 1 specifies the object to which the boundary condition is applied. Its format depends on the keyword option.
2. Card 2 reads in load curve IDs.
3. Card 3 reads in scale factors.

For each boundary condition to be specified, include one set of cards. This input ends at the next keyword ("*") card.

Card Summary:

Card 1a. Include this card if the MSURF keyword option is used.

MSPID	IWALL	DIR_X	DIR_Y	DIR_Z			

Card 1b. Include this card if the SEGMENT_SET keyword option is used.

SSID	IWALL	DIR_X	DIR_Y	DIR_Z			

Card 2. This card is required.

LC_PSTAT							

Card 3. This card is required.

SF_PSTAT							

Data Card Definitions:

Surface Part Card. Include this card when the MSURF keyword option is active.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID	IWALL	DIRX	DIRY	DIRZ			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT
IWALL	Artificial wall flag: EQ.0: Do not treat as an artificial wall if there is flow recirculation at this outlet boundary. EQ.1: Do treat this outlet boundary as an artificial wall whenever flow recirculation exists there.
DIRX, DIRY, DIRZ	If this vector is nonzero, it is used as the prescribed flow direction.

Set Card. Card 1b format used when the SEGMENT_SET keyword option is active.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID	IWALL	DIRX	DIRX	DIRY			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH
IWALL	Artificial wall flag: EQ.0: Do not treat as an artificial wall if there is flow recirculation at this outlet boundary. EQ.1: Do treat this outlet boundary as an artificial wall whenever flow recirculation exists there.
DIRX, DIRY, DIRZ	If this vector is nonzero, it is used as the prescribed flow direction.

Card 2	1	2	3	4	5	6	7	8
Variable	LC_PSTAT							
Type	I							
Remarks	1							

VARIABLE	DESCRIPTION
LC_PSTAT	Load curve ID (see *DEFINE_CURVE) to describe the total pressure as a function of time or function ID (see *DEFINE_FUNCTION) to give the total pressure as a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time). EQ.0: Total pressure is a constant with value SF_PSTAT.

Card 3	1	2	3	4	5	6	7	8
Variable	SF_PSTAT							
Type	F							
Default	1.0							

VARIABLE	DESCRIPTION
SF_PSTAT	Scale factor for LC_PSTAT

Remarks:

1. **Consistent boundary values.** On each centroid or set of centroids, the variable (P_{static}) that is given a value must be consistent and make the model well-posed (meaning be such that the solution of the model exists, is unique, and is physical).

DUALCESE_BOUNDARY_PRESCRIBED_PHASE_CHANGE_*OPTION

Available options are:

MSURF

SEGMENT_SET

Purpose: For the dual CESE compressible flow solver, set boundary values for velocity, density, pressure, temperature, and other values in the phase-change multiphase model. Boundary values are applied at the centroid of outside elements connected with this boundary.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Sets:

A set of data cards for this keyword consists of 3 of the following cards:

1. Card 1 specifies the object to which the boundary condition is applied. Its format depends on the keyword option.
2. Cards 2 and 3 provide load curve IDs.
3. Cards 4 and 5 provide scale factors.

For each boundary condition to be specified, include one set of cards. This input ends at the next keyword ("*") card.

Card Summary:

Card 1a. Include this card for the MSURF keyword option.

MSPID		DIRX	DIRY	DIRZ			

Card 1b. Include this card for the SEGMENT_SET keyword option.

SSID		DIRX	DIRY	DIRZ			

Card 2. This card is required.

LCIDU	LCIDV	LCIDW	LCIDRHO	LCIDP	LCIDT	LCIDY1	LCIDY2

Card 3. This card is required.

LCIDY3							

Card 4. This card is required.

SFU	SFV	SFW	SFRHO	SFP	SFT	SFY1	SFY2

Card 5. This card is required.

SFY3							

Data Card Definitions:

Surface Part Card. Card 1 used when the MSURF keyword option is active.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID		DIRX	DIRY	DIRZ			
Type	I		F	F	F			
Default	none		0.0	0.0	0.0			

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT
DIRX, DIRY, DIRZ	If nonzero, vector giving the prescribed flow direction

Set Card. Card 1 used when the SEGMENT_SET keyword option is active.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID		DIRX	DIRY	DIRZ			
Type	I		F	F	F			
Default	none		0.0	0.0	0.0			

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH
DIRX, DIRY, DIRZ	If nonzero, vector giving the prescribed flow direction

Card 2	1	2	3	4	5	6	7	8
Variable	LCIDU	LCIDV	LCIDW	LCIDRHO	LCIDP	LCIDT	LCIDY1	LCIDY2
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0
Remarks	1	1	1	1	1	1	1	1

VARIABLE	DESCRIPTION
LCIDU	Load curve or defined function ID to describe the *xx*-component of the velocity as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The *x*-component of velocity is a constant with value SFU. EQ.-1: The *x*-component of velocity is computed by the solver.
LCIDV	Load curve or defined function ID to describe the *yy*-component of the velocity as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The *y*-component of velocity is a constant with value SFV. EQ.-1: The *y*-component of velocity is computed by the solver.
LCIDW	Load curve or defined function ID to describe the *zz*-component of the velocity as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively.

VARIABLE	DESCRIPTION
	EQ.0: The z-component of velocity is a constant with value SFW.
	EQ.-1: The z-component of velocity is computed by the solver.
LCIDRHO	Load curve or defined function ID to describe the mixture density as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively.
	EQ.0: The density is a constant with value SFRHO.
	EQ.-1: The density is computed by the solver.
LCIDP	Load curve or defined function ID to describe the pressure as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively.
	EQ.0: The pressure is a constant with value SFP.
	EQ.-1: The pressure is computed by the solver.
LCIDT	Load curve or defined function ID to describe the temperature as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively.
	EQ.0: The temperature is a constant with value SFT.
	EQ.-1: The temperature is computed by the solver.
LCIDY1	Load curve or defined function ID to describe the mass fraction of material 1 as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively.
	EQ.0: The mass fraction is a constant with value SFY1.
	EQ.-1: The mass fraction is computed by the solver.
LCIDY2	Load curve or defined function ID to describe the mass fraction of material 2 as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively.
	EQ.0: The mass fraction is a constant with value SFY2.
	EQ.-1: The mass fraction is computed by the solver.

Card 3	1	2	3	4	5	6	7	8
Variable	LCIDY3							
Type	I							
Default	0							
Remarks	1							

VARIABLE	DESCRIPTION
LCIDY3	Load curve or defined function ID to describe the mass fraction of material 3 as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The mass fraction is a constant with value SFY3. EQ.-1: The mass fraction is computed by the solver.

Card 4	1	2	3	4	5	6	7	8
Variable	SFU	SFV	SFW	SFRH0	SFP	SFT	SFY1	SFY2
Type	F	F	F	F	F	F	F	F
Default	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

VARIABLE	DESCRIPTION
SFU	Scale factor for LCIDU
SFV	Scale factor for LCIDV
SFW	Scale factor for LCIDW
SFRHO	Scale factor for LCIDRHO
SFP	Scale factor for LCIDP

VARIABLE	DESCRIPTION
SFT	Scale factor for LCIDT
SFY1	Scale factor for LCIDY1
SFY2	Scale factor for LCIDY2

Card 5	1	2	3	4	5	6	7	8
Variable	SFY3							
Type	F							
Default	1.0							

VARIABLE	DESCRIPTION
SFY3	Scale factor for LCIDY3

Remarks:

1. Consistent boundary values. On each centroid or set of element centroids, the variables $(v_x, v_y, v_z, \rho, P, T, \ldots)$ that are given values must be consistent and make the model well-posed (meaning be such that the solution of the model exists, is unique, and is physical).

DUALCESE_BOUNDARY_PRESCRIBED_TWO-PHASE_*OPTION

Available options include:

MSURF

SEGMENT_SET

Purpose: For the dual CESE compressible flow solver, set boundary values for velocity, density, pressure, temperature, and other values in the two-phase multiphase model. Boundary values are applied at the centroid of elements connected with this boundary.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Sets:

A set of data cards for this keyword consists of 3 of the following cards:

1. Card 1 specifies the object to which the boundary condition is applied. Its format depends on the keyword option.
2. Card 2 provides load curve IDs.
3. Card 3 provides scale factors.

For each boundary condition to be specified, include one set of cards. This input ends at the next keyword ("*") card.

Card Summary:

Card 1a. Include this card if the keyword option is set to MSURF.

MSPID		DIRX	DIRY	DIRZ			

Card 1b. Include this card if the keyword option is set to SSID.

SSID		DIRX	DIRY	DIRZ			

Card 2. This card is required.

LC_Z1	LC_U	LC_V	LC_W	LC_D1	LC_D2	LC_P	LC_T

Card 3. This card is required.

SF_Z1	SF_U	SF_V	SF_W	SF_D1	SF_D2	SF_P	SF_T

Data Card Definitions:

Surface Part Set Card. Card 1 format used when the MSURF keyword option is active.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID		DIRX	DIRY	DIRZ			
Type	I		F	F	F			
Default	none		0.0	0.0	0.0			

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT
DIRX, DIRY, DIRZ	If this vector is nonzero, then it is used as the prescribed flow direction.

Segment Set Card. Card 1 format used when the SEGMENT_SET keyword option is active.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID		DIRX	DIRY	DIRZ			
Type	I		F	F	F			
Default	none		0.0	0.0	0.0			

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH

VARIABLE	DESCRIPTION
DIRX, DIRY, DIRZ	If this vector is nonzero, then it is used as the prescribed flow direction.

Load Curve Card. See Remark 1.

Card 2	1	2	3	4	5	6	7	8
Variable	LC_Z1	LC_U	LC_V	LC_W	LC_D1	LC_D2	LC_P	LC_T
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
LC_Z1	Load curve or defined function ID to describe the volume fraction of material 1 as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The volume fraction is a constant with value SF_Z1. EQ.-1: The volume fraction is computed by the solver.
LC_U	Load curve or defined function ID to describe the x-component of the velocity as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The x-component of velocity is a constant with value SF_U. EQ.-1: The x-component of velocity is computed by the solver.
LC_V	Load curve or defined function ID to describe the y-component of the velocity as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The y-component of velocity is a constant with value SF_V. EQ.-1: The y-component of velocity is computed by the solver.

VARIABLE	DESCRIPTION
LC_W	Load curve or defined function ID to describe the z-component of the velocity as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The z-component of velocity is a constant with value SF_W. EQ.-1: The z-component of velocity is computed by the solver.
LC_D1	Load curve or defined function ID to describe the density of material 1 as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The density is a constant with value SF_D1. EQ.-1: The density is computed by the solver.
LC_D2	Load curve or defined function ID to describe the density of material 2 as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The density is a constant with value SF_D2. EQ.-1: The density is computed by the solver.
LC_P	Load curve or defined function ID to describe the pressure as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The pressure is a constant with value SF_P. EQ.-1: The pressure is computed by the solver.
LC_T	Load curve or defined function ID to describe the temperature as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The temperature is a constant with value SF_T. EQ.-1: The temperature is computed by the solver.

Scale Factor Card.

Card 3	1	2	3	4	5	6	7	8
Variable	SF_Z1	SF_U	SF_V	SF_W	SF_D1	SF_D2	SF_P	SF_T
Type	F	F	F	F	F	F	F	F
Default	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

VARIABLE	DESCRIPTION
SF_Z	Scale factor for LC_Z1
SF_U	Scale factor for LC_U
SF_V	Scale factor for LC_V
SF_W	Scale factor for LC_W
SF_D1	Scale factor for LC_D1
SF_D2	Scale factor for LC_D2
SF_P	Scale factor for LC_P
SF_T	Scale factor for LC_T

Remark:

1. **Consistent boundary values.** On each centroid or set of element centroids, the variables ($v_x, v_y, v_z, \rho, P, T, \ldots$) that are given values must be consistent and make the model well-posed (meaning that the solution of the model exists, is unique, and is physical).

DUALCESE_BOUNDARY_PRESCRIBED_VN_*OPTION

Available options include:

MSURF

SEGMENT_SET

Purpose: For the dual CESE compressible flow solver, set boundary values for the normal velocity, density, pressure, and temperature. Boundary values are applied at the centroid of elements connected with this boundary. This boundary condition differs from *DUALCESE_BOUNDARY_PRESCRIBED in that the normal velocity is prescribed instead of each velocity component.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Sets:

A set of data cards for this keyword consists of 3 of the following cards:

1. Card 1 specifies the object to which the boundary condition is applied. Its format depends on the keyword option.
2. Card 2 reads in load curve IDs.
3. Card 3 reads in scale factors.

For each boundary condition to be specified, include one set of cards. This input ends at the next keyword ("*") card.

Card Summary:

Card 1a. Include this card if the MSURF keyword option is used.

MSPID	IDCOMP						

Card 1b. Include this card if the SEGMENT_SET keyword option is used.

SSID	IDCOMP						

Card 2. This card is required.

LC_VN			LC_RHO	LC_P	LC_T		

Card 3. This card is required.

SF_VN			SF_RHO	SF_P	SF_T		

Data Card Definitions:

Surface Part Card. Card 1 format used when the MSURF keyword option is active.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID	IDCOMP						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT
IDCOMP	For inflow boundaries in problems involving chemical reacting flows, the chemical mixture of the fluid entering the domain is defined with a *CHEMISTRY_COMPOSITION card with this ID [Not yet available].

Set Card. Card 1 format used when the SEGMENT_SET keyword option is active.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID	IDCOMP						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH

VARIABLE	DESCRIPTION
IDCOMP	For inflow boundaries in problems involving chemical reacting flows, the chemical mixture of the fluid entering the domain is defined with a *CHEMISTRY_COMPOSITION card with this ID [Not yet available].

Load Curve Card. See Remark 1.

Card 2	1	2	3	4	5	6	7	8
Variable	LC_VN			LC_RHO	LC_P	LC_T		
Type	I			I	I	I		

VARIABLE	DESCRIPTION
LC_VN	Load curve or function ID to describe the normal velocity as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The normal velocity is a constant with value SF_VN. EQ.-1: The normal velocity is computed by the solver.
LC_RHO	Load curve ID to describe the density as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The density is a constant with value SF_RHO. EQ.-1: The density is computed by the solver.
LC_P	Load curve ID to describe the pressure as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The pressure is a constant with value SF_P. EQ.-1: The pressure is computed by the solver.
LC_T	Load curve ID to describe the temperature as a function of time or a function of position, velocity, temperature, pressure, and time, f(x, y, z, vx, vy, vz, temp, pres, time), respectively. EQ.0: The temperature is a constant with value SF_T. EQ.-1: The temperature is computed by the solver.

Scale Factor Card.

Card 3	1	2	3	4	5	6	7	8
Variable	SF_VN			SF_RHO	SF_P	SF_T		
Type	F			F	F	F		
Default	1.0			1.0	1.0	1.0		

VARIABLE	DESCRIPTION
SF_VN	Scale factor for LC_VN
SF_RHO	Scale factor for LC_RHO
SF_P	Scale factor for LC_P
SF_T	Scale factor for LC_T

Remarks:

1. **Consistent boundary values.** On each centroid or set of centroids, the variables (V_N, ρ, P, T) that are given values must be consistent and make the model well-posed (meaning be such that the solution of the model exists, is unique, and is physical).

DUALCESE_BOUNDARY_REFLECTIVE_*OPTION

Available options are:

MSURF

SEGMENT_SET

Purpose: Define a reflective boundary condition for the dual CESE compressible flow solver. This boundary condition can be applied to a symmetrical surface or a solid wall of the computational domain.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Summary:

Card 1a. Include this card for the MSURF keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

MSPID							

Card 1b. Include this card for the SEGMENT_SET keyword option. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

SSID							

Data Card Definitions:

Surface Part Card. Card 1 format used when the MSURF keyword option is active. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT

Set Card. Card 1 format used when the SEGMENT_SET keyword option is active. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH

Remarks:

This boundary condition has the same effect as a solid wall boundary condition for inviscid flows.

DUALCESE_BOUNDARY_SLIDING_*OPTION

Available options are:

MSURF

SEGMENT_SET

Purpose: Define a fluid surface that translates in the main direction of mesh movement (in the moving mesh dual CESE solvers). This is useful in piston-type applications.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh.

Card Summary:

Card 1a. Include this card for the MSURF keyword option. Provide as many cards as necessary. This input card ends at the next keyword ("*") card.

MSPID							

Card 1b. Include this card for the SEGMENT_SET keyword option. Provide as many cards as necessary. This input card ends at the next keyword ("*") card.

SSID							

Data Card Definitions:

Surface Part Card. Card 1a format used when the MSURF keyword option is active. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT

Set Card. Card 1b format used when the SEGMENT_SET keyword option is active. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1b	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH

DUALCESE_BOUNDARY_SOLID_WALL_*OPTION1_{OPTION2}

For *OPTION1* the choices are:

MSURF

SEGMENT_SET

For *OPTION2* the choices are:

<BLANK>

ROTATE

Purpose: Define a solid wall boundary condition for the dual CESE compressible flow solver. This boundary condition can be applied at a solid boundary that is the physical boundary for the flow field. For inviscid flow, this will be a slip boundary condition, while for viscous flows, it is a no-slip boundary condition.

The MSURF option should be used when the dual CESE mesh has been created using *MESH cards. The SEGMENT_SET option should be used when *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCKMESH specifies the dual CESE mesh. The ROTATE keyword option allows the boundary condition to rotate around an axis with a variable speed given by a load curve.

Card Summary:

Card Sets. The following sequence of cards comprises a *single set*. LS-DYNA will continue reading data card sets until the next keyword ("*") card is encountered.

Card 1a. Include this card for the MSURF keyword option *without* the ROTATE keyword option.

MSPID	LCID	VX	VY	VZ			

Card 1b. Include this card for the MSURF keyword option *with* the ROTATE keyword option.

MSPID	LCID	XP	YP	ZP	NX	NY	NZ

Card 1c. Include this card for the SEGMENT_SET keyword option *without* the ROTATE keyword option.

SSID	LCID	VX	VY	VZ			

Card 1d. Include this card for the SEGMENT_SET keyword option *with* the ROTATE keyword option.

SSID	LCID	XP	YP	ZP	NX	NY	NZ

Data Card Definitions:

Surface Part Card without Rotation. Card 1 format used when the MSURF keyword option is active *without* the ROTATE keyword option

Card 1a	1	2	3	4	5	6	7	8
Variable	MSPID	LCID	VX	VY	VZ			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			
Remarks		2	2	2	2			

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT cards
LCID	Load curve ID scales the velocity vector specified with (VX, VY, VZ) to give the solid wall boundary movement. If not defined, the solid wall boundary moves with a constant velocity vector specified by (VX, VY, VZ).
VX, VY, VZ	Velocity vector of the solid wall boundary condition: LCID.EQ.0: Constant velocity vector specified with VX, VY, and VZ. LCID.NE.0: VX, VY, and VZ give the velocity vector that is scaled by LCID.

Surface Part Card with Rotation. Card 1 format used when the MSURF keyword option is active *with* the ROTATE keyword option.

Card 1b	1	2	3	4	5	6	7	8
Variable	MSPID	LCID	XP	YP	ZP	NX	NY	NZ
Type	I	I	F	F	F	F	F	F
Default	none	0	0.0	0.0	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
MSPID	Mesh surface part ID that is referenced by *MESH_SURFACE_ELEMENT
LCID	Load curve ID for specifying the rotating speed frequency in Hz. This input is required.
XP, YP, ZP	Coordinates for a point on the axis of rotation
NX, NY, NZ	Unit vector for specifying the direction of the axis of rotation. This is not used for the 2D case.

Segment Set Card without Rotation. Card 1 format used when the SEGMENT_SET keyword option is active *without* the ROTATE keyword option

Card 1c	1	2	3	4	5	6	7	8
Variable	SSID	LCID	VX	VY	VZ			
Type	I	I	F	F	F			
Default	none	0	0.0	0.0	0.0			
Remarks		2	2	2	2			

VARIABLE	DESCRIPTION
SSID	ID of the segment set created with *DUALCESE_SEGMENTSET

VARIABLE	DESCRIPTION
LCID	Load curve ID scales the velocity vector specified with (VX, VY, VZ) to give the solid wall boundary movement. If not defined, the solid wall boundary moves with a constant velocity vector specified by (VX, VY, VZ).
VX, VY, VZ	Velocity vector of the solid wall boundary condition: LCID.EQ.0: Constant velocity vector specified with VX, VY, and VZ. LCID.NE.0: VX, VY, and VZ give the velocity vector that is scaled by LCID.

Segment Set Card with Rotation. Card 1 format used when the SEGMENT_SET keyword option is active *with* the ROTATE keyword option.

Card 1d	1	2	3	4	5	6	7	8
Variable	SSID	LCID	XP	YP	ZP	NX	NY	NZ
Type	I	I	F	F	F	F	F	F
Default	none	0	0.0	0.0	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
SSID	Segment set ID for the segment set created with *DUALCESE_SEGMENTSET or *DUALCESE_BLOCKMESH
LCID	Load curve ID for specifying the rotating speed frequency in Hz. This input is required.
XP, YP, ZP	Coordinates for a point on the axis of rotation
NX, NY, NZ	Unit vector for specifying the direction of the axis of rotation. This is not used for the 2D case.

Remarks:

1. **Boundary movement restrictions.** In this solid-wall condition, the boundary movement can only be in the tangential direction of the wall and should not affect the fluid domain size and mesh during the calculation. Otherwise, an FSI

or moving mesh solver should be used. Also, this moving boundary condition only affects viscous flows (no-slip boundary condition).

2. **Fixed solid wall boundary condition.** If LCID = 0 and Vx = Vy = Vz = 0.0 (default), this will be a regular solid wall boundary condition.

*DUALCESE_CHEMISTRY_D3PLOT

Purpose: Cause mass fractions of the listed chemical species to be added to the dual CESE d3plot output. This is only used when chemically-reacting flow is being solved with the dual CESE solver (see *DUALCESE_SOLVER_CHEMICALLY_REACTING).

Card 1	1	2	3	4	5	6	7	8
Variable	MODELID							
Type	I							
Default	none							

Species Cards. Include one card for each species to be included in the d3plot database. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	SPECIES							
Type	A							

VARIABLE	DESCRIPTION
MODELID	Identifier of a Chemkin-compatible chemistry model (see *CHEMISTRY_MODEL)
SPECIES	Name of a chemical species that is defined in the chemistry model identified by MODELID (see *CHEMISTRY_MODEL)

*DUALCESE_CONTROL_LIMITER

Purpose: Set some stability parameters used by the dual CESE compressible flow solver on the current dual CESE model.

Card 1	1	2	3	4	5	6	7	8
Variable		ALPHA	BETA	EPSR				
Type		F	F	F				
Default		0.0	0.0	0.0				
Remarks		1, 4	2, 4	3, 4				

VARIABLE	DESCRIPTION
ALPHA	Material viscosity/damping control coefficient, α. It must be ≥ 0.
BETA	Numerical viscosity control coefficient, β. Note that $0 \leq \beta \leq 1$.
EPSR	Stability control coefficient, ε. It must be ≥ 0.

Remarks:

1. **Material viscosity/damping control coefficient.** α controls the material viscosity/damping added for stability. Larger values of α give more stability, but less accuracy. Usually $\alpha = 1.0$ or 2.0 is enough for normal shock problems.

2. **Numerical viscosity control coefficient.** BETA is used for accuracy. For instance, when the flow field is very smooth, choosing it so that $0.0 < \beta < 1.0$ gives a more accurate result. Larger values of β provide more stability. For problems with shock waves, $\beta = 1.0$ is recommended.

3. **Stability control coefficient.** EPSR is a pseudo-relaxation parameter and is only needed in some special cases. For example, a positive value for EPSR should only be used when a simulation is still unstable, even when a large ALPHA is used. In other words, EPSR = 0.0 means that there will be no relaxation for stability. Larger values of ε give more stability, but less accuracy.

4. **Parameter indepedence.** Note that ALPHA, BETA, and EPSR are all independent of one another.

*DUALCESE_CONTROL_MESH_MOV

Purpose: Specify the algorithm for calculating the mesh movement (morphing) of a given DUALCESE part in an FSI problem. This keyword is for the moving mesh version of dual CESE.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	IALG	NINTER	RELERR	MXDISPR			
Type	I	I	I	F	F			
Default	none	9	100	10^{-3}	10^{-2}			

VARIABLE	DESCRIPTION
ID	ID for this mesh motion algorithm
IALG	Mesh motion algorithm: EQ.9: IDW scheme (default)
NITER	Number of linear solver iterations (when using a linear solver specified in IALG). No linear solvers have been implemented at this time, so this field is ignored.
RELERR	Relative error for determining convergence when using a linear solver specified in IALG. No linear solvers have been implemented at this time, so this field is ignored.
MXDISPR	Maximum displacement relative to element size to use as a criterion for avoiding the full calculation of the motion of the DUALCESE part on a given time step. If the full calculation can be avoided, the elements touching an FSI interface are still morphed, but it is assumed that this approximation will not lead to elements that are overly distorted.

*DUALCESE_CONTROL_SOLVER

Purpose: Set general purpose control variables for the dual CESE compressible flow solver.

> **WARNING:** As of version R14, this keyword is deprecated. Instead, in order to select the appropriate equations to solve, you should use *DUALCESE_SOLVER_SELECTION along with the corresponding *DUALCESE_-SOLVER_... card. Also, instead of this card, use *DU-ALCESE_MESH_GEOMETRY for the geometry-related details.

Card 1	1	2	3	4	5	6	7	8
Variable	EQNS	IGEOM	IFRAME	MIXTYPE	IDC	ISNAN		
Type	A	A	A	A	F	I		
Default	EULER	none	FIXED	optional	0.25	0		
Remarks		1			2			

VARIABLE	DESCRIPTION
EQNS	Select the equations being solved with the dual CESE solver: EQ.NS: Navier-Stokes equations EQ.EULER: Euler equations
IGEOM	Sets the geometric dimension: EQ.2D: Two-dimensional (2D) problem EQ.3D: Three-dimensional (3D) problem EQ.AXI: 2D axisymmetric
IFRAME	Choose the frame of reference: EQ.FIXED: Usual non-moving reference frame (default). EQ.ROT: Non-inertial rotating reference frame. IFRAME = ROTATING may also be used.

VARIABLE	DESCRIPTION
MIXTYPE	Select the mix or multiphase model solver (if any): EQ.<blank>: No mix or multiphase model (default) EQ.HYBRID: Hybrid multiphase model solver EQ.TWO-PHASE: Two-phase multiphase solver
IDC	Contact interaction detection coefficient (for FSI problems)
ISNAN	Flag to check for NaN in the dual CESE solver solution arrays at the completion of each time step. This option can be useful for debugging purposes. There is a cost overhead when this option is active. EQ.0: No checking. EQ.1: Checking is active.

Remarks:

1. **Mesh and Boundary Conditions for 2D Problems.** If you want to use the 2D (IGEOM = 2D) or 2D axisymmetric (IGEOM=AXI) solver, the mesh should only be distributed in the *xy*-plane with the boundary conditions given only at the *xy* domain boundaries. Otherwise, a warning message will be given, and the 3D solver will be triggered instead.

   The 2D axisymmetric case will work only if the 2D mesh and corresponding boundary conditions are properly defined with the *x* and *y* coordinates corresponding to the radial and axial directions, respectively.

2. **Contact Interaction Detection Coefficient.** IDC is the same type of variable that is input on the *ICFD_CONTROL_FSI card. For an explanation, see Remark 1 for the *ICFD_CONTROL_FSI card.

3. **Dual CESE Solver and Restarts.** The dual CESE solver will support restarts beginning with release R17.

*DUALCESE_CONTROL_TIMESTEP

Purpose: Set the time step control parameters for the dual CESE compressible flow solver.

Card 1	1	2	3	4	5	6	7	8
Variable	IDDT	CFL	DTINT	DTMIN	DTMAX			
Type	I	F	F	F	F			
Default	0	0.7 or 1.0	1.E-12	1.E-12	1.E10			

VARIABLE	DESCRIPTION
IDDT	Time step option: EQ.0: Fixed time step size of DTINT, the given initial time step size NE.0: The time step size will be calculated based on the given CFL number and the flow solution at the previous time step.
CFL	CFL (Courant–Friedrichs–Lewy) number when IDDT ≠ 0 or a scaling factor on the time step size when IDDT = 0. For IDDT ≠ 0, 0.0 < CFL ≤ 1.0, and the default is 0.7 and. For IDDT = 0, the default value is 1.0. For either IDDT selection, if a negative value is entered, then |CFL| refers to a load curve ID. This curve gives the CFL number or scaling factor as a function of time.
DTINT	Time step value if IDDT = 0. The actual time step used by the solver is scaled by the CFL field.
DTMIN	Minimum time step value
DTMAX	Maximum time step value

*DUALCESE_DARCY-FORCHHEIMER_EQ

Purpose: Provide the parameters of the Darcy-Forchheimer equation. This equation is also called the Forchheimer correction of Darcy's Law. This equation models the porous medium for compressible porous media flows, such as supersonic spacecraft parachute flows.

Card 1	1	2	3	4	5	6	7	8
Variable	EQID	KP	BETA	LC_KP	LC_BETA			
Type	I	F	F	I	I			
Default	none	none	none	0	0			

VARIABLE	DESCRIPTION
EQID	Equation ID of the porous medium model (see *DUALCESE_POROUS_SPECIFY_JUMP)
KP	Permeability of the porous medium, k_p. See Remark 1.
BETA	Inertial resistance coefficient of the porous medium, β. See Remark 1.
LC_KP	Load curve to modify the permeability (KP) of the porous medium as a function of time. If used, KP is a scale factor for the curve.
LC_BETA	Load curve to modify the inertial resistance coefficient (BETA) as a function of time. If used, BETA is a scale factor for the curve.

Remarks:

1. **Darcy-Forchheimer equation.** The Darcy-Forchheimer equation is given by:

$$\frac{\partial P}{\partial n} = \frac{\mu_f}{k_p} v_n + \beta \rho_f v_n^2$$

where

$\frac{\partial P}{\partial n}$	Pressure gradient (in the normal direction) across the porous medium
ρ_f	Fluid density

μ_f Fluid dynamic viscosity

v_n Wall normal velocity of the flow through the porous medium

k_p Permeablity of the porous media

β Inertial resistance coefficient

2. **Restrictions for the structure.** The porous media model can only be used in FSI problems with thin shell structures (such as fabric). It is intended for modeling supersonic spacecraft parachutes and airbags.

3. **Porous media model.** Currently, only a jump condition method is available for modeling porous media flow.

*DUALCESE_DATABASE_HISTORY_DRAG_LIFT

Purpose: Enable output of drag and lift-related data for all pieces of structural parts participating in FSI with the dual CESE immersed boundary method (IBM). This data output goes to binary database binout. At most, one instance of this keyword may appear in the input deck.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	DT	LCUR	IOOPT	REFDENS	REFVEL	REFPRESS	REFAREA	REFLENG
Type	F	I	I	F	F	F	F	F
Default	0.0	0	0	none	none	none	none	none

Directions Card.

Card 2	1	2	3	4	5	6	7	8
Variable	DRAGDIRX	DRAGDIRY	DRAGDIRZ	LIFTDIRX	LIFTDIRY	LIFTDIRZ		
Type	F	F	F	F	F	F		
Default	1.0	0.0	0.0	0.0	1.0	0.0		

Moment Options Card.

Card 3	1	2	3	4	5	6	7	8
Variable	DRGAREA	MOMNTD	MOMNTL	MOMNTC				
Type	I	I	I	I				
Default	0	0	0	0				

VARIABLE	DESCRIPTION
DT	Time interval between outputs. If DT is zero, no output is generated. DT is ignored if LCUR > 0.
LCUR	Optional curve ID specifying the time interval between outputs. Use *DEFINE_CURVE to define the curve. The abscissa is time, and the ordinate is the time interval between outputs.
IOOPT	Flag to govern the behavior of the output frequency load curve defined by LCUR: EQ.1: When output is generated at time t_n, the next output time t_{n+1} is computed as $t_{n+1} = t_n + \mathrm{LCUR}(t_n)$. This is the default behavior. EQ.2: When output is generated at time t_n, the next output time t_{n+1} is computed as $t_{n+1} = t_n + \mathrm{LCUR}(t_{n+1})$. EQ.3: Output is generated for each abscissa point in the load curve definition. The actual value of the load curve is ignored.
REFDENS	An appropriate reference density
REFVEL	An appropriate reference velocity
REFPRESS	An appropriate reference pressure
REFAREA	An appropriate reference area
REFLENG	An appropriate reference length
DRAGDIRX	x-component of the drag direction
DRAGDIRY	y-component of the drag direction
DRAGDIRZ	z-component of the drag direction
LIFTDIRX	x-component of the lift direction
LIFTDIRY	y-component of the lift direction
LIFTDIRZ	z-component of the lift direction

VARIABLE	DESCRIPTION
DRGAREA	Flag specifying how to obtain the drag area needed for finding the drag-related coefficients: EQ.0: Use REFAREA in all the calculations requiring an enclosing surface area NE.0: Use LS-DYNA's dynamic calculation of the surface area of each piece/part to compute the various drag-related coefficients. REFAREA scales the computed surface area before being used to calculate the drag-related coefficients.
MOMNTD	Flag determining whether to compute the torque in the drag direction: EQ.0: Do not compute the torque. NE.0: Compute the torque.
MOMNTL	Flag determining whether to compute the torque in the lift direction: EQ.0: Do not compute the torque. NE.0: Compute the torque.
MOMNTC	Flag determining whether to compute the torque in the direction that is the cross product of the drag and lift directions: EQ.0: Do not compute the torque. NE.0: Compute the torque.

*DUALCESE_DATABASE_HISTORY_ELEMENT_SET

Purpose: Enable output of dual CESE solver data at the element centroids of the selected dual CESE mesh elements. This element output goes to binary database binout. At most, one instance of this keyword may appear in the input deck.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	ESID	DT	LCUR	IOOPT				
Type	I	F	I	I				
Default	0	0.	0	0				

VARIABLE	DESCRIPTION
ESID	ID of a dual CESE element set (see *DUALCESE_ELEMENTSET)
DT	Time interval between outputs. If DT is zero, no output is generated.
LCUR	Optional curve ID specifying the time interval between outputs. Use *DEFINE_CURVE to define the curve. The abscissa is time, and the ordinate is time interval between outputs.
IOOPT	Flag to govern behavior of the output frequency load curve defined by LCUR: EQ.1: When output is generated at time t_n, the next output time t_{n+1} is computed as $t_{n+1} = t_n + \mathrm{LCUR}(t_n)$. This is the default behavior. EQ.2: When output is generated at time t_n, the next output time t_{n+1} is computed as $t_{n+1} = t_n + \mathrm{LCUR}(t_{n+1})$. EQ.3: Output is generated for each abscissa point in the load curve definition. The actual value of the load curve is ignored.

*DUALCESE_DATABASE_HISTORY_GLOBALS

Purpose: Enable output of global dual CESE solver data. The output goes to binary database binout. At most, one instance of this keyword may appear in the input deck.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	DT	LCUR	IOOPT					
Type	F	I	I					
Default	0.0	0	0					

VARIABLE	DESCRIPTION
DT	Time interval between outputs. If DT is zero, no output is generated.
LCUR	Optional curve ID specifying the time interval between outputs. Use *DEFINE_CURVE to define the curve; the abscissa is time, and the ordinate is time interval between outputs.
IOOPT	Flag to govern behavior of the output frequency load curve defined by LCUR: EQ.1: When output is generated at time t_n, the next output time t_{n+1} is computed as $t_{n+1} = t_n + \mathrm{LCUR}(t_n)$. This is the default behavior. EQ.2: When output is generated at time t_n, the next output time t_{n+1} is computed as $t_{n+1} = t_n + \mathrm{LCUR}(t_{n+1})$. EQ.3: Output is generated for each abscissa point in the load curve definition. The actual value of the load curve is ignored.

*DUALCESE_DATABASE_HISTORY_NODE_SET

Purpose: Enable output of dual CESE solver data at the selected dual CESE mesh nodes. This nodal output goes to binary database binout. At most, one instance of this keyword may appear in the input deck.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID	DT	LCUR	IOOPT				
Type	I	F	I	I				
Default	0	0.	0	0.				

VARIABLE	DESCRIPTION
NSID	ID of a dual CESE node set (see *DUALCESE_NODESET)
DT	Time interval between outputs. If DT is zero, no output is generated.
LCUR	Optional curve ID specifying the time interval between outputs. Use *DEFINE_CURVE to define the curve; the abscissa is time, and the ordinate is time interval between outputs.
IOOPT	Flag to govern behavior of the output frequency load curve defined by LCUR:

EQ.1: When output is generated at time t_n, the next output time t_{n+1} is computed as

$$t_{n+1} = t_n + \mathrm{LCUR}(t_n) \, .$$

This is the default behavior.

EQ.2: When output is generated at time t_n, the next output time t_{n+1} is computed as

$$t_{n+1} = t_n + \mathrm{LCUR}(t_{n+1}) \, .$$

EQ.3: Output is generated for each abscissa point in the load curve definition. The actual value of the load curve is ignored.

*DUALCESE_DATABASE_HISTORY_POINT_SET

Purpose: Enable output of dual CESE solver data at the selected points inside the dual CESE mesh. This point-based output goes to binary database binout. At most, one instance of this keyword may appear in the input deck.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	PNTSID	DT	LCUR	IOOPT				
Type	I	F	I	I				
Default	0	0.	0	0.				

VARIABLE	DESCRIPTION
PNTSID	ID of a dual CESE point set (see *DUALCESE_POINTSET)
DT	Time interval between outputs. If DT is zero, no output is generated.
LCUR	Optional curve ID specifying the time interval between outputs. Use *DEFINE_CURVE to define the curve; the abscissa is time, and the ordinate is time interval between outputs.
IOOPT	Flag to govern behavior of the output frequency load curve defined by LCUR:

EQ.1: When output is generated at time t_n, the next output time t_{n+1} is computed as

$$t_{n+1} = t_n + \mathrm{LCUR}(t_n) \, .$$

This is the default behavior.

EQ.2: When output is generated at time t_n, the next output time t_{n+1} is computed as

$$t_{n+1} = t_n + \mathrm{LCUR}(t_{n+1}) \, .$$

EQ.3: Output is generated for each abscissa point in the load curve definition. The actual value of the load curve is ignored.

*DUALCESE_DATABASE_HISTORY_SEGMENT_SET

Purpose: Enable output of dual CESE solver data averaged on each of the specified dual CESE segment sets. This segment set-based output goes to binary database binout. At most, one instance of this keyword may appear in the input deck.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	DT	LCUR	IOOPT					
Type	F	I	I					
Default	0.0	0	0					

Segment Sets Card. Define as many cards as necessary. Input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	SSID1	SSID2	SSID3	SSID4	SSID5	SSID6	SSID7	SSID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
DT	Time interval between outputs. If DT is zero, no output is generated.
LCUR	Optional curve ID specifying the time interval between outputs. Use *DEFINE_CURVE to define the curve; the abscissa is time, and the ordinate is time interval between outputs.
IOOPT	Flag to govern behavior of the output frequency load curve defined by LCUR:

VARIABLE	DESCRIPTION
	EQ.1: When output is generated at time t_n, the next output time t_{n+1} is computed as $$t_{n+1} = t_n + \mathrm{LCUR}(t_n) .$$ This is the default behavior.
	EQ.2: When output is generated at time t_n, the next output time t_{n+1} is computed as $$t_{n+1} = t_n + \mathrm{LCUR}(t_{n+1}) .$$
	EQ.3: Output is generated for each abscissa point in the load curve definition. The actual value of the load curve is ignored.
SSID*i*	i^{th} dual CESE segment set ID (see *DUALCESE_SEGMENTSET). For each of these segment sets, an average value of each dual CESE output variable is output to the binout file at the times selected by the fields in Card 1.

*DUALCESE_DEM_COUPLING

Purpose: Set the structural and thermal coupling options for a coupling between dual CESE and DEM particles.

Card 1	1	2	3	4	5	6	7	8
Variable	IMODE		DRAG	HCONV				
Type	I		F	F				
Default	0		0.0	0.0				

Card 2	1	2	3	4	5	6	7	8
Variable	IDRGM	EPSMIN		MUF				
Type	I	F		F				
Default	0	0.4		0.0				

This card is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	IPEN	EPSBLK	ALPHA					
Type	I	F	F					
Default	0	0.05	0.5					

This card is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	IRES	RATIO	GAMMA					
Type	I	F	F					
Default	0	3.0	0.5					

VARIABLE	DESCRIPTION
IMODE	DEM coupling mode for drag force exchange (see Remark 1): EQ.0: Off EQ.1: Two-way coupling between the DEM particles and the fluid EQ.2: Fluid to DEM particles only EQ.3: DEM particles to fluid only
DRAG	Constant drag coefficient, C_d, that is used when IDRGM = 0
HCONV	Convective heat transfer coefficient between the fluid and DEM particles (see Remark 3)
IDRGM	Drag model selection (see Remark 2): EQ.0: Constant drag coefficient defined by DRAG EQ.1: Gidaspow model EQ.2: Wen-Yu model EQ.3: Ergun model
EPSMIN	Minimum void fraction clamp, ε_{min}
MUF	Dynamic viscosity, μ_f, for various IDRGM models. If it is left as 0.0 and the Navier-Stokes solver is used for dual CESE, the value from the Navier-Stokes material will be used, provided that it is defined. For all other dual CESE solvers, MUF must be defined if the selected IDRGM requires it.
IPEN	Additional penalty treatment for a cluster of particles (see Remark 4):

VARIABLE	DESCRIPTION
	EQ.0: Off EQ.1: Two-way coupling between the DEM particles and the fluid EQ.2: Fluid to DEM particles only EQ.3: DEM particles to fluid only
EPSBLK	Void fraction threshold for particle cluster identification and treatment
ALPHA	Penalty relaxation factor, α
IRES	Resolved single particle treatment (see Remark 5): EQ.0: Off EQ.1: Two-way coupling between the DEM particles and the fluid EQ.2: Fluid to DEM particles only EQ.3: DEM particles to fluid only
RATIO	Particle-size-to-cell-size ratio threshold
GAMMA	Relaxation factor on single particle treatment

Remarks:

1. **DEM coupling terms.** Consider the complete fluid system consisting of the continuity equation (one degree of freedom), the momentum equation (three degrees of freedom for the three velocity components), and the energy equation (one degree of freedom). The source term vector for DEM coupling can then be expressed as:

$$S_{\text{drag}} = \frac{1}{\varepsilon_{\text{f}} V_{\text{cell}}} \begin{pmatrix} 0. \\ -F_{p,\text{drag}_x} \\ -F_{p,\text{drag}_y} \\ -F_{p,\text{drag}_z} \\ -\left(\mathbf{F}_{p,\text{drag}} \cdot \mathbf{v}_p + Q_p\right) \end{pmatrix}$$

where $\mathbf{F}_{p,\text{drag}}$ is the drag force (see Remark 2) between the cell and the particle (in the direction of relative velocity between the particle and the fluid), $\mathbf{v}_p$ is the particle velocity, Q_p is the convective heat flux between particle and fluid (see

Remark 3), ε_f is the local void fraction, and V_{cell} is the volume of the cell. The local void fraction is expressed as:

$$\varepsilon_f = \max(\varepsilon_{\min}, 1 - \frac{1}{V_{\text{cell}}} \sum_{p \text{ in cell}} \frac{4}{3}\pi r_p^3)$$

with r_p the particle radius of particle p.

Note that the source term is applied to the fluid and the drag force is applied to the DEM particles. Thus, the source term is only transferred to the fluid when IMODE = 1 or 3, and the drag force is only is applied to the DEM particles when IMODE = 1 or 2.

2. **Drag models.** IDRGM specifies the drag model. For IDRGM = 0, the drag force coefficient, C_d, is provided with input field DRAG, and the drag force is expressed as:

$$\mathbf{F}_{p,\text{drag}} = 0.5 C_d \rho_f |\mathbf{v}_{\text{rel}}| \mathbf{v}_{\text{rel}} \pi r_p^2 \ ,$$

where $\mathbf{v}_{\text{rel}}$ is the relative velocity between fluid and particle.

For IDRGM = 1, the drag force is expressed as:

$$\mathbf{F}_{p,\text{drag}} = \beta V_p \mathbf{v}_{\text{rel}}$$

with

$$\beta = \begin{cases} 150\dfrac{(1-\varepsilon_f)^2\mu_f}{\varepsilon_f(2r_p)^2} + 1.75\dfrac{(1-\varepsilon_f)\rho_f|\mathbf{v}_{\text{rel}}|}{2r_p} & \text{if } \varepsilon_f < 0.8 \\ \dfrac{3}{4}C_d\dfrac{(1-\varepsilon_f)\rho_f|\mathbf{v}_{\text{rel}}|}{2r_p} & \text{if } \varepsilon_f \geq 0.8 \end{cases}$$

and

$$C_d = \begin{cases} \dfrac{24}{\text{Re}_p}(1 + 0.15\text{Re}_p^{0.687}) & \text{if } \text{Re}_p < 1000. \\ 0.44 & \text{if } \text{Re}_p \geq 1000. \end{cases}$$

In the above, Re_p is the local particle-based Reynolds number:

$$\text{Re}_p = \frac{2r_p|\mathbf{v}_{\text{rel}}|\rho_f\varepsilon_f}{\mu_f} \ .$$

IDRGM = 2 is the same as IDRGM = 1, except

$$\beta = \frac{3}{4}C_d\frac{(1-\varepsilon_f)\rho_f|\mathbf{v}_{\text{rel}}|}{2r_p}$$

Similarly, IDRGM = 3 is the same as IDRGM = 1, except

$$\beta = 150\frac{(1-\varepsilon_{\mathrm{f}})^2\mu_{\mathrm{f}}}{\varepsilon_{\mathrm{f}}(2r_p)^2} + 1.75\frac{(1-\varepsilon_{\mathrm{f}})\rho_{\mathrm{f}}|\mathbf{v}_{\mathrm{rel}}|}{2r_p}$$

3. **Thermal coupling.** The heat term exchanged between fluid and particle is expressed as:

$$Q_p = h_{\mathrm{conv}}(T_{\mathrm{fluid}} - T_p)\pi r_p^2$$

and automatically exchanged when DEM thermal solver is active.

4. **Optional penalty treatment for particle clusters.** When many small particles densely pack a fluid cell, drag correlations are extrapolated beyond their validity. Drag correlations assume the fluid permeates the particle assembly (meaning particles are independent of each other) and that every cell has a nonzero fluid velocity. For nearly impermeable clusters, these assumptions are no longer valid. Visually, the consequences in the simulation results are that particles seem to have a "too weak" effect on the fluid. The fluid seems to flow through particle clusters rather than around it, meaning that deviation and blocking effects are underestimated. Turning on the IPEN option separates the particles into two groups. For each cell where the void fraction is lower than EPSBLK ($\varepsilon_{\mathrm{f}} < \varepsilon_{\mathrm{blk}}$), the particles in that cell will be flagged as "cluster" particles, and a different penalization term will be applied rather than the pure drag-based correlation of IMODE. Following the same five-component structure as in Remark 1, the penalization source vector for a cell with cluster particles is:

$$S_{\mathrm{pen}} = \frac{1}{\varepsilon_{\mathrm{f}} V_{\mathrm{cell}}}\begin{pmatrix} 0. \\ -F_{p,\mathrm{pen}_x} \\ -F_{p,\mathrm{pen}_y} \\ -F_{p,\mathrm{pen}_z} \\ -(\mathbf{F}_{p,\mathrm{pen}}\cdot\mathbf{v}_{\mathrm{s}} + Q_p) \end{pmatrix}$$

where $\mathbf{v}_{\mathrm{s}}$ is the average velocity of all particles that belong to the cluster falling in the fluid cell and $\mathbf{F}_{p,\mathrm{pen}}$ is the penalty force expressed as:

$$\mathbf{F}_{p,\mathrm{pen}} = \alpha\frac{\rho_{\mathrm{f}}\varepsilon_{\mathrm{f}}}{\mathrm{dt}} V_{\mathrm{cell}}(\mathbf{v}_{\mathrm{f}} - \mathbf{v}_{\mathrm{s}}) \ .$$

Here, dt is the fluid timestep and α is a relaxation factor (ALPHA). $\alpha = 1$ means the fluid and particle cluster velocities should adjust with a single step. This often leads to unstable results. We recommend setting α such that $0 \leq \alpha < 1$. Lower values yield smoother results while higher values are more aggressive (stronger penalty effect but risk of instability). The force term is distributed among the particles that are part of the cluster.

Note that the source term is applied to the fluid and the drag force is applied to the DEM particles. Thus, the source term is only transferred to the fluid when

IPEN = 1 or 3, and the drag force is only is applied to the DEM particles when IPEN = 1 or 2.

5. **Resolved single particle mode.** The general assumption for drag correlations and penalty treatments in general is that DEM particles are small compared to the fluid mesh size and that multiple particles can fall within a fluid cell. However, for the exceptional case study where the reverse would be true (multiple fluid cells fall within a single particle radius), nonzero IRES invokes a "full resolution" mode. If the diameter of the particle (d_p) divided by the characteristic cell dimension (h_c) becomes larger than RATIO, then this particle will be flagged as "resolved" and receive a separate treatment from the drag/penalty approaches. Instead the source term is now applied directly to the cells flagged as "covered by particle" (if IRES = 1 or 3):

$$S_{\text{res}} = \frac{\gamma \rho_f}{dt} \begin{pmatrix} 0. \\ v_{p_x} - v_{f_x} \\ v_{p_y} - v_{f_y} \\ v_{p_z} - v_{f_z} \\ 0.5\left(\mathbf{v}_p^2 - \mathbf{v}_f^2\right) - \dfrac{Q_p}{N_c V_c} \end{pmatrix}$$

Like the α parameter discussed in Remark 4, γ acts as a relaxation factor that controls the robustness and accuracy of the simulation with values smaller than 1 recommended. Each DEM particle flagged as "resolved" receives the integrated force from the fluid summed over all cells inside the sphere (if IRES = 1 or 2):

$$\mathbf{F}_{p,\text{res}} = \gamma \sum_{\substack{\text{covered} \\ \text{fluid cells}}} \rho_f \frac{\mathbf{v}_f - \mathbf{v}_p}{dt} V_{\text{cell}}$$

The thermal heat flux term that is exchanged between the cells and the particle is defined as:

$$Q_p = h_{\text{conv}} 4\pi r_p^2 \left(-T_p + \sum_{\substack{\text{covered} \\ \text{fluid cells}}} \frac{T_f}{N_c} \right)$$

*DUALCESE_D3PLOT

Purpose: Specify the flow variables to be added to the dual CESE d3plot output.

Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	FLOW_VAR							
Type	A							

VARIABLE	DESCRIPTION
FLOW_VAR	Name of a flow variable to output to the d3plot file. The currently supported variables are listed in the table below.

Flow Variables:

This table lists the supported flow variables.

FLOW_VAR	DESCRIPTION
DENSITY	Density
VELOCITY	Velocity
MOMENTUM	Momentum
VORTICITY	Vorticity
TOTAL_ENERGY	Total energy
INTERNAL_ENERGY	Internal energy
PRESSURE	Pressure
TEMPERATURE	Temperature
MACH_NUMBER	Flow Mach number
SCHLIEREN_NUMBER	Quantity for capturing or highlighting the shock structure in a compressible flow

FLOW_VAR	DESCRIPTION
VOID_FRACTION	Void fraction (only for the dual CESE cavitation solver)
VOLUME_FRACTION	Volume fraction of the different materials in a hybrid or two-phase multiphase model
REACTANT_MASS_FRACTION	Mass fraction of the reactant (material α) with respect to the explosive material (material 2) in a hybrid multiphase model
LIQUID_MASS_FRACTION	Mass fraction of the liquid component of the fluid in the phase change model
VAPOR_MASS_FRACTION	Mass fraction of the vapor component of the fluid in the phase change model
OTHER_GASES_MASS_FRACTION	Mass fraction of the other gases in the fluid in the phase change model

*DUALCESE_D3PLOT_FLUID_SSID

Purpose: Generate surface d3plot output for the dual CESE solver on a specified dual CESE mesh segment set. These surfaces may be on the outside of the dual CESE fluid mesh that is in contact with the structural volume element parts.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							

VARIABLE	DESCRIPTION
SSID	ID of a segment set created with *DUALCESE_SEGMENTSET

Dual CESE variables to output. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	FLOW_VAR							
Type	A							

VARIABLE	DESCRIPTION
FLOW_VAR	Name of a flow variable to output to the d3plot file. The currently supported variables are listed in the table below.

Flow Variables:

This table lists the supported flow variables.

FLOW_VAR	DESCRIPTION
DENSITY	Density
VELOCITY	Velocity
MOMENTUM	Momentum

FLOW_VAR	DESCRIPTION
VORTICITY	Vorticity
TOTAL_ENERGY	Total energy
INTERNAL_ENERGY	Internal energy
PRESSURE	Pressure
TEMPERATURE	Temperature
MACH_NUMBER	Flow Mach number
SCHLIEREN_NUMBER	Quantity for capturing or highlighting the shock structure in a compressible flow
VOID_FRACTION	Void fraction (only for the dual CESE cavitation solver)
VOLUME_FRACTION	Volume fraction of the different materials in a hybrid or two-phase multiphase model
REACTANT_MASS_FRACTION	Mass fraction of the reactant (material α) with respect to the explosive material (material 2) in a hybrid multiphase model

*DUALCESE_ELE2D

Purpose: Define three and four node elements.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	EID	PID	N1	N2	N3	N4				
Type	I	I	I	I	I	I				
Default	none	none	none	none	none	none				

VARIABLE	DESCRIPTION
EID	Element ID. Choose a unique number with respect to other elements.
PID	Part ID, see *DUALCESE_MESH_PART.
N1	Nodal point 1
N2	Nodal point 2
N3	Nodal point 3
N4	Nodal point 4

*DUALCESE_ELE3D

Purpose: Define three-dimensional fluid volume elements. These can be 4 node tetrahedra, 5 node pyramids, 6 node wedges (prisms), and 8 node hexahedra.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	EID	PID	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none	none	none
Remarks	1									

VARIABLE	DESCRIPTION
EID	Element ID. A unique number must be chosen.
PID	Part ID, see *DUALCESE_MESH_PART.
N1	Nodal point 1
N2	Nodal point 2
N3	Nodal point 3
⋮	⋮
N8	Nodal point 8

Remarks:

1. **Node numbering.** Four, five, six, and eight node elements are allowed as numbered below. This ordering must be followed, or code termination will occur during the initialization phase with a negative volume message. In the case of a pyramid element, the base of the pyramid must follow the ordering used for the hexahedron. See *ELEMENT_SOLID for a figure showing the positions of the nodes in 4, 6, and 8 node elements.

4-noded tetrahedron	N1, N2, N3, N4, N4, N4, N4, N4
5-noded pyramid	N1, N2, N3, N4, N5, N5, N5, N5

6-noded pentahedron	N1, N2, N3, N4, N5, N5, N6, N6
8-noded hexahedron	N1, N2, N3, N4, N5, N6, N7, N8

*DUALCESE_ELEMENTSET

Purpose: Define a set of dual CESE mesh elements.

Card 1	1	2	3	4	5	6	7	8
Variable	ESID							
Type	I							
Default	none							

Element ID Cards. List of elements in the set, where the element IDs are defined with *DUALCESE_ELE2D or *DUALCESE_ELE3D cards. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	EID1	EID2	EID3	EID4	EID5	EID6	EID7	EID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
ESID	Set ID. All dual CESE element sets should have a unique set ID.
EID*i*	Element ID *i*

*DUALCESE_ENSIGHT_DOMAIN

Purpose: Specify the variable data output to a file in the Ansys EnSight™ Gold format along with the domain from which it is output. This output can be on the full dual CESE mesh, a subset of the dual CESE mesh, the FSI interface, or a subset of the FSI interface. The subset can be selected through various dual CESE keywords or *DUALCESE_ENSIGHT_REGION.

To output a file in this format, at least one instance of *DUALCESE_ENSIGHT_TIMESEQ is required in the dual CESE model. Each instance of the *DUALCESE_ENSIGHT_TIMESEQ keyword creates a new time sequence with an independent output frequency and duration. *DUALCESE_ENSIGHT_TIMESEQ must reference one or more *DUALCESE_ENSIGHT_DOMAIN keywords.

Card 1	1	2	3	4	5	6	7	8
Variable	DOMAIN_TYPE							
Type	A							
Default	none							

Domain Details Card.

Card 2	1	2	3	4	5	6	7	8
Variable	DOMID	REFID	REDUCT	REGID				
Type	I	I	A	I				
Default	none	optional	<BLANK>	optional				

Variable Name Card. Provide as many cards as necessary. This input ends at the next keyword ("*") card

Card 3	1	2	3	4	5	6	7	8
Variable	VARIABLE_NAME							
Type	A							
Remarks	1							

VARIABLE	DESCRIPTION
DOMAIN_TYPE	The type of domain for which output may be generated. The currently supported types are: DUALCESE_FSI_FACES, DUALCESE_FSI_INTERFACE, ELEMENT, ELEMENT_SET, NODE, NODE_SET, POINT_SET, and SEGMENT_SET. See Table 5-1.
DOMID	Dual CESE domain ID associated with this domain that is used by *DUALCESE_ENSIGHT_TIMESEQ.
REFID	ID for referring to dual CESE element, node, or segment sets to obtain a subset of the dual CESE mesh. The type of set referenced by REFID must match the DOMAIN_TYPE. For instance, for DOMAIN_TYPE = NODE or NODE_SET, this field refers to a *DUALCESE_NODESET. This field is not required and is not used by all values of DOMAIN_TYPE. See Table 5-1 for details.
REGID	ID of a dual CESE region specified with *DUALCESE_ENSIGHT_REGION. This region can limit the specified domain. It is optional and is available for all values of DOMAIN_TYPE. See Table 5-1.
REDUCT	A function that operates on the entire domain and returns a single value for scalar variables, three values for vector variables, or 6 values for symmetric tensor variables. The following are the supported functions: EQ.<BLANK>: No reduction (default) EQ.none: Same as <BLANK> EQ.avg: The average by component

VARIABLE	DESCRIPTION	
	EQ.average:	Same as avg
	EQ.min:	The minimum by component
	EQ.minimum:	Same as min
	EQ.max:	The maximum by component
	EQ.maximum:	Same as max
	EQ.sum:	The sum by component
VARIABLE_NAME	Name of a single output variable. See Table 5-2.	

Table 5-1. The following table describes the available settings of DOMAIN_TYPE.

DOMAIN_TYPE	Description
DUALCESE_FSI_FACES	Output data from the faces of the mechanics parts in the dual CESE FSI interface. REFID is not used with this DOMAIN_TYPE. Data for all mechanics faces in the FSI interface are output unless REGID refers to a *DUALCESE_ENSIGHT_REGION that selects a region of the FSI interface. Note that the data for each face is obtained at the centroid.
DUALCESE_FSI_INTERFACE	Output data from the mechanics nodes in the dual CESE FSI interface. REFID is not used with this DOMAIN_TYPE. Data for all mechanics nodes in the FSI interface are output unless REGID refers to a *DUALCESE_ENSIGHT_REGION that selects a region of the FSI interface.
ELEMENT	Output data from dual CESE elements (see *DUALCESE_ELE2D and *DUALCESE_ELE3D). REFID is optional. If REFID is used, it must refer to a *DUALCESE_ELEMENTSET. If REFID and REGID are not specified, then the data is obtained using all the elements in the dual CESE mesh. If REFID is not specified but REGID is, the data is obtained using all the elements of the dual CESE mesh within the region. Note that the data for each element is obtained at the centroid.
NODE	Output data from dual CESE nodes (see *DUALCESE_NODE2D and *DUALCESE_NODE3D). REFID is optional. If REFID is used, it must refeer to a *DUALCESE_NODESET. If REFID and REGID are

DOMAIN_TYPE	Description
	not specified, then the data is obtained using all the nodes in the dual CESE mesh. If REFID is not specified but REGID is, the data is obtained using all the nodes of the dual CESE mesh within the region.
NODE_SET	Output data from a dual CESE node set (see *DUALCESE_NODESET). REFID is required.
SEGMENT_SET	Output data from a dual CESE segment set (see *DUALCESE_SEGMENTSET). REFID is required. Note that the data for each segment is obtained at the centroid.

Table 5-2. The following table lists the supported choices for VARIABLE_NAME, where some variables are only available for specific domain types.

VARIABLE_NAME	Description
DENSITY	Density. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.
VELOCITY	Velocity. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.
MOMENTUM	Momentum. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.
VORTICITY	Vorticity. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.
TOTAL_ENERGY	Total energy. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.
INTERNAL_ENERGY	Internal energy. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.
PRESSURE	Pressure. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, ELEMENT_SET, SEGMENT_SET, and POINT_SET.

VARIABLE_NAME	Description
TEMPERATURE	Temperature. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.
MACH_NUMBER	Flow Mach number. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.
SCHLIEREN_NUMBER	Quantity for capturing or highlighting the shock structure in a compressible flow. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.
VOID_FRACTION	Void fraction (only for the dual CESE cavitation solver). This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.
VOLUME_FRACTION	Volume fraction of the different materials in a hybrid or two-phase model. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.
REACTANT_MASS_FRACTION	Mass fraction of the reactant (material α) with respect to the explosive material (material 2) in a hybrid multiphase model. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.
LIQUID_MASS_FRACTION	Mass fraction of the liquid component of the fluid in the phase change model. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.
VAPOR_MASS_FRACTION	Mass fraction of the vapor component of the fluid in the phase change model. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.
OTHER_GASES_MASS_FRACTION	Mass fraction of the other gases in the fluid in the phase change model. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.
FSI_ACCELERATION	Net acceleration due to FSI effects on the mechanics FSI nodes. This variable is available for DOMAIN_TYPE DUALCESE_FSI_INTERFACE.

VARIABLE_NAME	Description
FSI_FACE_HEATFLUX	Heat flux due to heat flow on the mechanics FSI nodes that lie on a conjugate heat transfer interface. This variable is available for DOMAIN_TYPE DUALCESE_FSI_FACES.
FSI_FACE_PRESSURE	Fluid pressure on the mechanics FSI faces. This variable is available for DOMAIN_TYPE DUALCESE_FSI_FACES.
SPECIESMF:<species name >	Mass fraction of the 'species name' chosen from the list of species names in the Chemkin file defining the chemical reaction network (see: *CHEMISTRY_MODEL). This variable is only computed when the dual CESE solver is specified by *DUALCESE_SOLVER_CHEMICALLY_REACTING. This variable is available for DOMAIN_TYPES: NODE, NODE_SET, ELEMENT, and SEGMENT_SET.

*DUALCESE_ENSIGHT_REGION

Purpose: Specify a subset of the dual CESE mesh or FSI interface for which variable data is output to a file in Ansys EnSight Gold format. This region can change during the simulation. This keyword is referenced by *DUALCESE_ENSIGHT_DOMAIN.

Card Summary:

Card Sets. Card 1 is required and appears once in an instance of this keyword. Include as many sets of Cards 2 through 5 as needed to define the whole region. These cards can both add or subtract from the region. Card 5 is optional in the set and can be excluded if not needed. The next keyword ("*") card terminates this input.

Card 1. This card is required.

REGID							

Card 2. This card is required.

REGION_TYPE

Card 3a. Include this card if REGION_TYPE = BOX.

X1	Y1	Z1	X2	Y2	Z2		

Card 3b. Include this card if REGION_TYPE = SPHERE.

X0	Y0	Z0	RADIUS				

Card 3c. Include this card if REGION_TYPE = CYLINDER.

XC1	YC1	ZC1	XC2	YC2	ZC2	RADCYL	LENCYL

Card 3d.1. Include this card if REGION_TYPE = ELLIPSOID.

XC3	YC3	ZC3	AX3	AY3	AZ3	BX3	BY3

Card 3d.2. Include this card if REGION_TYPE = ELLIPSOID.

BZ3	RA3	RB3	RC3				

Card 4. This card is required.

SIDE	INCL						

Card 5. This card is optional.

DIST	LCID	ISETUSE					

Data Card Definitions:

Region ID Card. This card is required.

Card 1	1	2	3	4	5	6	7	8
Variable	REGID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
REGID	ID of the dual CESE region used for output

Region Type Card. Provide Card 2, one type 3 card if needed for the cases of analytic geometry, Card 4, and optionally Card 5 to define a region. Any number of this sequence of cards can be provided to form an overall region definition.

Card 2	1	2	3	4	5	6	7	8
Variable	REGION_TYPE							
Type	A							

VARIABLE	DESCRIPTION
REGION_TYPE	Type of region for selecting a portion of the dual CESE mesh. The available options are: BOX, SPHERE, CYLINDER, ELLIPSOID, DISTANCE_FROM_FSI_INTERFACE, and DISTANCE_FROM_CHT_INTERFACE. DISTANCE_FROM_FSI_INTERFACE is the region defined by the fluid-structure interaction (FSI) interface to a distance specified by DIST and LCID from the interface (see Figure 5-1). Similarly, DISTANCE_FROM_CHT_INTERFACE is the region defined by the conjugate heat transfer (CHT) interface to a distance specified by DIST and LCID from the interface.

Box Region Card. Include this card if REGION_TYPE = BOX.

Card 3a	1	2	3	4	5	6	7	8
Variable	X1	Y1	Z1	X2	Y2	Z2		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
X1, Y1, Z1	Position of one corner of the BOX geometry
X2, Y2, Z2	Position of the opposite corner from (X1, Y1, Z1) for the BOX geometry

Spherical Region Card. Include this card if REGION_TYPE = SPHERE.

Card 3b	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0	RADIUS				
Type	F	F	F	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
X0, Y0, Z0	Position of the center of the SPHERE geometry
RADIUS	Radius of the sphere in the SPHERE geometry

Cylindrical Region Card. Include this card if REGION_TYPE = CYLINDER.

Card 3c	1	2	3	4	5	6	7	8
Variable	XC1	YC1	ZC1	XC2	YC2	ZC2	RADCYL	LENCYL
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
XC1, YC1, ZC1	A point on the cylindrical axis
XC2, YC2, ZC2	A vector that defines the direction of the axis of the cylinder
RADCYL	Radius of the cylinder
LENCYL	Length of the cylinder

Ellipsoidal Region Card 1. Include this card if REGION_TYPE = ELLIPSOID.

Card 3d.1	1	2	3	4	5	6	7	8
Variable	XC3	YC3	ZC3	AX3	AY3	AZ3	BX3	BY3
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Ellipsoidal Region Card 2. Include this card if REGION_TYPE = ELLIPSOID.

Card 3d.2	1	2	3	4	5	6	7	8
Variable	BZ3	RA3	RB3	RC3				
Type	F	F	F	F				
Default	0.0	0.0	0.0	0.0				

VARIABLE	DESCRIPTION
XC3, YC3, ZC3	Coordinates of the center of the ellipsoid
AX3, AY3, AZ3	A vector in the direction of the first axis of the ellipsoid (axis **a**)
BX3, BY3, BZ3	A vector, $\tilde{\mathbf{b}}$, in the plane of the first and second axes of the ellipsoid. The third axis of the ellipsoid (axis **c**) will be in the direction of $\mathbf{a} \times \tilde{\mathbf{b}}$, and finally, the second axis $\mathbf{b} = \mathbf{c} \times \mathbf{a}$.
RA3, RB3, RC3	The semi-axis lengths of the ellipsoid

Region Details Card 1. This card affects which side of the geometry the region comes from. It is required.

Card 4	1	2	3	4	5	6	7	8
Variable	SIDE	INCL						
Type	A	A						
Default	INSIDE	INCLUDE						

VARIABLE	DESCRIPTION
SIDE	Flag to specify which side of the geometry the region comes from: EQ.INSIDE: Region comes from inside the geometry (inside). For DISTANCE_FROM_FSI_INTERFACE and DISTANCE_FROM_CHT_INTERFACE, inside is the region that includes the FSI/CHT interface out to a distance specified with DIST and LCID. See Figure 5-1. EQ.OUTSIDE: Region comes from outside the geometry.
INCL	Flag to include region in or exclude this region from the overall region: EQ.INCLUDE: Include this region in the overall region (default). EQ.EXCLUDE: Exclude this region from the overall region.

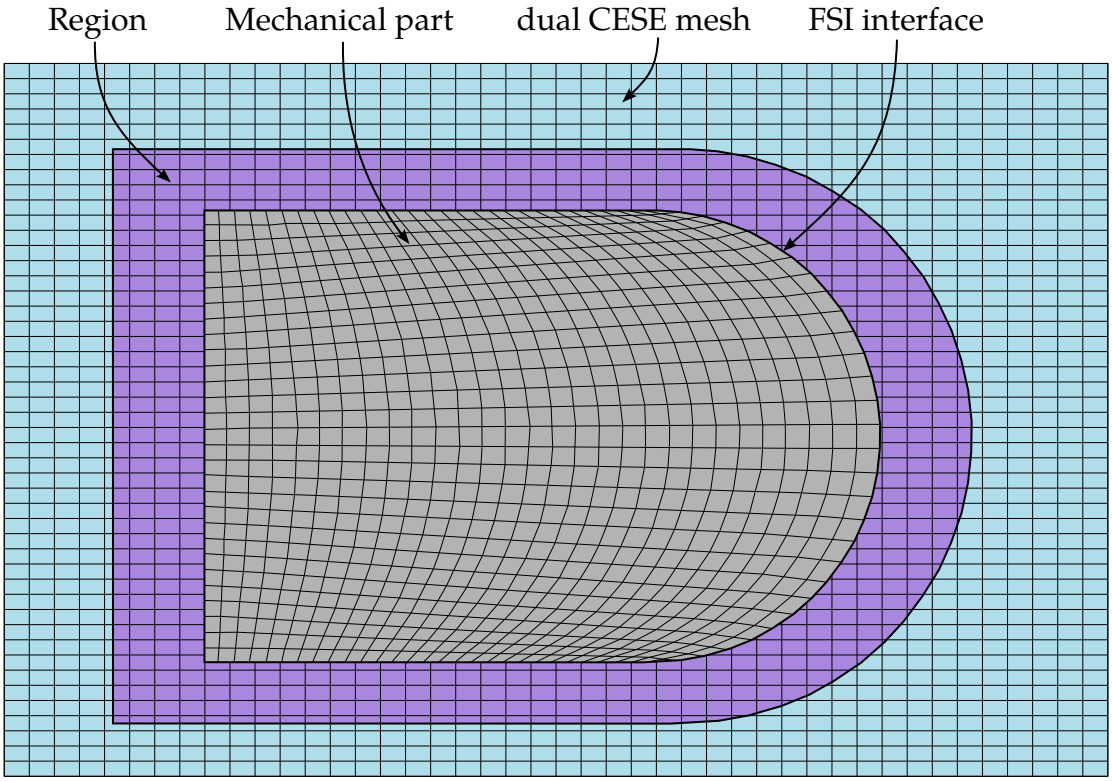


Figure 5-1. Example of REGION_TYPE set to DISTANCE_FROM_FSI_INTERFACE when SIDE = INSIDE and INCL = INCLUDE.

VARIABLE	DESCRIPTION
	SUBTRACT can also be used as an alternative name to EXCLUDE.

Region Details Card 2. This card is optional.

Card 5	1	2	3	4	5	6	7	8
Variable	DIST	LCID						
Type	F	I						
Default	0.0	0						

VARIABLE	DESCRIPTION
DIST	Distance measure from the selected mesh object. This field only applies to DISTANCE_FROM_FSI_INTERFACE and DISTANCE_FROM_CHT_INTERFACE.

VARIABLE	DESCRIPTION
LCID	Load curve that adjusts the distance from the selected mesh object as a function of time. This field only applies to DISTANCE_FROM_FSI_INTERFACE and DISTANCE_FROM_CHT_INTERFACE.

*DUALCESE_ENSIGHT_TIMESEQ

Purpose: Specify the frequency of data output by the dual CESE solvers for the specified model domains to a file in the Ansys Ensight Gold format. To output a file in this format, at least one instance of this keyword is required in the dual CESE model.

Each instance of the *DUALCESE_ENSIGHT_TIMESEQ keyword creates a new time sequence with an independent output frequency and duration. This keyword must reference one or more *DUALCESE_ENSIGHT_DOMAIN keywords. The *DUALCESE_ENSIGHT_DOMAIN specifies the variable data for output and the domain from which it is output.

Card 1	1	2	3	4	5	6	7	8
Variable	TSEQID	TSEQTYPE	DT	LCDT	LCOPT	NPLTC	TBEG	TEND
Type	I	A	F	I	I	I	F	F
Default	none	NODE	0.0	0	0	0	none	0.0
Remarks			1	1	1	1		

Domain IDs. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	DOMID1	DOMID2	DOMID3	DOMID4	DOMID5	DOMID6	DOMID7	DOMID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
TSEQID	ID of this time sequence
TSEQTYPE	Type of time sequence. These are the geometric objects on which the data will be output.
	EQ.ELEMENT: The data is output for the elements in the domains. The DOMAIN_TYPE (see *DUALCESE_ENSIGHT_DOMAIN) for the domains

VARIABLE	DESCRIPTION
	must be ELEMENT or ELEMENT_SET.
	EQ.FSI: The data is output for the FSI nodes in the domains. The DOMAIN_TYPE (see *DUALCESE_ENSIGHT_DOMAIN) for the domains must be DUALCESE_FSI_INTERFACE.
	EQ.FSI_FACE: The data is output for the FSI faces in the domains. The DOMAIN_TYPE (see *DUALCESE_ENSIGHT_DOMAIN) for the domains must be DUALCESE_FSI_FACES.
	EQ.NODE: The data is output for the nodes in the domains, and the DOMAIN_TYPE (see *DUALCESE_ENSIGHT_DOMAIN) for the domains must be NODE or NODE_SET. This is the default.
	EQ.TIMEHIST: The data for the variables for each domain is output as a single value (or three values for vectors) for each output time. This time sequence is available for every value of DOMAIN_TYPE (see *DUALCESE_ENSIGHT_DOMAIN), but REDUCT must not be blank.
DT	Time interval between outputs (ignored if LCDT or NPLTC is nonzero)
LCDT	Optional load curve ID specifying the time interval between dumps
LCOPT	Flag to govern the behavior of the plot frequency load curve, LCDT:
	EQ.1: At the time each plot is generated, the load curve value is added to the current time to determine the next plot time (this is the default behavior).
	EQ.2: At the time each plot is generated, the next plot time T is computed so that T equals the current time plus the load curve value at the time T.
	EQ.3: A plot is generated for each ordinate point in the load curve definition. The actual value of the load curve is ignored.
NPLTC	DT = ENDTIM/NPLTC. ENDTIM is set on *CONTROL_TERMINATION. This calculated value of DT overrides DT specified

VARIABLE	DESCRIPTION
	in the first field.
TBEG	Problem time at which to begin writing output to this time sequence
TEND	Problem time at which to terminate writing output to this time sequence
DOMID1, …	Domain ID specifying the domain over which variable output is to be performed in this time sequence. Each DOMID refers to the domain ID in an instance of *DUALCESE_ENSIGHT_DOMAIN.

Remarks:

1. **Output time interval order of precedence.** If LCDT is nonzero, then it is used, and DT and NPLTC are ignored. If LCDT is zero and NPLTC is nonzero, then NPLTC determines the snapshot time increment. If LCDT and NPLTC are both zero, then the minimum nonzero time increment specified by DT is used to determine the snapshot times.

*DUALCESE_EOS_CAV_HOMOG_EQUILIB

Purpose: Define the coefficients in the equation of state (EOS) for the homogeneous equilibrium cavitation model.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	RHOVAP	RHOLIQ	AVAP	ALIQ	MUVAP	MULIQ	PSATVAP
Type	I	F	F	F	F	F	F	F
Default	none	0.8	880.0	334.0	1386.0	1.435e-5	1.586e-4	1.2e+4

VARIABLE	DESCRIPTION
EOSID	Equation of state ID
RHOVAP	Density of the saturated vapor, ρ_{vap}
RHOLIQ	Density of the saturated liquid, ρ_{liq}
AVAP	Sound speed of the saturated vapor, a_{vap}
ALIQ	Sound speed of the saturated liquid, a_{liq}
MUVAP	Dynamic viscosity of the vapor, μ_{vap}
MULIQ	Dynamic viscosity of the liquid, μ_{liq}
PSATVAP	Pressure of the saturated vapor, P_{SatVap}

Remarks:

1. **EOS Validity.** In this homogeneous equilibrium cavitation model, a barotropic equation of state is used. This model can be used in small scale and high-speed cavitation flows, but it is not good for large-scale, low-speed cavitation calculations.

*DUALCESE_EOS_COCHRAN_CHAN

Purpose: Define a Cochran-Chan type of EOS that provides a means to represent a condensed phase explosive in a dual CESE multiphase model.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	A	B	EPS1	EPS2	GAMMA0	RHO0	E0
Type	I	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	0.0

Card 2	1	2	3	4	5	6	7	8
Variable	CV							
Type	F							
Default	none							

VARIABLE	DESCRIPTION
EOSID	Equation of state ID for the dual CESE solver
A	Model parameter (in pressure units), A
B	Model parameter (in pressure units), B
EPS1	Model constant (dimensionless), ε_1
EPS2	Model constant (dimensionless), ε_2
GAMMA0	Gruneisen coefficient
RHO0	Initial or reference density, ρ_0
E0	Represents the heat of detonation released during the reactions, or the constant rate of afterburn energy added (E0 = 0.0 is the default), e_0
CV	Heat capacity, C_v

Remarks:

The Cochran-Chan EOS like the JWL EOS (*DUALCESE_EOS_JWL) is a type of Mie-Gruneisen EOS. The equations of state of a Mie-Gruneisen form are given by:

$$P(\rho, e) = P_{\text{ref}} + \Gamma(\rho)\rho[e - e_{\text{ref}}(\rho)]$$

Here $\Gamma(\rho)$ is the Gruneisen coefficient. For the Cochran-Chan EOS reference pressure and energy are given by:

$$P_{ref}(\rho) = A\left(\frac{\rho_0}{\rho}\right)^{-\varepsilon_1} - B\left(\frac{\rho_0}{\rho}\right)^{-\varepsilon_2}$$

$$e_{\text{ref}}(\rho) = \frac{A}{\rho_0(1-\varepsilon_1)}\left(\frac{\rho_0}{\rho}\right)^{1-\varepsilon_1} + \frac{B}{\rho_0(1-\varepsilon_2)}\left(\frac{\rho_0}{\rho}\right)^{1-\varepsilon_2} - e_0$$

*DUALCESE_EOS_COOLPROP

Purpose: Define an equation of state (EOS) to be evaluated using the COOLPROP EOS library.

> **WARNING:** This feature is currently not working.

Note that the COOLPROP library is not provided by ANSYS. You need to download a 64-bit version of the shared library from a public repository, such as:

https://sourceforge.net/projects/coolprop/files/CoolProp/6.3.0/shared_library/Linux/64bit/

> **WARNING:** Since the *MODULE capability is not yet working in the Windows build of LS-DYNA, do *not* attempt to use a Windows DLL version of the COOLPROP shared library.

To use the COOLPROP shared library with this keyword card, load this shared library into LS-DYNA using the *MODULE capability. The following *MODULE card needs to appear before a *DUALCESE_MODEL card (not inside the file hierarchy of any file specified with a *DUALCESE_MODEL card):

```
*MODULE_LOAD
UserA                    DUALCESE COOLPROP
 < path to installed COOLPROP shared library >
```

Card Summary:

Card 1. This card is required.

EOSID	NCOMP	TYPE	PHASE	TABULAR			

Card 2. Include as many cards as needed to specify mole fractions for the NCOMP components of the fluid.

MOL_FR1	MOL_FR2	MOL_FR3	MOL_FR4	MOL_FR5	MOL_FR6	MOL_FR7	MOL_FR8

Card 3. Include this card when the TABULAR field is active on Card 1.

N_T	N_DEN	DEN_LOW	DEN_HIGH	T_LOW	T_HIGH		

Card 4. This card is required.

FLUIDNAME

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	NCOMP	TYPE	PHASE	TABULAR			
Type	I	I	A	A	A			
Default	none	none	none	GAS	optional			

VARIABLE	DESCRIPTION
EOSID	ID for this EOS
NCOMP	Number of components in the fluid composition
TYPE	Fluid type: EQ.PURE: A single component fluid (default) EQ.PSEUDOPURE: A predefined fluid mixture EQ.MIXTURE: A fluid mixture with NCOMP components
PHASE	Phase of the fluid. EQ.GAS: Gas phase EQ.LIQUID: Liquid phase
TABULAR	Type of lookup tables to build for this EOS: EQ.<BLANK>: No table lookup (default) EQ.P_EIN: Build tables of pressure and internal energy, both as a function of density and temperature.

COOLPROP Parameters by Fluid Component. Repeat this card as many times as needed to input mole fractions for the NCOMP components of the fluid.

Card 2	1	2	3	4	5	6	7	8
Variable	MOL_FR1	MOL_FR2	MOL_FR3	MOL_FR4	MOL_FR5	MOL_FR6	MOL_FR7	MOL_FR8
Type	F	F	F	F	F	F	F	F

VARIABLE	DESCRIPTION
MOL_FR*i*	Mole fraction of the i^{th} component

COOLPROP EOS Table Density and Temperature Ranges. This card is included when the TABULAR option on Card 1 is active.

Card 3	1	2	3	4	5	6	7	8
Variable	N_T	N_DEN	DEN_LOW	DEN_HIGH	T_LOW	T_HIGH		
Type	I	I	F	F	F	F		
Default	none	none	none	none	none	none		
Remarks	1	1	2	2	2	2		

VARIABLE	DESCRIPTION
N_T	Number of temperature values in the tables
N_DEN	Number of density values (on a log scale) in the tables
DEN_LOW	Minimum density available in the tables (in model units)
DEN_HIGH	Maximum density available in the tables (in model units)
T_LOW	Minimum temperature available in the tables (in model units)
T_HIGH	Maximum temperature available in the tables (in model units)

Name of CoolProp fluid. This card is required.

Card 4	1	2	3	4	5	6	7	8
Variable	FLUIDNAME							
Type	A							

VARIABLE	DESCRIPTION
FLUIDNAME	Name of a fluid that has an EOS in CoolProp. For a list of the supported pure and pseudo-pure fluids, see: http://www.coolprop.org/fluid_properties/PurePseudoPure.html#list-of-fluids Note that the predefined fluid mixtures are not supported at this time.

Remarks:

1. **Number of Values in the Lookup Tables.** The number of density and temperature values in the tables should not be too few to give good resolution of the EOS. Note that the cost of building the EOS from these tables rises with these numbers, as well as the computer memory required. Nevertheless, if these numbers are too small (< 20), then the accuracy may suffer, while larger numbers of density and temperature points improves the accuracy.

2. **Valid Value Ranges for the Lookup Tables.** For many equations of state in the CoolProp library, there is a range of valid densities and temperatures. Thus, the low and high limits for the table densities and temperatures should not lie outside these ranges. Please refer to the CoolProp documentation for that information.

*DUALCESE_EOS_IDEAL_GAS

Purpose: Define the coefficients $C_v$and C_p in the equation of state for an ideal gas in the dual CESE fluid solver.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	CV	CP	E0				
Type	I	F	F	F				
Default	none	717.5	1004.5	0.0				
Remarks		1	1	2				

VARIABLE	DESCRIPTION
EOSID	Equation of state ID
CV	Specific heat at constant volume, C_v
CP	Specific heat at constant pressure, C_p
E0	Represents the heat of detonation released during the reactions, or the constant rate of afterburn energy added (E0 = 0.0 is the default), e_0

Remarks:

1. **Units.** As with other solvers in LS-DYNA, you are responsible for unit consistency. For example, if you want to use dimensionless variables, CV and CP should also be replaced by the corresponding dimensionless ones. If the dual CESE model has a specified system of units either directly from the *DUALCESE_MODEL card or inherited from the overall problem input, then these values need to be given in that unit system.

2. **E0.** E0 is used only with the hybrid multiphase solver where the EOS of the reactant is specified by this ideal gas EOS in the *DUALCESE_EOS_SET card.

*DUALCESE_EOS_INFLATOR1

Purpose: Define an EOS using C_p and C_v thermodynamic expansions for an inflator gas mixture with a single temperature range.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID							
Type	I							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	CP0	CP1	CP2	CP3	CP4			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

Card 3	1	2	3	4	5	6	7	8
Variable	CV0	CV1	CV2	CV3	CV4			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

VARIABLE	DESCRIPTION
EOSID	Equation of state ID for the dual CESE solver
CP0, …, CP4	Coefficients of temperature-dependent specific heat at constant pressure $$C_p(T) = C_{p_0} + C_{p_1}T + C_{p_2}T^2 + C_{p_3}T^3 + C_{p_4}T^4$$

VARIABLE	DESCRIPTION
CV0, …, CV4	Coefficients of temperature-dependent specific heat at constant volume $$C_v(T) = C_{v0} + C_{v1}T + C_{v2}T^2 + C_{v3}T^3 + C_{v4}T^4$$

Remarks:

These coefficient expansions for the specific heats over the entire temperature range are generated by the zero-dimensional inflator model solver. See *CHEMISTRY_CONTROL_INFLATOR and *CHEMISTRY_INFLATOR_PROPERTIES for details related to running that solver.

*DUALCESE_EOS_INFLATOR2

Purpose: Define an EOS using C_p and C_v thermodynamic expansions for an inflator gas mixture with two temperature ranges, one below 1000 Kelvin, and the other above 1000 Kelvin.

Card Summary:

Card 1. This card is required.

EOSID							

Card 2. This card is required. This card with Card 3 specifies C_p. This card gives the coefficients for T < 1000 K.

CP10	CP11	CP12	CP13	CP14			

Card 3. This card is required. This card gives the coefficients for T > 1000 K.

CP20	CP21	CP22	CP23	CP24			

Card 4. This card is required. This card with Card 5 defines C_v. This card gives the coefficients for T < 1000 K.

CV10	CV11	CV12	CV13	CV14			

Card 5. This card is required. This card gives the coefficients for T > 1000 K.

CV20	CV21	CV22	CV23	CV24			

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
EOSID	Equation of state ID for the dual CESE solver

Coefficients for the expansion to determine specific heat at constant pressure for T < 1000 K.

Card 2	1	2	3	4	5	6	7	8
Variable	CP10	CP11	CP12	CP13	CP14			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

Coefficients for the expansion to determine specific heat at constant pressure for T > 1000 K.

Card 3	1	2	3	4	5	6	7	8
Variable	CP20	CP21	CP22	CP23	CP24			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

VARIABLE	DESCRIPTION
CP10, …, CP14	Coefficients of temperature-dependent specific heat at constant pressure valid for T < 1000 K
CP20, …, CP24	Coefficients of temperature-dependent specific heat at constant pressure valid for T > 1000 K

Cards 2 and 3 give C_p over the two temperature ranges:

$$C_p(T) = \begin{cases} \text{CP10} + \text{CP11} \times T + \text{CP12} \times T^2 + \text{CP13} \times T^3 + \text{CP14} \times T^4 & \text{for } T < 1000 \text{ K} \\ \text{CP20} + \text{CP21} \times T + \text{CP22} \times T^2 + \text{CP23} \times T^3 + \text{CP24} \times T^4 & \text{for } T > 1000 \text{ K} \end{cases}$$

Card for the Expansion of Specific Heat at Constant Volume. Valid for T < 1000 K

Card 4	1	2	3	4	5	6	7	8
Variable	CV10	CV11	CV12	CV13	CV14			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

Card for the Expansion of Specific Heat at Constant Volume. Valid for T > 1000 K

Card 5	1	2	3	4	5	6	7	8
Variable	CV20	CV21	CV22	CV23	CV24			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

VARIABLE	DESCRIPTION
CV10, …, CV14	Coefficients of temperature-dependent specific heat at constant volume valid for T < 1000 K
CV20, …, CV24	Coefficients of temperature-dependent specific heat at constant volume valid for T > 1000 K

Cards 4 and 5 give C_v over the two temperature ranges:

$$C_v(T) = \begin{cases} \text{CV10} + \text{CV11} \times T + \text{CV12} \times T^2 + \text{CV13} \times T^3 + \text{CV14} \times T^4 & \text{for } T < 1000 \text{ K} \\ \text{CV20} + \text{CV21} \times T + \text{CV22} \times T^2 + \text{CV23} \times T^3 + \text{CV24} \times T^4 & \text{for } T > 1000 \text{ K} \end{cases}$$

Remarks:

These coefficient expansions for the specific heats over two temperature ranges are generated by the zero-dimensional inflator model solver. See *CHEMISTRY_CONTROL_INFLATOR and *CHEMISTRY_INFLATOR_PROPERTIES for details related to running that solver.

*DUALCESE_EOS_JWL

Purpose: Define a JWL-type EOS that provides a means to represent a condensed phase explosive in a dual CESE multiphase model.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	A	B	R1	R2	GAMMA0	RHO0	E0
Type	I	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	0.0

Card 2	1	2	3	4	5	6	7	8
Variable	CV							
Type	F							
Default	none							

VARIABLE	DESCRIPTION
EOSID	Equation of state ID for the dual CESE solver.
A	Model parameter, A (in pressure units)
B	Model parameter, B (in pressure units)
R1	Model constant, R_1 (dimensionless)
R2	Model constant, R_2 (dimensionless)
GAMMA0	Gruneisen coefficient
RHO0	Initial or reference density, ρ_0
E0	Represents the heat of detonation released during the reactions or the constant rate of afterburn energy added (E0 = 0.0 for standard JWL EOS), e_0
CV	Heat capacity, C_v

Remark:

The equations of state of a Mie-Gruneisen form are given by:

$$P(\rho, e) = P_{\text{ref}} + \Gamma(\rho)\rho[e - e_{\text{ref}}(\rho)]$$

Here $\Gamma(\rho)$ is the Gruneisen coefficient. Equations of state of this type are very popular in condensed phase explosive modeling. Depending on the form of the reference pressure and energy functions, different EOS types can be retrieved. The JWL EOS is one type with reference pressure and energy given by:

$$P_{\text{ref}}(\rho) = A\exp\left(\frac{-R_1\rho_0}{\rho}\right) + B\exp\left(\frac{-R_2\rho_0}{\rho}\right)$$

$$e_{\text{ref}}(\rho) = \frac{A}{\rho_0 R_1}\exp\left(\frac{-R_1\rho_0}{\rho}\right) + \frac{B}{\rho_0 R_2}\exp\left(\frac{-R_2\rho_0}{\rho}\right) - e_0$$

*DUALCESE_EOS_NASG

Purpose: Define a Noble-Abel Stiffened-Gas (NASG) type fluid EOS for use in the phase-change dual CESE solver. See M'etayer and Saurel [2016] for details.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	GAMMA	CV	PINF	Q	QP	B	W
Type	I	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
EOSID	Equation of state ID
GAMMA	Ratio of specific heats, γ
CV	Specific heat at constant volume
PINF	Parameter for a given phase, p_∞ (dimensions: $\left[\frac{M}{LT^2}\right]$). See Le M'etayer and Saurel [2016].
Q	Parameter for a given phase, q (dimensions: $\left[\frac{L^2}{T^2}\right]$). q is the heat bond for the phase. See Le M'etayer and Saurel [2016].
QP	Parameter for a given phase, q' (dimensions: $\left[\frac{L^2}{T^2\theta}\right]$, where θ represents the dimension of temperature). See Le M'etayer and Saurel [2016].
B	Parameter for a given phase, b (dimensions: $\left[\frac{L^3}{M}\right]$). b indicates the fluid's covolume. See Le M'etayer and Saurel [2016].
W	Molar mass of this fluid

Remarks:

The NASG EOS given by M'etayer and Saurel [2016] has the following form:

$$p(\nu, e) = \frac{(\gamma - 1)(e - q)}{\nu - b} - \gamma p_\infty$$
$$T(p, \nu) = \frac{(\nu - b)(p + p_\infty)}{C_v(\gamma - 1)}$$
$$g(p, T) = (\gamma C_v - q')T - C_v T \ln\left[\frac{T^\gamma}{(p + p_\infty)^{\gamma - 1}}\right] + bp + q$$
$$c(p, \nu) = \sqrt{\gamma\nu(p + p_\infty)}$$

where γ, C_v, p_∞, q, q' and b are parameters needed for each phase. ν, e, g and c are for each fluid phase the specific volume, the specific internal energy, the specific Gibbs free energy, and the sound speed, respectively. $g = h - Ts$ with h and s as the specific enthalpy and specific entropy, respectively.

References:

[1] Le M'etayer, O. and R. Saurel, "The Nobel-Abel Stiffened-Gas equation of state," *Physics of Fluids*, 28 (4), (2016).

*DUALCESE_EOS_REFPROP

Purpose: Define an equation of state (EOS) to be evaluated using the REFPROP EOS library

Note that the REFPROP library is provided by ANSYS; it is REFPROP v10.0 from NIST.

> **WARNING:** Since the *MODULE capability is not yet working in the Windows build of LS-DYNA, do *not* attempt to use a Windows DLL version of the REFPROP shared library that comes with REFPROP v10.0.

To use the REFPROP shared library with this keyword card, load this shared library into LS-DYNA using the *MODULE capability. The following *MODULE card needs to appear before a *DUALCESE_MODEL card (not inside the file hierarchy of any file specified with a *DUALCESE_MODEL card):

```
*MODULE_LOAD
UserA                    DUALCESE REFPROP
 < path to the installed REFPROP shared library >
```

In addition, for REFPROP to be able to find the appropriate EOS data, *DUALCESE_EOS_REFPROP_PATH must also be given somewhere inside a *DUALCESE_MODEL file hierarchy to point to the place in your filesystem where REFPROP has been installed.

Card Summary:

Card 1. This card is required.

EOSID	NCOMP	TYPE	PHASE	TABULAR			

Card 2. Include as many cards as needed to specify mole fractions for the NCOMP components of the fluid.

MOL_FR1	MOL_FR2	MOL_FR3	MOL_FR4	MOL_FR5	MOL_FR6	MOL_FR7	MOL_FR8

Card 3. This card is included when the TABULAR field is active on Card 1.

N_T	N_DEN	DEN_LOW	DEN_HIGH	T_LOW	T_HIGH		

Card 4. This card is required.

FLUIDNAME

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	NCOMP	TYPE	PHASE	TABULAR			
Type	I	I	A	A	A			
Default	none	none	none	GAS	optional			

VARIABLE	DESCRIPTION
EOSID	ID for this EOS
NCOMP	Number of components in the fluid composition
TYPE	Fluid type: EQ.PURE: A single component fluid (default) EQ.PSEUDOPURE: A predefined fluid mixture EQ.MIXTURE: A fluid mixture with NCOMP components
PHASE	Phase of the fluid: EQ.GAS: Gas phase EQ.LIQUID: Liquid phase
TABULAR	Type of lookup tables to build for this EOS: EQ.<BLANK>: No table lookup (default) EQ.P_EIN: Build tables of pressure and internal energy, both as a function of density and temperature.

REFPROP Parameters by Fluid Component. Repeat this card as many times as needed to input mole fractions for the NCOMP components of the fluid.

Card 2	1	2	3	4	5	6	7	8
Variable	MOL_FR1	MOL_FR2	MOL_FR3	MOL_FR4	MOL_FR5	MOL_FR6	MOL_FR7	MOL_FR8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
MOL_FR*i*	Mole fraction of the i^{th} component of the fluid.

.

REFPROP EOS Table Density and Temperature Ranges. This card is included when the TABULAR field is active on Card 1.

Card 3	1	2	3	4	5	6	7	8
Variable	N_T	N_DEN	DEN_LOW	DEN_HIGH	T_LOW	T_HIGH		
Type	I	I	F	F	F	F		
Default	none	none	none	none	none	none		
Remarks	1	1	2	2	2	2		

VARIABLE	DESCRIPTION
N_T	Number of temperature values in the tables
N_DEN	Number of density values (on a log scale) in the table
DEN_LOW	Minimum density available in the tables (in model units)
DEN_HIGH	Maximum density available in the tables (in model units)
T_LOW	Minimum temperature available in the tables (in model units)
T_HIGH	Maximum temperature available in the tables (in model units)

Name of REFPROP fluid. This card is required.

Card 4	1	2	3	4	5	6	7	8
Variable	FLUIDNAME							
Type	A							

VARIABLE	DESCRIPTION
FLUID-NAME	Name of a fluid that has an EOS in REFPROP. For a list of the supported pure and pseudo-pure fluids, see the directory of supported fluids that comes with the REFPROP v10.0 library from ANSYS. Note that the predefined fluid mixtures are not supported at this time.

Remarks:

1. **Number of Values in the Lookup Table.** The number of values of density and temperature axes of the tables should not be too few to give good resolution of the EOS. Note that the cost of building the EOS from these tables rises with these numbers, as well as the computer memory required. Nevertheless, if these numbers are too small (< 20), then the accuracy may suffer, while larger numbers of density and temperature points improves the accuracy.

2. **Valid Value Ranges for the Lookup Tables.** For many equations of state in the REFPROP library, a range of densities and temperatures are valid. Thus, the low and high limits for the table densities and temperatures should not lie outside these ranges. Please refer to the REFPROP documentation for that information.

*DUALCESE_EOS_REFPROP_PATH

Purpose: Provide the file path to the directory where the REFPROP EOS system is installed.

Note that in any problem where a *DUALCESE_EOS_REFPROP card is used, you must also provide a *DUALCESE_EOS_REFPROP_PATH card somewhere inside a *DUALCESE_MODEL file hierarchy to point to the place in your filesystem where REFPROP has been installed so that the appropriate EOS data can be loaded.

Card 1	1	2	3	4	5	6	7	8
Variable	PATH							
Type	A							

VARIABLE	DESCRIPTION
PATH	Path giving the directory where the REFPROP data is installed.

*DUALCESE_EOS_SET

Purpose: Define a set of equations of state that are used together to compute the thermodynamic state of a multiphase fluid for the dual CESE solver.

> **WARNING:** As of version R14, this keyword is deprecated. Instead, use *DUALCESE_SOLVER_HYBRID_MULTIPHASE for the hybrid multiphase case and *DUALCESE_-SOLVER_TWO-PHASE_MULTIPHASE for the two-phase multiphase case.

Include one card for each dual CESE multiphase mesh. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSSID	EOSINID	EOSRCTID	EOSPRDID				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
EOSSID	Set ID of the EOS mixture of a given multiphase fluid
EOSINID	EOS ID of the inert component of the multiphase mixture
EOSRCTID	EOS ID of the reactant phase of the multiphase mixture
EOSPRDID	EOS ID of the product phase of the multiphase mixture

*DUALCESE_EOS_STIFFENED_GAS

Purpose: Define a stiffened gas type fluid EOS for use by the dual CESE solver.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	GA	BT					
Type	I	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
EOSID	Equation of state ID for this dual CESE solver EOS
GA	Adiabatic exponent, γ. Must be > 1.0.
BT	Reference pressure, β. Must be ≥ 0.0

Remark:

The stiffened gas equation of state:

$$p(\rho, e) = (\gamma - 1)\rho e - \gamma\beta$$

provides a fundamental characterization of material properties of fluids. Here e is the internal energy per unit mass, ρ is the density, and γ and β are two thermodynamic constants. γ and β can be determined by a fitting procedure from laboratory data. A typical set of parameter values for water are: $\gamma = 7$ and $\beta = 3000$ atm while for human blood are: $\gamma = 5.527$ and $\beta = 614.6$ MPa. In addition to the modelling of a liquid, it is often used to describe other type of materials, including many compressible solids of practical importance.

*DUALCESE_EOS_VAN_DER_WAALS_GENERALIZED

Purpose: Define a Van Der Waals generalized type fluid EOS for use in the dual CESE solver.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	A	B	GA	BT			
Type	I	F	F	F	F			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
EOSID	Equation of state ID
A	van der Waals gas constant for molecular cohesive forces, a
B	van der Waals gas constant for the finite size of molecules, b
GA	Ratio of specific heats, γ. Must be > 1.0.
BT	Reference pressure, β. Must be ≥ 0.0. See *DUALCESE_EOS_-STIFFENED_GAS.

Remark:

The generalized Van der Waals equation of state can be written as:

$$p(\rho, e) = \frac{(\gamma - 1)}{1 - b\rho}\left(\rho e - \beta + a\rho^2\right) - \left(\beta + a\rho^2\right)$$

Here e denotes the specific internal energy, γ is the ratio of specific heats ($\gamma > 1$), β is a reference pressure, and the quantities a and b are the van der Waals gas constants for molecular cohesive forces and the finite size of molecules, respectively ($a \geq 0$, $0 \leq b < 1/\rho$). This EOS is often used to deal with possible real-gas effect (without phase transition) when both the temperature and pressure are high. When $\beta = 0$, the generalized van der Waals equation of state becomes a standard Van der Waals equation of state. If $a = b = 0$, the stiffened gases EOS will be recovered.

References:

[1] G. Allaire, S. Clerc, S. Kokh, A five-equation model for the simulation of interfaces between compressible fluids. J. Comp. Phys. 181 (2) (2002) 577-616.

[2] K.M. Shyue, A fluid-mixture type algorithm for compressible multicomponent flow with van der Waals equation of state, J. Comp. Phys. 156, 43 (1999)

*DUALCESE_FSI_EXCLUDE

Purpose: Provide a list of mechanics solver parts that are not involved in the dual CESE FSI calculation. This keyword is intended for increasing computational efficiency by excluding parts that will not involve significant FSI interactions with the dual CESE compressible fluid solver.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID1	PID2	PID3	PID4	PID5	PID6	PID7	PID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
PID*n*	IDs of mechanics parts that will be excluded from the FSI interaction calculation with the dual CESE solver

*DUALCESE_FSI_EXCLUDE_PARTSET

Purpose: Provide a list of mechanics solver part sets for parts that are not involved in the dual CESE FSI calculation. This keyword is intended for increasing computational efficiency by excluding parts that will not involve significant FSI interactions with the dual CESE compressible fluid solver.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID1	PSID2	PSID3	PSID4	PSID5	PSID6	PSID7	PSID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
PSID*n*	IDs of mechanics part sets that will be excluded from the FSI interaction calculation with the dual CESE solver

*DUALCESE_GRAVITY

Purpose: Apply a gravity load to the dual CESE fluid. It is only avilable for single fluid solvers (meaning EQNS = EULER or NS) for both FSI (including FSI-IBM and FSI-MMM) and no FSI. Please note that this keyword must be accompanied by *LOAD_BODY_VECTOR, *LOAD_BODY_X, *LOAD_BODY_Y, or *LOAD_BODY_Z. However, these *LOAD_BODY_... keywords cannot be in the dual CESE model portion of the input deck (see Dual CESE Keyword Deck Structure for details).

Card 1	1	2	3	4	5	6	7	8
Variable	IGRAV							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
IGRAV	Flag for applying a gravity load: EQ.0: No gravity load applied (default) EQ.1: Apply gravity load

Remarks:

1. **Including gravity.** In most compressible cases, the gravity terms are not needed (the default is no gravity). Including gravity only has a significant effect for certain cases, such as buoyancy driven flow, stratified flow, etc.

*DUALCESE_INCLUDE_MODEL

Purpose: Provide the filename of a file containing additional keywords belonging to a dual CESE model. Any number of these *DUALCESE_INCLUDE_MODEL keywords may be used in a single dual CESE model, where at the top level the overall model begins with a *DUALCESE_MODEL card.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A							

VARIABLE	DESCRIPTION
FILENAME	Filename of the keyword file containing more of the dual CESE model. This card is only allowed inside a file that is given in one instance of a *DUALCESE_MODEL keyword card.

*DUALCESE_INITIAL

Purpose: Specify constant initial conditions for flow variables at the centroid of each dual CESE fluid element.

Card 1	1	2	3	4	5	6	7	8
Variable	U	V	W	RH	P	T		IFUNC
Type	F	F	F	F	F	F		I
Default	0	0.0	0.0	1.225	0.0	0.0		none

VARIABLE	DESCRIPTION
U, V, W	x, y, and z velocity components, respectively
RHO	Density, ρ
P	Pressure, P
T	Temperature, T
IFUNC	Option to define initial conditions using *DEFINE_FUNCTION cards: EQ.0: Not in use. EQ.1: All values for initial velocity, pressure, density, and temperature now refer to *DEFINE_FUNCTION IDs. In these functions, the following parameters are allowed: f(x,y,z), meaning that each variable's initial profile is a function of position.

Remarks:

1. **Required input.** Usually, only two of ρ, P, and T are needed to be specified (besides the velocity). If all three are given, only ρ and P will be used.

2. **Applicable elements.** These initial conditions will be applied in those elements that have not been assigned a value by *DUALCESE_INITIAL_*OPTION* cards for individual elements or sets of elements.

*DUALCESE_INITIAL_SET

Purpose: Specify initial conditions for the flow variables at the centroid of each element in an element subset of the dual CESE mesh.

Card 1	1	2	3	4	5	6	7	8
Variable	ESID	IFUNC						
Type	I	I						
Default	none	none						

Card 2	1	2	3	4	5	6	7	8
Variable	U	V	W	RHO	P	T		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
ESID	Element set ID (see *DUALCESE_ELEMENTSET)
IFUNC	Option to define initial conditions using *DEFINE_FUNCTION cards: EQ.0: Not in use. EQ.1: All values for initial velocity, pressure, density, and temperature now refer to *DEFINE_FUNCTION IDs. In these functions, the following parameters are allowed: f(x,y,z), meaning that each variable's initial profile is a function of position.
U, V, W	x, y, and z velocity components, respectively
RHO	Density, ρ
P	Pressure, P

VARIABLE	DESCRIPTION
T	Temperature, T

Remarks:

1. **Required input.** Usually, only two of ρ, P and T are needed to be specified (besides the velocity). If all three are given, only ρ and P will be used.

2. **Keyword priority.** The priority of this card is higher than *DUALCESE_INITIAL. Thus, if an element is assigned an initial value by this card, *DUALCESE_INITIAL will no longer apply to that element.

*DUALCESE_INITIAL_REGION

Purpose: Specify initial conditions for the flow variables at the centroid of each element in a region of the dual CESE mesh specified with *DEFINE_REGION. This initialization method is only used by the single-fluid dual CESE solvers.

Card 1	1	2	3	4	5	6	7	8
Variable	REGID	IFUNC						
Type	I	I						
Default	none	none						

Card 2	1	2	3	4	5	6	7	8
Variable	U	V	W	RHO	P	T		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
REGID	Region ID (see *DEFINE_REGION).
IFUNC	Flag to specify initial conditions using *DEFINE_FUNCTION: EQ.0: Not in use. EQ.1: All values for initial velocity, pressure, temperature, etc. now refer to *DEFINE_FUNCTION IDs. In these functions, the following parameters are allowed: f(x, y, z). Thus, each variable's initial profile is a function of position.
U, V, W	x, y, and z components of velocity, respectively
RHO	Density, ρ
P	Pressure, P

VARIABLE	DESCRIPTION
T	Temperature, T

Remarks:

1. **Required input.** Usually, only two of ρ, P, and T are needed to be specified (besides the velocity). If all three are given, only ρ and P will be used.

2. **Keyword priority.** The priority of this card is higher than *DUALCESE_INITIAL. Thus, if an element is assigned an initial value by this keyword, *DUALCESE_INITIAL will no longer apply to that element. However, the priority of *DUALCESE_INITIAL_SET is higher than this keyword, meaning any element initialized by *DUALCESE_INITIAL_SET will not be initialized by this keyword.

*DUALCESE_INITIAL_CHEMISTRY

Purpose: Initialize the chemistry and fluid state in every element of the dual CESE mesh that has not already been initialized by one of the other *DUALCESE_INITIAL_CHEMISTRY_... keywords. This keyword is only used when chemistry is solved with the dual CESE solver.

Card 1	1	2	3	4	5	6	7	8
Variable	CHEMID	COMPID						
Type	I	I						
Default	none	none						

Card 2	1	2	3	4	5	6	7	8
Variable	UIC	VIC	WIC		PIC	TIC	HIC	
Type	F	F	F		F	F	F	
Default	none	none	none		none	none	none	

VARIABLE	DESCRIPTION
CHEMID	Identifier of the chemistry control card to use (see *CHEMISTRY_CONTROL_FULL)
COMPID	Identifier of chemical composition to use (see *CHEMISTRY_COMPOSITION)
UIC	*X*-component of the fluid velocity
VIC	*Y*-component of the fluid velocity
WIC	*Z*-component of the fluid velocity
PIC	Initial fluid pressure
TIC	Initial fluid temperature

VARIABLE	DESCRIPTION
HIC	Initial fluid enthalpy

Remarks:

1. **Applicable elements.** These initial conditions apply only in those elements that have not been assigned a value by other *DUALCESE_INITIAL_CHEMISTRY_... keywords.

*DUALCESE_INITIAL_CHEMISTRY_REGION

Purpose: Initialize the chemistry and fluid state at the element centroid of each element in a region of the dual CESE mesh that is specified with *DEFINE_REGION. This keyword is used only when chemistry is solved with the dual CESE solver.

Card 1	1	2	3	4	5	6	7	8
Variable	REG_ID	CHEMID	COMPID					
Type	I	I	I					
Default	none	none	none					

Card 2	1	2	3	4	5	6	7	8
Variable	UIC	VIC	WIC		PIC	TIC	HIC	
Type	F	F	F		F	F	F	
Default	none	none	none		none	none	none	

VARIABLE	DESCRIPTION
REG_ID	Region identifier (see *DEFINE_REGION).
CHEMID	Identifier of the chemistry control card to use (see *CHEMISTRY_CONTROL_FULL)
COMPID	Identifier of chemical composition to use (see *CHEMISTRY_COMPOSITION)
UIC	*X*-component of the fluid velocity
VIC	*Y*-component of the fluid velocity
WIC	*Z*-component of the fluid velocity
PIC	Initial fluid pressure
TIC	Initial fluid temperature

VARIABLE	DESCRIPTION
HIC	Initial fluid enthalpy

Remarks:

1. **Keyword priority.** The priority of this keyword is higher than *DUALCESE_INITIAL_CHEMISTRY and *DUALCESE_INITIAL_CHEMISTRY_SET. Thus, if an element is assigned an initial value by this keyword, *DUALCESE_INITIAL_CHEMISTRY and *DUALCESE_INITIAL_CHEMISTRY_SET no longer apply to that element.

*DUALCESE_INITIAL_CHEMISTRY_SET

Purpose: Initialize the chemistry and fluid state in every element of the specified dual CESE element set in the dual CESE mesh. This keyword is used only when chemistry is solved with the dual CESE solver.

Card 1	1	2	3	4	5	6	7	8
Variable	SETID	CHEMID	COMPID					
Type	I	I	I					
Default	none	none	none					

Card 2	1	2	3	4	5	6	7	8
Variable	UIC	VIC	WIC		PIC	TIC	HIC	
Type	F	F	F		F	F	F	
Default	none	none	none		none	none	none	

VARIABLE	DESCRIPTION
SETID	Identifier of a dual CESE element set defined by *DUALCESE_ELEMENTSET
CHEMID	Identifier of the chemistry control card to use (see *CHEMISTRY_CONTROL_FULL)
COMPID	Identifier of chemical composition to use (see *CHEMISTRY_COMPOSITION)
UIC	*X*-component of the fluid velocity
VIC	*Y*-component of the fluid velocity
WIC	*Z*-component of the fluid velocity
PIC	Initial fluid pressure

VARIABLE	DESCRIPTION
TIC	Initial fluid temperature
HIC	Initial fluid enthalpy

Remarks:

1. **Keyword priority.** The priority of this keyword is higher than *DUALCESE_INITIAL_CHEMISTRY. Thus, if an element is assigned an initial value by this keyword, *DUALCESE_INITIAL_CHEMISTRY no longer applies to that element. However, the priority of *DUALCESE_INITIAL_CHEMISTRY_REGION is higher than this keyword, meaning any element initialized by *DUALCESE_INITIAL_CHEMISTRY_REGION is not initialized by this keyword.

*DUALCESE_INITIAL_HYBRID

Purpose: Specifies values to use for velocity, pressure, etc. for problem initialization of a hybrid multiphase model on the dual CESE mesh. Note that these values can be overridden in some mesh elements by use of the *DUALCESE_INITIAL_HYBRID_SET card.

The hybrid multiphase model involves two materials, separated by a material surface. These materials do not mix. The first material is inert while the second material is an explosive mixture, composed of reactants and products. This model is useful for simulations with high explosives. See Michael and Nikiforakis 2016 for details about this model.

Card 1	1	2	3	4	5	6	7	8
Variable	Z1	RA	UIC	VIC	WIC	RHO1	RHO_A	RHO_B
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Card 2	1	2	3	4	5	6	7	8
Variable	PIC	TIC		IFUNC				
Type	F	F		I				
Default	none	none		none				

VARIABLE	DESCRIPTION
Z1	Volume fraction of material 1 (or color function). This is usually a value of 0 or 1. For numerical stability, however, use a very small value instead of zero.
RA	Mass fraction of the reactant (material α) with respect to material 2 (the explosive mixture)
UIC, VIC, WIC	Multiphase flow velocity components in the x, y, and z-directions, respectively
RHO1	Density of material 1

VARIABLE	DESCRIPTION
RHO_A	Density of the reactant (material α)
RHO_B	Density of the product (material β)
PIC	Equilibrium multifluid pressure
TIC	Equilibrium multifluid temperature
IFUNC	Option to define initial conditions using *DEFINE_FUNCTION cards: EQ.0: Not in use. EQ.1: All values for initial velocity, pressure, and temperature now refer to *DEFINE_FUNCTION IDs. In these functions, the following parameters are allowed: f(x,y,z), meaning that each variable's initial profile is a function of position.

*DUALCESE_INITIAL_HYBRID_SET

Purpose: Specifies values to use for velocity, pressure, etc. for problem initialization of a hybrid multiphase model in an element subset of the dual CESE mesh. The values specified here override the values set on *DUALCESE_INITIAL_HYBRID for the element subset.

Card 1	1	2	3	4	5	6	7	8
Variable	ESID	IFUNC						
Type	I	I						
Default	none	none						

Card 2	1	2	3	4	5	6	7	8
Variable	Z1	RA	UIC	VIC	WIC	RHO1	RHO_A	RHO_B
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Card 3	1	2	3	4	5	6	7	8
Variable	PIC	TIC						
Type	F	F						
Default	none	none						

VARIABLE	DESCRIPTION
ESID	Element set ID (see *DUALCESE_ELEMENTSET)
IFUNC	Option to define initial conditions using *DEFINE_FUNCTION cards:

VARIABLE	DESCRIPTION
	EQ.0: Not in use. EQ.1: All values for initial velocity, pressure, and temperature now refer to *DEFINE_FUNCTION IDs. In these functions, the following parameters are allowed: f(x,y,z), meaning that each variable's initial profile is a function of position.
Z1	Volume fraction of material 1 (or color function). This is usually a value of 0 or 1. For numerical stability, however, use a small value instead of 0.
RA	Mass fraction of the reactant (material α) with respect to material 2 (the explosive mixture)
UIC, VIC, WIC	Multiphase flow velocity components in the x, y, and z-directions, respectively
RHO1	Density of material 1
RHO_A	Density of the reactant (material α)
RHO_B	Density of the product (material β)
PIC	Equilibrium multifluid pressure
TIC	Equilibrium multifluid temperature

*DUALCESE_INITIAL_PHASE_CHANGE

Purpose: Specify values to use for velocity, pressure, etc. for problem initialization of a phase-change multiphase model on the dual CESE mesh. Note that these values can be overridden in some mesh elements by use of the *DUALCESE_INITIAL_PHASE_CHANGE_SET card.

The phase-change multiphase model involves three fluids: a liquid, its vapor, and other gases.

Card 1	1	2	3	4	5	6	7	8
Variable	UIC	VIC	WIC	RHO	PIC	TIC	Y1	Y2
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Card 2	1	2	3	4	5	6	7	8
Variable	Y3			IFUNC				
Type	F			I				
Default	none			0				

VARIABLE	DESCRIPTION
UIC, VIC, WIC	Multiphase flow velocity components in the x, y, and z-directions, respectively
RHO	Mixture density
PIC	Equilibrium multifluid pressure
TIC	Equilibrium multifluid temperature
Y1	Mass fraction of fluid 1
Y2	Mass fraction of fluid 2

VARIABLE	DESCRIPTION
Y3	Mass fraction of fluid 3
IFUNC	Option to define initial conditions with *DEFINE_FUNCTION: EQ.0: Not in use. EQ.1: All values for initial velocity, pressure, and temperature now refer to *DEFINE_FUNCTION IDs. In these functions, the following parameters are allowed: f(x, y, z). Thus, each variable's initial profile is a function of position.

*DUALCESE_INITIAL_PHASE_CHANGE_SET

Purpose: Specify values to use for velocity, pressure, etc. for problem initialization of a phase- change multiphase model in an element subset of the dual CESE mesh. The values specified here override the values set on *DUALCESE_INITIAL_PHASE_CHANGE for the element subset.

Card 1	1	2	3	4	5	6	7	8
Variable	ESID	IFUNC						
Type	I	I						
Default	none	0						

Card 2	1	2	3	4	5	6	7	8
Variable	UIC	VIC	WIC	RHO	PIC	TIC	Y1	Y2
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Card 3	1	2	3	4	5	6	7	8
Variable	Y3							
Type	F							
Default	none							

VARIABLE	DESCRIPTION
ESID	Element set ID (see *DUALCESE_ELEMENTSET)
IFUNC	Option to define initial conditions with *DEFINE_FUNCTION: EQ.0: Not in use.

VARIABLE	DESCRIPTION
	EQ.1: All values for initial velocity, pressure, and temperature now refer to *DEFINE_FUNCTION IDs. In these functions, the following parameters are allowed: f(x, y, z). Thus, each variable's initial profile is a function of position.
UIC, VIC, WIC	Multiphase flow velocity components in the x, y, and z-directions, respectively
RHO	Mixture density
PIC	Equilibrium multifluid pressure
TIC	Equilibrium multifluid temperature
Y1	Mass fraction of fluid 1
Y2	Mass fraction of fluid 2
Y3	Mass fraction of fluid 3

*DUALCESE_INITIAL_TWO-PHASE

Purpose: Specify values to use for velocity, pressure, etc. for problem initialization of a two-phase multifluid model on the dual CESE mesh. Note that these values can be over-ridden in some mesh elements by use of the *DUALCESE_INITIAL_TWO-PHASE_SET card.

Card 1	1	2	3	4	5	6	7	8
Variable	Z1	UIC	VIC	WIC	RHO_1	RHO_2	PIC	TIC
Type	F	F	F	F	F	F	F	F
Default	none	none	0.	0.	0.	none	none	none

Card 2	1	2	3	4	5	6	7	8
Variable	IFUNC							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
Z1	Volume fraction of material 1 (or color function)
UIC, VIC, WIC	Multiphase flow velocity components in the x, y, and z-directions respectively.
RHO_1	Density of fluid 1
RHO_2	Density of fluid 2
PIC	Equilibrium multifluid pressure
TIC	Equilibrium multifluid temperature
IFUNC	Option to define initial conditions using *DEFINE_FUNCTION cards:

VARIABLE	DESCRIPTION
	EQ.0: Not in use.
	EQ.1: All values for initial velocity, pressure, and temperature now refer to *DEFINE_FUNCTION IDs. In these functions, the following parameters are allowed: f(x,y,z), meaning that each variable's initial profile is a function of position.

*DUALCESE_INITIAL_TWO-PHASE_SET

Purpose: Specify values to use for velocity, pressure, etc. for problem initialization of a two-phase multifluid model in an element subset of the dual CESE mesh.

Card 1	1	2	3	4	5	6	7	8
Variable	ESID	IFUNC						
Type	I	I						
Default	none	none						

Card 2	1	2	3	4	5	6	7	8
Variable	Z1	UIC	VIC	WIC	RHO_1	RHO_2	PIC	TIC
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
ESID	Element set ID (see *DUALCESE_ELEMENTSET)
IFUNC	Option to define initial conditions using *DEFINE_FUNCTION cards: EQ.0: Not in use. EQ.1: All values for initial velocity, pressure, and temperature now refer to *DEFINE_FUNCTION IDs. In these functions, the following parameters are allowed: f(x,y,z), meaning that each variable's initial profile is a function of position.
Z1	Volume fraction of material 1 (or color function)
UIC, VIC, WIC	Multiphase flow velocity components in the x, y, and z-directions, respectively

VARIABLE	DESCRIPTION
RHO_1	Density of material 1
RHO_2	Density of material 2
PIC	Equilibrium multiphase flow pressure
TIC	Equilibrium multiphase flow temperature

*DUALCESE_MAT_GAS

Purpose: Define the fluid (gas) properties in a viscous flow for the dual CESE solver. In this model, the dynamic viscosity is determined using Sutherland's formula for viscosity, and the thermal conductivity is determined using the Prandtl Number.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	C1	C2	PRND				
Type	I	F	F	F				
Default	none	1.458E-6	110.4	0.72				

VARIABLE	DESCRIPTION
MID	Material ID
C1, C2	Two coefficients in the Sutherland's formula for viscosity: $$\mu = \frac{C_1 T^{3/2}}{T + C_2} \ .$$ Here C_1 and C_2 are constants for a given gas. For example, for air at moderate temperatures $$C_1 = 1.458 \times 10^{-6}\,\mathrm{kg/msK^{1/2}}, \quad C_2 = 110.4\ \mathrm{K}$$
PRND	Prandtl Number (used to determine the coefficient of thermal conductivity). It is approximately constant for most gases. For air at standard conditions PRND = 0.72.

Remarks:

1. **Inviscid flows.** C1 and C2 are only used to calculate the viscosity in viscous flows, so for inviscid flows, this material card is not needed.

2. **Thermal coupling.** The Prandtl number is used to extract the thermal conductivity. It is only needed when thermal coupling with the structure is being done. However, only the IBM-FSI single-fluid solvers support a conjugate heat transfer capability .

3. **Unit consistency.** As with other solvers in LS-DYNA, you are responsible for unit consistency. For example, if dimensionless variables are used, C_1 and C_2 should be replaced by the corresponding dimensionless ones. If the dual CESE model has a specified system of units either directly from the *DUALCESE_-MODEL card or inherited from the overall problem input, then these values need to be given in that unit system. Also, note that the formulas here require the temperature be given in either Kelvin or Rankine units.

*DUALCESE_MAT_GAS_0

Purpose: Define the fluid (gas) properties in a viscous flow for the dual CESE solver. In this model, the dynamic viscosity and thermal conductivity are specified constants.

Material Definition Cards. Include one card for each instance of this material type. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	MU	K					
Type	I	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
MID	Material ID
MU	Fluid dynamic viscosity. MU = 1.81×10^{-5} kg/ms for air at 15°C.
K	Thermal conductivity of the fluid

Remarks:

1. **Fields that depend on the problem's physics.** The viscosity is only used in viscous flows, so it is not necessary to define it for inviscid flows. The thermal conductivity is only used when coupling is activated between the structure and thermal solver to calculate the heat transfer. However, only the IBM-FSI single-fluid solvers support a conjugate heat transfer capability

2. **Unit consistency.** As with other solvers in LS-DYNA, you are responsible for unit consistency. For example, if dimensionless variables are used, MU should be replaced by the corresponding dimensionless one. If the dual CESE model has a specified system of units either directly from the *DUALCESE_MODEL card or inherited from the overall problem input, then these values need to be given in that unit system.

*DUALCESE_MAT_GAS_2

Purpose: Define the fluid (gas) properties in a viscous flow for the dual CESE solver. This model determines dynamic viscosity and thermal conductivity by combining Sutherland's formula with the Power law for dilute gases.

Material Definition Cards. Include one card for each instance of this material type. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	MU0	SMU	K0	SK	T0		
Type	I	F	F	F	F	F		
Default	none	1.716E-5	111.	0.0241	194.0	273.0		

VARIABLE	DESCRIPTION
MID	Material ID
MU0 / SMU	Two coefficients appearing in the equation derived by combining Sutherland's formula with the Power law for dilute gases: $$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0}\right)^{3/2} \frac{T_0 + S_\mu}{T + S_\mu} \ .$$ Here μ is the dynamic viscosity, μ_0 is a reference value, and S_μ is an effective temperature called the Sutherland constant which is characteristic of the gas. For air at moderate temperatures, $$\mu_0 = 1.716 \times 10^{-5}\,\mathrm{Ns/m^2}\,, \qquad S_\mu = 111\ \mathrm{K}$$
K0/SK	Two coefficients appearing in the equation derived by combining Sutherland's formula with the Power law for dilute gases: $$\frac{k}{k_0} = \left(\frac{T}{T_0}\right)^{3/2} \frac{T_0 + S_k}{T + S_k} \ .$$ Here k is the thermal conductivity, k_0 is a reference value, and S_k is the Sutherland constant, which is characteristic of the gas. For air at moderate temperatures, $$k_0 = 0.0241\ \mathrm{W/m}\,, \qquad S_k = 194\ \mathrm{K}$$
T0	Reference temperature, T_0. The default value (273.0) is for air in K.

Remarks:

1. **Fields that depend on the problem's physics.** The viscosity is only used for viscous flows, so it is not necessary to define it for inviscid flows. The thermal conductivity is only used to calculate the heat transfer between the structure and the thermal solver when coupling is activated. However, only the IBM-FSI single-fluid solvers support a conjugate heat transfer capability.

2. **Unit consistency.** As with other solvers in LS-DYNA, you are responsible for unit consistency. For example, if dimensionless variables are used, MU should be replaced by the corresponding dimensionless one. If the dual CESE model has a specified system of units either directly from the *DUALCESE_MODEL card or inherited from the overall problem input, then these values need to be given in that unit system. Also, note that the formulas here require the temperature be given in either Kelvin or Rankine units.

*DUALCESE_MAT_LIQUID

Purpose: Define the fluid (liquid) properties in a viscous flow for the dual CESE solver. The parameters are used to calculate the viscosity of the liquid. This material model is usually accompanied by a liquid EOS (see *DUALCESE_EOS_NASG). This keyword is only available for single fluid, viscous flows for either no FSI or FSI-IBM.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	A	B			K		
Type	I	F	F			F		
Default	none	1.732E-6	1863			0.579		

VARIABLE	DESCRIPTION
MID	Material ID
A, B	Two coefficients in the Andrade's formula for the liquid's viscosity: $$\mu = A \exp\left(\frac{B}{T}\right)$$ Here T is the absolute temperature of the liquid. See Remark 1. The default values taken from the values for water at 10°C: $A = 1.732 \times 10^{-6}$ N·s/m^2 and $B = 1863$ K.
K	Thermal conductivity of the liquid. The default value is taken from that for water at 10°C: 0.579 W/m·K.

Remarks:

1. **Properties in Andrade's formula.** The values of A and B need to be determined for the liquid by fitting the equation to experimental viscosity data at different temperatures.

*DUALCESE_MESH_GEOMETRY

Purpose: Set general-purpose control variables for the dual CESE compressible flow solver.

Card 1	1	2	3	4	5	6	7	8
Variable	GEOM	IDC	SFNBCKT					
Type	A	F	F					
Default	none	0.25	1.0					
Remarks	1	2						

VARIABLE	DESCRIPTION
GEOM	Sets the geometric dimension: EQ.2D: Two-dimensional (2D) problem EQ.3D: Three-dimensional (3D) problem EQ.AXI: 2D axisymmetric
IDC	Contact interaction detection coefficient (for FSI problems)
SFNBCKT	Scale factor to increase (> 1.0) or decrease (< 1.0) the number of buckets used internally in the dual CESE IBM FSI solver bucket sort search routines

Remarks:

1. **Mesh and boundary conditions for 2D problems.** If you want to use the 2D (GEOM = 2D) or 2D axisymmetric (GEOM=AXI) solver, the mesh should only be distributed in the *xy*-plane with the boundary conditions given only at the *xy* domain boundaries. Otherwise, a warning message will be given, and the 3D solver will be triggered instead.

   The 2D axisymmetric case will work only if the 2D mesh and corresponding boundary conditions are properly defined with the *x* and *y* coordinates corresponding to the radial and axial directions, respectively.

2. **Contact interaction detection coefficient.** IDC is the same type of variable that is input on the *ICFD_CONTROL_FSI card. For an explanation, see Remark 1 for the *ICFD_CONTROL_FSI card.

3. **Dual CESE solver and restarts.** The dual CESE solver supports restarts beginning with release R17.

*DUALCESE_MESH_PART

Purpose: Define dual CESE solver mesh parts, that is, specify the dual CESE FSI algorithm to use for a mesh part. In other words, this keyword allows you to restrict the type of solver (immersed boundary FSI, moving mesh FSI, or non-FSI Eulerian) used on a region of a dual CESE mesh. The part ID specified with this keyword corresponds to the part ID specified on *DUALCESE_ELE2D, *DUALCESE_ELE3D, or *DUALCESE_BLOCK-MESH.

Part Cards. Include one card for each dual CESE part. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	FSITYPE	MMSHID	ICHT				
Type	I	A	I	I				
Default	none	optional	0	0				

VARIABLE	DESCRIPTION
PID	Part ID (must be different from any other *DUALCESE_MESH_-PART part ID). It is important to note that *DUALCESE_PART and *DUALCESE_PART_MULTIPHASE cards should not be used when the newer *DUALCESE_MESH_PART card is used.
FSITYPE	FSI type to use on this part: EQ.<BLANK>: If left blank, no FSI performed. EQ.IBM: Immersed boundary FSI solver EQ.MOVMESH: Moving mesh FSI solver (FSITYPE = MMM may also be used for the same effect)
MMSHID	ID for the mesh motion algorithm to use for the moving mesh FSI solver on this part (region of the current dual CESE mesh). This ID refers to the ID of an instantiation of *DUALCESE_CONTROL_-MESH_MOV.
ICHT	Conjugate heat transfer flag that applies when using the IBM FSI solver: EQ.0: Do not perform conjugate heat transfer when using the IBM FSI solver.

VARIABLE	DESCRIPTION
	EQ.1: Perform conjugate heat transfer when using the IBM FSI solver.

*DUALCESE_MODEL

Purpose: Set the units used by a dual CESE compressible flow problem, along with the name of the file specifying the dual CESE model. There can be any number of such models (each with a separate mesh), and each such model must be in a different file.

Card 1	1	2	3	4	5	6	7	8
Variable	UNITSYS	FILENAME						
Type	A	A						

VARIABLE	DESCRIPTION
UNITSYS	Name of the unit system of this dual CESE model (defined with *UNIT_SYSTEM). EQ.<BLANK>: Use same units as the presumed units of the entire problem.
FILENAME	Filename of the keyword file containing the dual CESE model. Note that only *DUALCESE_... keyword cards are allowed in this file.

*DUALCESE_NODE2D

Purpose: Define a node and its coordinates in the global coordinate system. The nodal point ID must be unique relative to other nodes defined with *DUALCESE_NODE2D or *DUALCESE_NODE3D cards.

Node Cards. Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	NID	X		Y						
Type	I	F		F						
Default	none	0.		0.						

VARIABLE	DESCRIPTION
NID	Node number
X	x coordinate
Y	y coordinate

*DUALCESE_NODE3D

Purpose: Define a node and its coordinates in the global coordinate system. The nodal point ID must be unique relative to other nodes defined with *DUALCESE_NODE3D or *DUALCESE_NODE2D cards.

Node Cards. Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	NID	X		Y		Z				
Type	I	F		F		F				
Default	none	0.		0.		0.				

VARIABLE	DESCRIPTION
NID	Node number
X	*x* coordinate
Y	*y* coordinate
Z	*z* coordinate

*DUALCESE_NODESET

Purpose: Define a nodal set of dual CESE mesh nodes.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID							
Type	I							
Default	none							

Node ID Cards. Set one value per node in the set. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	NID1	NID2	NID3	NID4	NID5	NID6	NID7	NID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
NSID	Set ID of new node set. All dual CESE node sets should have a unique set ID.
NID*i*	Node ID *i*

*DUALCESE_PART

Purpose: Define dual CESE solver parts, that is, specify the dual CESE material and EOS information for a part. It also provides a means to restrict the type of solver used on a region of a dual CESE mesh, meaning an immersed boundary FSI solver, a moving mesh FSI solver, or a non-FSI Eulerian solver can be specified for just this part.

> **WARNING:** As of version R14, this keyword is deprecated. Instead, for the part ID and FSI solver details, use *DUALCESE_MESH_PART. For the material properties, use the *DUALCESE_SOLVER_... card corresponding to the choice made with *DUALCESE_SOLVER_SELECTION.

Part Cards. Include one card for each dual CESE part. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	MID	EOSID	FSITYPE	MMSHID			
Type	I	I	I	A	I			
Default	none	none	none	optional	0			

VARIABLE	DESCRIPTION
PID	Part ID (must be different from any other *DUALCESE_PART, *DUALCESE_PART_MULTIPHASE, *DUALCESE_MESH_PART part ID)
MID	Material ID referring to a *DUALCESE_MAT_… material (see Remark 1)
EOSID	Equation of state ID referring to a *DUALCESE_EOS_… EOS
FSITYPE	FSI type to use on this part: EQ.<BLANK>: If left blank, no FSI performed. EQ.IBM: Immersed boundary FSI solver EQ.MOVMESH: Moving mesh FSI solver (FSITYPE = MMM may also be used for the same effect)

VARIABLE	DESCRIPTION
MMSHID	ID for the mesh motion algorithm to use for the moving mesh FSI solver on this part (region of the current dual CESE mesh). This ID refers to a *DUALCESE_CONTROL_MESH_MOV card ID.

Remarks:

1. **MID for Inviscid Flows.** Since material coefficients are only used in viscous flows, MID can be left blank for inviscid flows.

*DUALCESE_PART_MULTIPHASE

Purpose: Define dual CESE multiphase solver parts, that is, specify the dual CESE material and EOS information for a part. It also provides a means to restrict the type of solver used on a region of a dual CESE mesh, meaning an immersed boundary FSI solver, a moving mesh FSI solver, or a non-FSI Eulerian solver can be specified for just this part.

> **WARNING:** As of version R14, this keyword is deprecated. Instead, for the part ID and FSI solver details, use *DUALCESE_MESH_PART. For the material properties, use the *DUALCESE_SOLVER_... card corresponding to the choice made with *DUALCESE_SOLVER_SELECTION.

Part Cards. Include one card for each dual CESE multiphase solver part. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	REACT_ID	EOSSID	MID	FSITYPE	MMSHID		
Type	I	I	I	I	A	I		
Default	none	none	none	none	optional	0		

VARIABLE	DESCRIPTION
PID	Part ID (must be different from any PID on a *DUALCESE_PART, *DUALCESE_PART_MULTIPHASE, *DUALCESE_MESH_PART card)
REACT_ID	ID of chemical reaction rate model (see *DUALCESE_REACTION_RATE_... cards)
EOSSID	Set ID of multiphase EOS set specification (see *DUALCESE_EOS_SET)
MID	Material ID defined by a *DUALCESE_MAT_… card
FSITYPE	FSI type to use on this part: EQ.<BLANK>: If left blank, no FSI is performed. EQ.IBM: Immersed boundary FSI solver EQ.MOVMESH: Moving mesh FSI solver (FSITYPE = MMM

VARIABLE	DESCRIPTION
	may also be used for the same effect)
MMSHID	ID of the mesh motion algorithm to use for the moving mesh FSI solver on this part (region of the current dual CESE mesh). This ID refers to a *DUALCESE_CONTROL_MESH_MOV card ID.

*DUALCESE_POINTSET

Purpose: Define a list of points used to output at specified sample times variables from the chosen dual CESE solver to binary database binout (see *DUALCESE_DATABASE_HISTORY_POINT_SET) or to a file in Ansys Ensight Gold format (see *DUALCESE_ENSIGHT_DOMAIN).

Card 1	1	2	3	4	5	6	7	8
Variable	PNTSID							
Type	I							
Default	none							

Point Cards. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	X	Y	Z					
Type	F	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
PNTSID	ID for this point set, which can be used by *DUALCESE_DATABASE_HISTORY_POINT_SET ir *DUALCESE_ENSIGHT_DOMAIN
X, Y, Z	Coordinates of a point. As many points as desired can be specified

*DUALCESE_POINT_SOURCE

Purpose: For the dual CESE compressible flow solver, define values at a point source (inside the fluid mesh) for mass flow rate, magnitude of velocity, density, pressure, and temperature.

Card Sets:

A set of data cards for this keyword consists of 3 of the following cards:

1. Card 1 specifies the position and direction where an injected point source is to be applied.
2. Card 2 reads in load curve IDs.
3. Card 3 reads in scale factors.

For each point source to be specified include one set of cards. This input ends at the next keyword ("*") card.

Card Summary:

Card 1. This card is required.

NSID							

Card 2. This card is required.

LC_MRTE	LC_V	LC_RHO	LC_P	LC_T			

Card 3. This card is required.

SF_MRTE	SF_V	SF_RHO	SF_P	SF_T			

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	NSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
NSID	The ID of a *DUALCESE_POINT_SOURCE_STRUCTNODE_SET card to determine where the point source values in this card should be set.

Card 2	1	2	3	4	5	6	7	8
Variable	LC_MRTE	LC_V	LC_RHO	LC_P	LC_T			
Type	I	I	I	I	I			
Default	0	0	0	0	0			
Remarks	1	1	1	1	1			

VARIABLE	DESCRIPTION
LC_MRTE	Load curve ID (see *DEFINE_CURVE) to describe the mass flow rate as a function of time. EQ.0: Mass flow rate is a constant with value SF_MRTE. EQ.-1: Mass flow rate is computed by the solver.
LC_V	Load curve ID to describe the magnitude of the velocity as a function of time. EQ.0: The magnitude of velocity is a constant with value SF_V. EQ.-1: The magnitude of velocity is computed by the solver.

VARIABLE	DESCRIPTION
LC_RHO	Load curve ID to describe the density as a function of time. EQ.0: Density is a constant with value SF_RHO. EQ.-1: Density is computed by the solver.
LC_P	Load curve ID to describe the pressure as a function of time. EQ.0: Pressure is a constant with value SF_P. EQ.-1: Pressure is computed by the solver.
LC_T	Load curve ID to describe the temperature as a function of time. EQ.0: Temperature is a constant with value SF_T. EQ.-1: Temperature is computed by the solver.

Card 3	1	2	3	4	5	6	7	8
Variable	SF_MRTE	SF_V	SF_RHO	SF_P	SF_T			
Type	F	F	F	F	F			
Default	1.0	1.0	1.0	1.0	1.0			

VARIABLE	DESCRIPTION
SF_MRTE	Scale factor for LC_MRTE
SF_V	Scale factor for LC_V
SF_RHO	Scale factor for LC_RHO
SF_P	Scale factor for LC_P
SF_T	Scale factor for LC_T

Remarks:

1. **Consistent flow injection values.** On each point source, the variables (*mrate*, *v*, ρ, *P*, *T*) that are given values must be consistent and make the model well-posed (meaning be such that the solution of the model exists, is unique, and is physical).

*DUALCESE_POINT_SOURCE_TWO-PHASE

Purpose: For the dual CESE compressible two-phase multiphase flow solver, this keyword specifies values at a point source (inside the fluid mesh) for the mass flow rate, volume fraction of material 1, magnitude of the velocity, density of material 1, density of material 2, pressure, and temperature.

Card Sets:

A set of data cards for this keyword consists of 3 of the following cards:

1. Card 1 specifies the position and direction where an injected point source is to be applied.
2. Card 2 reads in load curve IDs.
3. Card 3 reads in scale factors.

For each point source to be specified, include one set of cards. This input ends at the next keyword ("*") card.

Card Summary:

Card 1. This card is required.

NSID							

Card 2. This card is required.

LC_MRTE	LC_Z1	LC_V	LC_D1	LC_D2	LC_P	LC_T	

Card 3. This card is required.

SF_MRTE	SF_Z1	SF_V	SF_D1	SF_D2	SF_P	SF_T	

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	NSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
NSID	ID of a *DUALCESE_POINT_SOURCE_STRUCTNODE_SET where the point source values in this card should be set.

Card 2	1	2	3	4	5	6	7	8
Variable	LC_MRTE	LC_Z1	LC_V	LC_D1	LC_D2	LC_P	LC_T	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	
Remarks	1	1	1	1	1	1	1	

VARIABLE	DESCRIPTION
LC_MRTE	Load curve ID (see *DEFINE_CURVE) to describe the mass flow rate as a function of time. EQ.0: Constant mass flow rate with value SF_MRTE EQ.-1: The solver computes the mass flow rate.
LC_Z1	Load curve ID to describe the magnitude of the volume fraction of material 1 as a function of time. EQ.0: Constant volume fraction for material 1 with value SF_Z1 EQ.-1: The solver computes the volume fraction of material 1.

VARIABLE	DESCRIPTION
LC_V	Load curve ID to describe the magnitude of the velocity as a function of time. EQ.0: Constant magnitude of velocity with value SF_V EQ.-1: The solver computes the magnitude of velocity.
LC_D1	Load curve ID to describe the density of material 1 as a function of time. EQ.0: Constant density for material 1 with value SF_D1 EQ.-1: The solver computes the density of material 1.
LC_D2	Load curve ID to describe the density of material 2 as a function of time. EQ.0: Constant density for material 2 with value SF_D2 EQ.-1: The solver computes the density of material 2.
LC_P	Load curve ID to describe the pressure as a function of time. EQ.0: Constant pressure is a constant with value SF_P EQ.-1: The solver computes the pressure.
LC_T	Load curve ID to describe the temperature as a function of time. EQ.0: Constant temperature with value SF_T EQ.-1: The solver computes the temperature.

Card 3	1	2	3	4	5	6	7	8
Variable	SF_MRTE	SF_Z1	SF_V	SF_D1	SF_D2	SF_P	SF_T	
Type	F	F	F	F	F	F	F	
Default	1.0	1.0	1.0	1.0	1.0	1.0	1.0	

VARIABLE	DESCRIPTION
SF_MRTE	Scale factor for LC_MRTE
SF_Z1	Scale factor for LC_Z1

VARIABLE	DESCRIPTION
SF_V	Scale factor for LC_V
SF_D1	Scale factor for LC_D1
SF_D2	Scale factor for LC_D2
SF_P	Scale factor for LC_P
SF_T	Scale factor for LC_T

Remarks:

1. **Consistent flow injection values.** On each point source, the variables (*mrate*, $z1, v, \rho 1, \rho 2, P, T$) that are given values must be consistent and make the model well-posed (meaning be such that the solution of the model exists, is unique, and is physical).

*DUALCESE_POINT_SOURCE_STRUCTNODE_SET

Purpose: Provide the position, direction, and size (area of the orifice) of a flow to be injected into a background flow field.

Node Set ID Card.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID							
Type	I							
Default	none							

Orifice Definition Card. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	NID	VID	ORIFA					
Type	I	I	F					
Default	none	none	1.0					

VARIABLE	DESCRIPTION
NSID	ID of this set of orifice definitions
NID	Structural node that gives the position of the orifice
VID	ID of a vector defined by *DEFINE_VECTOR that gives the direction of the flow away from the position defined by NID
ORIFA	Surface area of the orifice

*DUALCESE_POROUS_MODEL

Purpose: Specify the method for treating porous media flow in porous media fluid-structure interaction. Although the literature contains several methods, currently only the jump condition method is available. Other methods may be added in the future.

Card 1	1	2	3	4	5	6	7	8
Variable	METHOD							
Type	C							
Default	JUMP							

VARIABLE	DESCRIPTION
METHOD	Method for modeling the porous-media interface: EQ.JUMP: Jump method (default). See *DUALCESE_POROUS_SPECIFY_JUMP.

Remarks:

1. **Availability.** The porous media flow capability is available for single-fluid FSI-IBM and two-phase FSI-IBM solvers including for viscous and inviscid flows.

*DUALCESE_POROUS_SPECIFY_JUMP

Purpose: Specify the properties of the porous media used in the jump method (see *DUALCESE_POROUS_MODEL). These properties include the structural part or part set of the porous medium, the thickness of the medium, and the model for mimicking the porosity of the materials.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	THICKNSS	EQID					
Type	I	F	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
PID	Structural part ID of the porous medium
THICKNSS	Thickness of the medium
EQID	ID of the equation used to model the porous medium (see *DUALCESE_DARCY-FORCHHEIMER_EQ)

*DUALCESE_REACTION_RATE_IG

Purpose: Define a reaction rate law for the Ignition and Growth model to describe the conversion of reactants to products in the modeling of a condensed phase explosive in a dual CESE multiphase model. See Michael and Nikiforakis 2016 and Tarver 2005 for details about this law.

Card 1	1	2	3	4	5	6	7	8
Variable	REACT_ID	IGN	AA	BB	XX	GROW1	CC	DD
Type	I	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Card 2	1	2	3	4	5	6	7	8
Variable	YY	GROW2	EE	GG	ZZ	IGMAX	G1MAX	G2MAX
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
REACT_ID	ID of reaction rate law
IGN	Reaction ignition term parameter, I
AA	Reaction ignition term parameter, a
BB	Reaction ignition term parameter, b
XX	Reaction ignition term parameter, x
GROW1	Reaction growth term parameter, G_1
CC	Reaction growth term parameter, c
DD	Reaction growth term parameter, d
YY	Reaction growth term parameter, y

VARIABLE	DESCRIPTION
GROW2	Reaction completion term parameter, G_2
EE	Reaction completion term parameter, e
GG	Reaction completion term parameter, g
ZZ	Reaction completion term parameter, z
IGMAX	Maximum mass fraction of the product for reaction ignition term, Φ_{IGmax}
G1MAX	Maximum mass fraction of the product for reaction growth term, Φ_{G1max}
G2MAX	Maximum Φ for reaction completion term, Φ_{G2max}

Remarks:

Using the notation of Michael and Nikiforakis 2016, the reaction rate law can be given as:

$$\frac{d\Phi}{dt} = I(1-\Phi)^b(\rho - 1 - a)^x H(\Phi_{\text{IGmax}} - \Phi) + G_1(1-\Phi)^c\Phi^d p^y H(\Phi_{\text{G1max}} - \Phi) + G_2(1-\Phi)^e\Phi^g p^z H(\Phi - \Phi_{\text{G2max}})$$

where H is the Heaviside function. Here Φ is the mass fraction of the products, p is the pressure, and ρ is the density of the explosive mixture. Note that the pressure is assumed to be in equilibrium between the phases. I, G_1, G_2, a, b, c, d, e, g, x, y, and z are constants. They depend on the explosive as well as the part of the detonation process being modeled; see Michael and Nikiforakis 2016 for details.

The constants Φ_{IGmax}, Φ_{G1max}, and Φ_{G2max} in the Heavyside functions indicate when each of the three stages of the reaction are dominant. It was developed by Tarver 2005 to describe the reaction of pressed solid explosives during shock initiation and detonation. Tarver 2005 developed this reaction law for pressed solid explosives to describe the observed reaction stages during shock initiation and detonation See Tarver 2005 for a description of each stage of the reaction.

*DUALCESE_REACTION_RATE_IG_REDUCED

Purpose: Define a reduced form reaction rate law compared to that defined with *DUALCESE_REACTION_RATE_IG for describing the conversion of reactants to products in the modeling of a condensed phase explosive in a dual CESE multiphase model. See Michael and Nikiforakis 2016 for details about this reaction rate law.

Card 1	1	2	3	4	5	6	7	8
Variable	REACT_ID	GROW1	CC	DD	YY	PHI0		
Type	I	F	F	F	F	F		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
REACT_ID	ID of reaction rate law
GROW1	Reaction growth term parameter, G_1
CC	Reaction growth term parameter, c
DD	Reaction growth term parameter, d
YY	Reaction growth term parameter, y
PHI0	Additional parameter to account for the non-zero amount of reaction when the mass fraction of the products, ϕ, is zero

Remarks:

Michael and Nikiforakis 2016 simplified the Ignition and Growth model reaction law to a pressure dependent law with a single stage to make the reduced model. They excluded the first and third terms from the Ignition and Growth model to makes this law. However, to account for a finite amount of reaction from ignition when ϕ is zero, they added a constant ϕ_0. This reduced law has the form:

$$\frac{d\phi}{dt} = G_1(1-\phi)^c(\phi+\phi_0)^d p^y \quad .$$

All the parameters are the same as the Ignition and Growth model except Φ_0. See *DUALCESE_REACTION_RATE_IG and Michael and Nikiforakis 2016 for details.

*DUALCESE_REACTION_RATE_P_DEPEND

Purpose: Define an explicitly pressure-dependent reaction rate law for describing the conversion of reactants to products in the modeling of a condensed phase explosive in a dual CESE multiphase model. This law is from Banks et al 2008.

Card 1	1	2	3	4	5	6	7	8
Variable	REACT_ID	SIGMA	NU	N				
Type	I	F	F	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
REACT_ID	ID of reaction rate law
SIGMA	A positive constant, σ
NU	A positive constant, ν
N	A positive constant, n

Remarks:

Using the notation of Michael and Nikiforakis 2016, this simple pressure dependent reaction rate law by Banks et al 2008 can be stated as:

$$\frac{d\phi}{dt} = \sigma \phi^{\nu} p^{n}$$

Here ϕ is the mass fraction of the products, p is the pressure, and σ, ν, and n are positive constants found from experimental data. As discussed by Banks et al 2008, this type of reaction rate law that depends explicitly on pressure is often preferred for modeling the detonation of solid explosives because the constants can be found by fitting experimental data and other physical variables are difficult to measure.

*DUALCESE_SEGMENTSET

Purpose: Define a set of segments. For three-dimensional geometries, a segment can be triangular or quadrilateral. For two-dimensional geometries, a segment is a line defined by two nodes.

Card 1	1	2	3	4	5	6	7	8
Variable	SID							
Type	I							
Default	none							

Segment Cards. For each segment in the set include one card of this format. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4				
Type	I	I	I	I				

VARIABLE	DESCRIPTION
SID	Set ID. All segment sets should have a unique set ID.
N1	Nodal point n_1
N2	Nodal point n_2
N3	Nodal point n_3. To define a line segment, set N3 = N2.
N4	Nodal point n_4. To define a triangular segment, set N4 = N3. To define a line segment, set N4 = N2.

Table 5-3. Face definitions for volume dual CESE elements

FACE	Hexahedron	Pentahedron	Pyramid	Tetrahedron
1	N1, N5, N8, N4	N1, N2, N5	N1, N4, N3, N2	N1, N2, N4
2	N2, N3, N7, N6	N4, N6, N3	N1, N2, N5	N2, N3, N4

FACE	Hexahedron	Pentahedron	Pyramid	Tetrahedron
3	N1, N2, N6, N5	N1, N4, N3, N2	N2, N3, N5	N1, N3, N2
4	N4, N8, N7, N3	N2, N3, N6, N5	N3, N4, N5	N1, N4, N3
5	N1, N4, N3, N2	N1, N5, N6, N4	N4, N1, N5	
6	N5, N6, N7, N8			

*DUALCESE_SOLVER_SELECTION

Purpose: Set general-purpose control variables for the dual CESE compressible flow solver.

Card 1	1	2	3	4	5	6	7	8
Variable	EQNS	SUBCYC	LCSBCYC	MXSBCYC	TOLFSI	LCTOLFSI		
Type	A	I	I	I	F	I		
Default	EULER	0	optional	20	0.001	optional		
Remark	1	2						

VARIABLE **DESCRIPTION**

EQNS — Select the equations being solved with the dual CESE solver:

- EQ.CAVITATION: Cavitation solver
- EQ.EULER: Euler equations
- EQ.HYBRID: Hybrid multiphase model
- EQ.NS: Navier-Stokes equations
- EQ.PHASE-CHNG: Phase change model
- EQ.TWO-PHASE: Two-phase multiphase model
- EQ.CHEM_REACT: Chemically-reacting flow solver

SUBCYC — Subcycling flag:

- EQ.0: Do not do FSI subcycling against the structural solver.
- EQ.1: Do FSI subcycling against the structural solver.

LCSBCYC — Optional load curve for adjusting subcycling frequency as a function of time

MXSBCYC — Maximum number of subcycling steps (structural steps) before the fluid solver must update

TOLFSI — Error tolerance threshold that will stop further subcycling steps. The maximum allowed single-time-step structural displacement is this tolerance multiplied by the maximum element length in the

VARIABLE	DESCRIPTION
	dual CESE mesh. If LCTOLFSI is used, TOLFSI acts as a scale factor on the load curve.
LCTOLFSI	Optional load curve to adjust TOLFSI as a function of time

Remarks:

1. **Dual CESE solver and restarts.** The dual CESE solver supports restarts beginning with release R17.

2. **Subcycling.** In FSI, the structural and dual CESE solvers are loosely coupled. If subcycling is not enabled, then the next time step for the solvers is the same. Before each step, each solver estimates its time step and the smaller value is used by both solvers. With subcycling enabled, the estimated time steps are compared. If the structural solver's time step is ast least two times smaller than the dual CESE solver's, the structural solver subcycles, meaning it can perform at least one more time step update before the dual CESE solver updates. The number of structural subcycling steps is limited by MXSBCYC or LCSBCYC. Additionally, if the error tolerance set with TOLFSI/LCTOLFSI is exceeded, then the subcycling stops and the dual CESE solver updates.

*DUALCESE_SOLVER_CAV_EQNS

Purpose: Set the properties of the cavitating flow mixture being solved with the cavitation solver.

Property Card. Include one card.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
EOSID	Equation of state (EOS) ID that must refer to an EOS defined with *DUALCESE_EOS_CAV_HOMOG_EQUILIB

*DUALCESE_SOLVER_CHEMICALLY_REACTING

Purpose: Select the options that apply to the chemically-reacting flow solver. This solver uses a Chemkin chemistry model input with the *CHEMISTRY keywords. The type of flow physics can be either the Navier-Stokes equations or the Euler equations. This solver can study the flow field due to both mixing and chemical reactions or just due to mixing.

Property Card. Include one card.

Card 1	1	2	3	4	5	6	7	8
Variable	ID_CHEM	FLOWTYP	MIX_FLAG					
Type	I	A	I					
Default	none	NS	0					

VARIABLE	DESCRIPTION
ID_CHEM	ID of the *CHEMISTRY_CONTROL_FULL card; this is the only *CHEMISTRY solver currently available for use with the *DUALCESE chemically-reacting flow solver. Also, this ID is the same as the chemistry model ID of the *CHEMISTRY_MODEL card that selects the Chemkin-compatible input model to be used. This chemistry model provides species information including chemical reactions, thermodynamics information, and species transport properties.
FLOWTYP	Flow physics: EQ.NS: Navier-Stokes solver (default) EQ.EULER: Euler equations solver
MIX_FLAG	Flag for determining whether only mixing is studied: EQ.0: Perform the full calculation (chemical reactions and mixing). EQ.1: Perform only the mixing (transport) aspect of the calculation, meaning that the chemical reactions are ignored (not computed).

*DUALCESE_SOLVER_EULER_EQNS

Purpose: Set the properties of the gas flow being solved with the Euler equations solver.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
EOSID	Equation of state (EOS) ID that refers to an EOS defined by one of the following keywords: *DUALCESE_EOS_IDEAL_GAS, *DUALCESE_EOS_INFLATOR1, or *DUALCESE_EOS_INFLATOR2.

*DUALCESE_SOLVER_HYBRID_MULTIPHASE

Purpose: Define the properties of the fluid and condensed phase materials via equations-of-state that are used together to compute the thermodynamic state in the hybrid multi-phase dual CESE solver. Also, specify the single-step reaction rate model for the chemical reactions.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSINID	EOSRCTID	EOSPRDID	REACT_ID				
Type	I	I	I	I				
Default	none	none	none	none				
Remark	1	1	1	2				

VARIABLE	DESCRIPTION
EOSINID	EOS ID of the inert component of the multiphase mixture
EOSRCTID	EOS ID of the reactant phase of the multiphase mixture
EOSPRDID	EOS ID of the product phase of the multiphase mixture
REACT_ID	ID of chemical reaction rate model

Remarks:

1. **Supported EOS Types.** Each EOS can be one of the following types:

   *DUALCESE_EOS_IDEAL_GAS

   *DUALCESE_EOS_STIFFENED_GAS

   *DUALCESE_EOS_VAN_DER_WAALS_GENERALIZED

   *DUALCESE_EOS_COCHRAN_CHAN

   *DUALCESE_EOS_JWL.

2. **Supported Reaction Rate Types.** The reaction rate can be one of the following types:

*DUALCESE_REACTION_REAT_IG

*DUALCESE_REACTION_RATE_IG_REDUCED

*DUALCESE_REACTION_RATE_P_DEPEND

*DUALCESE_SOLVER_NAVIER_STOKES

Purpose: Set the properties of the fluid flow being solved with the Navier-Stokes equations solver. That is, specify the dual CESE material and EOS information to be used.

Property Card. Include one card.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	MID						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
EOSID	Equation of state (EOS) ID referring to an EOS defined by one of the following card types: *DUALCESE_EOS_IDEAL_GAS, *DUALCESE_EOS_INFLATOR1, or *DUALCESE_EOS_INFLATOR2.
MID	Material ID referring to a *DUALCESE_MAT_… material card (see Remark 1)

Remarks:

1. **MID for Inviscid Flows.** Since material coefficients are only used in viscous flows, MID can be left blank for inviscid flows.

*DUALCESE_SOLVER_PHASE_CHANGE

Purpose: Define the properties of the fluid and condensed phase materials via equations of state that are used together to compute the thermodynamic state in the phase change multiphase dual CESE solver.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSIDL	EOSIDV	EOSIDO					
Type	I	I	I					
Default	none	none	none					
Remarks	1	1	1					

VARIABLE	DESCRIPTION
EOSIDL	Equation of state (EOS) ID of the liquid component in the multiphase mixture
EOSIDV	Equation of state (EOS) ID of the vapor component in the multiphase mixture
EOSIDO	Equation of state (EOS) ID of the other gases in the multiphase mixture

Remarks:

1. **Available EOS models.** Each EOS can be one of the following types:

   *DUALCESE_EOS_NASG

*DUALCESE_SOLVER_TWO-PHASE_MULTIPHASE

Purpose: Define the properties of the fluid phase materials through equations-of-state that are used together to compute the thermodynamic state in the two-phase multiphase dual CESE solver. This solver does not compute phase changes.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID1	EOSID2						
Type	I	I						
Default	none	none						
Remarks	1	1						

VARIABLE	DESCRIPTION
EOSID1	EOS ID of the first inert component of the multiphase mixture
EOSID2	EOS ID of the second inert component of the multiphase mixture

Remarks:

1. **Supported EOS Types.** Each EOS can be one of the following types:

   *DUALCESE_EOS_IDEAL_GAS

   *DUALCESE_EOS_STIFFENED_GAS

   *DUALCESE_EOS_VAN_DER_WAALS_GENERALIZED

   *DUALCESE_EOS_COCHRAN_CHAN

   *DUALCESE_EOS_JWL

*EM

The *EM keyword cards provide input for the electromagnetism module. This module is for solving 3D eddy-current, inductive heating or resistive heating problems. It can be coupled with the mechanical and thermal solvers. Typical applications include magnetic metal forming and welding. This module also includes coupling the boundary element method to the finite element method coupling. We intend this coupling for simulations involving a conductor interacting with air so that the air does not need to be meshed. The conductor is modeled with finite elements while the air is modeled with boundary elements.

*EM_2DAXI

*EM_BOUNDARY

*EM_BOUNDARY_PRESCRIBED

*EM_CIRCUIT

*EM_CIRCUIT_CONNECT

*EM_CIRCUIT_ROGO

*EM_CIRCUIT_SOURCE

*EM_CONTACT

*EM_CONTACT_RESISTANCE

*EM_CONTACT_SUBDOM

*EM_CONTROL

*EM_CONTROL_CONTACT

*EM_CONTROL_COUPLING

*EM_CONTROL_COUPLING_P2D

*EM_CONTROL_EROSION

*EM_CONTROL_EP

*EM_CONTROL_MAGNET

*EM_CONTROL_SOLUTION

*EM_CONTROL_SWITCH

*EM_CONTROL_SWITCH_CONTACT

*EM_CONTROL_TIMESTEP

*EM_DATABASE_CIRCUIT

*EM_DATABASE_CIRCUIT0D

*EM_DATABASE_CIRCUITSOURCE

*EM_DATABASE_ELOUT

*EM_DATABASE_FIELDLINE

*EM_DATABASE_GLOBALENERGY

*EM_DATABASE_NODOUT

*EM_DATABASE_PARTDATA

*EM_DATABASE_POINTOUT

*EM_DATABASE_ROGO

*EM_DATABASE_TIMESTEP

*EM_EOS_BURGESS

*EM_EOS_MEADON

*EM_EOS_PERMEABILITY

*EM_EOS_TABULATED1

*EM_EOS_TABULATED2

*EM_EP_CELLMODEL_DEFINE_FUNCTION

*EM_EP_CELLMODEL_FENTONKARMA

*EM_EP_CELLMODEL_FIZHUGHNAGUMO

*EM_EP_CELLMODEL_STEP

*EM_EP_CELLMODEL_TENTUSSCHER

*EM_EP_CELLMODEL_TOMEK

*EM_EP_CELLMODEL_TOR_ORD

*EM_EP_CELLMODEL_TROVATO

*EM_EP_CELLMODEL_USERMAT

*EM_EP_CREATEFIBERORIENTATION

*EM_EP_ECG

*EM_EP_EIKONAL

*EM_EP_ISOCH

*EM_EP_LAPLACE_DIRICHLET

*EM_EP_PURKINJE_NETWORK

*EM_EP_TENTUSSCHER_STIMULUS

*EM_EP_TENTUSSCHER_STIMULUS2

*EM_EXTERNAL_CHARGE

*EM_EXTERNAL_ECVEL

*EM_EXTERNAL_FIELD

*EM_ISOPOTENTIAL

*EM_ISOPOTENTIAL_CONNECT

*EM_ISOPOTENTIAL_ROGO

*EM_MAT_001

*EM_MAT_002

*EM_MAT_003

*EM_MAT_004

*EM_MAT_005

*EM_MAT_006

*EM_MAT_007

*EM_OUTPUT

*EM_OUTPUT_BINARY

*EM_OUTPUT_FORCES

*EM_OUTPUT_VTK

*EM_P2D_BATMAC

*EM_P2D_MESHLESS

*EM_PERMANENT_MAGNET

*EM_POINT_SET

*EM_RANDLES_BATMAC

*EM_RANDLES_EXOTHERMIC_REACTION

*EM_RANDLES_MESHLESS

*EM_RANDLES_TSHELL

*EM_RANDLES_SHORT

*EM_RANDLES_SOLID

*EM_ROTATION_AXIS

*EM_SOLVER_BEM

*EM_SOLVER_BEM_REASSEMBLY

*EM_SOLVER_BEMMAT

*EM_SOLVER_FEM

*EM_SOLVER_FEMBEM

*EM_SOLVER_FEMBEM_MONOLITHIC

*EM_2DAXI

Purpose: Sets up the electromagnetism solver as 2D axisymmetric instead of 3D, on a given part, in order to save computational time as well as memory.

The electromagnetism is solved in 2D on a given cross section of the part (defined by a segment set), with a symmetry axis defined by its direction (at this time, it can be the x, y, or z axis). The EM forces and Joule heating are then computed over the full 3D part by rotations. The part needs to be compatible with the symmetry, i.e. each node in the part needs to be the child of a parent node on the segment set, by a rotation around the axis. Only the conductor parts (with a *EM_MAT_… of type 2 or 4) should be defined as 2D axisymmetric.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	SSID			STARSSID	ENDSSID	NUMSEC	
Type	I	I			I	I	I	
Default	none	none			none	none	none	

VARIABLE	DESCRIPTION
PID	Part ID of the part to be solved using 2D axisymmetry
SSID	Segment Set ID : Segment that will define the 2D cross section of the part where the EM field is solved
STARSSID, ENDSSID	Used by the 2D axisymmetric solver to make the connection between two corresponding boundaries on each side of a slice when the model is a slice of the full 360 circle.
NUMSEC	Number of Sectors. This field gives the ratio of the full circle to the angular extension of the mesh. This has to be a power of two. For example, NUMSEC = 4 means that the mesh of the part represents one fourth of the total circle. If this value is set to 0, then the value from *EM_ROTATION_AXIS is used instead.

Remarks:

1. At this time, *either* all or none of the conductor parts should be 2D axisymmetric. In the future, a mix between 2D axisymmetric and 3D parts will be allowed.

*EM_BOUNDARY

Purpose: Define some boundary conditions for the electromagnetism problems.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	BTYPE						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
SSID	Segment Set Id
BTYPE	EQ.9: The faces of this segment set are eliminated from the BEM calculations (used for example for the rear or side faces of a workpiece).

*EM_BOUNDARY_PRESCRIBED

Purpose: Prescribe a local boundary condition applied on nodes.

Card 1	1	2	3	4	5	6	7	8
Variable	BPID	BPTYPE	SETTYPE	SETID	VAL	LCID		SYSTYPE
Type	I	I	I	I	F	I		I
Default	none	none	none	none	0.	0		0

Optional Card.

Card 2	1	2	3	4	5	6	7	8
Variable	BIRTHT	DEATHT			VAL2	LCID2		
Type	F	F			F	I		
Default	0.	1.e28			0.	0		

VARIABLE	DESCRIPTION
ISOID	ID of the prescribed boundary
BPTYPE	Prescribed boundary type: EQ.1: Short (scalar potential set to 0.) EQ.2: Prescribed resistance (Robin B.C.) EQ.3: Prescribed scalar potential (Dirichlet B.C.) EQ.4: Prescribed current density (Neumann B.C.)
SETTYPE	Set type: EQ.1: Segment set EQ.2: Node set EQ.3: Fluid part (see *ICFD_PART)
SETID	Set ID

VARIABLE	DESCRIPTION
VAL	Value of the resistance, current density or potential depending on BPTYPE. Ignored if LCID is defined.
LCID	Load curve ID defining the value of the resistance, voltage, or current as a function of time. If a negative value is entered, a *DEFINE_FUNCTION will be expected. The following parameters are allowed: f(time, emdt, curr, pot, cond, temp, potglob, currglob, areaglob, area, x, y, z). Pot/curr/area and potglob/currglob/areaglob are the local value of the scalar potential/current/area and the global averaged value on the prescribed boundary, respectfully. Cond is the local electrical conductivity, and x, y, and z are the local coordinates.
SYSTYPE	Flag for the type of system on which the boundary condition is applied (applies only for cardiac electrophysiology when *EM_BOUNDARY_PRESCRIBED is used with EMSOL = 11 or 12): EQ.0: Applied on extracellular potential EQ.1: Applied on transmembrane potential
BIRTHT / DEATHT	Birth and death times for that prescribed boundary
VAL2	Value of the impedance in radiofrequency problems to be used in conjunction with BPTYE = 2.
LCID2	Load curve ID for defining the impedance value as a function of time. It is only available in radiofrequency problems with BPTYPE = 2.

Remarks:

1. **Supported solvers.** This keyword is currently only available for the resistive heating solver (EMSOL = 3) and the electrophysiology solvers (EMSOL = 11, 12, and 13).

*EM_CIRCUIT

Purpose: Define an electrical circuit.

The solver type, set with EMSOL of *EM_CONTROL, determines how the current density in a circuit is modeled. For example, for an eddy current solver, the diffusion of the current in the circuit is considered. To model a circuit with uniform current through its cross-section, see *EM_CIRCUIT_SOURCE.

Card 1	1	2	3	4	5	6	7	8
Variable	CIRCID	CIRCTYP	LCID	R/F	L/A	C/T0	V0	T0
Type	I	I	I	F	F	F	F	F
Default	none	none	none	none	none	none	none	0.

Card 2	1	2	3	4	5	6	7	8
Variable	SIDCURR	SIDVIN	SIDVOUT	PID				
Type	I	I	I	I				
Default	none	none	none	none				

Optional Card. This card is only read if CIRCTYP = 30.

Card 3	1	2	3	4	5	6	7	8
Variable	VCUT	R3	L3	R2				
Type	F	F	F	F				
Default	0.	0.	0.	0.				

VARIABLE	DESCRIPTION
CIRCID	Circuit ID

VARIABLE	DESCRIPTION
CIRCTYP	Circuit type : EQ.1: Imposed current as a function of time set by a load curve. EQ.2: Imposed voltage as a function of time specified by a load curve. If a negative value is entered for LCID, its absolute value will refer to a DEFINE FUNCTION for a user-defined circuit equation. If a DEFINE_FUNCTION is used, the following parameters are accepted: f(time, emdt, curr, curr1, curr2, pot1, pot2). emdt is the current timestep; curr, curr1, and curr2 refer to the current value at t, $t-1$, and $t-2$, respectfully; and pot1 and pot2 refer to the scalar potential at $t-1$ and $t-2$, respectfully. EQ.3: R, L, C, V0 circuit. EQ.11: Imposed current defined by an amplitude A, frequency F and initial time t_0: $I = A\sin[2\pi F(t-t_0)]$ EQ.12: Imposed voltage defined by an amplitude A, frequency F and initial time t_0: $V = A\sin[2\pi F(t-t_0)]$ EQ.21: Imposed current defined by a load curve over one period and a frequency F. See Remark 3. EQ.22: Imposed voltage defined by a load curve over one period and a frequency F. See Remark 3. EQ.30: Modified version of R, L, C, V0 circuit that includes a diode. See Figure 6-1.
LCID	Load curve ID for CIRCTYP = 1, 2, 21, and 22
R/F	Value of the circuit resistance for CIRCTYP = 3 and 30. For CIRCTYP = 30, this part of the circuit is labeled R1 in Figure 6-1. Value of the frequency for CIRCTYP = 11, 12, 21, and 22. For CIRCTYP = 11 and 12, to have the frequency specified as a function of time with a load curve, a negative value can be entered with the absolute value corresponding to the load curve ID.
L/A	Value of the circuit inductance for CIRCTYP = 3 and 30. For CIRCTYP = 30, this part of the circuit is labeled L1 in Figure 6-1. Value of the amplitude for CIRCTYP = 11 and 12. To have the amplitude specified as a function of time with a load curve, a negative value can be entered with the absolute value corresponding to the load curve ID.
C/T0	Value of the circuit capacity for CIRCTYP = 3 and 30. For CIRC-

VARIABLE	DESCRIPTION
	TYP = 30, this part of the circuit is labeled C1 in Figure 6-1. Value of the initial time t_0 for CIRCTYP = 11 and 12. For the frequency domain solvers (EMSOL = 4 and 9 on *EM_CONTROL), T0 specifies the phase time shift for CIRCTYP = 1 and 2.
V0	Value of the circuit initial voltage for CIRCTYP = 3 and 30.
T0	Starting time for CIRCTYP = 3. The default is the beginning of the run.
SIDCURR	Segment set ID for the current. It uses the orientation given by the normal of the segments. To use the opposite orientation, use a '–' (minus) sign in front of the segment set ID. CIRCTYP.EQ.1/11/21: The current is imposed through this segment set. CIRCTYP.EQ.3/30: The current needed by the circuit equations is measured through this segment set. CIRCTYP.EQ.2/12/22: Optional segment set that the current is forced through. See Remark 2.
SIDVIN	Segment set ID for input voltage when CIRCTYP = 2, 3, 30, 12, or 22 or for input current when CIRCTYP = 1, 11, or 21. The input voltage or current is oriented to enter the structural mesh, irrespective of the orientation of the segments.
SIDVOUT	Segment set ID for output voltage when CIRCTYP = 2, 3, 30, 12, or 22 or for output current when CIRCTYP = 1, 11, or 21. The output voltage or current is oriented to leave the structural mesh, irrespective of the orientation of the segment.
PID	Part ID associated with the circuit. It is only mandatory when including the second line of *EM_ROTATION_AXIS for the EM 2D axisymmetric solver.
VCUT	Cut off voltage for the diode, allowing the current to flow or not. Setting this field to a very large negative value effectively means that the diode circuit branch is never triggered, causing CIRCTYP = 30 to revert to an R, L, C circuit with behavior similar to CIRCTYP = 3. See Figure 6-1.
R3/L3/R2	Optional circuit resistance and inductance values for the different branches for CIRCTYP = 30. See Figure 6-1.

Table 6-1. Correspondence between circuit type and card entries. "M" indicates mandatory, "M*" mandatory with exceptions (see Remark 1), "O" optional, and "-" ignored.

	Circuit type (CIRCTYP)									
Variable	1:	Imposed current	2:	Imposed voltage	3:	R, L, C	11:	F, A, t0	12:	F, A, t0
LCID		M		M		-		-		-
R/L/C/V0		-		-		M		-		-
F		-		-		-		M		M
A/T0		-		-		-		M		M
SIDCURR		M		O		M		M		O
SIDVIN		M*		M		M		M*		M
SIDVOUT		M*		M		M		M*		M

	Circuit type (CIRCTYP)					
Variable	21:	LCID, F	22:	LCID, F	30:	R, L, C with diode
LCID		M		M		-
R/L/C/V0		-		-		M
F		M		M		-
A/T0		-		-		-
SIDCURR		M		O		M
SIDVIN		M*		M		M
SIDVOUT		M*		M		M

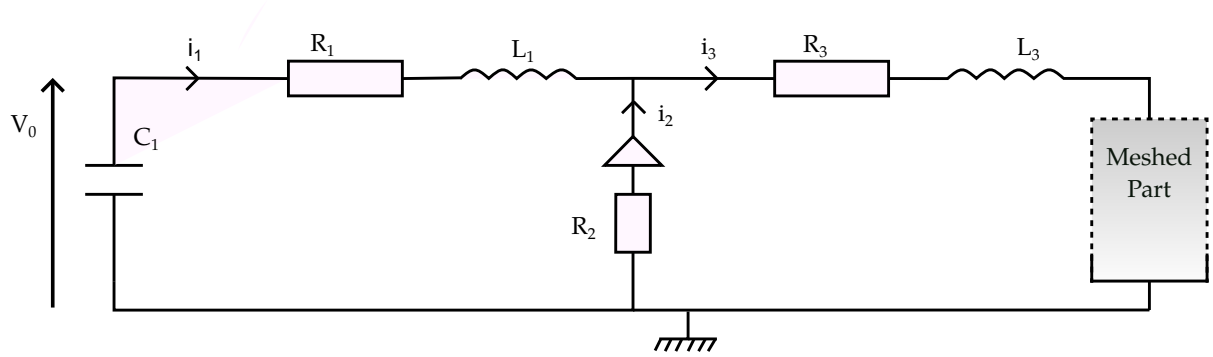


Figure 6-1. CIRCTYP = 30 circuit sketch

Remarks:

1. **Imposed current with closed loop geometry.** For a circuit with an imposed current (CIRCTYP 1, 11, or 21) in a closed loop geometry (torus), SIDVIN and SIDVOUT cannot be specified. Thus, only SIDCURR is necessary.

2. **SIDCURR with imposed tension.** For a circuit with an imposed tension (CIRCTYP 2, 12, 22), it is also possible to set SIDCURR. This can be useful in circuits where various flow paths are possible for the current to force the entire current to go through SIDCURR.

3. **CIRCTYP = 21 and 22.** Circuit types 21 and 22 are for cases where the periodic current/tension does not exactly follow a perfect sinusoidal. You must provide the frequency and the shape of the current/tension over one period through a load curve.

4. **EMSOL = 4 and 9.** Only CIRCTYP = 1 and 2 are available for the frequency-based current solvers. The values in the load curve represent the amplitude peak values. A time shift can optionally be defined using T0.

5. **CIRCTYP = 30.** This circuit type is only available for the Eddy current solver. When used, the output includes the current and voltage values of the different branches.

*EM_CIRCUIT_CONNECT

Purpose: Connect several circuits together by imposing a linear constraint on the global currents of circuit pairs.

$$c_1 i_1 + c_2 i_2 = 0.$$

This feature is especially useful for 2D axisymmetric models involving spiral or helical coils.

Card 1	1	2	3	4	5	6	7	8
Variable	CONID	CONTYPE	CIRC1	CIRC2	C1	C2		
Type	I	I	I	I	F	F		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
CONID	ID of the circuit connect
CONTYPE	Connection type between circuits: EQ.1: If CIRC2 already has a global condition on its current, i_2, (see *EM_CIRCUIT), that global condition takes precedence over the constraint defined by the circuit connect. EQ.2: The condition defined by circuit connect takes precedence in the case of conflicting constraints (for example CIRC2 also has an imposed current defined by *EM_CIRCUIT).
CIRC1	ID of the first circuit
CIRC2	ID of the second circuit
C1/C2	Values of the linear constraints

*EM_CIRCUIT_ROGO

Purpose: Define Rogowsky coils to measure a global current versus time through a segment set or a node set.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	ROGID	SETID	SETTYPE	CURTYP	PID			
Type	I	I	I	I	I			
Default	0	0	0	0	0			

VARIABLE	DESCRIPTION
ROGID	Rogowsky coil ID
SETID	Segment or node set ID
SETTYPE	Type of set: EQ.1: Segment set EQ.2: Node set (not available yet}
CURTYP	Type of current measured: EQ.1: Volume current EQ.2: Surface current (not available yet} EQ.3: Magnetic field flux (B field times Area)
PID	Part ID. It is only available for 2D axisymmetric Eddy current cases. See Remark 2.

Remarks:

1. **Output.** An ASCII file em_rogo_*xxx*, with *xxx* being the ROGID, is generated for each *EM_CIRCUIT_ROGO card. This file contains the value of the current or the magnetic field versus time for that Rogowsky coil.

2. **Special input for 2D axisymmetric Eddy current problems.** When the second line of *EM_ROTATION_AXIS is used, a simplified input for 2D axisymmetric

Eddy current problems becomes available. This case does not require specifying segment sets through the cross sections. Instead, if segment sets are unavailable, provide the part ID of the 2D axisymmetric part as PID to associate it with a Rogowsky coil.

*EM_CIRCUIT_SOURCE

Purpose: Define a circuit with uniform current through its cross-section.

This feature generally helps model stranded conductors carrying a source current (in which case Amperes become Ampere-turns). This feature can also help save computational time in models with a low-frequency current and where the diffusion of the EM fields is a very fast process. In contrast, when using *EM_CIRCUIT, the current density in a circuit is modeled in accordance with the solver type defined in EMSOL of *EM_CONTROL. For example, using an eddy current solver involves considering the diffusion of the current in the circuit. See Remark 2 for a discussion of the available circuit types for source circuits.

Card 1	1	2	3	4	5	6	7	8
Variable	CIRCID	CIRCTYP	LCID	R/F	T/A	T0		
Type	I	I	I	F	F	F		
Default	none	none	none	none	none	none		

Card 2	1	2	3	4	5	6	7	8
Variable	SIDCURR	SIDVIN	SIDVOUT	PARTID	IFREQST			
Type	I	I	I	I	I			
Default	none	none	none	none	1			

VARIABLE	DESCRIPTION
CIRCID	Circuit ID
CIRCTYP	Circuit type : EQ.1: Imposed current as a function of time specified by a load curve. EQ.2: Imposed voltage as a function of time given by a load curve. EQ.11: Imposed current defined by an amplitude A, frequency

VARIABLE	DESCRIPTION
	F, and initial time t_0: $I = A\sin[2\pi F(t - t_0)]$
LCID	Load curve ID for CIRCTYP = 1 or 2. See Remark 3.
R/F	Value of the frequency for CIRCTYP = 11. To have the frequency specified as a function of time with a load curve, a negative value can be entered with the absolute value corresponding to the load curve ID. Value of the resistance when using an imposed voltage (CIRCTYPE = 2).
T/A	Value of the amplitude for CIRCTYP = 11. To have the amplitude specified as a function of time with a load curve, a negative value can be entered with the absolute value corresponding to the load curve ID. See Remark 3. Number of turns when using an imposed voltage (CIRCTYPE = 2).
T0	Value of the initial time, t_0, for CIRCTYP = 11. Phase time shift for the frequency domain solvers (EMSOL = 4, 7, and 9 *EM_CONTROL) when CIRCTYP = 1 or 2.
SIDCURR	Segment set ID for the current. It uses the orientation given by the normal of the segments. To use the opposite orientation, use a '–' (minus) sign in front of the segment set ID or reorient the face normals.
SIDVIN	Segment set ID for input voltage when CIRCTYP = 2 or for input current when CIRCTYP = 1 or 11. This field must be left blank if the stranded conductor is a torus shape (see Remark 1).
SIDVOUT	Segment set ID for output voltage when CIRCTYP = 2 or for output current when CIRCTYP = 1 or 11. This field must be left blank if the stranded conductor is a torus shape (see Remark 1).
PARTID	Part ID associated with the circuit
IFREQST	Frequency for recomputing the source terms. The source terms are recalculated every IFREQST time steps. By default, the source terms are recomputed every EM time step. See Remark 4. LT.0: |IFREQST| is a load curve ID giving the frequency for recomputing as a function of time.

Remarks:

1. **Imposed current with closed loop geometry.** For a closed loop geometry (torus), SIDVIN and SIDVOUT cannot be defined. Thus, only SIDCURR is necessary.

2. **Circuit types for source circuits.** Source circuits with imposed currents (CIRCTYP = 1 or 11) can be associated with an insulator material (MTYPE = 1) in *EM_MAT_001. In this case, the solver relies on the Biot-Savart integration method to retrieve the influence of the source circuit on other conductors. This method is fast, but it is not possible to calculate the force on the coil. Circuits conforming to these conditions can be used with both the Richardson (*EM_SOLVER_FEMBEM) and Monolithic solvers (*EM_SOLVER_FEMBEM_MONOLITHIC) for coupling the FEM-BEM solvers.

   To extract the magnetic force on the coil when using imposed current, the source circuit must be associated with a conductor material (MTYPE = 2) that has a 0.0 value for the conductivity. In this case, the stranded circuit becomes part of the FEM/BEM system, and resolution might be slower. This approach is only valid with the monolithic solver (see *EM_SOLVER_FEMBEM_MONOLITHIC).

   Source circuits also support imposed voltage with CIRCTYP = 2 (associated with a resistance value and a number of turns or windings). The source circuit must be used with the monolithic solver (*EM_SOLVER_FEMBEM_MONOLITHIC) and become part of the FEM/BEM system. Therefore, it must be associated with a conductor material (MTYPE = 2) with a 0. value for the conductivity in its associated EM material (see *EM_MAT_001). Defining source circuits by their voltage instead of their current is very useful in cases where the current is unknown, such as for actuators or electric motors.

3. **Ampere versus Ampere-turns.** For an imposed current (CIRCTYP = 1 or 11), the number of windings can be directly applied by multiplying the current value by the number of turns. The imposed current value corresponds to Ampere-turns. For example, a stranded coil with ten turns and a current of two Amperes means that imposing a value of twenty in the load curve (if CIRCTYP = 1) or its periodic amplitude (if CIRCTYP = 11). For imposed voltage, providing Ampere-turns is not available. Thus, the number of turns must be specified directly in the keyword (as well as the total resistance of the circuit).

4. **When to change IFREQST.** In cases where conductors are not moving with respect to one another, using high values of IFREQST avoids recomputing the source fields which can save some calculation time.

5. **EMSOL = 4, 7, 9.** For the frequency domain solvers (EMSOL = 4, 7, and 9 on *EM_CONTROL), only CIRCTYPE = 1 and 2 are available. The values contained

in the load curve represent the peak value of the imposed current or voltage, while a time shift can optionally be defined using T0.

*EM_CONTACT

Purpose: To optionally define and specify options on electromagnetic contacts between two sets of parts. Generally, it is used with the *EM_CONTACT_RESISTANCE. Fields left empty on this card default to the value of the equivalent field for *EM_CONTROL_-CONTACT.

Contact Definition Cards. Include one card for each contact definition. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	CONTID	DTYPE	PSIDR	PSIDT	EPS1	EPS2	EPS3	D0
Type	I	I	I	I	F	F	F	F
Default	none	0	none	none	0.3	0.3	0.3	none

VARIABLE	DESCRIPTION
CONTID	Electromagnetic contact ID
DTYPE	Detection type (see remarks): EQ.0: Contact type 0 (default) EQ.1: Contact type 1
PSIDR	Reference surface part set ID
PSIDT	Tracked surface part set ID
EPS*i*	Contact coefficients for contact detection conditions. See the discussion below.
D0	Parameter for contact condition 3 when DTYPE = 1 (see remarks)

Remarks:

In these remarks, we discuss the conditions for contact detection. For reference, Figure 0-1 illustrates which geometric values help determine contact. In this figure and discussion, f_r is a face of the reference surface, and f_t is a face of the tracked surface. Contact is detected when *all of the following three conditions are satisfied*:

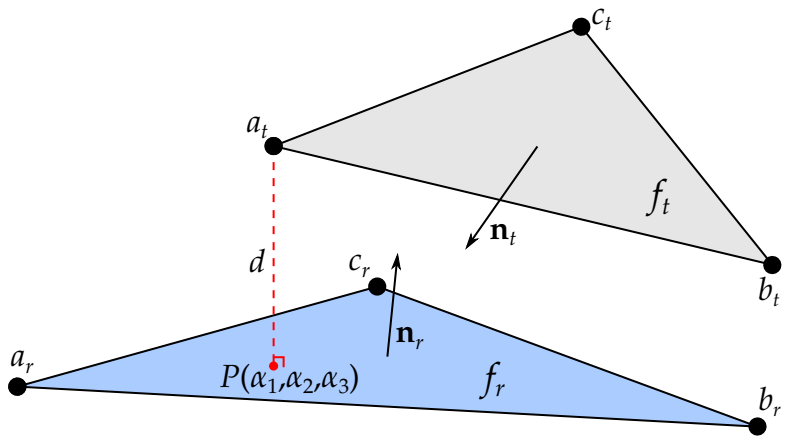


Figure 0-1. Contact detection conditions between two faces.

1. Contact condition 1:

$$\mathbf{n}_r \cdot \mathbf{n}_t \leq -1 + \varepsilon_1$$

Here $\mathbf{n}_r$ and $\mathbf{n}_t$ are the normal vectors of faces f_r and f_t, respectively, and ε_1 is an input parameter. See Figure 0-1.

2. Contact condition 2:

$$-\varepsilon_2 \leq \alpha_1 \leq 1 + \varepsilon_2$$
$$-\varepsilon_2 \leq \alpha_2 \leq 1 + \varepsilon_2$$
$$-\varepsilon_2 \leq \alpha_3 \leq 1 + \varepsilon_2$$

$(\alpha_1, \alpha_2, \alpha_3)$ are the local coordinates of point P (see Figure 0-1). P is the projection of point a_2 on face f_r. ε_2 is an input parameter.

3. Contact condition 3 depends on the contact type. For either possible condition, let d be the distance between P and a_t (see Figure 0-1).

   a) For contact type 0:

$$d \leq \varepsilon_3 S_r \ ,$$

   where ε_3 is an input parameter and S_r is the minimum side length for f_r:

$$S_r = \min[d(a_r, b_r), d(b_r, c_r), d(c_r, a_r)] \ .$$

   b) For contact type 1:

$$d \leq D_0 \ ,$$

   where D_0 is an input parameter.

*EM_CONTACT_RESISTANCE

Purpose: Calculate the electric contact resistance of a previously defined EM contact in *EM_CONTACT or as a general contact resistance if not associated with any specific contact (see Remark 2 and CONTID = 0).

Card 1	1	2	3	4	5	6	7	8
Variable	CRID	CONTID	CRTYPE		JHRTYPE			
Type	I	I	I		I			
Default	none	none	none		none			

Cards 2	1	2	3	4	5	6	7	8
Variable	DFID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
CRID	Resistive contact ID
CONTID	EM contact ID defined in *EM_CONTACT. EQ.0: This contact resistance applies to all detected contacts that are not part of any *EM_CONTACT. See Remark 2.
CRTYPE	Contact resistance type: EQ.1: Electric contact resistance defined with a *DEFINE_FUNCTION.
DFID	*DEFINE_FUNCTION ID (see Remark 1)
JHRTYPE	Indicates how the Joule heating calculated by the contact resistance shall be taken into account:

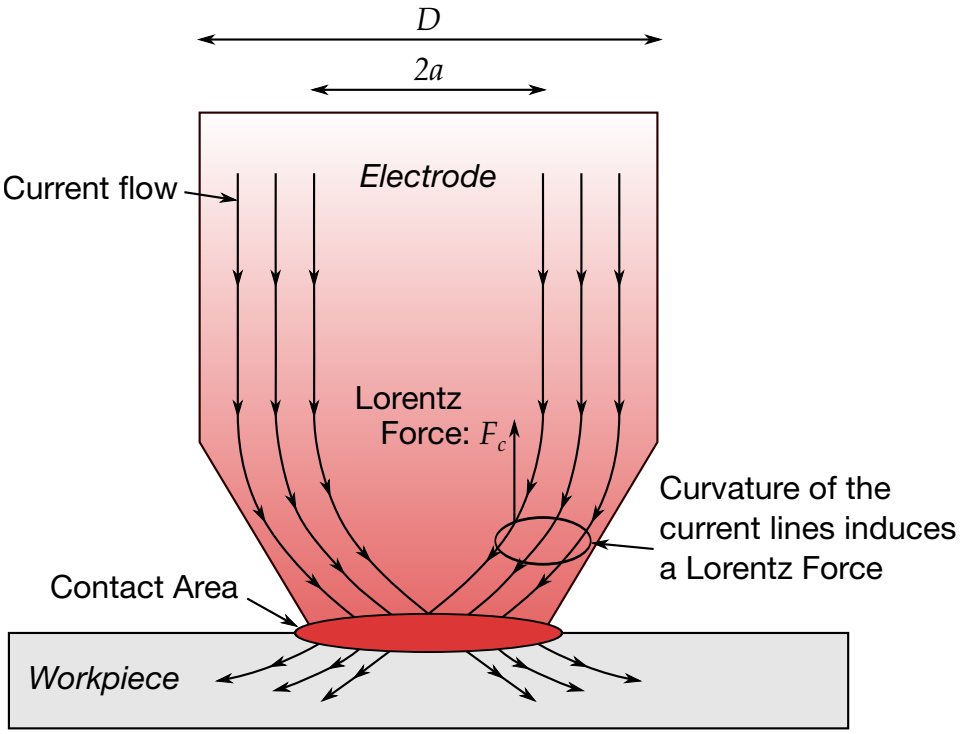


Figure 6-2. Electrode coming into contact with workpiece (RSW application).

VARIABLE	DESCRIPTION
	EQ.0: No addition. The Joule heating calculated by the contact resistance is not taken into account.
	EQ.1: The Joule heating coming from the contact resistance is divided and distributed evenly among all elements neighboring the contact surface.

Remarks:

1. ***DEFINE_FUNCTION parameters.** In the *DEFINE_FUNCTION, the following parameters are allowed: f(time, emdt, arealoc, areatot, ctdist, currloc, currglob, rho_mst, rho_slv, cond_mst, cond_slv, ctpress_mst, ctpress_slv, temp_mst, temp_slv, vmstress_mst, vmstress_slv, press_mst, press_slv). Here,

time	current time step
emdt	EM time step
arealoc	Local area associated with each face in contact
areatot	Total contact area
ctdist	Contact distance between the two faces in contact

rho_mst	Density of the elements associated with the reference surface side of the contact
rho_slv	Density of the elements associated with the tracked surface side of the contact
cond_mst	Electrical conductivity of the elements associated with the reference surface side of the contact
cond_slv	Electrical conductivity of the elements associated with the tracked surface side of the contact
ctpress_mst	Contact pressure of the elements associated with the reference surface side of the contact
ctpress_slv	Contact pressure of the elements associated with the tracked surface side of the contact
temp_mst	Temperature of the elements associated with the reference surface side of the contact
temp_slv	Temperature of the elements associated with the tracked surface side of the contact
vmstress_mst	von Mises stress of the elements associated with the reference surface side of the contact
vmstress_slv	von Mises stress of the elements associated with the tracked surface side of the contact
press_mst	Pressure of the elements associated with the reference surface side of the contact
press_slv	Pressure of the elements associated with the tracked surface side of the contact

2. **General contact resistance.** If CONTID is 0, this *EM_CONTACT_RESISTANCE applies to all detected EM contacts that are not part of any *EM_CONTACT. This feature can be helpful in cases where a given part comes into contact with itself (self-contact). Note that CCONLY must be set to 0 on *EM_CONTROL_CONTACT for contacts not part of a *EM_CONTACT to be detected.

3. **Supported contact types.** CTYPE on *EM_CONTROL_CONTACT must be greater than or equal to 0 for the contact resistance to apply because contact resistance is only supported for penalty contacts.

*EM_CONTACT_SUBDOM

Purpose: Optional card used for defining a specific region where EM contact will be active. This allows saving some calculation time by limiting the contact search area. Must be used in conjunction with *EM_CONTROL_CONTACT.

Card 1	1	2	3	4	5	6	7	8
Variable	SDTYPE	MVTYPE	LCIDX/NID	LCIDY	LCIDZ			
Type	I	I	I	I	I			
Default	none	0	none	none	none			

Card 2	1	2	3	4	5	6	7	8
Variable	R	PMINX	PMINY	PMINZ	PMAXX	PMAXY	PMAXZ	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
SDTYPE	Subdomain definition type : EQ.1: Defined by box. EQ.2: Defined by cylinder. EQ.3: Defined by sphere.
MVTYPE	Movement type of subdomain : EQ.0: Static subdomain (Default). EQ.1: Domain translates in the three directions by the velocities given by LCIDX,LCIDY,LCIDZ. EQ.2: Domain follows the displacements of the node ID given by NID.

VARIABLE	DESCRIPTION
LCIDX/NID	Time dependent load curve ID for the translational velocity in the X direction for MVTYPE = 1, Node ID for MVTYPE = 2.
LCIDY/Z	Time dependent load curve IDs for MVTYPE = 1 in the Y and Z directions.
R	Radius of the sphere if SDTYPE = 3 or the cylinder if SDTYPE = 2.
PMINX/Y/Z	Point of minimum coordinates if SDTYPE = 1. Origin point if SDTYPE = 3. Axis head point if SDTYPE = 2.
PMAXX/Y/Z	Point of maximum coordinates if SDTYPE = 1. Axis tail point if SDTYPE = 2.

*EM_CONTROL

Purpose: Enable the EM solver and set its options.

Card 1	1	2	3	4	5	6	7	8
Variable	EMSOL	NUMLS	BIOT	DIMTYPE	NPERIO	EPS0	NCYLFEM	NCYLBEM
Type	I	F	I	I	I	F	I	I
Default	none	none	0	0	2	8.85E-12	5000	5000

VARIABLE	DESCRIPTION
EMSOL	Electromagnetism solver selector: EQ.-1: Turns the EM solver off after reading the EM keywords. EQ.1: Eddy current and magnetostatics solver EQ.2: Periodic inductive heating solver (see Remark 3) EQ.3: Resistive heating solver EQ.4: Frequency-based Eddy current solver (see Remark 3) EQ.5: Periodic resistive heating solver (see Remark 3) EQ.6: FEM-BEM electrostatic solver (see Remark 6) EQ.7: Helmholtz wave equation solver (see Remark 4) EQ.8: Quasistatic electrostatics solver (see Remark 5) EQ.9: Radiofrequency (RF) heating solver (see Remark 5) EQ.11: Electrophysiology monodomain EQ.12: Electrophysiology bidomain EQ.13: Electrophysiology monodomain coupled with bidomain EQ.14: Pure eikonal model. Activation times are computed and output in VTK format to the `/vtk` directory. See *EM_EP_EIKONAL. EQ.15: Reaction eikonal (RE-) model based on [1]. See *EM_EP_EIKONAL. EQ.16: Reaction eikonal (RE+) model based on [1]. See *EM_EP_EIKONAL.

VARIABLE	DESCRIPTION
NUMLS	Number of local EM steps in a whole period for EMSOL = 2 and 5. For EMSOL = 4, 7, and 9, NUMLS is the global frequency. If a negative value is entered in either case, it gives the value as a function of the macro time. For EMSOL = 3 when the EM solver is coupled with the P2D battery solver, NUMLS is the number of subcycles that the P2D model solves before the next EM update (see Remark 4 of *BATTERY_ECHEM_CONTROL_INTEGRATION).
BIOT	Biot-Savart type calculations: EQ.0: No extra Biot-Savart type calculations EQ.1: When the resistive heat solver is selected (EMSOL = 3), this option triggers a Lorentz force calculation in beam elements that have been defined as conductors. For a magnetostatic solve (EMSOL = 1), this option causes the output of a binary file embiotout (see Remark 7). EQ.2: This option causing reading in the embiotout binary file for the Biot-Savart solve. See Remark 7.
DIMTYPE	EM dimension type: EQ.0: 3D solve EQ.1: 2D planar with zero thickness shell elements EQ.2: 2D axisymmetric (Y-axis only) with zero thickness elements
NPERIO	Number of periods for which the last is used to calculate the average Joule heat rate when EMSOL = 2. NPERIO = 2 means that two periods of NUMLS steps will be calculated. Only the last period of NPERIO is used for the average Joule heat calculation. See Remark 1.
EPS0	Optional modification of vacuum permittivity, ε_0. See Remark 6.
NCYLFEM	Number of electromagnetism cycles between the recalculation of FEM matrices. If negative, the absolute value refers to a load curve that gives the number of electromagnetism cycles as a function of time. See Remark 2.

VARIABLE	DESCRIPTION
NCYLBEM	Number of electromagnetism cycles between the recalculation of BEM matrices. If negative, the absolute value refers to a load curve that gives the number of electromagnetism cycles as a function of time. See Remark 2.

Remarks:

1. **Number of periods and the average Joule heating calculation.** The purpose of using more than one period to calculate the average Joule heating is to allow the different fields to adopt the correct amplitude and time shift when starting from 0.0 at $t = 0$. NPERIO = 2 means that two periods are calculated, of which only last one, being the second one in this case, is used for the average Joule heat calculation. In some cases, larger values might be required to achieve good accuracy, and conversely, in others, NPERIO = 1 might yield sufficient accuracy.

2. **Order of precedence for NCYLFEM and NCYLBEM.** You can set NCYLFEM and NCYLBEM with *EM_CONTROL_SOLUTION, *EM_SOLVER_FEM/BEM, or *EM_CONTROL. *EM_CONTROL_SOLUTION has the highest priority for setting these fields, while *EM_CONTROL has the lowest priority. If a field is left as the default on *EM_CONTROL_SOLUTION, the solver looks at the setting of that field on *EM_SOLVER_FEM/BEM. If left as default on *EM_SOLVER_FEM/BEM, the solver looks at the setting on *EM_CONTROL. When the conductor parts are deforming or undergoing changes in their EM material properties (conductivity for example), the FEM matrices should be recalculated more often, so NCYLFEM may need to be changed. NCYLBEM controls the BEM matrices recomputation frequency as well as the EM contact detection. It is therefore important to use the field in Eddy current cases when conductors are moving respective to one another as well as in case involving changing EM contact.

3. **Solver for inductive heating.** Both EMSOL = 2 and 4 can be used for inductive heating problems. EMSOL = 4 requires using the monolithic solver. It is a frequency-based solver, so the solve is usually faster than solving EM quantities in the time domain over the full period. However, only EMSOL = 2 is available for nonlinear magnetic permeability cases. EMSOL = 5 is similar to EMSOL = 2 but restricted to resistive-heating-type applications because it does not consider inductive-diffusive effects.

4. **Helmholtz equation.** When EMSOL = 7 is selected, the equation solved on the FEM system is the frequency-domain wave equation:

$$\nabla \times \left(\frac{1}{\mu}\nabla \times \hat{\mathbf{E}}\right) - \omega^2 \varepsilon \hat{\mathbf{E}} = -j\omega \hat{\mathbf{J}}_{\text{source}}$$

In the above, $\hat{\mathbf{E}}$ and $\hat{\mathbf{J}}_{\text{source}}$ are the phasor representations of **E** (the electric field) and $\mathbf{J}_{\text{source}}$ (the current density source). BEM is used for the far-field domain and interactions between conductors.

5. **Dielectrics.** EMSOL = 8 and 9 solve the quasi-static electrostatic equation in the time domain and frequency domains, respectively. EMSOL = 8 solves the following equation:

$$\nabla \cdot (\sigma \nabla \varphi) + \frac{\partial}{\partial t}[\nabla \cdot (\varepsilon \nabla \varphi)] = 0 \ ,$$

where σ is electrical conductivity, ε is the permittivity, and φ is the scalar potential. For EMSOL = 9, the equation becomes:

$$\nabla \cdot \left((\sigma + j\omega\varepsilon)\nabla\varphi\right) \ = 0 \ ,$$

where ω is the angular frequency and j is the imaginary unit.

Both those solvers are purely FEM. The choice between EMSOL = 8 and 9 depends on the time scale of the application. EMSOL = 9 can include dielectric heating (via the introduction of a complex permittivity ε) while the time domain formulation cannot. Note that for $\varepsilon = 0.$, EMSOL = 8 reduces to the resistive heating solver (same as EMSOL = 3), while EMSOL = 9 becomes equivalent to EMSOL = 5.

6. **Vacuum permittivity.** The default permittivity, ε_0, is set to $8.8541878128e^{-12}$, which is valid in SI units. However, if a different unit system is selected, that default value may need to be adjusted by a scale factor. Note that the EM solver uses a "shifted" EM unit system where the fundamental SI unit for electromagnetics, the Ampere, has been replaced by the "Henry/meter" (allowing the vacuum permeability, μ_0, to become a dimensionless constant $4\pi e^{-7}$). All EM quantities must therefore be adjusted accordingly. Changing EPS0 only affects solvers that use ε, meaning EMSOL = 6, 7, 8, and 9. Relative permittivity is defined on *EM_MAT_001, *EM_MAT_002, and *EM_MAT_004.

7. **Two step solve**. In order to study the magnetic field generated by a magnetic source involving permanent magnets or ferromagnets on its environment, a two-step solving approach can be used. In the first solve, the magnets and other contributing magnetic parts are modeled, and the magnetostatic problem is solved. Selecting BIOT = 1 allows the solver to automatically output relevant quantities from the solution to a binary file. This binary file will allow a subsequent Biot-Savart evaluation during the second step. For the second solve, select BIOT = 2 to cause the solver to automatically look for the previously generated binary file and use it to do a Biot-Savart evaluation on all parts that have been defined as conductors. This approach can be useful for avoiding solving the complete magnetostatic problem each time its effects on a different part of the domain need to be studied.

8. **Summary of physics modeled by solver.** The following table summarizes the physics modeled by solver for EMSOL = 1 through 9.

Solver	Ohmic losses (σ)	Eddy currents – current diffusion	Dielectric effects (ε)	Dielectric losses	Magnetic effects (linear μ)	Magnetization (permanent magnets, nonlinear μ)	Magnetic losses	Time integration	Formulation
1	yes	yes	no	no	yes	yes	no	time	FEM-BEM
2	yes	yes	no	no	yes	no	no	time avg.	FEM-BEM
3	yes	no	no	no	no	no	no	time	FEM
4	yes	yes	no	no	yes	no	yes	freq.	FEM-BEM
5	yes	no	no	no	no	no	no	time avg.	FEM
6	no	no	yes	no	no	no	no	time	FEM-BEM
7	yes	no	yes	yes	yes	no	yes	freq.	FEM-BEM
8	yes	no	yes	no	no	no	No	time	FEM
9	yes	no	yes	yes	no	no	no	freq.	FEM

References:

[1] A. Neic et al, " Efficient computation of electrograms and ECGs in human whole heart simulations using a reaction-eikonal model", Journal of Computational Physics 346 (2017) 191–211

*EM_CONTROL_CONTACT

Purpose: Activate the electromagnetism contact algorithms, which detect contact between conductors. Electromagnetic fields flow from one conductor to another when found to be in contact.

Card 1	1	2	3	4	5	6	7	8
Variable	EMCT	CCONLY	CTYPE	DTYPE	EPS1	EPS2	EPS3	D0
Type	I	I	I	I	F	F	F	F
Default	0	0	0	0	0.3	0.3	0.3	none

VARIABLE	DESCRIPTION
EMCT	EM contact activation flag: EQ.0: No contact detection EQ.1: Contact detection
CCONLY	Determines parts for which EM contact is active: EQ.0: Contact detection between all active parts associated with a conducting material. (Default) EQ.1: Only look for EM contact between parts associated through the *EM_CONTACT card. In some cases, this option can reduce the calculation time.
CTYPE	Contact type: EQ.-1: Node-to-node contact based on constraints on the scalar potential. See Remark 1. EQ.0: Node-to-node penalty-based contact on the scalar potential. EQ.1: Discrete mortar penalty contact on the scalar potential. EQ.2: Continuous mortar penalty contact on the scalar potential and the vector potential (when active).
DTYPE	Detection type. If *EM_CONTACT is not defined, the solver will look for global contact options in *EM_CONTROL_CONTACT. See *EM_CONTACT for details.

VARIABLE	DESCRIPTION
	EQ.0: Contact type 0 (default). EQ.1: Contact type 1.
EPS*i*	Global contact coefficients used if the equivalent fields in *EM_-CONTACT are empty. See *EM_CONTACT for details.
D0	Global parameter for contact condition 3 when DTYPE = 1. See *EM_CONTACT for details.

Remarks:

1. **Contact types.** In versions prior to R12, CTYPE = 0 was the default EM contact for the resistive heating solver, while CTYPE = -1 was the default EM contact for the Eddy current solver. In versions R12 and later, CTYPE = 0 became the default for all solvers. We recommend CTYPE = 1 and 2 for the best accuracy.

2. **Contact between BEM surfaces with the Eddy current solver.** When the Eddy current solver is active and contact occurs between BEM surfaces, the solver automatically removes the faces on the contact surface and internally stitches the two BEM surfaces together to achieve a continuous closed BEM mesh.

*EM_CONTROL_COUPLING

Purpose: Control couplings between various solvers with the EM solver.

Card 1	1	2	3	4	5	6	7	8
Variable	THCPL	SMCPL	THLCID	SMLCID	THCPLFL	SMCPLFL		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		

Optional Card.

Card 2	1	2	3	4	5	6	7	8
Variable	SMMOD	DFX	DFY	DFZ		FSIB	INTERP	
Type	I	I	I	I		I	I	
Default	0	none	none	none		0	0	

Optional Card.

Card 3	1	2	3	4	5	6	7	8
Variable	NID1	NID2	NID3					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
THCPL	Coupling to the thermal solver. When turned on, the EM solver will transfer the Joule heating terms to the solid mechanics thermal solver. EQ.0: Coupling on.

VARIABLE	DESCRIPTION
	EQ.1: Coupling off. EQ.2: Coupling on. It forces all EM heating terms to be expressed at the element level. See Remark 2.
SMCPL	Coupling to the solid mechanics solver. When turned on, the EM solver will transfer forces to the solid mechanics solver. EQ.0: Coupling on. Lorentz force density is transferred. EQ.1: Coupling off. EQ.2: Coupling on. Magnetic force surface density is transferred. More accurate representation of EM forces in cases involving magnets or nonlinear ferromagnets. See *EM_SOLVER_FEMBEM_MONOLITHIC. EQ.3: Coupling on. Magnetic force surface density is transferred to magnets and ferromagnets, while Lorentz force density is transferred to regular conductors.
THLCID	Optional load curve ID. When defined, the heat rate transferred to the thermal solver will be scaled by the value returned by THLCID.
SMLCID	Optional load curve ID. When defined, the forces transferred to the solid mechanics solver will be scaled by the value returned by SMLCID.
THCPLFL	Coupling to the heat equation when EM quantities are solved on fluid elements. When turned on, the EM solver will transfer the Joule heating terms to the ICFD solver. EQ.0: Coupling off. EQ.1: Coupling on.
SMCPLFL	Interaction between the solid mechanics solver and the ICFD solver when EM quantities are solved on fluid elements. EQ.0: The fluid pressure will be passed to the solid mechanics solver (default). EQ.1: The fluid pressure is replaced by the electrostatic pressure. EQ.2: The fluid and electrostatic pressure are passed on to the solid mechanics solver.

VARIABLE	DESCRIPTION
	EQ.4: The EM solver will send the Lorentz Force as a volumetric source term to the fluid solver.
SMMOD	Coupling to the solid mechanics solver. When turned on, the EM solver will transfer forces to the solid mechanics solver. EQ.0: Off. EQ.1: Force calculation at the element level is decided by *DEFINE_FUNCTION. See DFX, DFY and DFZ. EQ.2: Force calculation at the element level is decided by the usermat routine. See dyn21em.f and `user_getEMForceArray` routine.
DFX/DFY/DFZ	*DEFINE_FUNCTION IDs for the force three components if SMMOD = 1. Arguments for the *DEFINE_FUNCTIONs are the same as in *EM_EOS_TABULATED2. See Remark 2 of *EM_EOS_TABULATED2.
FSIB	Solid mechanics – ICFD FSI boundaries. When turned on, the EM solver uses the coupled FSI interface to impose the continuity of the scalar potential between the two domains. See Remark 1. EQ.0: Off EQ.1: On
INTERP	Flag for EM force interpolation by the solid mechanics solver. By default, the solid mechanics solver interpolates at each time the EM force from the last two EM steps. See Remark 3. EQ.0: Time interpolation on (default) EQ.1: No time interpolation. Use the force from the last EM solve.
NID1/NID2/NID3	Three node IDs defining a local coordinate system. See Remark 4.

Remarks:

1. **FSI coupling.** *ICFD_CONTROL_FSI and *ICFD_BOUNDARY_FSI control the coupling between the ICFD and the solid mechanics solvers. When FSI exists and the FSIB flag is turned on, the EM solver reuses those boundaries to impose the necessary constraints on the scalar potential to ensure continuity. The solid

and fluid domains must have been previously defined as conductors (see *EM_-MAT_001).

2. **Heating term at the element level.** By default, the EM-thermal heat rate is expressed at the node level before being sent to the thermal solver. However, for some applications where sharp gradients are present, such as battery applications involving Randles circuits, an alternative approach is to express the heat rate term at the element level using THCPL = 2. THCPL = 2 results in a smoother and more diffusive solution.

3. **Force interpolation.** In typical EM Eddy current applications, such as magnetic metal forming or welding, the mechanical time step is much smaller than the EM time step. The EM solver completes its solve with its larger time step and waits for the mechanical solver to catch up. During each of its solves, the mechanical solver interpolates the force between the two last EM steps. However, for specific magnetostatic problems (specifically rotating problems), this time interpolation can introduce errors. INTERP, therefore, exists to turn off the time interpolation and have the mechanical solver use the force from the last EM solve.

4. **Rotating problems.** In conjunction with INTERP = 1 described in Remark 3, optional Card 3 enables defining a local reference frame. If the frame is rotating, the mechanical solver applies the corresponding rotation to the force vectors obtained from the EM solver. This feature was developed specifically for magnetostatic rotating problems and is subject to further development.

*EM_CONTROL_COUPLING_P2D

Purpose: Control aspects of the coupling to the P2D battery electrochemistry solver (see *BATTERY).

Card 1	1	2	3	4	5	6	7	8
Variable	LUNIT	MUNIT	TUNIT					
Type	F	F	F					
Default	1.0	1.0	1.0					

Swelling and Thermal Card. This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	ISWELL	IP2DTH						
Type	I	I						
Default	0	0						

Optimization Card. This card is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	ISKIP	NBIN	NSTALE	SKIPTOL	TOPT			
Type	I	I	I	F	F			
Default	0	0	0	1.0E-3	0.0			

VARIABLE	DESCRIPTION
LUNIT	Length unit conversion factor, α_L. See Remark 1.
MUNIT	Mass unit conversion factor, α_M. See Remark 1.
TUNIT	Time unit conversion factor, α_T. See Remark 1.

VARIABLE	DESCRIPTION
ISWELL	Flag to enable strain swelling calculation (see Remark 2): EQ.0: No strain swelling calculation EQ.1: Perform strain swelling calculation
IP2DTH	Flag to turn off coupling the P2D heating terms with the thermal solver: EQ.0: Coupling with the thermal solver is enabled. EQ.1: Coupling with the thermal solver is disabled.
ISKIP	Flag to skip the P2D solve when the current change is less than SKIPTOL: EQ.0: This optimization is disabled. EQ.1: Skip the P2D solve when the current change is less than SKIPTOL.
NBIN	Number of groups to bin the circuits into based on current density, N_b EQ.0: Binning is disabled.
NSTALE	Number of P2D solve skips permitted before forcing a P2D solve for a circuit EQ.0: Unlimited skips
SKIPTOL	Relative current change threshold
TOPT	Time at which optimization becomes active: GE.0.0: Time at which the optimization methods become active. LT.0.0: |TOPT| refers to a load curve ID (see *DEFINE_CURVE and *DEFINE_CURVE_FUNCTION) specifying whether optimization is active as a function of time. If the ordinate value is positive, then optimization is active, while negative values mean optimization is off.

Remarks:

1. **Unit consistency.** The P2D solver uses fixed SI units (kg, m, s, and A) internally, but the EM solver uses the model's unit system, which can be any consistent unit

system. Without conversion, quantities shared between the P2D and EM solvers, such as current density, voltage, resistance, and power, have inconsistent units. LUNIT, MUNIT, and TUNIT provide the unit conversion between the EM solver and the P2D solver. If the model is built in standard SI units, this keyword is not required (all defaults are 1.0). Examples of settings for LUNIT, MUNIT, and TUNIT are shown below for different unit systems that could be used by the EM solver:

	mm/tonne/s	mm/g/ms	mm/kg/ms
LUNIT, α_{L}	1.0E-3	1.0E-3	1.0E-3
MUNIT, α_{M}	1.0E3	1.0E-3	1.0
TUNIT, α_{T}	1.0	1.0E-3	1.0E-3

From these three inputs, the derived voltage scaling factor is given by:

$$\alpha_{\mathrm{V}} = \alpha_{\mathrm{M}} \alpha_{\mathrm{L}}^{2} \alpha_{\mathrm{T}}^{-3}$$

The current density, i_{app}, from the EM solver is scaled by α_{L}^{-2} before being given to the P2D solver. The effective resistance, R_{eff}, is scaled by $\alpha_{\mathrm{V}} \alpha_{\mathrm{L}}^{2}$ prior to being returned to the EM solver. The heat source terms from the P2D solver are scaled by $\alpha_{L}^{2} \alpha_{\mathrm{V}}^{-1}$ before being given to the thermal solver.

2. **Swelling.** When ISWELL = 1, a through-thickness intercalation swelling strain is computed for each circuit after every P2D solve. The calculation requires the partial molar volumes Ω_{a} and Ω_{c} specified on *BATTERY_ECHEM_MAT_ANODE and *BATTERY_ECHEM_MAT_CATHODE, respectively.

   In continuum mechanics, an eigenstrain is any mechanical deformation in a material that is not caused by external mechanical stress, with thermal expansion often given as a familiar example. In the context of battery swelling, only the active-material fraction, $\varepsilon_{\mathrm{swell},r}$, of each electrode (with r as a for anode and c for cathode) swells and the electrode-level eigenstrain is expressed as:

$$\varepsilon_{\mathrm{swell},r} = \varepsilon_{\mathrm{am},r} \frac{\Omega_r}{3.} c_{\mathrm{s},r,\max} (\bar{s}_r - s_r)$$

   In the above, r is the region which is a for anode or c for cathode, $\bar{s}_r$ is the current average stoichiometry returned by the P2D solve for the region, s_r is the initial stoichiometry for the region (see *BATTERY_ECHEM_MAT_ANODE and *BATTERY_ECHEM_MAT_CATHODE), $c_{\mathrm{s},r,\max}$ is the maximum solid phase lithium concentration for the region (see *BATTERY_ECHEM_MAT_ANODE and *BATTERY_ECHEM_MAT_CATHODE), and $\varepsilon_{\mathrm{am},r}$ is the active material volume fraction for the region (see Remark 3 of *BATTERY_ECHEM_CONTROL_SOLVER). The total swelling strain per circuit is computed as:

$$\varepsilon_{\mathrm{swell}} = \frac{L_{\mathrm{a}} \varepsilon_{\mathrm{swell,a}} + L_{\mathrm{c}} \varepsilon_{\mathrm{swell,c}}}{L_a + L_c}$$

where L_a and L_c are the lengths of the anode and cathode, respectively.

3. **Optimization.** The parameters on Card 3 reduce the number of P2D solves per EM step. These optimization parameters are useful when using a BatMac model (see *EM_P2D_BATMAC) that has thousands of circuits. Without these optimization mechanisms, the solver loops over all local circuits and solves the P2D system on each one, which can be computationally expensive The mechanisms available on this card complement each other.

   When ISKIP = 1, before solving, the current density of each circuit at step n is compared to its value from the previous step $n-1$. If the relative change is below SKIPTOL:

$$\frac{|I^n - I^{n-1}|}{\max(|I^n|, |I^{n-1}|)} < \text{SKIPTOL} \ ,$$

   then the P2D solve is skipped and the previous step's R_{eff}^{n-1}, i_{app}^{n-1}, and V_{cell}^{n-1} are reused. Although the full P2D system (concentration profiles, radial diffusion, etc.) is not solved, the electrode utilization is advanced through a lightweight forward-Euler step based on Faraday's law. This keeps SOC tracking, OCV evolution, and swelling strain calculations consistent even for circuits that are skipped over multiple consecutive steps. A per circuit staleness counter tracks the number of consecutive skips.

   Among the circuits that are not skipped, the code groups them into N_b bins based on their current density if NBIN > 1. Bins are constructed per CPU and per cell ID, so circuits with different electrochemistry are never grouped together. The P2D solve is only performed on a representative circuit for each bin. All other circuits in the same bin reuse the representative circuit's $R_{\text{eff}}^{\text{rep}}$, $i_{\text{app}}^{\text{rep}}$, and $V_{\text{cell}}^{\text{rep}}$. Unlike the solve-skip case, binned circuits also receive the representative's full P2D state, meaning solid-concentration history, pore-wall flux, and radial concentration profiles are copied from the representative to each binned member. This ensures that the electrochemical state of binned circuits advances in time rather than remaining frozen, reducing the discontinuity when a circuit later receives a fresh solve (via NSTALE or bin reassignment). A circuit that has never been solved ($R_{\text{eff}} = 0.$) is always forced through a full P2D solve and cannot be binned.

   To prevent any circuit from drifting indefinitely without a fresh P2D solve, each circuit maintains a staleness counter that increments every time the circuit is skipped or uses a bin representative's result. When the counter reaches NSTALE, a full P2D solve is performed on the circuit, regardless of the skip tolerance or bin assignment. The counter is then reset to zero. Setting NSTALE = 0 disables the limit.

During the initial transient phase (current ramp-up, first loading step, etc.), every circuit needs a full P2D solve to establish its electrochemical baseline. The optional parameter TOPT specifies the simulation time at which the optimization logic becomes active. For $t < t_{opt}$, all skip and bin logic is disabled, and every circuit is solved. By default, TOPT = 0.0, so the optimisation is available from the start. Note that regardless of t_{opt}, a circuit that has never been solved ($R_{eff} = 0.$) is always forced through a full P2D solve rather than being skipped or binned. If a negative value is entered for TOPT, the absolute value refers to a *DEFINE_CURVE or *DEFINE_CURVE_FUNCTION ID. This curve specifies whether the optimization is active as a function of time. If the ordinate value is positive, then optimization is active, while negative values mean optimization is off.

*EM_CONTROL_EP

Purpose: Enable the EM solver for electrophysiology and set its options.

Card 1	1	2	3	4	5	6	7	8
Variable	EPSOL	NSPLITRI	ACTUSIG	IONSOLVR				
Type	I	I	I	I				

This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	SPLTEND	SPLDT						
Type	F	F						
Default	0.0	0.0						

VARIABLE	DESCRIPTION
EPSOL	Electrophysiology solver selector: EQ.11: Monodomain EQ.12: Bidomain EQ.13: Monodomain coupled with bidomain EQ.14: Pure eikonal model. Activation times are computed and output in VTK format to the `/vtk` directory. See *EM_EP_EIKONAL. EQ.15: Reaction eikonal (RE-) model based on [1]. See *EM_EP_EIKONAL. EQ.16: Reaction eikonal (RE+) model based on [1]. See *EM_EP_EIKONAL.
NSPLITRI	Split ratio between the ionic cell model time step and the monodomain time step. In other words, we will do "NSPLITRI" cell model time steps for each diffusion time step.
ACTUSIG	Number of EP steps between electrical conductivity updates

VARIABLE	DESCRIPTION
IONSOLVR	Ionic solver type: EQ.0: Euler explicit EQ.1: VODE (unstable) EQ.2: Spline. Use only for the eikonal solvers (EPSOL = 14, 15, or 16) or for EPSOL = 11 or 12. It solves each type of cell model only once. The resulting transmembrane potentials are shifted in time for each node with a delay equal to the activation time of the node.
SPLTEND	Optional end time for building the spline
SPLDT	Time step for building the spline

*EM_CONTROL_EROSION

Purpose: Allow the EM solver to take into account eroded elements.

Card 1	1	2	3	4	5	6	7	8
Variable	ECTRL							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
ECTRL	Erosion search: EQ.0: Off. The EM solver ignores eroded elements and still considers them part of the EM problem. EQ.1: On. The EM solver looks for elements that are potentially eroded. If they are eroded, the elements are removed from the EM solve by updating its matrix system.

*EM_CONTROL_MAGNET

Purpose: Control how often the magnetization vector for magnets is recomputed. By default, the magnetization vector for each magnet is computed only once at the beginning of the calculation (see Remark 1).

Card 1	1	2	3	4	5	6	7	8
Variable	MCOMP	NCYCM						
Type	I	I						
Default	0	0						

VARIABLE	DESCRIPTION
MCOMP	Flag controlling whether the magnetization vector is recomputed. EQ.0: Magnetization vector not recomputed (see Remark 1). EQ.1: Magnetization vector recomputed at the frequency controlled by NCYCM.
NCYCM	Magnetization vector recomputation frequency. A value of 1 means recomputation at every EM time step. If a negative value is entered, |NCYCM| is the ID of a load curve giving the value as a function of time.

Remark:

1. **MCOMP.** In most applications involving magnets, the magnetization vector associated to each magnet needs only to be calculated once at the beginning of the analysis. The magnetization direction is then scaled by the Coercive force value (See EM_PERMANENT_MAGNET) to correctly estimate the contribution of each magnet. In certain specific applications, for examples in cases involving magnet deformations or changes in magnet boundary conditions, it may be needed to periodically update the initially computed magnetization vector. This can be achieved by setting MCOMP to 1 and defining NCYCM.

*EM_CONTROL_SOLUTION

Purpose: Specify different conditions under which reassembly of the FEM and BEM matrices occurs.

Card 1	1	2	3	4	5	6	7	8
Variable	NCYLFEM	NCYLBEM	AUTOFEM	AUTOBEM	TOL1FEM	TOL2FEM	TOL1BEM	TOL2BEM
Type	I	I	I	I	F	F	F	F
Default	5000	5000	0	0	0.3	0.1	0.3	0.1

Optional Card.

Card 2	1	2	3	4	5	6	7	8
Variable	NID1	NID2	NID3	NID4	NID5	NID6	ANG	DIST
Type	I	I	I	I	I	I	F	F
Default	0	0	0	0	0	0	1.e16	1.e16

VARIABLE	DESCRIPTION
NCYLFEM	Number of electromagnetism cycles between the recalculation of FEM matrices. See Remark 1. LT.0: |NCYLFEM| refers to a load curve, giving the number of cycles as a function of time.
NCYLBEM	Number of electromagnetism cycles between the recalculation of BEM matrices. See Remark 1. LT.0: |NCYLBEM| refers to a load curve, giving the number of cycles as a function of time.
AUTOFEM	In addition to NCYLFEM, AUTOFEM triggers an automatic recomputation of the FEM matrices based on an error calculation of the conductors' relative deformation and electrical conductivity changes. See TOL1FEM and TOL2FEM. See Remark 2. EQ.0: Autorecomputation off

VARIABLE	DESCRIPTION
	EQ.1: Autorecomputation on
AUTOBEM	In addition to NCYLBEM, AUTOBEM triggers an automatic recomputation of the BEM matrices based on an error calculation of the conductors' relative displacements. See TOL1BEM and TOL2BEM. See Remark 2. EQ.0: Autorecomputation off EQ.1: Autorecomputation on EQ.2: Autorecomputation on. Skip EM solve. See Remark 4.
TOL1FEM	Only used if AUTOFEM = 1. If a conducting element experiences a deformation or a conductivity change that reaches an error larger than TOL1FEM, the solver reassembles the FEM matrices. If negative, the absolute value refers to a load curve that gives the tolerance as a function of time.
TOL2FEM	Only used if AUTOFEM = 1. If (number of conducting elements) × TOL2FEM elements experience a deformation or a conductivity change that reaches an error larger than TOL2FEM, the solver recomputes the FEM matrices. If negative, the absolute value refers to a load curve that gives TOL2FEM as a function of time.
TOL1BEM	Only used if AUTOBEM = 1. If a conducting element experiences a displacement that reaches an error larger than TOL1BEM, then LS-DYNA reassembles the BEM matrices. If negative, the absolute value refers to a load curve that gives the tolerance as a function of time.
TOL2BEM	Only used if AUTOBEM = 1. If (number of conducting elements) × TOL2BEM elements experience a displacement that reaches an error larger than TOL2BEM, then the BEM matrices are recomputed. If negative, the absolute value refers to a load curve that gives TOL2BEM as a function of time.
NID1, NID2, NID3	Local coordinate system specified with NID1, NID2, and NID3. See Remark 3.
NID4, NID5, NID6	Local coordinate system specified with NID4, NID5, and NID6. See Remark 3.
ANG	Angle (in degrees) tolerance criterion that triggers a BEM matrix recomputation. See Remark 3. LT.0.0: |ANG| is a load curve ID giving the angle tolerance

VARIABLE	DESCRIPTION
	criterion as a function of time.
DIST	Distance tolerance criterion that triggers a BEM matrix recomputation. See Remark 3.
	LT.0.0: |DIST| is a load curve ID giving the distance tolerance criterion as a function of time.

Remarks:

1. **Order of precedence for NCYLFEM and NCYLBEM.** You can set NCYLFEM and NCYLBEM with *EM_CONTROL_SOLUTION, *EM_SOLVER_FEM/BEM, or *EM_CONTROL. *EM_CONTROL_SOLUTION has the highest priority for setting these fields, while *EM_CONTROL has the lowest priority. If you leave one of these fields as the default value on *EM_CONTROL_SOLUTION, the solver then looks at the setting of that field on *EM_SOLVER_FEM/BEM. If left as the default on *EM_SOLVER_FEM/BEM, the solver looks at the setting on *EM_CONTROL.

2. **EM contact with recomputation.** When EM contact occurs and you have enabled automatic FEM or BEM matrix recomputation, the solver simultaneously reassembles both the FEM and BEM matrices when any criterion is reached to maintain consistency. When you do not enable automatic recomputation, we recommend the FEM and BEM systems have the same value for the recomputation cycle.

3. **Rotating and translating parts.** The optional second line triggers a BEM matrix recomputation only if the rotation or distance between the system composed of NID1/NID2/NID3 and NID4/NID5/NID6 exceeds ANG (in degrees) or DIST. This feature is useful in cases when one conductor part rotates or translates based on another part and an automatic criterion is desired.

4. **AUTOBEM = 2.** Selecting this option causes skipping the EM solve entirely when no BEM matrix recomputation is triggered. This option is especially useful in magnetostatic cases, meaning the EM solution is independent of previous EM stages and a solve needs to occur only once a certain BEM matrix recomputation threshold is met (usually in combination with the use of NID1/NID2/NID3 and NID4/NID5/NID6). This option provides significant calculation gains in magnetostatic problems that involve moving pieces.

*EM_CONTROL_SWITCH

Purpose: Create a control "switch" that will shut down the solver based on information from a load curve. LS-DYNA supports complex types of curves (See *DEFINE_CURVE_-FUNCTION) that allow the setting up of complex on/off switches, for instance, by using a nodal temperature value.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID	FEMCOMP	BEMCOMP					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
LCID	Load curve or define curve function ID. Negative values switch the solver off; positive values switch it back on.
FEMCOMP	Flag to determine if LS-DYNA recomputes the FEM matrices each time the EM solver is turned back on: EQ.0: Recompute FEM matrices. EQ.1: Do not recompute FEM matrices.
BEMCOMP	Flag to determine if LS-DYNA recomputes the BEM matrices each time the EM solver is turned back on: EQ.0: Recompute BEM matrices. EQ.1: Do not recompute the BEM matrices.

*EM_CONTROL_SWITCH_CONTACT

Purpose: Define a control "switch" that will shut down electromagnetic contact detection. This feature can save some calculation time in cases where you know when contact between conductors will occur or stop occurring.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
LCID	Load curve ID for the curve giving the state as a function of time. Negative values on the ordinate switch the contact detection off, while positive values switch it back on.

*EM_CONTROL_TIMESTEP

Purpose: Controls the EM time step and its evolution.

Card 1	1	2	3	4	5	6	7	8
Variable	TSTYPE	DTCONST	LCID	FACTOR	TSMIN	TSMAX	RLCSF	MECATS
Type	I	F	I	F	F	F	I	I
Default	none	none	none	1.0	none	none	25	0

VARIABLE	DESCRIPTION
TSTYPE	Time step type: EQ.1: Constant time step given in DTCONST EQ.2: Time step as a function of time given by a load curve specified in LCID EQ.3: Automatic time step computation, depending on the solver type. This time step is then multiplied by FACTOR EQ.5: EM time step forced to be the same as the thermal time step in problems with the EM solver coupled to the thermal solver. In other words, the thermal solver determines the EM time step.
DTCONST	Constant value for the time step for TSTYPE = 1
LCID	Load curve ID giving the time step as a function of time for TSTYPE = 2
FACTOR	Multiplicative factor applied to the time step for TSTYPE = 3
TSMIN	Minimum time step. When TSMIN is defined, the EM time step cannot drop below TSMIN. A negative value will refer to a time-dependent load curve.
TSMAX	Maximum time step. When TSMAX is defined, the EM time step cannot increase beyond TSMAX. A negative value will refer to a time-dependent load curve.
RLCSF	RLC Circuit time step scale factor. See Remark 2.

VARIABLE	DESCRIPTION
MECATS	Mechanical time step handling in cases where the EM solver time step becomes smaller (see Remark 3): EQ.0: Default. The EM time step will go below the solid mechanics timestep, and several EM solves will occur between two solid mechanics time steps to ensure time consistency. EQ.1: The solid mechanics time step will adapt and decrease to the EM time step value so that only one EM solve occurs between two solid mechanics solves.

Remarks:

1. **Automatic time step for Eddy current solvers.** For eddy current solvers, the automatic time step is based on the diffusion equation for the magnetic field:

$$\sigma \frac{\partial \vec{A}}{\partial t} + \vec{\nabla} \times \frac{1}{\mu} \vec{\nabla} \times \vec{A} + \sigma \vec{\nabla} \varphi = \vec{j}_S \ .$$

   It is computed as the minimal elemental diffusion time step over the elements. For a given element, the elemental diffusion time step is given as $dt_e = l_e^2/(2D)$, where:

   - D is the diffusion coefficient $D = 1/(\mu_0 \sigma_e)$,
   - σ_e is the element's electrical conductivity,
   - μ_0 is the permeability of free space,
   - l_e is the minimal edge length of the element (minimal size of the element).

2. **Automatic time step with RLC circuit.** When an automatic time step is defined and an RLC circuit is present, the EM solver will perform an additional check and calculate an approximation of the first current period based on a 0-D circuit solve. It will then limit the timestep by a factor $T_{\text{period}}/(4 \times \text{RLCSF})$. The default value of RLCSF ensures that 25 EM timesteps will be calculated for the first quarter period.

3. **MECATS.** In general, we recommend avoiding scenarios where the EM time step becomes smaller than the solid mechanics time step, a situation often resulting from ill-defined input decks and parameters. However, this can happen when conducting elements have large deformations and an automatic EM time step is selected. For this case, choose between the two MECATS options.

4. **Macro time step.** If EMSOL = 2 or 4 in *EM_CONTROL, the EM time step defined here represents a "macro" time step, meaning the time when new time-

averaged quantities such as Joule heating Power or Magnetic Forces are updated. If material properties are constant and no displacement of parts is expected, this macro time step can be equal to the termination time.

*EM_DATABASE_CIRCUIT

Purpose: Enable the output of EM data for every circuit defined.

Output options card

Card 1	1	2	3	4	5	6	7	8
Variable	OUTLV	DTOUT						
Type	I	F						
Default	1	0.						

VARIABLE	DESCRIPTION
OUTLV	Determines if LS-DYNA creates the output file: EQ.0: Do not generate the output file. EQ.1: Generate the output file.
DTOUT	Time interval to print the output. If DTOUT is equal to 0.0, LS-DYNA uses the EM timestep.

Remarks:

1. **File name.** The file name for this database is em_circuit_*XXX*.dat with *XXX* as the circuit ID.

2. **ResistanceD.** LS-DYNA calculates *ResistanceD* in the following way:

   a) Imposes a scalar potential difference of 1 at the circuit's boundaries SIDVIN and SIDVOUT.

   b) Solves $\nabla^2\varphi = 0$ with $\varphi_{\text{SIDVIN}} = 1$ and $\varphi_{\text{SIDVOUT}} = 0$ at SIDCURR. We do not consider diffusive effects, meaning the current density can be written as $\mathbf{j} = \nabla\varphi$ and the total current as $I = \mathbf{j} \cdot \mathbf{n}dA$.

   c) Estimates the resistance using $R_D = U/I$. U is the potential difference between φ_{SIDVIN} and and φ_{SIDVOUT}, that is, equal to 1. The circuit's geometry and conductivity solely determine this R_D resistance. It is, therefore, equivalent to the resistance as commonly defined in the circuit equations:

$$R_D = L/\sigma S \ ,$$

where L is the length of the circuit and S is its surface area.

3. **ResistanceJ.** LS-DYNA calculates *ResistanceJ* with the data provided during the EM solve: $R_J = J/I^2$ where J and I are, respectively, the joule heating and the current. Compared with *ResistanceD*, *ResistanceJ* is not so much a resistance calculation since it accounts for the resistive effects (when using the Eddy current solver). Rather, it corresponds to the resistance that the circuit would need in order to get the same Joule heating in the context of a circuit equation. If all EM fields are diffused or you are using the resistive heating solver, *ResistanceJ* should be close to *ResistanceD*.

4. **Mutual inductances.** Only the mutual inductances between the first three circuits defined are output.

*EM_DATABASE_CIRCUIT0D

Purpose: Enable the output of 0D EM data for every circuit defined.

Output options card

Card 1	1	2	3	4	5	6	7	8
Variable	OUTLV	DTOUT						
Type	I	F						
Default	0	0.						

VARIABLE	DESCRIPTION
OUTLV	Determines if LS-DYNA creates the output file. EQ.0: Does not generate the output file. EQ.1: Generates the output file.
DTOUT	Time interval to print the output. If DTOUT equals 0.0, then LS-DYNA uses the EM time step.

Remarks:

1. **File name.** The file name for this database is em_circuit0D_*XXX*.dat with *XXX* as the circuit ID.

2. **0D solution.** At the start of the run, based on the initial values of the meshes resistances and inductances, the solver calculates the results for a so-called "0D" solution, which does not consider the current's diffusion, the part's displacements, or the EM material property changes. It is, therefore, a crude approximation. LS-DYNA extrapolates the initial results through time which it outputs at DTOUT intervals. This result can be useful in some cases, especially in RLC circuits if you need a rough idea of how the source current will behave.

3. **Cost.** Since the calculation of this 0D circuit can take time depending on the size of the problem, you should only use it in cases where the output results help with comprehension of the analysis.

4. **Effect of this keyword.** This keyword does not influence the results of the EM run itself.

*EM_DATABASE_CIRCUITSOURCE

Purpose: Output additional data specific to source circuits.

Card 1	1	2	3	4	5	6	7	8
Variable	CIRCID	IMP						
Type	I	I						
Default	none	0						

VARIABLE	DESCRIPTION
CIRCID	Source circuit ID for which total magnetic flux is calculated and output to em_CircuitSource_CIRCID.dat. See Remark 1.
IMP	Include the source circuit with ID CIRCID in the impedance matrix calculation and output: EQ.0: Not included. EQ.1: Included. See Remark 2.

Remarks:

1. **Magnetic flux.** This keyword causes an additional vector product operation on the current density and vector potential to give the magnetic flux for the specified source circuits. For the frequency domain solver, the output is decomposed into its real and imaginary components. This output is useful for retrieving coil inductance variations and resistance variations or induced voltage. It is frequently used in crack detection problems in conjunction with the frequency domain Eddy current solver. Since it relies on the vector potential output, the source circuit coil must be defined as a conductor. See Remark 2 in *EM_CIRCUIT_SOURCE).

2. **Impedance matrix.** IMP = 1 only applies when the frequency domain solver is also activated (EMSOL = 4 in *EM_CONTROL). For each CIRCID for which IMP = 1, the solver performs a separate solve with each CIRCID considered in turn as an emissive and receptive coil. This method enables establishing the complete impedance matrix. Its real and imaginary parts are output in ASCII files em_indMtx.dat and em_resMtx.dat, respectively. The results of em_resMtx.dat are multiplied by ω, where $\omega = 2\pi F$, for convenience purposes.

*EM_DATABASE_ELOUT

Purpose: Enable the output of EM data on elements.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	OUTLV	DTOUT						
Type	I	F						
Default	0	0.						

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	ELSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
OUTLV	Determines if LS-DYNA creates the output file: EQ.0: Does not generate the output file. EQ.1: Generates the output file.
DTOUT	Time interval to print the output. If DTOUT equals 0.0, LS-DYNA uses the EM time step.
ELSID	Solid elements set ID

Remarks:

1. **File name.** The file name for this database is em_elout.dat.

*EM_DATABASE_FIELDLINE

Purpose: The EM solver uses a BEM method to calculate the EM fields between conductors. With this method, the magnetic field in the air or vacuum between conductors is therefore not explicitly computed. However, in some cases, it may be interesting to visualize some magnetic field lines for a better analysis. This keyword allows the output of field line data. It has no influence on the results of the EM solve.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	FLID	PSID	DTOUT	NPOINT				
Type	I	I	F	I				
Default	none	none	0.	100				

Remaining cards are optional.†

Card 2	1	2	3	4	5	6	7	8
Variable	INTEG	H	HMIN	HMAX	TOLABS	TOLREL		
Type	I	F	F	F	F	F		
Default	2	0.	0.	1E10	1E-3	1E-5		

Card 3	1	2	3	4	5	6	7	8
Variable	BTYPE							
Type	I							
Default	2							

VARIABLE	DESCRIPTION
FLID	Field line set ID
PSID	Point Set ID associated to the field line set (See *EM_POINT_SET). The coordinates given by the different points will be the starting points of the field lines.
DTOUT	Time interval to print the output. If DTOUT is equal to 0.0, then the EM time step will be used.
NPOINT	Number of points per field line. The points are regularly spaced.
INTEG	Type of numerical integrator used to compute the field lines : EQ.1: RK4, Runge Kutta 4. See Remark 2. EQ.2: DOP853, Dormand Prince 8(5,3). See Remark 2.
H	Value of the step size. In case of an integrator with adaptive step size, it is the initial value of the step size.
HMIN	Minimal step size value. Only used in the case of an integrator with adaptive step size.
HMAX	Maximal step size value. Only used in the case of an integrator with adaptive step size.
TOLABS	Absolute tolerance of the integrator. Only used in the case of an integrator with adaptive step size.
TOLREL	Relative tolerance of the integrator. Only used in the case of an integrator with adaptive step size.
BTYPE	Method to compute the magnetic field : EQ.1: Direct method (every contribution is computed by the Biot Savart Law and summed up : very slow). EQ.2: Multipole method (approximation of the direct method using the multipole expansion). EQ.3: Multicenter method (approximation of the direct method using a weighted subset of points only in order to compute the magnetic field).

Remarks:

1. **File Names.** The file name for this database is em_fieldLine_*XX*_*YYY*.dat where *XX* is the field line ID and *YYY* is the point set ID defined in *EM_POINT_SET.

2. **Integrators.** The Runge Kutta 4 integrator is an explicit iterative method for solving ODEs. It is a fourth order method with a constant step size. The Dormand Prince 8(5,3) integrator is an explicit iterative method for solving IDEs. Particularly, this integrator is an embedded Runge Kutta integrator of order 8 with an adaptive step size. This integrator allows a step size control which is done though an error estimate at each step. The Dormand Prince 8(5,3) is a Dormand Prince 8(6) for which the 6^{th} order error estimator has been replaced by a 5^{th} order estimator with 3^{rd} order correction in order to make the integrator more robust.

*EM_DATABASE_GLOBALENERGY

Purpose: Enable the output of global EM energies.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	OUTLV	DTOUT						
Type	I	F						
Default	0	0.						

VARIABLE	DESCRIPTION
OUTLV	Determines if the output file should be dumped. EQ.0: No output file is generated. EQ.1: The output file is generated.
DTOUT	Time interval to print the output. If DTOUT is equal to 0.0, then the EM timestep will be used.

Remarks:

1. **Database Name.** The file name for this database is em_globEnergy.dat.

2. **Output Data.** The output file includes the global EM energies of the mesh, the air, and the source circuit. It also includes the global kinetic energy and the global plastic work energy.

*EM_DATABASE_NODOUT

Purpose: Enable the output of EM data on nodes.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	OUTLV	DTOUT						
Type	I	F						
Default	0	0.						

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	NSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
OUTLV	Determines if the output file should be dumped: EQ.0: No output file is generated. GE.1: The output file is generated and has this value in the file name as an ID. See Remark 1.
DTOUT	Time interval to print the output. If DTOUT is 0.0, the EM time step is used.
NSID	Node set ID

Remarks:

1. **File name.** The file name for this database is em_nodout_VAL.dat with VAL as the value assigned to OUTLV.

*EM_DATABASE_PARTDATA

Purpose: This keyword enables the output of EM data for every part defined. .

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	OUTLV	DTOUT						
Type	I	F						
Default	0	0.						

VARIABLE	DESCRIPTION
OUTLV	Determines if the output file should be dumped. EQ.0: No output file is generated. EQ.1: The output file is generated.
DTOUT	Time interval to print the output. If DTOUT is equal to 0.0, then the EM timestep will be used.

Remarks:

1. The file name for this database is em_partData_*XXX*.dat with *XXX* the part ID.
2. Outputs the part EM energies of the part as well as the Lorentz force. Also outputs the part kinetic energy and the part plastic work energy.

*EM_DATABASE_POINTOUT

Purpose: Enable the output of EM data on point sets.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	OUTLV	DTOUT						
Type	I	F						
Default	0	0.						

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	PTSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
OUTLV	Determines if LS-DYNA creates the output file: EQ.0: Do not generate the output file. EQ.1: Generate the output file.
DTOUT	Time interval to print the output. If DTOUT equals 0.0, LS-DYNA uses the ICFD time step.
PTSID	Point set ID (See *EM_POINT_SET card).

Remarks:

1. **File name.** The file name for this database is em_pointout.dat.

*EM_DATABASE_ROGO

Purpose: This keyword enables the output of EM data for every circuit defined. .

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	OUTLV	DTOUT						
Type	I	F						
Default	1	0.						

VARIABLE	DESCRIPTION
OUTLV	Determines if the output file should be dumped. EQ.0: No output file is generated. EQ.1: The output file is generated.
DTOUT	Time interval to print the output. If DTOUT is equal to 0.0, then the EM timestep will be used.

Remarks:

1. The file name for this database is em_rogoCoil_*XXX*.dat where *XXX* is the rogo Coil ID.

*EM_DATABASE_TIMESTEP

Purpose: This keyword enables the output of EM data regarding the EM timestep.

Output options card.

Card 1	1	2	3	4	5	6	7	8
Variable	OUTLV							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
OUTLV	Determines if the output file should be dumped. EQ.0: No output file is generated. EQ.1: The output file is generated.

Remarks:

1. The file name for this database is em_timestep.dat.

2. Outputs the run's EM tim estep versus the time step calculated using the EM CFL condition as criteria (autotimestep). This can be useful in cases with big deformations and/or material property changes and a fixed time step is being used in case that time step becomes to big compared to the stability time step.

*EM_EOS_BURGESS

Purpose: Define the parameters for a Burgess model giving the electrical conductivity as as a function of the temperature and the density, see:

T.J. Burgess, "Electrical resistivity model of metals", 4th International Conference on Megagauss Magnetic-Field Generation and Related Topics, Santa Fe, NM, USA, 1986

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	V0	GAMMA	THETA	LF	C1	C2	C3
Type	I	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Card 2	1	2	3	4	5	6	7	8
Variable	C4	K	EXPON	LGTUNIT	TIMUNIT	TEMUNI	ADJUST	
Type	F	F	I	F	F	I	I	
Default	none	none	none	none	none	none	none	

In the following, UUS stands for User Units System and BUS for Burgess Units

VARIABLE	DESCRIPTION
EOSID	ID of the EM_EOS (specified by an *EM_MAT card)
V0	Reference specific volume V_0 (UUS).
GAMMA0	Reference Gruneisen value γ_0.(no units).
THETA	Reference melting temperature $\theta_{m,0}$ in eV (BUS).
LF	Latent heat of fusion L_F in kJoule/mol (BUS).
C1	C1 constant (BUS)
C2	C2 constant (no units)

VARIABLE	DESCRIPTION
C3	C3 constant (no units)
C4	C4 constant (no units)
K	Parameter k (no units).
EXPON	Exponent in equations (2) (see remarks)
LGTUNIT	Length units for UUS (relative to meter, i.e. = 1.e-3 if UUS in mm).
TIMUNIT	Time units for UUS (relative to seconds).
TEMUNIT	Temperature units EQ.1: temperature in Celsius EQ.2: temperature in Kelvins
ADJUST	Conductivity modification EQ.0: (default) The conductivity is given by the Burgess formula. EQ.1: The conductivity is adjusted so that it is equal to the conductivity defined in *EM_MAT card σ_{mat} at room temperature: $$\sigma(\theta) = \sigma_{\text{Burgess}}(\theta)\frac{\sigma_{\text{mat}}}{\sigma_{\text{Burgess}}(\theta_{\text{room}})}$$

Remarks:

1. The Burgess model gives the electrical resistivity vs temperature and density for the solid phase, liquid phase and vapor phase. At this time, only the solid and liquid phases are implemented. To check which elements are in the solid and in the liquid phase, a melting temperature is first computed by:

$$\theta_m = \theta_{m,0}\left(\frac{V}{V_0}\right)^{-\frac{1}{3}} e^{(2\gamma_0-1)(1-\frac{V}{V_0})}$$

   a) If T < θ_m: solid phase model applies.

   The solid phase electrical resistivity corresponds to the Meadon model:

$$\eta_S = (C_1 + C_2\theta^{C_3})f_c\left(\frac{V}{V_0}\right), \quad (1)$$

where θ is the temperature, V is the specific volume, and V_0 is the reference specific volume (zero pressure, solid phase). In (1), the volume dependence is given by:

$$f_c\left(\frac{V}{V_0}\right) = \begin{cases} \left(\frac{V}{V_0}\right)^{2\gamma-1} & \text{EXPON.EQ.} -1 \quad \text{(most materials)} \\ \left(\frac{V}{V_0}\right)^{2\gamma+1} & \text{EXPON.EQ.} +1 \quad \text{(tungsten)} \\ \left(\frac{V}{V_0}\right)^{2\gamma} & \text{EXPON.EQ. 0} \quad \text{(stainless steel)} \end{cases} \tag{2}$$

with

$$\gamma = \gamma_0 - \left(\gamma_0 - \frac{1}{2}\right)\left(1 - \frac{V}{V_0}\right) \tag{3}$$

b) If $T > \theta_m$: liquid phase model:

$$\eta_L = (\eta_L)_{\theta_m}\left(\frac{\theta}{\theta_m}\right)^{C_4} \tag{4}$$

with

$$(\eta_L)_{\theta_m} = \Delta\eta(\eta_S)_{\theta_m}$$

where

$$\Delta\eta = \begin{cases} ke^{0.69L_F/\theta_m} & k > 0 \\ 1 + 0.0772(2 - \theta_m) & k = -1 \quad \text{(tungsten)} \\ 1 + 0.106(0.846 - \theta_m) & k = -2 \quad \text{(stainless steel SS-304)} \end{cases} \tag{5}$$

The following table reports some sets of parameters given by Burgess in his paper:

Parameter	Cu	Ag	Au	W	Al(2024)	SS(304)
V_0(cm³/gm)	0.112	0.0953	0.0518	0.0518	0.370	0.1265
γ_0	2.00	2.55	3.29	1.55	2.13	2.00
$\theta_{m,0}$ (BUS)	0.117	0.106	0.115	0.315	0.0804	0.156
L_F (BUS)	0.130	0.113	0.127	0.337	0.107	0.153
C_1 (BUS)	-4.12e-5	-3.37e-5	-4.95e-5	-9.73e-5	-5.35e-5	0

Parameter	Cu	Ag	Au	W	Al(2024)	SS(304)
C_2	0.113	0.131	0.170	0.465	0.233	0.330
C_3	1.145	1.191	1.178	1.226	1.210	0.4133
EXPON	-1	-1	-1	+1	-1	0
C_4	0.700	0.672	0.673	0.670	0.638	0.089
k	0.964	0.910	1.08	-1.	0.878	-2.

*EM_EOS_MEADON

Purpose: Define the parameters for a Meadon model, giving the electrical conductivity as a function of the temperature and the density; see:

T.J. Burgess, "Electrical resistivity model of metals", 4th International Conference on Megagauss Magnetic-Field Generation and Related Topics, Santa Fe, NM, USA, 1986

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	C1	C2	C3	TEMUNI	V0	GAMMA	EXPON
Type	I	F	F	F	I	F	F	I
Default	none	none	none	none	none	none	none	none

Card 2	1	2	3	4	5	6	7	8
Variable	LGTUNIT	TIMUNIT	ADJUST					
Type	F	F	I					
Default	none	none	none					

In the following, UUS stands for User Units System and BUS for Burgess Units.

VARIABLE	DESCRIPTION
EOSID	ID of the EM_EOS
C1	C1 constant (BUS)
C2	C2 constant (no units)
C3	C3 constant (no units)
TEMUNIT	Temperature units EQ.1: temperature in Celsius EQ.2: temperature in Kelvins

VARIABLE	DESCRIPTION
V0	Reference specific volume V0 (UUS).
GAMMA0	Reference Gruneisen value γ_0.(no units).
EXPON	Exponent in equations (7)
LGTUNIT	Length units for UUS (relative to meter, i.e. = 1.e-3 if UUS in mm).
TIMUNIT	Time units for UUS (relative to seconds).
ADJUST:	EQ.0: (default) the conductivity is given by the Burgess formula. EQ.1: The conductivity is adjusted so that it is equal to the conductivity defined in the *EM_MAT card σ_{mat}at room temperature: $$\sigma(\theta) = \sigma_{\text{Burgess}}(\theta)\frac{\sigma_{\text{mat}}}{\sigma_{\text{Burgess}}(\theta_{\text{room}})}$$

Remarks:

1. The Meadon model is a simplified Burgess model with the solid phase equations only.

   The electrical resistivity is given by:

   $$\eta_S = \left(C_1 + C_2\theta^{C_3}\right) f_c\left(\frac{V}{V_0}\right) \tag{6}$$

   where θ is the temperature, V is the specific volume, and V_0 is the reference specific volume (zero pressure, solid phase).

   In (6), the volume dependence is given by:

   $$f_c\left(\frac{V}{V_0}\right) = \begin{cases} \left(\frac{V}{V_0}\right)^{2\gamma-1} & \text{EXPON.EQ.}-1 \quad \text{(most materials)} \\ \left(\frac{V}{V_0}\right)^{2\gamma+1} & \text{EXPON.EQ.}+1 \quad \text{(tungsten)} \\ \left(\frac{V}{V_0}\right)^{2\gamma} & \text{EXPON.EQ.0} \quad \text{(stainless steel)} \\ 1 & \text{VO.EQ. 0} \quad \text{(default value for } V_0 \text{ is zero)} \end{cases} \tag{7}$$

   (In this last case, only EOSID, C1, C2, C3, TEMUNIT, TIMUNIT and LGTUNIT need to be defined)

   with,

$$\gamma = \gamma_0 - \left(\gamma_0 - \frac{1}{2}\right)\left(1 - \frac{V}{V_0}\right) \tag{8}$$

The following table reports some sets of parameters given by Burgess in his paper:

Parameter	Cu	Ag	Au	W	Al(2024)	SS(304)
V_0(cm³/gm)	0.112	0.0953	0.0518	0.0518	0.370	0.1265
γ_0	2.00	2.55	3.29	1.55	2.13	2.00
C_1 (BUS)	-4.12e-5	-3.37e-5	-4.95e-5	-9.73e-5	-5.35e-5	0
C_2	0.113	0.131	0.170	0.465	0.233	0.330
C_3	1.145	1.191	1.178	1.226	1.210	0.4133
EXPON	-1	-1	-1	+1	-1	0

*EM_EOS_PERMEABILITY

Purpose: Define the parameters for the behavior of a material's permeability.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	EOSTYPE	LCID	BS/POLY	MUR0/ REQ	C	TC	
Type	I	I	I	F	F	F	F	
Default	none	none	none	0.	0.	0.	0.	

VARIABLE	DESCRIPTION
EOSID	ID of the *EM_EOS
EOSTYPE	Define the type of EOS:

EQ.1: Permeability defined by B as a function of H curve ($B = \mu H$). The abscissa of the curve is H, and the ordinate is B.

EQ.2: Permeability defined by a H function of B curve ($H = B/\mu$). The abscissa of the curve is B, and the ordinate is H.

EQ.3: Permeability defined by a load curve where B-H curves ($B = \mu H$) may be defined as a function of von Mises stress. The abscissa of the curve is von Mises stress, and the ordinate is a load curve ID for the B-H load curves. The abscissa for each B-H load curve is H, and the ordinate is B.

EQ.4: Permeability defined by a load curve where B-H load curves ($B = \mu H$) may be defined as a function of temperature. The abscissa of the curve is temperature, and the ordinate is load curve ID for the B-H load curves. The abscissa for each B-H load curve is H, and the ordinate is B.

EQ.5: Permeability defined by a load curve where B-H curves ($B = \mu H$) may be defined as a function of an equivalent stress model calculated from uniaxial stress. The abscissa of the curve is uniaxial stress; the ordinate is a load curve ID for the B-H load

VARIABLE	DESCRIPTION
	curves. The abscissa for each B-H load curve is H, and the ordinate is B. See Remark 4.
	EQ.7: Permeability given by an analytical arctan law defining B as a function of H and optionally temperature. See Remark 1.
	EQ.8: Permeability given by the Froelich law defining B as a function of H and optionally temperature. See Remark 2.
	EQ.31: Same as 3, except the signed von Mises stress is used.
LCID	Load curve ID for EOSTYPE = 1, 2, 3, and 4. Ignored if EOSTYPE = 7 or 8.
BS/MUR0/C/TC	Saturation magnetic flux (B_s), relative permeability at a reference temperature (μ_{r0}), constant value (C), and Curie temperature (T_c), respectively. These parameters are used by the analytical laws (EOSTYPE = 7 and 8). See Remarks 1 and 2.
REQ	Equivalent tensile strain/stress at which the permeability is at a maximum, r_{eq}. This parameter is only used by EOSTYPE = 5 and is optional. See Remark 4.
POLY	Applies a polynomial smoothing on the input load curve for EOSTYPE = 1 and 2. See Remark 3.
	EQ.0: Off
	EQ.1: On

Remarks:

1. **Arctan analytical law.** For EOSTYPE = 7, the solver applies the following analytical law:

$$B(H,T) = \mu_0 H + \frac{2B_s}{\pi}\text{atan}\left(\frac{\pi}{2B_s}H\mu_0(\mu_{r0}-1.)\right) \times F(T) \times \text{SF}$$

SF is an optional coefficient defined by the material properties (see Remark 3 on *EM_MAT_002). $F(T)$ provides temperature dependence. B_s and μ_{r0} are input parameters. μ_0 is the permeability of free space and is hard-coded in the solver. By default, SF = 1 and $F(T) = 1$. If C is defined, $F(T)$ becomes:

$$F(T) = \max\left(0.,1. - \exp\left(\frac{T - T_c}{C}\right)\right)$$

T_c is an input parameters.

2. **Froelich analytical law.** For EOSTYPE = 8, the solver applies the following analytical law:

$$B(H,T) = \mu_0 H + H\frac{B_s}{1. + \mu_{r0}H} \times F(T) \times \text{SF}$$

SF is an optional coefficient defined by material properties (see Remark 3 on *EM_MAT_002). $F(T)$ provides temperature dependence. By default, SF = 1 and $F(T) = 1$. If C is defined, $F(T)$ becomes:

$$F(T) = \max\left(0.,1. - \exp\left(\frac{T - T_c}{C}\right)\right)$$

T_c is an input parameters.

3. **Polynomial approach.** For EOSTYPE = 1 or 2, in some cases the input B-H curve contains few points or has some ill-defined areas. In such scenarios, it can be useful to reconstruct the curve using a polynomial approach. Accuracy can increase with reduced computational time. The smoothed load curve is output to a keyword file em_polyCurve_LCID.dat for reference. It is not necessarily recommended to use this feature if the original B-H curve is well defined.

4. **Equivalent stress model.** For EOSTYPE = 5, an equivalent stress scalar value is calculated based on the stress tensor, $\boldsymbol{\sigma}$, taking into account the local direction of the magnetic flux field vector, $\mathbf{B} = B\mathbf{b}$, for magneto-strictive applications (Rasilo et al [2019]). Here, $\mathbf{b}$ is the direction vector, and B is the amplitude. If the parameter r_{eq} (REQ) is defined, the following expression is used:

$$\sigma_{\text{eq}} = \begin{cases} r_{\text{eq}} - \sqrt{\left(\mathbf{b}^t\left(r\mathbf{I} - \frac{3}{2}\mathbf{d}\right)^2\mathbf{b}\right)}, & \text{if } \mathbf{b}^t\mathbf{d}\mathbf{b} \le \frac{2r_{\text{eq}}}{3} \\ r_{\text{eq}} + \sqrt{\left(\mathbf{b}^t\left(r\mathbf{I} - \frac{3}{2}\mathbf{d}\right)^2\mathbf{b}\right)}, & \text{otherwise.} \end{cases}$$

Here, $\mathbf{d} = \boldsymbol{\sigma} - (1./3.)(\text{tr}\,\boldsymbol{\sigma})\mathbf{I}$. If r_{eq} is not defined or set to 0., the following expression is used:

$$\sigma_{\text{eq}} = \frac{3}{2}\mathbf{b}^t\mathbf{d}\mathbf{b}.$$

References:

Rasilo, P., U. Aydin, F. Martin, A. Belahcen, R. Kouhia, and L. Daniel, "Equivalent Strain and Stress Models for the Effect of Mechanical Loading on the Permeability of Ferromagnetic Materials," *IEEE Trans. Magn*, vol. 55, No. 6, pp. 1-4 June 2019.

*EM_EOS_TABULATED1

Purpose: Define an EM material property as a function of temperature by using a load curve. The interpretation of the EM material property depends on the *EM_MAT_... referencing this EOS. It can be conductivity, permeability, or permittivity.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	LCID						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
EOSID	ID of the EM EOS
LCID	Load curve ID. See Remark 1.

Remarks:

1. **Suggestions for the load curve.** The load curve describes the electrical conductivity or permeability (ordinate) as a function of the temperature (abscissa). You need to make sure the temperature and the EM material property given by the load curve are in the correct units. Also, we advise giving some bounds to the load curve (the EM material property at very low and very high temperatures) to avoid bad extrapolations of the EM material property if the temperature gets out of the load curve bounds.

*EM_EOS_TABULATED2

Purpose: Define an EM material property through a load curve, table, or function. The interpretation of the EM material property depends on the *EM_MAT_... referencing this EOS. It can be conductivity, permeability, or permittivity.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	LCID	IFLAG	IDFV				
Type	I	I	I	I				
Default	none	none	0	0				

VARIABLE	DESCRIPTION
EOSID	ID of the EM EOS
LCID	Load curve ID (see Remark 1), function ID (see *DEFINE_FUNCTION), table ID or 2D table ID. For the arguments for the *DEFINE_FUNCTION, see Remark 2.
IFLAG	If LCID is a table ID or 2D table ID, the EM material property is a function of temperature and material density. This flag dictates how the solver interprets the table. In other words, it specifies which property (temperature or material density) is the value for the table and which is the abscissa for load curves in the table: EQ.0: Temperature (value) indexes each EM material property (ordinate) versus material density (abscissa) load curve. EQ.1: Material density indexes each EM material property (ordinate) versus temperature (abscissa) load curve.
IDFV	Number of additional parameters (up to 10) that can be added to the list of *DEFINE_FUNCTION arguments. See Remark 2.

Remarks:

1. **Suggestions for the load curve.** The load curve describes the EM material property (ordinate) as a function of time (abscissa). You need to make sure the time and the EM material property given by the load curve are in the correct units. Also, we advise giving some bounds to the load curve (EM material

property at $t = 0$ and after a long time) to avoid bad extrapolations of the EM material property if the run time exceeeds the load curve bounds.

2. ***DEFINE_FUNCTION arguments.** LCID can also refer to a *DEFINE_FUNCTION ID. If a *DEFINE_FUNCTION is used, the following parameters are allowed: f(vx, vy, vz, temp, pres, vol, mass, Ex, Ey, Ex, Bx, By, Bz, Fx, Fy, Fz, JHrate, vmstress, efstrain, ieleuser, murel, time, x, y, z). (Fx, Fy, Fz) refers to the Lorentz force vector, JHrate is the Joule heat rate, vmstress is the is the von Mises stress, efstrain is the effective strain, ieleuser is the element ID, and murel is the is the relative permeability.If IDFV is nonzero, then the additional parameter names are save1, save2,.., save[IDFV]. IDFV cannot exceed 10. Adding an additional parameter enables you to have your own trackable local variable per element when defining complex conductivity behavior. For example, save1 could be used to represent and track a humidity factor that decreases with temperature. Note that for the variable value to be saved from the function, a "*" must proceed the variable name in the function. See *DEFINE_FUNCTION for an example.

*EM_EP_CELLMODEL_DEFINE_FUNCTION

Purpose: Define a user-defined ionic cell model for Electro-Physiology.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	NSTATE	FSWITCH					
Type	I	I	I					
Default	none	none	none					

Card 2	1	2	3	4	5	6	7	8
Variable	DVDT	DU1DT	DU2DT	DU3DT	DU4DT	DU5DT	DU6DT	DU7DT
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

Card 3	1	2	3	4	5	6	7	8
Variable	V0	U1	U2	U3	U4	U5	U6	U7
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
MID	Material ID defined in the *MAT section. If a negative value is entered, the cell model is defined on a node set instead of a part, and -MID is the node set where the cell model is defined.
FSWITCH	Flag for the ODE definition (see Remark 1): EQ.0: Functions EQ.1: Derivatives

VARIABLE	DESCRIPTION
NSTATE	Number of state variables $u_1, u_2, \dots, u_n$. The maximum value is 7 (see Cards 2 and 3).
DVDT	Function ID (see *DEFINE_FUNCTION) for evolution of V (function g in the equations in Remark 1).
DUiDT	Function ID (see *DEFINE_FUNCTION) for evolution of u_i (function f_i in the equations in Remark 1)
V0	Function ID (see *DEFINE_FUNCTION) for initial value of $V(x,y,z)$
Ui	Function ID (see *DEFINE_FUNCTION) for initial value of $u_i(x,y,z)$

Remarks:

1. **Model.** This keyword enables specifying a user-defined cell model through defined functions (see *DEFINE_FUNCTION). This model traces the evolution of the transmembrane potential, V, and n state variables: $u_1, u_{2,} \dots u_n$. Their temporal evolution depends upon FSWITCH.

   a) If FSWITCH = 0:

$$
\begin{aligned}
V(t) &= g\big(t, dt, V(t-1), u_1(t-1), u_2(t-1), \dots, u_n(t-1)\big) \\
u_1(t) &= f_1\big(t, dt, V(t-1), u_1(t-1), u_2(t-1), \dots, u_n(t-1)\big) \\
u_2(t) &= f_2\big(t, dt, V(t-1), u_1(t-1), u_2(t-1), \dots, u_n(t-1)\big) \\
&\vdots \\
u_n(t) &= f_n\big(t, dt, V(t-1), u_1(t-1), u_2(t-1), \dots, u_n(t-1)\big)
\end{aligned}
$$

   b) If FSWITCH = 1:

$$
\begin{aligned}
\frac{\partial V(t)}{\partial t} &= g\big(t, dt, V(t-1), u_1(t-1), u_2(t-1), \dots, u_n(t-1)\big) \\
\frac{\partial u_1(t)}{\partial t} &= f_1\big(t, dt, V(t-1), u_1(t-1), u_2(t-1), \dots, u_n(t-1)\big) \\
\frac{\partial u_2(t)}{\partial t} &= f_2\big(t, dt, V(t-1), u_1(t-1), u_2(t-1), \dots, u_n(t-1)\big) \\
&\vdots \\
\frac{\partial u_n(t)}{\partial t} &= f_3\big(t, dt, V(t-1), u_1(t-1), u_2(t-1), \dots, u_n(t-1)\big)
\end{aligned}
$$

2. **Benchmarks.** You can use this model to perform the electrophysiology benchmarks presented in Pathmanathan and Gray [2014].

References:

[1] Pathmanathan, P. and R.A. Gray, “Verification of computational models of cardiac electro-physiology,” Int J Numer Method Biomed Eng, vol. 30, No. 5, pp. 524-544, (2014).

*EM_EP_CELLMODEL_FENTONKARMA

Purpose: Define a Fenton-Karma ionic cell model for Electro-Physiology.

Card 1	1	2	3	4	5	6	7	8
Variable	MID							
Type	I							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	TAUD	TAUR	TAUSI	TAUO	TAUVP	TAUVM	TAUWP	TAUWM
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Card 3	1	2	3	4	5	6	7	8
Variable	UC	UCSI	K					
Type	F	F	F					
Default	none	none	none					

Card 4	1	2	3	4	5	6	7	8
Variable	U0	V0	W0					
Type	F	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
MID	Material ID defined in *MAT. If a negative value is entered, the cell model is defined on a node set instead of a part, and -MID is the node set where the cell model is defined.
TAUD	Time constant τ_d described in Equation 5
TAUR	Time constant τ_r described in Equation 6
TAUSI	Time constant τ_{si} described in Equation 7
TAU0	Time constant τ_0 described in Equation 6
TAUVP	Time constant τ_{vp} described in Equation 3
TAUVM	Time constant τ_{vm} described in Equation 3
TAUWP	Time constant τ_{wp} described in Equation 4
TAUWM	Time constant τ_{wm} described in Equation 4
UC	Threshold potential, u_c for activation of J_{fi} (the fast inward current) in Equations 3, 4, 5, and 6
UCSI	Threshold potential u_c^{si} for activation of J_{si} (the slow inward current) in Equation 7
K	Constant k in Equation 7
U0/V0/W0	Initial values of u, v, and w, respectively

Remarks:

The Fenton-Karma model is a simplified ionic model with three membrane currents that approximates well the restitution properties and spiral wave behavior of more complex ionic models of cardiac action potential (Beeler-Reuter and others). It was introduced in [1].

The total current flowing through the membrane is given by:

$$I_{\text{ion}} = -C_m \frac{\partial V}{\partial t} = -J_{\text{fi}} \tag{1}$$

where V is the transmembrane potential, C_m is the specific capacitance of the cell membrane, and J_{fi} is the fast inward current.

The model depends on three state variables, u, v, and w, and three membrane currents, J_{fi}, J_{so} (slow outward current), and J_{si} (slow inward current), through the following equations:

$$\frac{du}{dt} = -J_{\text{fi}} - J_{\text{so}} - J_{\text{si}} \tag{2}$$

$$\frac{dv}{dt} = \frac{\Theta(u_c - u)(1 - v)}{\tau_{vm}} - \frac{\Theta(u - u_c)v}{\tau_{vp}} \tag{3}$$

$$\frac{dw}{dt} = \frac{\Theta(u_c - u)(1 - w)}{\tau_{wm}} - \frac{\Theta(u - u_c)w}{\tau_{wp}} \tag{4}$$

$$J_{\text{fi}} = -\frac{\Theta(u_c - u)(1 - u)(u - u_c)}{\tau_d} \tag{5}$$

$$J_{\text{so}} = \frac{u\,\Theta(u_c - u)}{\tau_o} + \frac{u\,\Theta(u - u_c)}{\tau_r} \tag{6}$$

$$J_{\text{si}} = -\frac{w(1 + \tanh\left[k(u - u_c^{si}\right])}{2\tau_{si}} \tag{7}$$

In the above Θ is the Heaviside step function.

References:

[1] Fenton, F. & A. Karma, "Vortex dynamics in three-dimensional continuous myocardium with fiber rotation. Filament instability and fibrillations," *Chaos, Solitons, and Fractals*, Vol. 8, No. 1, pp. 661-686, (1998).

[2] https://www.ibiblio.org/e-notes/html5/fk.html

*EM_EP_CELLMODEL_FITZHUGHNAGUMO

Purpose: Define a Fitzhugh-Nagumo ionic cell model for Electro-Physiology.

Card 1	1	2	3	4	5	6	7	8
Variable	MID							
Type	I							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	ALPHA	BETA	GAMMA	C	MU1	MU2		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

Card 3	1	2	3	4	5	6	7	8
Variable	V	R						
Type	F	F						
Default	none	none						

VARIABLE	DESCRIPTION
MID	Material ID defined in *MAT. If a negative value is entered, the cell model is defined on a node set instead of a part, and -MID is the node set where the cell model is defined.
ALPHA	Excitation constant α described in Equation 1
BETA	Excitation constant β described in Equation 2

VARIABLE	DESCRIPTION
GAMMA	Excitation constant γ described in Equation 2
C	Excitation constant c described in Equation 1
MU1	Excitation constant μ_1 described in Equation 2
MU2	Excitation constant μ_2 described in Equation 2
V	Initial value of V
R	Initial value of r

Remarks:

In the Fitzhugh-Nagumo model, the excitation is defined by a cubic polynomial along with one recovery variable, r. The transmembrane current, I_{ion}, is given by:

$$I_{\text{ion}} = -C_m \frac{\partial V}{\partial t} = -cV\ (V-\alpha)(V-1) - rV \tag{1}$$

Here V is the transmembrane potential, C_m is the specific capacitance of the cell membrane, and c and α are excitation constants.

The recovery variable r evolves according to:

$$\frac{dr}{dt} = (\gamma + \frac{r\mu_1}{\mu_2 + V})(-r - cV(V-\beta-1)) \tag{2}$$

where β, γ, μ_1 and μ_2 are excitation constants.

References:

[1] Aliev, R.R. and Panfilov, A.V., "A simple two-variable model of cardiac excitation," *Chaos, Solitons, and Fractals*, Vol 7, No 3, pp 293-301, (1996).

[2] Pullan, A.J., Cheng, L.K., and Buist, M.L., Mathematically Modelling the Electrical Activity of the Heart, World Scientific Publishing Co. Pte. Ltd., Singapore, pp 132-133, (2005).

[3] Baillargeon, B. et al., "The Living Heart Project: A robust and integrative simulator for human heart function," *European Journal of Mechanics - A/Solids*. Vol 48, pp 38-47, (2014).

*EM_EP_CELLMODEL_STEP

Purpose: Define a simple step as an ionic cell model for Electro-Physiology.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	DURATION	RESTP	DEPOLP				
Type	I	F	F	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
MID	Material ID defined in *MAT
DURATION	Duration of the depolarization potential
RESTP	Resting potential
DEPOLP	Depolarization potential

Remarks:

Figure 6-3 illustrates the step ionic cell model. When the Electro-Physiology wave reaches the node, the membrane potential steps to the depolarization potential and lasts for the specified duration. Then, the membrane potential returns to the resting potential.

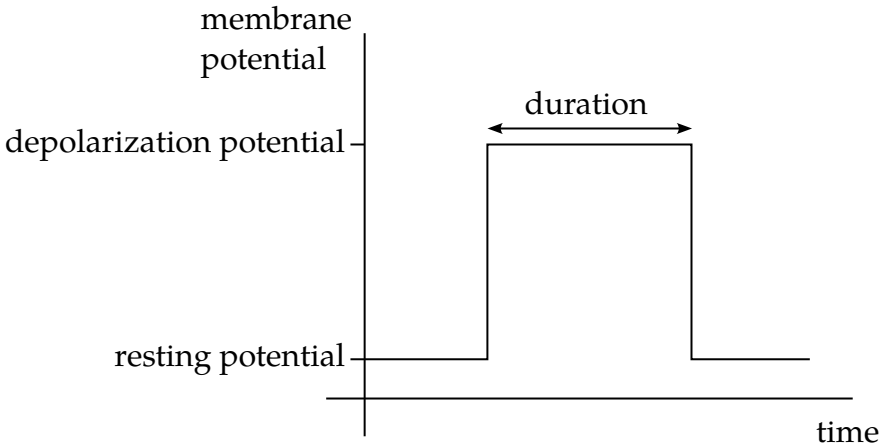


Figure 6-3. Illustration of the step ionic cell model

*EM_EP_CELLMODEL_TENTUSSCHER

Purpose: Define a ten Tusscher ionic cell model for Electro-Physiology.

Card Summary:

Card 1. This card is required.

MID							

Card 2. This card is required.

R	T	F	CM	VC	VSR	VSS	PKNA

Card 3. This card is required.

KO	NAO	CAO					

Card 4. This card is required.

GK1	GKR	GKS	GNA	GBNA	GCAL	GBCA	GTO

Card 5. This card is required.

GPCA	GPK						

Card 6. This card is required.

PNAK	KMK	KMNA	KNACA	KSAT	ALPHA	GAMMA	KMCA

Card 7. This card is required.

KMNAI	KPCA						

Card 8. This card is required.

K1	K2	K3	K4	EC	MAXSR	MINSR	

Card 9. This card is required.

VREL	VLEAK	VXFER	VMAXUP	KUP			

Card 10. This card is required.

BUFC	KBUFC	BUFSR	KBUFSF	BUFSS	KBUFSS		

Card 11. This card is required.

V	KI	NAI	CAI	CASS	CASR	RPRI	

Card 12. This card is required.

XR1	XR2	XS	M	H	J	D	F

Card 13. This card is required.

F2	FCASS	S	R				

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	MID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MID	Material ID defined in *MAT section. If a negative value is entered, the cell model is defined on a node set instead of a part, and -MID is the node set where the cell model is defined.

Card 2	1	2	3	4	5	6	7	8
Variable	R	T	F	CM	VC	VSR	VSS	PKNA
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
R	Gas constant ($J \times K^{-1} \times Mol^{-1}$)

VARIABLE	DESCRIPTION
T	Temperature (K)
F	Faraday constant (C × $mmol^{-1}$)
CM	Cell capacitance for unit surface area (μF × Cm^{-2})
VC	Cytoplasmic volume (μm^3)
VSR	Sarcoplasmic reticulum volume (μm^3)
VSS	Subspace volume (μm^3)
PKNA	Relative I_{Ks} permeability to Na^+

Card 3	1	2	3	4	5	6	7	8
Variable	KO	NAO	CAO					
Type	F	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
KO	Extracellular K^+ concentration (mM)
NAO	Extracellular Na^+ concentration (mM)
CAO	Extracellular Ca^{2+} concentration (mM)

Card 4	1	2	3	4	5	6	7	8
Variable	GK1	GKR	GKS	GNA	GBNA	GCAL	GBCA	GTO
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Card 5	1	2	3	4	5	6	7	8
Variable	GPCA	GPK						
Type	F	F						
Default	none	none						

VARIABLE	DESCRIPTION
GK1, GKR, GKS, GNA, GBNA, GCAL, GBCA, GTO, GPCA, GPK	Maximal I_{K1}, I_{Kr}, I_{Ks}, I_{Na}, I_{bNa}, I_{CaL}, I_{bCa}, I_{to}, I_{pCa}, and I_{pK} conductance, respectively (units: nS × pF^{-1})

Card 6	1	2	3	4	5	6	7	8
Variable	PNAK	KMK	KMNA	KNACA	KSAT	ALPHA	GAMMA	KMCA
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Card 7	1	2	3	4	5	6	7	8
Variable	KMNAI	KPCA						
Type	F	F						
Default	none	none						

VARIABLE	DESCRIPTION
PNAK	P_{NaK}, parameter for calculating the Na^+/K^+ pump current (units: pA × pF^{-1}). See Reference [1].

VARIABLE	DESCRIPTION
KMK, KMNA	K_{mK} and K_{mNa}, parameters for calculating the Na^+/K^+ pump current (units: millimolar). See Reference [1].
KNACA, KSAT, ALPHA, GAMMA, KMNAI	k_{NaCa}, k_{sat}, α, γ, and K_{mNai}, parameters for calculating the Na^+/Ca^{2+} exchanger current (units: millimolar). See Reference [1].
KPCA	K_{pCa}, parameter for calculating Ca^{2+} pump current (units: millimolar). See Reference [1].

Card 8	1	2	3	4	5	6	7	8
Variable	K1	K2	K3	K4	EC	MAXSR	MINSR	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	none	none	none	
Ref	2	2	2	2	2	2	2	

VARIABLE	DESCRIPTION
K1	R to O and RI to I I_{rel} transition rate ($mM^{-2} \times ms^{-1}$)
K2	O to I and R to RI I_{rel} transition rate ($mM^{-1} \times ms^{-1}$)
K3	O to R and I to RI I_{rel} transition rate (ms^{-1})
K4	I to O and RI to I I_{rel} transition rate (ms^{-1})
EC	Ca_{SR} half-saturation constant of k_{casr} (mM)
MAXSR/MINSR	Maximum and minimum values of k_{casr}

Card 9	1	2	3	4	5	6	7	8
Variable	VREL	VLEAK	VXFER	VMAXUP	KUP			
Type	F	F	F	F	F			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
VREL, VLEAK, VXFER, VMAXUP	Maximal I_{rel}, I_{leak}, I_{xfer}, and I_{up} conductance (mM × ms^{-1}), respectively. See Reference [2].
KUP	Half-saturation constant of I_{up} (mM). See Reference [2].

Card 10	1	2	3	4	5	6	7	8
Variable	BUFC	KBUFC	BUFSR	KBUFSF	BUFSS	KBUFSS		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		
Ref	2	2	2	2	2	2		

VARIABLE	DESCRIPTION
BUFC	Total cytoplasmic buffer concentration (mM)
KBUFC	Ca_i half-saturation constant for cytoplasmic buffer (mM)
BUFSR	Total sarcoplasmic buffer concentration (mM)
KBUFSR	Ca_{SR} half-saturation constant for sarcoplasmic buffer (mM)
BUFSS	Total subspace buffer concentration (mM)
KBUFSS	Ca_{SS} half-saturation constant for subspace buffer (mM)

Card 11	1	2	3	4	5	6	7	8
Variable	V	KI	NAI	CAI	CASS	CASR	RPRI	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	none	none	none	
Ref	2	1	1	2	2	2	2	

VARIABLE	DESCRIPTION
V	Initial value of transmembrane potential (mV)
KI	Initial value of K_i, used in potassium dynamics (mM)
NAI	Initial value of Na_i, used in sodium dynamics (mM)
CAI	Initial value of Ca_i. used in calcium dynamics (mM)
CASS	Initial value of Ca_{SS}, used in calcium dynamics (mM)
CASR	Initial value of Ca_{SR}, used in calcium dynamics (mM)
RPRI	Initial value of R', used in calcium dynamics

Card 12	1	2	3	4	5	6	7	8
Variable	XR1	XR2	XS	M	H	J	D	F
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none
Ref	1	1	1	1	1	1	2	2

VARIABLE	DESCRIPTION
XR1	Initial value of x_{r1}, used to compute the rapid time dependent K^+ current

VARIABLE	DESCRIPTION
XR2	Initial value of x_{r2}, used to compute the rapid time dependent K^+ current
XS	Initial value of x_s, used to compute slow time dependent K^+ current
M	Initial value of m, used to compute the fast Na^+ current
H	Initial value of h, used to compute the fast Na^+ current
J	Initial value of j, used to compute the fast Na^+ current
D	Initial value of d, used to compute the L-type Ca^{2+} current
F	Initial value of f, used to compute the L-type Ca^{2+} current

Card 13	1	2	3	4	5	6	7	8
Variable	F2	FCASS	S	R				
Type	F	F	F	F				
Default	none	none	none	none				
Ref	2	2	1	1				

VARIABLE	DESCRIPTION
F2	Initial value of f_2, used to compute the L-type Ca^{2+} current
FCASS	Initial value of f_{cass}, used to compute the L-type Ca^{2+} current
S	Initial value of s, used to compute the transient outward current
R	Initial value of r, used to compute the transient outward current

Remarks:

This is a model of the action potential of human ventricular cells that, while including a high level of electrophysiological detail, is computationally cost-effective enough to be applied in large-scale spatial simulations for the study of reentrant arrhythmias. Please see the references for details. This model is based on [2].

References:

[1] ten Tusscher, K.H.W.J., D. Noble, P.J. Noble, and A.V. Panfilov, “A model for human ventricular tissue,” Am J Physiol Heart Circ Physiol, vol 286, no 4, pp H1573-H1589, (2004).

[2] ten Tusscher, K.H.W.J. and A.V. Panfilov, “Alternans and spiral breakup in human ventricular tissue model,” Am J Physiol Heart Circ Physiol, vol 291, no 3, pp H1088-H1100, (2006).

*EM_EP_CELLMODEL_TOMEK

Purpose: Define a ToR-ORd model for cardiac electrophysiology [1].

> **WARNING:** As of version R15, this keyword is deprecated. Instead, use *EM_EP_CELLMODEL_TOR_ORD which is an updated version of this keyword.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	PHIEND	PHIMYO					
Type	I	F	F					
Default	none	0.0	0.0					

VARIABLE	DESCRIPTION
MID	Material ID: refers to MID in the *PART card. If a negative value is entered, the cell model is defined on a node set instead of a part, and -MID is the node set where the cell model is defined.
PHIEND	Value between 0.0 and 1.0 giving the ratio of the cardiac tissue to be considered endocardial in the ToR-Ord cell model.
PHIMYO	Value between 0.0 and 1.0 giving the ratio of the cardiac tissue to be considered myocardial in the ToR-Ord cell model.

References:

[1] Tomek J., Bueno-Orovio A., Passini E., Zhou X., Minchole A., Britton O., Bartolucci C., Severi S., Shrier A., Virag L., Varro A., and Rodriguez B., “Development, calibration, and validation of a novel human ventricular myocyte model in health, disease, and drug block,” Elife (2019).

*EM_EP_CELLMODEL_TOR_ORD

Purpose: Define a ToR-ORd model for cardiac electrophysiology [1].

> **NOTE:** This keyword is an updated version of *EM_EP_CELLMODEL_-TOMEK. For versions R15 and later, *EM_EP_CELLMODEL_-TOR_ORD is the preferred keyword, and *EM_EP_CELLMOD-EL_TOMEK is deprecated.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	LDID	PHIEND	PHIMYO				
Type	I	I	F	F				
Default	none	none	0.0	0.0				

VARIABLE	DESCRIPTION
MID	Material ID. It refers to MID in the *PART card. If a negative value is entered, the cell model is defined on a node set instead of a part, and -MID is the node set where the cell model is defined.
LDID	ID of the *EM_EP_LAPLACE_DIRICHLET solution to be used for transmural depth definition
PHIEND	Value between 0.0 and 1.0 giving the ratio of the cardiac tissue to be considered endocardial in the ToR-Ord cell model.
PHIMYO	Value between 0.0 and 1.0 giving the ratio of the cardiac tissue to be considered myocardial version in the ToR-Ord cell model.

References:

[1] Tomek J., Bueno-Orovio A., Passini E., Zhou X., Minchole A., Britton O., Bartolucci C., Severi S., Shrier A., Virag L., Varro A., and Rodriguez B., “Development, calibration, and validation of a novel human ventricular myocyte model in health, disease, and drug block,” Elife (2019).

*EM_EP_CELLMODEL_TROVATO

Purpose: Define a Trovato cellular action potential model for Electro-Physiology. This model is used to model cells Purkinje fibers ([Trovato et al., 2020]).

Card 1	1	2	3	4	5	6	7	8
Variable	MID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MID	Material ID defined in *MAT section. If a negative value is entered, the cell model is defined on a node set instead of a part, and -MID is the node set where the cell model is defined.

References:

Trovato C, Passini E, Nagy N, Varró A, Abi-Gerges N, Severi S, Rodriguez B., "Human Purkinje in silico model enables mechanistic investigations into automaticity and pro-arrhythmic abnormalities," *J Mol Cell Cardiol*. 2020 May;142:24-38.

*EM_EP_CELLMODEL_USERMAT

Purpose: Specify a user material for an ionic cell model to be used in electrophysiology simulations.

Card 1	1	2	3	4	5	6	7	8
Variable	MID							
Type	I							

VARIABLE	DESCRIPTION
MID	Material ID. A unique number must be specified (see *PART). If a negative value is entered, the cell model is defined on a node set instead of a part, and -MID is the node set where the cell model is defined.

*EM_EP_CREATEFIBERORIENTATION

Purpose: Define fiber orientation by solving a Laplace-Dirichlet system defined by *EM_EP_LAPLACE_DIRICHLET. This feature is based on [1].

Card 1	1	2	3	4	5	6	7	8
Variable	PSID	LDID1	LDID2	ALPHA	BETA	IEXPORT	IPRERUN	
Type	I	I	I	F	F	I	I	
Default	none	none	none	0.0	0.0	0	0	

VARIABLE	DESCRIPTION
PSID	Part set ID of the part set on which the system is solved
LDID1	ID of the Laplace system that is solved in the transmural direction
LDID2	ID of the Laplace system that is solved in the apicobasal direction
ALPHA	Helical angle with respect to the counterclockwise circumferential direction in the heart when looking from the base towards the apex. LT.0: |ALPHA| is the ID for the *DEFINE_FUNCTION giving the helical angle. See Remark 1 for available arguments.
BETA	Angle with respect to the outward transmural axis of the heart. LT.0: |BETA| is the ID for the *DEFINE_FUNCTION giving the angle. See Remark 1 for available arguments.
IEXPORT	Selects whether result files (ELEMENT_SOLID_ORTHO.k and vtk files) are exported: EQ.0: Not exported EQ.1: Exported
IPRERUN	Select whether the run is stopped after creating fibers: EQ.0: Do not stop after fiber creation EQ.1: Stop after fiber creation

Remarks:

1. ***DEFINE_FUNCTION Arguments.** The arguments for the function defined with *DEFINE_FUNCTION may include the following: f(x_ele, y_ele, z_ele, phi_len, phi_thi). Here phi_len and phi_thi are the potentials corresponding to the potentials solved in LDID1 and LDID2, respectively.

References:

[1] Bayer, J.D., Blake, R. C., Plank, G., and Trayanova, N. A., "A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models," Annals of biomedical engineering, 40(10), 2243-2254 (2012).

*EM_EP_ECG

Purpose: Compute pseudo-ECGs on a set of virtual points. LS-DYNA exports a file named em_ECG_{ECGID}.dat. This feature is available for EMSOL = 11, 12, 13, 15, and 16 (see *EM_CONTROL).

Card 1	1	2	3	4	5	6	7	8
Variable	ECGID	PSID						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
ECGID	ID of the ECG computation
PSID	Point set ID containing the list of virtual points on which the pseudo-ECGs are computed

*EM_EP_EIKONAL

Purpose: Set up the eikonal (EMSOL = 14) or reaction eikonal, RE- (EMSOL = 15) or RE+ (EMSOL = 16), models.

Card 1	1	2	3	4	5	6	7	8
Variable	EIKID	EIKPST	EIKSNS	EIKSDF				
Type	I	I	I	I				
Default	none	0	none	0				

RE Card. Additional parameters for the RE- and RE+ models (EMSOL = 15 and 16)

Card 2	1	2	3	4	5	6	7	8
Variable	FTYPE	FT	FA					
Type	I	F	F					
Default	none	none	none					

Card 3	1	2	3	4	5	6	7	8
Variable	SOLVETYP							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
EIKID	ID of the eikonal solver
EIKPST	Part set ID on which the eikonal solve is performed EQ.0: Perform the eikonal solve on the whole conducting mesh

VARIABLE	DESCRIPTION
EIKSNS	Node set ID where the seed (meaning initial activation time values) is set
EIKSDF	*DEFINE_FUNCTION ID for defining the seed value. Accepted arguments by the *DEFINE_FUNCTION are the node coordinates using the names "x_node, y_node, z_node." If no *DEFINE_-FUNCTION is used (EIKSDF = 0), the activation time is set to 0.0 on EIKSNS.
FTYPE	Type of foot current: EQ.1: Neic model. FA gives the foot current between the activation time and the activation time plus FT.
FT	Duration of the foot current
FA	Amplitude of the foot current
SOLVETYP	Type of eikonal solve: EQ.0: Simple eikonal that only computes activation times. It can only simulate for one cycle. EQ.1: Time-stepping multifront eikonal. This version supports simulating several cycles and reentries.

References:

[1] A. Neic et al., "Efficient computation of electrograms and ECGs in human whole heart simulations using a reaction-eikonal model," Journal of Computational Physics, 346 (2017), 191–211.

[2] P.L'Eplattenier et Al: An Eikonal Solver for Cardiac Electrophysiology in LS-DYNA, 2025 Ansys EMEA Transportation Summit and LS-DYNA Conference, October 2025.

*EM_EP_ISOCH

Purpose: Compute isochrones of activation/repolarization of a cardiac electrophysiology wave of one or several beats.

Card 1	1	2	3	4	5	6	7	8
Variable	ISOCHID	IDEPOL	DPLTHR	IREPOL	RPLTHR	CYCLEMIN	APDMIN	
Type	I	I	F	I	F	F	F	
Default	none	0	none	0	none	0.0	0.0	

VARIABLE	DESCRIPTION
ISOCHID	ID of the isochrone
IDEPOL	Flag to activate the computation of depolarization: EQ.0: Off EQ.1: On
DPLTHR	Amplitude threshold used for measuring depolarization
IREPOL	Flag to activate the computation of repolarization times: EQ.0: Off EQ.1: On
RPLTHR	Amplitude threshold used for measuring repolarization
CYCLEMIN	Minimum time between two consecutive depolarizations (used in the EP eikonal model with time stepping)
APDMIN	Minimum time between depolarization and repolarization (used in the EP eikonal model with time stepping)

*EM_EP_LAPLACE_DIRICHLET

Purpose: Define the boundary conditions of the Laplace-Dirichlet system to be solved to define fiber orientation, based on [1]. The system is solved by using *EM_EP_CREATE-FIBERORIENTATION. This keyword was formerly called *EM_EP_FIBERINITIAL in versions R14 and earlier.

Include as many of this card as needed. The next keyword ("*") card terminates this input.

Card 1	1	2	3	4	5	6	7	8
Variable	LDID	PID	STYPE	SID1	SID0			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
LDID	ID of the Laplace system to solve (define a new ID with each new card)
PID	Part ID on which the system is solved
STYPE	Set type for the boundary condition: EQ.1: Segment set EQ.2: Node set
SID1	Set on which a potential of value 1 is prescribed
SID0	Set on which a potential of value 0 is prescribed

References:

[1] Bayer, J.D., Blake, R. C., Plank, G., and Trayanova, N. A., "A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models," Annals of biomedical engineering, 40(10), 2243-2254 (2012).

*EM_EP_PURKINJE_NETWORK

Purpose: Define a Purkinje network that consists of conductive beams and lies on a given surface, based on Costabal et al [2016].

Card 1	1	2	3	4	5	6	7	8
Variable	PURKID	BUILDNET	SSID	MID	POINTSTX	POINTSTY	POINTSTZ	EDGELEN
Type	I	I	I	I	F	F	F	F

Card 2	1	2	3	4	5	6	7	8
Variable	NGEN	NBRINIT	NSPLIT	INODEID	IEDGEID			
Type	I	I	I	I	I			

VARIABLE	DESCRIPTION
PURKID	ID for the Purkinje network
BUILDNET	Flag to create Purkinje network: EQ.0: Purkinje network not created. EQ.1: New Purkinje network created.
SSID	Segment set on which the Purkinje network is lying
MID	Material ID defined in the *MAT section.
POINTSTX	*X* coordinate of the tree origin
POINTSTY	*Y* coordinate of the tree origin
POINTSTZ	*Z* coordinate of the tree origin
EDGELEN	Edge length
NGEN	Number of generations of branches
NBRINIT	Number of branches attached to the tree origin

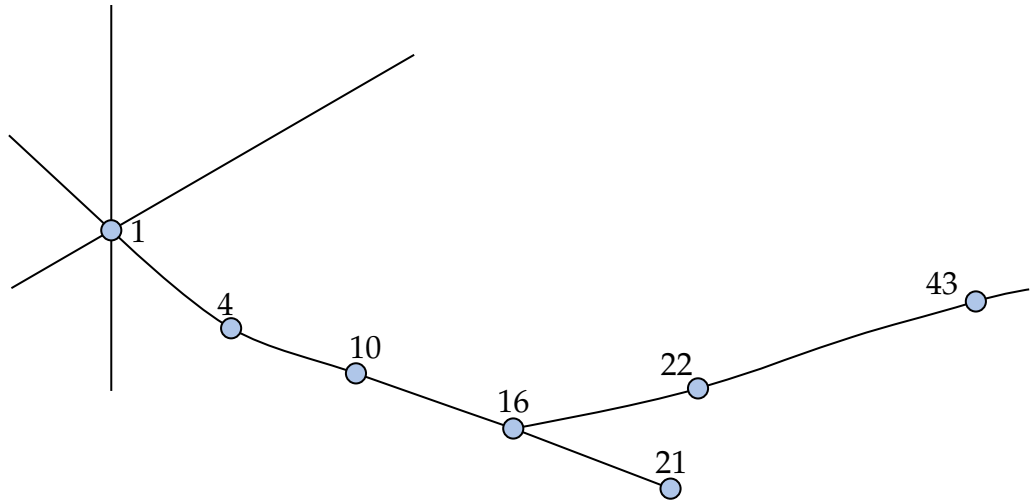


Figure 6-4. Example of part of a network. For this network, NSPLIT = 4.

VARIABLE	DESCRIPTION
NSPLIT	Number of nodes between two consecutive branchings as shown in Figure 6-4
INODEID	Initial node ID
IEDGEID	Initial edge ID. These edges are internally generated by LS-DYNA.

References:

Costabal, F.S., D. E. Hurtado, and E. Kuhl, "Generating Purkinje networks in the human heart," Journal of Biomechanics, vol. 49, issue 12, pp. 2455–2465, (2016).

*EM_EP_TENTUSSCHER_STIMULUS

Purpose: Define a stimulation pattern of a Tentusscher cell model (requires the use of *EM_EP_CELLMODEL_TENTUSSCHER).

Card 1	1	2	3	4	5	6	7	8
Variable	STIMID	SETTYPE	SETID					
Type	I	I	I					

Card 2	1	2	3	4	5	6	7	8
Variable	STIMSTRT	STIMT	STIMDUR	STIMAMP				
Type	F	F	F	F				

VARIABLE	DESCRIPTION
STIMID	ID of the stimulation
SETTYPE	Set type: EQ.1: Segment set EQ.2: Node set
SETID	Node set or segment set ID to be stimulated
STIMSTRT	Starting time of the stimulation
STIMT	Stimulation period
STIMDUR	Stimulation duration
STIMAMP	Stimulation amplitude (picoA/picoF)

*EM_EP_TENTUSSCHER_STIMULUS2

Purpose: Define a stimulation pattern for a cardiac cell model based on a load curve. This keyword is supported for all electro-physiology cell models, not just the ten Tusscher model.

Card 1	1	2	3	4	5	6	7	8
Variable	STIMID	SETTYPE	SETID	LCID				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
STIMID	ID of the stimulation
SETTYPE	Set type: EQ.1: Segment set EQ.2: Node set
SETID	Node set or segment set ID to be stimulated
LCID	Load curve ID for defining the stimulation current as a function of time

*EM_EXTERNAL_CHARGE

Purpose: Place a lumped point charge in the exterior of a FEM-BEM electrostatic model (EMSOL = 6 on *EM_CONTROL). See Remarks 1 and 2.

Card 1	1	2	3	4	5	6	7	8
Variable	Q	X	Y	Z	LCIDX	LCIDY	LCIDZ	
Type	F	F	F	F	I	I	I	
Default	none	0.	0.	0.	0	0	0	

VARIABLE	DESCRIPTION
Q	Charge value (units: Coulomb)
X, Y, Z	Point charge origin
LCID[X,Y,Z]	Optional load curve IDs that are applied on the X, Y, and Z coordinates, respectively, enabling defining moving point charges. These curves apply scaling factors on the point charge origin as a function of time.

Remarks:

1. **Charge effect on a BEM surface.** Each charge Q located at $\mathbf{r}_0$ produces the following incident field on BEM faces located at $\mathbf{r}$:

$$\mathbf{E}_{\text{inc}}(\mathbf{r}) = \frac{Q}{4\pi\varepsilon_0}\frac{\mathbf{r}-\mathbf{r}_0}{|\mathbf{r}-\mathbf{r}_0|^3}$$

2. **Point location.** The point must sit outside of any meshed conductor and ideally far enough away for it to be a valid point charge.

*EM_EXTERNAL_ECVEL

Purpose: Add an inertial velocity term to the vector potential FEM formulation to add velocity effects on Eddy currents. See Remarks 1 and 2.

Card 1	1	2	3	4	5	6	7	8
Variable	PID		NID1	NID2	OMEGA	VX	VY	VZ
Type	I		I	I	F	F	F	F
Default	none		none	none	0.	0.	0.	0.

VARIABLE	DESCRIPTION
PID	Part ID on which the velocity term will be added
NID1/NID2	Node IDs defining the origin (NID1) and rotation axis (pointing from NID2 to NID1)
OMEGA	Rotation velocity term (given in radians) GE.0.0: Constant value LT.0.0: |OMEGA| points to a load curve ID for a load curve giving the rotation velocity as a function of time.
V[X,Y,Z]	Translational velocity components: GE.0.0: Constant value LT.0.0: The absolute value of V[X,Y,Z] refers to a load curve ID for the load curve giving the translational velocity component as a function of time.

Remarks:

1. **Eddy current velocity effects.** In a typical EM solver application, the effects of velocity on Eddy currents are automatically taken into account by the translations and rotations of the different conductors in a Lagrangian formulation (thanks to the coupling with the mechanical solvers). However, it may be convenient to use an Eulerian formulation for certain applications, such as the study of homopolar generators (like a Faraday disk) or default detection in pipes (like

a Lenz's law apparatus). In this case, the conductors are static, and this keyword directly applies the velocity effects on the relevant conductors.

2. **Vector potential formulation.** When present, the velocity term appears in the FEM formulation for the vector potential **a**:

$$\sigma \frac{\partial \mathbf{a}}{\partial t} + \nabla \times \left(\frac{1}{\mu} \nabla \times \mathbf{a} \right) + \sigma \nabla \varphi - \sigma \mathbf{u} \times \mathbf{a} = \mathbf{J}_s$$

Here, σ is the material's conductivity, φ is the scalar potential, **u** is the Eulerian velocity, and $\mathbf{J}_s$ is the surface current.

*EM_EXTERNAL_FIELD

Purpose: Define the components of a time-dependent exterior field uniform in space applied to the conducting parts.

Card Summary:

Card 1. This card is required.

FIELDID	FTYPE	FDEF	LCIDX	LCIDY	LCIDZ		

Card 2a. This card is read when EMSOL = 4 (frequency-based Eddy current solver) on *EM_CONTROL for FTYPE = 1. It is optional.

			LCIDIX	LCIDIY	LCIDIZ		

Card 2b. Include this card when FTYPE = 4.

			LCIDPX	LCIDPY	LCIDPZ		

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	FIELDID	FTYPE	FDEF	LCIDX	LCIDY	LCIDZ		
Type	I	I	I	I	I	I		
Default	none	0	0	0	0	0		

VARIABLE	DESCRIPTION
FIELDID	External field ID
FTYPE	Field type: EQ.1: Magnetic flux density. It is supported for the eddy current and inductive heating solvers. EQ.2: Uniform external electric field density. It is only used by the FEM-BEM electrostatic solver (EMSOL = 6 on *EM_CONTROL). EQ.3: Charge density. This is supported for EMSOL = 3 and 6

VARIABLE	DESCRIPTION
	on *EM_CONTROL. See Remark 1.
	EQ.4: Planar electric field wave defined by direction and polarization (Helmholtz wave equation solver only, meaning EMSOL = 7 on *EM_CONTROL).
FDEF	Field defined by:
	EQ.1: Load curves giving the field as a function of time. This option is available for FTYPE = 1, 2, 3, and 4.
	EQ.2: Define function (see *DEFINE_FUNCTION). This option is available for FTYPE = 1 and 3. If a *DEFINE_FUNCTION is used, the following parameters are accepted: x, y, z, time. See Remark 2. For FTYPE = 3, the following additional parameters are accepted: emdt, pot, curr, sigma.
	EQ.4: Subroutine `usermat_getextfield` in the usermat package. This option is available for FTYPE = 1 only. See Remark 3. LCIDX, LCIDY, LCIDZ, LCIDIX, LCIDIY, and LCIDIZ are ignored in this case.
LCID[X,Y,Z]	Load curve (FDEF = 1) or *DEFINE_FUNCTION (FDEF = 2) IDs giving the (X, Y, Z) components of the magnetic field for FTYPE = 1. For FTYPE = 3, only LCIDY is used and should be a load curve (FDEF = 1) or *DEFINE_FUNCTION (FDEF = 2) ID giving the charge density. For FTYPE = 4, these load curves give the components of the planar wave direction.

Phase Shift Card. Additional input for frequency-based Eddy current cases (EMSOL = 4 in *EM_CONTROL).

Card 2a	1	2	3	4	5	6	7	8
Variable				LCIDIX	LCIDIY	LCIDIZ		
Type				I	I	I		
Default				optional	optional	optional		

VARIABLE	DESCRIPTION
LCIDI[X,Y,Z]	Load curve (FDEF = 1) or *DEFINE_FUNCTION (FDEF = 2) IDs defining the (X, Y, Z) components of the optional phase shift as a function of time for FTYPE = 1 when EMSOL = 4 on *EM_CONTROL. The unit is in degrees.

Plane Wave Polarization Card. Include this card for FTYPE = 4 (the planar electric field wave).

Card 2b	1	2	3	4	5	6	7	8
Variable				LCIDPX	LCIDPY	LCIDPZ		
Type				I	I	I		
Default				none	none	none		

VARIABLE	DESCRIPTION
LCIDP[X,Y,Z]	Load curve IDs defining the (X, Y, Z) components of the planar wave polarization as a function of time for FTYPE = 4

Remarks:

1. **Electrostatics problems.** FTYPE = 3 is mostly used in electrostatics problem configurations. Thus, it can be used with the resistive heating solver where the material's conductivity then represents permittivity (proceeding by analogy) or the FEM-BEM electrostatics solver where relative permittivity is defined in the material keyword and the charge density is applied to a meshed conductor part.

2. **Spatially varying magnetic fields.** Using FTYPE = 1 and FDEF = 2 enables defining a spatially varying field. The coordinates given as arguments by the *DEFINE_FUNCTION represent the coordinates at the midpoint of the BEM edges.

3. **Defining the field with usermat.** Using FTYPE = 1 and FDEF = 4 allows defining a spatially varying field in even greater detail by accessing the routine called `usermat_getextfield` in dyn21em.f when compiling LS-DYNA in usermat mode. With this subroutine, you can implement your own law and rules.

*EM_ISOPOTENTIAL

Purpose: Define an isopotential. In other words, constrain nodes so that they have the same scalar potential value. This keyword card can only be used with the type 3 EM solver.

Card 1	1	2	3	4	5	6	7	8
Variable	ISOID	SETTYPE	SETID	RDLTYPE				
Type	I	I	I	I				
Default	none	none	none	0				

VARIABLE	DESCRIPTION
ISOID	ID of the isopotential
SETTYPE	Set type: EQ.1: Segment Set EQ.2: Node Set EQ.3: Fluid surface part. See *ICFD_PART.
SETID	Set ID
RDLTYPE	Used for the battery application (see *EM_RANDLES_BATMAC or *EM_RANDLES_TSHELL). Selects which layers of the underlying battery cell are associated with the isopotential: EQ.0: Default. No specific treatment. EQ.1: Current Collector Positive EQ.2: Positive Electrode EQ.3: Separator EQ.4: Negative Electrode EQ.5: Current Collector Negative The function of a layer is defined in *EM_MAT_001.

*EM_ISOPOTENTIAL_CONNECT

Purpose: Define a connection between two isopotentials or between an isopotential and the ground.

Card Summary:

Card 1a. Include this card for CONTYPE = 1 (short circuit).

CONID	CONTYPE	ISOID1	ISOID2				

Card 1b. Include this card for CONTYPE = 2, 3, 4, 6, or 7.

CONID	CONTYPE	ISOID1	ISOID2	VAL	LCID		

Card 1c. Include this card for CONTYPE = 5 (meshless Randles circuit or meshless P2D circuit).

CONID	CONTYPE	ISOID1	ISOID2		RDLID	PSID	

Card 2a. Include this card for CONTYPE = 6 (RLC circuit).

L	C	V0					

Card 2b. Include this card for CONTYPE = 3 or 4 when the boundary condition includes a phase shift (frequency-based solvers, meaning EMSOL = 4, 7, and 9 on *EM_CONTROL, only).

				VAL2	LCID2		

Card 2c. This blank card is optional and is read in the cases not covered by Cards 2a and 2b. It must be included for those other cases if Card 3 is included.

Card 3. This card is optional. When used, always define Card 2 (even if fields can be left blank).

BT	DT						

Data Card Definitions:

Short Circuit Connection Card. Include this card if CONTYPE= 1.

Card 1a	1	2	3	4	5	6	7	8
Variable	CONID	CONTYPE	ISOID1	ISOID2				
Type	I	I	I	I				
Default	none	none	none	0				

VARIABLE	DESCRIPTION
CONID	Connection ID
CONTYPE	Connection type: EQ.1: Short circuit
ISOID1	ID of the first isopotential to be connected
ISOID2	Optional ID of the second isopotential to be connected

Resistance, Voltage Source, Current Source, and RLC Circuit Connection Card. Include this card if CONTYPE= 2, 3, 4, 6, or 7.

Card 1b	1	2	3	4	5	6	7	8
Variable	CONID	CONTYPE	ISOID1	ISOID2	VAL	LCID		
Type	I	I	I	I	F	I		
Default	none	none	none	0	none	0		

VARIABLE	DESCRIPTION
CONID	Connection ID
CONTYPE	Connection type: EQ.2: Resistance EQ.3: Voltage source

VARIABLE	DESCRIPTION
	EQ.4: Current source EQ.6: RLC circuit EQ.7: Power (See Remark 1)
ISOID1	ID of the first isopotential to be connected
ISOID2	Optional ID of the second isopotential to be connected
VAL	Value of the resistance, voltage, or current depending on CONTYPE. Ignored for CONTYPE = 2 through 4 if LCID ≠ 0.
LCID	Load curve ID giving the value of resistance, voltage, or current as a function of time. Only available for CONTYPE = 2 through 4. LT.0: |LCID| is a *DEFINE_FUNCTION ID. The following arguments are allowed: f(time, emdt, curr1, curr2, pot1, pot2, rmesh). pot1 and pot2 are the potential at the previous time step and at two previous time steps. curr1 and curr2 are the current at the previous time step and two previous time steps. rmesh is the mesh resistance calculated by the solver at this isopotential.

Meshless Randles or P2D Circuit Connection Card. Include this card if CONTYPE = 5.

Card 1c	1	2	3	4	5	6	7	8
Variable	CONID	CONTYPE	ISOID1	ISOID2		RDLID	PSID	
Type	I	I	I	I		I	I	
Default	none	none	none	0		none	0	

VARIABLE	DESCRIPTION
CONID	Connection ID
CONTYPE	Connection type: EQ.5: Meshless Randles circuit (used to represent a cell by one lumped Randles circuit) or meshless P2D circuit (single unit cell scaled to represent multiple unit cells in parallel)

VARIABLE	DESCRIPTION
ISOID1	ID of the first isopotential to be connected
ISOID2	Optional ID of the second isopotential to be connected
RDLID	ID of the Randles circuit defined by *EM_RANDLES_MESHLESS or ID for the P2D circuit defined by *EM_P2D_MESHLESS
PSID	Used if the variable R0TOTH of *EM_RANDLES_MESHLESS equals 1. This is a part set ID where the joule heating corresponding to the resistance r_0 in *EM_RANDLES_MESHLESS is added, averaged over the part set.

RLC Circuit Parameters Card. Only defined if CONTYPE = 6.

Card 2a	1	2	3	4	5	6	7	8
Variable	L	C	V0					
Type	F	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
L/C/V0	Circuit inductance, capacity, and initial voltage. VAL gives the resistance.

Optional Phase Shift Condition Card. Only defined if CONTYPE = 3 or 4 when using th0065 frequency domain solvers (EMSOL = 4, 7, or 9 on *EM_CONTROL).

Card 2b	1	2	3	4	5	6	7	8
Variable					VAL2	LCID2		
Type					F	I		
Default					0.	0		

VARIABLE	DESCRIPTION
VAL2/LCID2	Phase shift, φ, defined by a constant (VAL2) or by a load curve if LCID2 $\neq$ 0

Optional Blank Card. This blank card is optional and applies only in the cases not covered by Cards 2a and 2b. It must be included in those additional cases if Card 3 is included.

Card 2c	1	2	3	4	5	6	7	8
Variable								
Type								
Default								

Optional Birth/Death Time Card. This card is optional. Include this card if you wish to specify birth and death times for the isopotential connect condition.

Card 3	1	2	3	4	5	6	7	8
Variable	BT	DT						
Type	F	F						
Default	optional	optional						

VARIABLE	DESCRIPTION
BT	Birth time of the isopotential connection. If a negative value is entered, the absolute value points to a load curve ID specifying whether the connection is active as a function of time. Positive values in the ordinate means the condition is active, whereas a negative value means that the condition is deactivated.
DT	Death time of the isopotential connection. Only positive nonzero values are accepted.

Remarks:

1. **Power boundary condition.** This condition is only available when using the EM resistive heating solver (EMSOL = 3 in *EM_CONTROL). When active, the solver solves the resistive heating problem in two steps per time step. In the first solve, it applies a Dirichlet boundary condition with an imposed voltage of 1 to each power boundary condition. Once the first solve is complete, it retrieves the current on the boundary to update the voltage boundary condition to the value corresponding to the user-defined expected value for power and solves again.

   In classic resistive heating applications, the classic Ohm's law is used to determine the relationship between power, voltage, and current:

   $$P = VI$$

   In radiofrequency applications (meaning when complex numbers are present), power is found by calculating the sum of the real and imaginary parts:

   $$P = V_r I_r + V_i I_i$$

*EM_ISOPOTENTIAL_ROGO

Purpose: measures the total current flowing through a given section of the conductor and outputs it in an ASCII file called em_rogoCoil.dat

Card 1	1	2	3	4	5	6	7	8
Variable	ISOID	SETTYPE	SETID					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
ISOID	ID of the Rogo coil.
SETTYPE	Set type: EQ.1: Segment Set.
SETID	Set ID

*EM_MAT_001

Purpose: Define the electromagnetic material type and properties for a material whose permeability equals the free space permeability. Electromagnetic materials act like a *MAT_ADD, so they must be associated with a *MAT material or an ICFD part. See Remark 1.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	MTYPE	SIGMA	EOSID		EPLAMDA	DEATHT	RDLTYPE
Type	I	I	F	I		F	F	I
Default	none	none	none	optional		optional	10^{28}	0

Optional Dielectric Properties Card. Additional material parameters for the radiofrequency heating (EMSOL = 9 on *EM_CONTROL), quasistatic electrostatic (EMSOL = 8 on *EM_CONTROL), Helmholtz wave (EMSOL = 7 on *EM_CONTROL), and FEM-BEM electrostatic (EMSOL = 6 on on *EM_CONTROL) cases.

Card 2	1	2	3	4	5	6	7	8
Variable			EPSRR	EOSID2	EPSRI	EOSID3		
Type			F	I	F	I		
Default			0.	optional	0.	optional		

VARIABLE	DESCRIPTION
MID	Material ID. For all MTYPE except MTYPE = 3, MID must reference a *MAT material since the electromagnetic properties are added onto the *MAT properties. If MTYPE = 3, MID can be left blank for the electromagnetic properties to apply to the ICFD entire fluid or it can be the PID of a *ICFD_PART_VOL to apply to an ICFD fluid part. See Remark 1.
MTYPE	Defines the electromagnetism type of the material: EQ.0: Air or vacuum EQ.1: Insulator material. These materials have the same

VARIABLE	DESCRIPTION
	electromagnetism behavior as MTYPE = 0. EQ.2: Conductor carrying a source. In these conductors, the eddy-current problem is solved, yielding the actual current density. Typically, this material would correspond to the coil. In Electro-Physiology, it corresponds to the tissue when solving the monodomain equations (EMSOL = 11 or 13 on *EM_CONTROL). For this case, an *EM_EP_CELLMODEL must be associated with this *EM_MAT_001. EQ.3: Fluid conductor. In this case, MID refers to the ID given in *ICFD_PART_VOL. See Remark 2. EQ.4: Conductor not connected to any current or voltage source, where the eddy current problem is solved. Typically, this material would correspond to the workpiece.
SIGMA	Initial electrical conductivity of the material. For the eikonal solvers (EMSOL = 14 and 15 on *EM_CONTROL), SIGMA is the wave velocity.
EOSID	Optional ID of the EOS to be used for the electrical conductivity (see *EM_EOS cards).
EPLAMDA	Optional. When defined, this field activates the computation of extracellular potentials in the Purkinje network with the anisotropy ratio given by EPLAMDA.
DEATHT	Death time for the material. After DEATHT, the material will no longer be considered a conductor and will be removed from the EM solve. If a negative value is entered, a *DEFINE_FUNCTION will be expected. The following parameters are allowed: (vx, vy, vz, temp, vol, mass, Ex, Ey, Ez, Bx, By, Bz, Fx, Fy, Fz, JHrate, time). Fx, Fy, and Fz denote the components of the Lorentz force vector. A negative value returned by the *DEFINE_FUNCTION corresponds to a "dead" or inactive element. Once an element has been removed from the EM solve, it cannot return.
RDLTYPE	Used for composite tshell batteries modeled with *EM_RANDLES_TSHELL. RDLTYPE specifies the function of the layer associated with MID: EQ.0: Default. A conductor that is not part of a battery cell.

VARIABLE	DESCRIPTION
	EQ.1: Positive current collector EQ.2: Positive electrode EQ.3: Separator EQ.4: Negative electrode EQ.5: Negative current collector
EPSRR/EPSRI	Real and imaginary parts of the relative permittivity (dielectric constant and dielectric loss terms). See Remark 4.
EOSID2/EOSID3	Optional IDs defining equation of states for EPSRR and EPSRI, respectively.

Remarks:

1. **Input decks with EM materials.** EM materials must be associated with a *MAT material for MTYPE ≠ 3 or an ICFD fluid for MTYPE = 3. These EM properties are added onto the normal material properties, like *MAT_ADD material properties. Thus, when added onto a *MAT material, MID here must refer to the MID of the *MAT material. For example, an input deck can have the following to define a conductor:

```
*PART
2,1,20
*MAT_RIGID
20,&dens,&young,$nu
*EM_MAT_001
20,2,&cond
```

   In this case, if *MAT_RIGID is not included, the conduction properties are not applied.

   For MTYPE = 3, MID can refer to the PID of a *ICFD_PART_VOL or be left blank. If it is left blank, the EM properties apply to the entire ICFD fluid.

2. **Coupling ICFD to EM.** Only the resistive heating solver is currently available when coupling the ICFD solver with the EM solver (see *EM_CONTROL).

3. **Eikonal solvers.** For the eikonal solvers (EMSOL = 14 and 15 on *EM_CONTROL), this material is supported for beam elements.

4. **Dielectric properties.** Relative permittivity is a material parameter present in solvers that involve dielectric properties (EMSOL = 6, 7, 8, and 9 on *EM_CONTROL). For the time-based solvers (EMSOL = 6 and 8), EPSRR and EOSID2 directly define the relative permittivity. For the frequency-based solvers,

permittivity is a complex number ($\varepsilon = \varepsilon' - j\varepsilon''$) that can be divided into its real component (EPSRR or EOSID2) and its imaginary component (EPSRI or EOSID3). The imaginary component, ε'', of the relative permittivity enables calculating dielectric losses that are automatically coupled to the thermal solver:

$$P_d = 0.5\omega\varepsilon''(|\mathbf{E}|)^2$$

In the above, $|\mathbf{E}|$ is the norm of the complex electric field and ω is the angular frequency.

*EM_MAT_002

Purpose: Define an electromagnetic material type and properties with a permeability different than the free space permeability. Electromagnetic materials act like a *MAT_ADD, so they must be associated with a *MAT material. See Remark 1. This electromagnetic material is supported for EMSOL = 1, 2, 4, and 7 on *EM_CONTROL. We recommend using the monolithic solver (*EM_SOLVER_FEMBEM_MONOLITHIC); see Remark 2.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	MTYPE	SIGMA	EOSID	MUREL	EOSMU	DEATHT	
Type	I	I	F	I	F	I	F	
Default	none	none	none	none	none	optional	10^{28}	

Additional Permeability Properties Card. This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable		EOSIDSF			MULOSS	EOSLOSS		
Type		I			F	I		
Default		optional			optional	optional		

Dielectric Properties Card. This card is optional. It provides additional material parameters for Helmholtz wave equation (EMSOL = 7 on *EM_CONTROL) cases.

Card 3	1	2	3	4	5	6	7	8
Variable			EPSRR	EOSID2	EPSRI	EOSID3		
Type			F	I	F	I		
Default			0.	optional	0.	optional		

VARIABLE	DESCRIPTION
MID	Material identification. MID must reference a *MAT material since the electromagnetic properties are added onto the *MAT properties. See Remark 1.
MTYPE	Electromagnetism type of the material: EQ.0: Air or vacuum EQ.1: Insulator material. These materials have the same electromagnetism behavior as MTYPE = 0. EQ.2: Conductor carrying a source. In these conductors, the eddy current problem is solved, which gives the actual current density. Typically, this would correspond to the coil. EQ.4: Conductor not connected to any current or voltage source, where the Eddy current problem is solved. Typically, this would correspond to the workpiece.
SIGMA	Initial electrical conductivity of the material
EOSID	ID of the EOS to be used for the electrical conductivity (see *EM_EOS cards)
MUREL	Relative permeability, which is the ratio of the permeability of a specific medium to the permeability of free space ($\mu_r = \mu/\mu_0$). For frequency-based solvers, this is the real component of the relative permeability. See Remark 2.
EOSMU	ID of the EOS to be used to define the nonlinear behavior of μ. It is optional and only available for EMSOL = 1 on *EM_CONTROL. Note that if EOSMU is defined, MUREL will be used for the initial value only. See *EM_EOS_PERMEABILITY.
DEATHT	Death time for the material. After DEATHT, the material will no longer be considered a conductor and will be removed from the EM solve. If a negative value is entered, a *DEFINE_FUNCTION will be expected. The following parameters are allowed: (vx, vy, vz, temp, vol, mass, Ex, Ey, Ez, Bx, By, Bz, Fx, Fy, Fz, JHrate, time). Fx, Fy, and Fz refer to the components of the Lorentz force vector. A negative value returned by the *DEFINE_FUNCTION corresponds to a 'dead' or inactive element. Once an element has been removed from the EM solve, it cannot return.
EOSIDSF	For EMSOL = 1, this is an optional EOS for specifying a scale

VARIABLE	DESCRIPTION
	factor to the permeability EOS defined in EOSMU. This field only applies if EOSMU is defined, but it is optional. If used, the EOS must be either *EM_EOS_TABULATED1 or *EM_EOS_TABULATED2. See Remark 3. For frequency-based solver (EMSOL = 4 and 7 on *EM_CONTROL), this directly defines how the real component of the relative permeability should behave. See Remark 4.
MULOSS/ EOSLOSS	Optional magnetic losses are defined by the loss tangent. The loss tangent can either be a constant (MULOSS) or defined with an EOS (EOSLOSS). Magnetic losses are only present in frequency-based solvers (EMSOL = 4 or 7 on *EM_CONTROL). See Remark 4.
EPSRR/EPSRI	Real and imaginary parts of the relative permittivity (dielectric constant and dielectric loss terms). Only present in relevant frequency-based solvers (EMSOL = 7 on *EM_CONTROL). See Remark 5.
EOSRR/EOSRI	Optional IDs defining equation of states for EPSRR and EPSRI, respectively

Remarks:

1. **Input decks with EM materials.** EM materials must be associated with a *MAT material. These EM properties are added onto the normal material properties like *MAT_ADD material properties. Thus, MID here must refer to the MID of the *MAT material. For example, an input deck can have the following to define a conductor:

```
*PART
2,1,20
*MAT_RIGID
20,&dens,&young,$nu
*EM_MAT_002
20,2,&cond,&eosid,&murel
```

   In this case, if *MAT_RIGID is not included, the conduction properties are not applied.

2. **Solver recommendations.** If $\mu \neq \mu_0$, then it is recommended to switch to the monolithic solver (See *EM_SOLVER_FEMBEM_MONOLITHIC) for better stability.

3. **Scale factor applied on permeability.** For EMSOL = 1 on *EM_CONTROL, EOSIDSF is an equation of state for specifying a scale factor to the permeability specified with EOSMU. The scale factor is applied in the following way:

$$B(H,T) = \mu_0 H + G(H) \times \text{SF}$$

In other words, the nonlinear behavior of $B(H,T)$ is decomposed into a linear part $(\mu_0 H)$ and a nonlinear part $(G(H) \times \text{SF})$. The linear part represents what happens at very high H values, such as when saturation is reached. $G(H)$ in the nonlinear term depends on the curve or law specified with EOSMU. The scale factor, SF, is determined with EOSIDSF. In general, this scale factor is a function of temperature, but the capability is not limited to it (see options available in *EM_EOS_TABULATED1 and *EM_EOS_TABULATED2).

4. **Complex permeability.** For frequency-based solvers, relative permeability is a material property that can be defined as a complex number: $(\mu_r = \mu_r' - j\mu_r'')$. The imaginary part is typically represented by the loss tangent, which is simply the ratio between the imaginary and real parts $(\tan\delta = \mu_r''/\mu_r')$. MULOSS and EOSLOSS offer the option to define the loss tangent either as a constant (MULOSS) or through an EOS (EOSLOSS). When present, the magnetic losses trigger an extra magnetic heating term, also known as hysteresis energy loss, automatically coupled with the thermal solver:

$$P_m = 0.5\omega\mu''(|\mathbf{H}|)^2 \quad .$$

In the above, $|\mathbf{H}|$ is the norm of the complex magnetic field intensity. The real component of the real relative permeability, μ_r', can also be made a function of temperature or other quantities through the optional definition of EOSIDSF.

5. **Dielectric properties.** In frequency-based solvers, permittivity is a material property that can be defined as a complex number $(\varepsilon = \varepsilon' - j\,\varepsilon'')$. This complex number can be divided into its real component (EPSRR/EOSID2) and its imaginary component (EPSRI/EOSID3). The imaginary component, ε'', of the relative permittivity allows us to calculate dielectric losses that are automatically coupled to the thermal solver:

$$P_d = 0.5\omega\varepsilon''(|\mathbf{E}|)^2 \quad .$$

In the above, $|\mathbf{E}|$ is the norm of the complex electric field.

*EM_MAT_003

Purpose: Define an electromagnetic material type whose electromagnetic conductivity is defined by a (3 × 3) tensor matrix. Applications include composite materials. This material only applies to solid elements. Electromagnetic materials act like a *MAT_ADD, so they must be associated with a *MAT material. See Remark 1.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	MTYPE	SIGMA11	SIGMA22	SIGMA33	BETA	CM	
Type	I	I	F	F	F	F	F	

Card 2	1	2	3	4	5	6	7	8
Variable	SIGMA12	SIGMA13	SIGMA21	SIGMA23	SIGMA31	SIGMA32	AOPT	LAMBDA
Type	F	F	F	F	F	F	F	F

Card 3	1	2	3	4	5	6	7	8
Variable	XP	YP	ZP	A1	A2	A3	MACF	
Type	F	F	F	F	F	F	I	

Card 4	1	2	3	4	5	6	7	8
Variable	V1	V2	V3	D1	D2	D3		
Type	F	F	F	F	F	F		

VARIABLE	DESCRIPTION
MID	Material ID. . MID must reference a *MAT material since the electromagnetic properties are added onto the *MAT properties. See Remark 1.
MTYPE	Defines the electromagnetism type of the material:

VARIABLE	DESCRIPTION
	EQ.0: Air or vacuum EQ.1: Insulator material. These materials have the same electromagnetism behavior as MTYPE = 0. EQ.2: Conductor carrying a source. For these conductors, the EM solver solves the eddy current problem, which gives the actual current density. Typically, this type would correspond to the coil. In electrophysiology (EP), it corresponds to the tissue where the monodomain equations are solved for EMSOL = 11 or EMSOL = 13. For this case, an *EM_EP_CELLMODEL must be associated with this *EM_MAT_003. EQ.4: Conductor not connected to any current or voltage source, where the Eddy current problem is solved. Typically, this type would correspond to the workpiece. In electrophysiology (EP), for EMSOL = 11, 12 or 13, it corresponds to the bath surrounding the tissue for which only the external potential is found. In the case of electrophysiology, no *EM_EP_CELLMODEL should be associated with this material.
SIGMA11	The 1, 1 term in the 3 × 3 electromagnetic conductivity tensor matrix. For the eikonal solvers (EMSOL = 14 and 15 on *EM_CONTROL), this is the 1, 1 term in the 3 × 3 wave velocity tensor. Note that 1 corresponds to the **a** material direction. LT.0.0: |SIGMA11| corresponds to the ID of a *DEFINE_FUNCTION. See Remark 3 for available parameters.
SIGMA12	The 1, 2 term in the 3 × 3 electromagnetic conductivity tensor matrix. For the eikonal solvers (EMSOL = 14 and 15 on *EM_CONTROL), this is the 1, 2 term in the 3 × 3 wave velocity tensor. Note that 2 corresponds to the **b** material direction. LT.0.0: |SIGMA12| corresponds to the ID of a *DEFINE_FUNCTION. See Remark 3 for available parameters.
⋮	⋮
SIGMA33	The 3, 3 term in the 3 × 3 electromagnetic conductivity tensor matrix. For the eikonal solvers (EMSOL = 14 and 15 on *EM_CONTROL), this is the 3, 3 term in the 3 × 3 wave velocity tensor. Note that 3 corresponds to the **c** material direction. LT.0.0: |SIGMA33| corresponds to the ID of a *DEFINE_

VARIABLE	DESCRIPTION
	FUNCTION. See Remark 3 for available parameters.
BETA	Surface to volume ratio of the cell membrane (to be used only when EMSOL = 11 or 12 in *EM_CONTROL).
CM	Membrane capacitance per unit area (to be used only when EMSOL = 11 or 12 in *EM_CONTROL).
AOPT	Material axes option (see *MAT_002 for a more detailed description): EQ.0.0: Locally orthotropic with material axes determined by element nodes. The **a**-direction is from node 1 to node 2 of the element. The **b**-direction is orthogonal to the **a**-direction and is in the plane formed by nodes 1, 2, and 4. EQ.1.0: Locally orthotropic with material axes determined by a point in space, *P*, and the global location of the element center; this is the **a**-direction. EQ.2.0: Globally orthotropic with material axes determined by vectors defined below, as with *DEFINE_COORDINATE_VECTOR. EQ.3.0: Locally orthotropic material axes determined by a vector **v** and the normal vector to the plane of the element.The plane of a solid element is the midsurface between the inner surface and outer surface defined by the first four nodes and the last four nodes of the connectivity of the element, respectively. Thus, for solid elements, AOPT = 3 is only available for hexahedrons. **a** is determined by taking the cross product of **v** with the normal vector, **b** is determined by taking the cross product of the normal vector with **a**, and **c** is the normal vector. Then **a** and **b** are rotated about **c** by an angle BETA. BETA may be set in the keyword input for the element. EQ.4.0: Locally orthotropic in cylindrical coordinate system with the material axes determined by a vector, **v**, and an originating point, *P*, which define the centerline axis.
XP, YP, ZP	Coordinates of point, *P*, for AOPT = 1 and 4
A1, A2, A3	Components of vector, **a**, for AOPT = 2

VARIABLE	DESCRIPTION
MACF	Material axes change flag for solid elements: EQ.1: No change, default
V1, V2, V3	Components of vector, **v**, for AOPT = 3 and 4.
D1, D2, D3	Components of vector, **d**, for AOPT = 2
LAMBDA	Intra-to-extracellular conductivity ratio. When non-empty, the elliptic equation is solved to compute extracellular potentials (to be used only when EMSOL = 11, 15, or 16 in *EM_CONTROL).

Remarks:

1. **Input decks with EM materials.** EM materials must be associated with a *MAT material. These EM properties are added onto the normal material properties like *MAT_ADD material properties. Thus, MID here must refer to the MID of the *MAT material. For example, an input deck can have the following to define a conductor:

```
*PART
2,1,20
*MAT_RIGID
20,&dens,&young,$nu
*EM_MAT_003
20,2,&cond11,&cond22,&cond33
&cond12,&cond13,&cond21,&cond23,&cond31,&cond32
```

   In this case, if *MAT_RIGID is not included, the conduction properties are not applied.

2. **Material directions.** See the manual page for *MAT_002 for a description of how the principal material directions, {**a**, **b**, **c**}, are determined with AOPT. Note that *EM_MAT_003 only works for solid elements. The AOPT options illustrated in the AOPT figure of *MAT_002 can define the material directions for all elements of the parts that use the material.

3. **Function parameters.** The available parameters for the *DEFINE_FUNCTIONs are: f(time, emdt, x_ele, y_ele, z_ele, ieleuser). Here, time is the current EM time; emdt is the current EM time step; x_ele, y_ele, and z_ele are the location of the element; and ieleuser is the element ID.

4. **Eikonal solvers.** For the eikonal solvers (EMSOL = 14 and 15 on *EM_CONTROL), this material is supported for tetrahedral elements.

*EM_MAT_004

Purpose: Define the electromagnetic material type and properties for conducting shells in a 3D problem or in a 2D resistive heating problem. Electromagnetic materials act like a *MAT_ADD, so they must be associated with a *MAT material or an ICFD part. See Remark 1.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	MTYPE	SIGMA	EOSID	NELE	MUREL	EOSMU	DEATHT
Type	I	I	F	I	I	F	I	F
Default	none	none	none	none	1	1.	optional	10^{28}

Optional RF heating Card. Additional material parameters for radio-frequency heating (RF) cases.

Card 2	1	2	3	4	5	6	7	8
Variable			EPSRR	EOSID2	EPSRI	EOSID3		
Type			F	I	F	I		
Default			0.	0	0.	0		

VARIABLE	DESCRIPTION
MID	Material ID. For all MTYPE except MTYPE = 3, MID must reference a *MAT material since the electromagnetic properties are added onto the *MAT properties. If MTYPE = 3, MID can be left blank for the electromagnetic properties to apply to the entire ICFD fluid, or it can be the PID of a *ICFD_PART_VOL to apply to an ICFD fluid part. See Remark 1.
MTYPE	Defines the electromagnetism type of the material: EQ.0: Air or vacuum EQ.1: Insulator material. These materials have the same electromagnetism behavior as MTYPE = 0. EQ.2: Conductor carrying a source. In these conductors, the

VARIABLE	DESCRIPTION
	eddy current problem is solved, which gives the actual current density. Typically, this would correspond to the coil. EQ.3: Fluid conductor. In this case, MID refers to the ID given in *ICFD_PART_VOL. Note that this is only available for ICFD in 2D. See Remark 2. EQ.4: Conductor not connected to any current or voltage source, where the Eddy current problem is solved. Typically, this would correspond to the workpiece.
SIGMA	Initial electrical conductivity of the material
EOSID	ID of the EOS to be used for the electrical conductivity (see *EM_EOS cards).
NELE	Number of elements in the thickness of the shell. Note that you must make sure your mesh is fine enough to capture the inductive-diffusive effects correctly (see the skin depth definition).
MUREL	Relative permeability, which is the ratio of the permeability of a specific medium to the permeability of free space ($\mu_r = \mu/\mu_0$).
EOSMU	ID of the EOS to be used to define the nonlinear behavior of μ. Note that if EOSMU is defined, MUREL will be used for the initial value only. See *EM_EOS_PERMEABILITY.
DEATHT	Death time for the material. After DEATHT, the material will no longer be considered a conductor and will be removed from the EM solve. If a negative value is entered, a *DEFINE_FUNCTION will be expected. The following parameters are allowed: (vx, vy, vz, temp, vol, mass, Ex, Ey, Ez, Bx, By, Bz, Fx, Fy, Fz, JHrate, time). Fx, Fy, and Fz refer to the components of the Lorentz force vector. A negative value returned by the *DEFINE_FUNCTION corresponds to a 'dead' or inactive element. Once an element has been removed from the EM solve, it cannot return.
EPSRR/EPSRI	Real and imaginary parts of the relative permittivity (dielectric constant and dielectric loss terms). In cases that use the quasi-static electrostatic solver, EPSRR directly represents the relative permeability.

VARIABLE	DESCRIPTION
EOSID2/EOSID3	Optional IDs defining equations of state for EPSRR and EPSRI, respectively.

Remarks:

1. **Input decks with EM materials.** EM materials must be associated with a *MAT material for MTYPE ≠ 3 or an ICFD fluid for MTYPE = 3. These EM properties are added onto the normal material properties like *MAT_ADD material properties. Thus, when added onto a *MAT material, MID here must refer to the MID of the *MAT material. For example, an input deck can have the following to define a conductor:

```
*PART
2,1,20
*MAT_RIGID
20,&dens,&young,$nu
*EM_MAT_004
20,2,&cond,&eosid
```

   In this case, if *MAT_RIGID is not included, the conduction properties are not applied.

   For MTYPE = 3, MID can refer to the PID of a *ICFD_PART_VOL or be left blank. If it is left blank, the EM properties apply to the entire ICFD fluid.

2. **Coupling ICFD to EM.** Only the resistive heating solver is currently available when coupling the ICFD solver with the EM solver (see *EM_CONTROL).

*EM_MAT_005

Purpose: Define an electromagnetic material for which two material conductivities are specified per EM node and electromagnetic conductivities are defined by a (3 × 3) tensor matrix. Applications of this material include the Randles Batmac model (see Remark 2) and the electrophysiology bidomain model. This material is only supported for solid elements. Electromagnetic materials act like a *MAT_ADD, so they must be associated with a *MAT material. See Remark 1.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	MTYPE	SIGMAXXA	SIGMAYYA	SIGMAZZA			
Type	I	I	F	F	F			

Card 2	1	2	3	4	5	6	7	8
Variable	SIGMAXYA	SIGMAXZA	SIGMAYXA	SIGMAYZA	SIGMAZXA	SIGMAZYA		
Type	F	F	F	F	F	F		

Card 3	1	2	3	4	5	6	7	8
Variable			SIGMAXXB	SIGMAYYB	SIGMAZZB			
Type			F	F	F			

Card 4	1	2	3	4	5	6	7	8
Variable	SIGMAXYB	SIGMAXZB	SIGMAYXB	SIGMAYZB	SIGMAZXB	SIGMAZYB		LAMBDA
Type	F	F	F	F	F	F		F

Card 5	1	2	3	4	5	6	7	8
Variable	AOPT	XP	YP	ZP	A1	A2	A3	MACF
Type	F	F	F	F	F	F	F	I

Card 6	1	2	3	4	5	6	7	8
Variable	V1	V2	V3	D1	D2	D3		
Type	F	F	F	F	F	F		

VARIABLE	DESCRIPTION
MID	Material ID. MID must reference a *MAT material since the electromagnetic properties are added onto the *MAT properties. See Remark 1.
MTYPE	Defines the electromagnetism type of the material: EQ.0: Air or vacuum EQ.1: Insulator material. These materials have the same electromagnetism behavior as MTYPE = 0. EQ.2: In electrophysiology, it corresponds to the tissue, where the bidomain equations will be solved for EMSOL = 12 or EMSOL = 13. An *EM_EP_CELLMODEL must be associated with this material. EQ.4: In electrophysiology, it corresponds to the bath where only the external potential is solved for. No *EM_EP_CELLMODEL should be associated with this material. EQ.5: Material associated with *EM_RANDLES_BATMAC. See Remark 2.
SIGMAXXA/B	The 1, 1 term in the 3 × 3 electromagnetic conductivity tensor matrix for the two conductivities. For the BatMac model, A is for the potential on the positive current collector, and B is for the potential on the negative current collector. For the bidomain model in Electrophysiology, A is for the intracellular potential, and B is for the extracellular potential. Note that 1 corresponds to the **a** material direction.

VARIABLE	DESCRIPTION
	LT.0.0: |SIGMAXXA/B| corresponds to the ID of a *DEFINE_FUNCTION. See Remark 4 for available parameters.
SIGMAXYA/B	The 1, 2 term in the 3 × 3 electromagnetic conductivity tensor matrix for the two conductivities. Note that 2 corresponds to the **b** material direction. LT.0.0: |SIGMAXYA/B| corresponds to the ID of a *DEFINE_FUNCTION. See Remark 4 for available parameters.
⋮	⋮
SIGMAZZA/B	The 3, 3 term in the 3 × 3 electromagnetic conductivity tensor matrix for the two conductivities. Note that 3 corresponds to the **c** material direction. LT.0.0: |SIGMAZZA/B| corresponds to the ID of a *DEFINE_FUNCTION. See Remark 4 for available parameters.
LAMBDA	Intra-to-extracellular conductivity ratio. When non-empty, the elliptic equation is solved to compute extracellular potentials (to be used only when EMSOL = 15 or 16 in *EM_CONTROL).
AOPT	Material axes option (see *MAT_002 for a more detailed description): EQ.0.0: Locally orthotropic with material axes determined by element nodes. The **a**-direction is from node 1 to node 2 of the element. The **b**-direction is orthogonal to the **a**-direction and is in the plane formed by nodes 1, 2, and 4. EQ.1.0: Locally orthotropic with material axes determined by a point in space, *P*, and the global location of the element center; this is the **a**-direction. EQ.2.0: Globally orthotropic with material axes determined by vectors defined below, as with *DEFINE_COORDINATE_VECTOR. EQ.3.0: Locally orthotropic material axes determined by a vector **v** and the normal vector to the plane of the element.The plane of a solid element is the midsurface between the inner surface and outer surface defined

VARIABLE	DESCRIPTION
	by the first four nodes and the last four nodes of the connectivity of the element, respectively. Thus, for solid elements, AOPT = 3 is only available for hexahedrons. **a** is determined by taking the cross product of **v** with the normal vector, **b** is determined by taking the cross product of the normal vector with **a**, and **c** is the normal vector. Then **a** and **b** are rotated about **c** by an angle BETA. BETA may be set in the keyword input for the element.
	EQ.4.0: Locally orthotropic in cylindrical coordinate system with the material axes determined by a vector, **v**, and an originating point, *P*, which define the centerline axis.
XP, YP, ZP	Define coordinates of point **p** for AOPT = 1 and 4.
A1, A2, A3	Define components of vector **a** for AOPT = 2.
MACF	Material axes change flag for solid elements: EQ.1: No change, default
V1, V2, V3	Define components of vector **v** for AOPT = 3 and 4.
D1, D2, D3	Define components of vector **d** for AOPT = 2.

Remarks:

1. **Input decks with EM materials.** EM materials must be associated with a *MAT material. These EM properties are added onto the normal material properties like *MAT_ADD material properties. Thus, MID here must refer to the MID of the *MAT material. For example, an input deck can have the following to define a conductor:

```
*PART
2,1,20
*MAT_RIGID
20,&dens,&young,$nu
*EM_MAT_005
20,2,&cond11a,&cond22a,&cond33a
&cond12a,&cond13a,&cond21a,&cond23a,&cond31a,&cond32a
,,&cond11b,&cond22b,&cond33b
&cond12b,&cond13b,&cond21b,&cond23b,&cond31b,&cond32b
```

In this case, if *MAT_RIGID is not included, the conduction properties are not applied.

2. **Using this material with BatMac.** When this material is used in conjunction with the battery BatMac model, then the conductivities must be adjusted by the current collector thickness ratio over the total thickness of the cell. For example, if n_p is the number of positive current collectors, t_p the thickness of each individual positive current collector and Th the total thickness of the cell, then the conductivity for the positive current collector must be scaled by:

$$\frac{n_p \times t_p}{\mathrm{Th}} .$$

3. **Material directions.** See the manual page for *MAT_002 for a description of how the principal material directions, {**a**, **b**, **c**}, are determined with AOPT. Note that *EM_MAT_005 only works for solid elements. The AOPT options illustrated in the AOPT figure of *MAT_002 can define the material directions for all elements of the parts that use the material.

4. **Function parameters.** The available parameters for the *DEFINE_FUNCTIONs are: f(time, emdt, x_ele, y_ele, z_ele, ieleuser). Here, time is the current EM time; emdt is the current EM time step; x_ele, y_ele, and z_ele are the location of the element; and ieleuser is the element ID.

*EM_MAT_006

Purpose: Define two conductivities per EM node for special applications (Randles Batmac). Electromagnetic materials act like a *MAT_ADD, so they must be associated with a *MAT material. See Remark 1.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	MTYPE	SIGP	EOSP	SIGN	EOSN	DEATHT	
Type	I	I	F	I	F	I	F	
Default	none	none	none	none	none	none	10^{28}	

VARIABLE	DESCRIPTION
MID	Material identification. MID must reference a *MAT material since the electromagnetic properties are added onto the *MAT properties. See Remark 1.
MTYPE	Defines the electromagnetism type of the material: EQ.0: Air or vacuum EQ.1: Insulator material. These materials have the same electromagnetism behavior as MTYPE = 0. EQ.5: Material associated to *EM_RANDLES_BATMAC
SIGP/SIGN	Conductivities of the positive / negative current collector materials
EOSP/EOSN	Optional ID of the EOS to be used for the two conductivities
DEATHT	Death time for the material. After DEATHT, the material will no longer be considered a conductor and removed from the EM solve. If a negative value is entered, a |DEATHT| is a *DEFINE_FUNCTION ID. The following parameters are allowed: (vx, vy, vz, temp, vol, mass, Ex, Ey, Ez, Bx, By, Bz, Fx, Fy, Fz, JHrate, time). The vector (Fx, Fy, Fz) refers to the Lorentz force vector. A negative value returned by the *DEFINE_FUNCTION corresponds to a "dead" or inactive element. Once an element has been removed from the EM solve, it cannot return.

Remarks:

1. **Input decks with EM materials.** EM materials must be associated with a *MAT material. These EM properties are added onto the normal material properties like *MAT_ADD material properties. Thus, MID here must refer to the MID of the *MAT material. For example, an input deck can have the following to define an insulator:

```
*PART
2,1,20
*MAT_RIGID
20,&dens,&young,$nu
*EM_MAT_006
20,1,&condp,&eosp,&condn,&eosn
```

   In this case, if *MAT_RIGID is not included, the insulator properties are not applied.

2. **BatMac Model.** When this material is used in conjunction with the battery BatMac model, the conductivities must be adjusted by the current collector thickness ratio over the total thickness of the cell. For example, if n_p is the number of positive current collectors, t_p is the thickness of each individual positive current collector, and Th is the total thickness of the cell, the conductivity for the positive current collector must be scaled by: $n_p \times t_p/\mathrm{Th}$.

*EM_MAT_007

Purpose: Define the electromagnetic material type and properties for a type II superconductor material whose permeability equals the free space permeability. Electromagnetic materials act like a *MAT_ADD, so they must be associated with a *MAT material. See Remark 1. This electromagnetic material is supported for the Eddy current and magnetics solver (EMSOL = 1 on *EM_CONTROL). The monolithic solver must be used for a stable calculation. See *EM_SOLVER_FEMBEM_MONOLITHIC.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	MTYPE	IMOD	JC0	EC	N		SIGFL
Type	I	I	I	F	F	F		I
Default	none	none	none	0.	0.	0.		10^{14}

Optional Quenching Properties Card. Additional material parameters for superconductor degradation properties.

Card 2	1	2	3	4	5	6	7	8
Variable	JCMIN	BSCALE		JCEOSID	SIGN	SGNEOSID		
Type	F	F		I	F	I		
Default	0.	0.02		optional	0.01	optional		

VARIABLE	DESCRIPTION
MID	Material ID. For all MTYPE, MID must reference a *MAT material since the electromagnetic properties are added onto the *MAT properties. See Remark 1.
MTYPE	Electromagnetism type of the material: EQ.0: Air or vacuum EQ.1: Insulator material. These materials have the same electromagnetism behavior as MTYPE = 0. EQ.2: Conductor carrying a source. In these conductors, the electromagnetic problem is solved, yielding the actual

VARIABLE	DESCRIPTION
	current density. Typically, this material would correspond to the coil. EQ.4: Conductor not connected to any current or voltage source, where the electromagnetic problem is solved. Typically, this material would correspond to the workpiece.
IMOD	Select the type II superconductor model (see Remark 2): EQ.1: Power law EQ.2: Hyperbolic tangent
JC0	Reference critical current density, J_{c0}. See Remarks 2 and 3.
EC	Critical electric field, E_c. See Remark 2.
N	Power law exponent, n. See Remark 2.
SIGFL	Floor conductivity, σ_{fl}. See Remark 2.
JCMIN	Tripping current density value, $J_{c,min}$. See Remark 3.
BSCALE	Blend scale value, b_{scale}. See Remark 3.
JCEOSID	Optional ID of the EOS to be used for J_c (see *EM_EOS cards). See Remarks 2 and 3.
SIGN	Normal state conductivity, σ_n. This field is ignored if SGNEOSID is defined. See Remark 3.
SGNEOSID	Optional ID of the EOS to be used for SIGN (see *EM_EOS cards). See Remark 3.

Remarks:

1. **Input decks with EM materials.** EM materials must be associated with a *MAT material. These EM properties are added onto the normal material properties, like *MAT_ADD material properties. Thus, MID here must refer to the MID of the *MAT material. For example, an input deck can have the following to define a conductor:

```
*PART
2,1,20
*MAT_RIGID
20,&dens,&young,$nu
*EM_MAT_007
```

```
20,2,&imod
```

In this case, if *MAT_RIGID is not included, the conduction properties are not applied.

2. **Superconducting models.** The different superconductor models aim to provide the nonlinear relationship between the electric resistivity (i.e., inverse of conductivity $\rho = 1/\sigma$), the electric field, and the current density. IMOD = 1 is the classic power-law model defined as:

$$\rho_{\mathrm{pl}}(E) = \frac{E_{\mathrm{c}}}{J_{\mathrm{c}}}\left(\frac{|E|}{E_{\mathrm{c}}}\right)^{\frac{n-1}{n}}$$

$$\rho = \rho_{\mathrm{pl}}(E) + \frac{1}{\sigma_{\mathrm{fl}}}$$

$$J_{\mathrm{SC}} = \frac{E}{\rho}$$

IMOD = 2 is the hyperbolic tangent model defined as:

$$\beta = J_{\mathrm{c}} \tanh\left(\frac{|E|}{E_{\mathrm{c}}}\right)$$

$$J_{\mathrm{SC}} = \frac{\beta}{|E| + \frac{\beta}{\sigma_{\mathrm{fl}}}} E$$

The critical current density, J_{c}, can evolve function of temperature or other properties through the JCEOSID field. If JCEOSID is not defined, then $J_{\mathrm{c}} = J_{\mathrm{c0}}$. An example of a field dependent critical current law is:

$$J_{\mathrm{c}}(B) = \frac{J_{\mathrm{c0}}}{\left(1 + \frac{|B|}{B_0}\right)^{\alpha}}$$

3. **Property degradation.** Superconductors are highly sensitive to external environmental factors, mainly temperature. The second card of this keyword supports taking such behavior into account. A nonzero JCEOSID refers to an EOS for J_{c} (temperature dependent or function of mechanical properties). JCMIN is a minimum current density value that controls the "switch" between superconductor behavior and regular conductor behavior. If we introduce a smoothing weight, w:

$$w = \begin{cases} 0 & t \le 0. \\ t^2(3-2t) & 0. < t < 1. \\ 1 & t \ge 1. \end{cases}$$

$$t = \frac{J_{\mathrm{SC}} - J_{\mathrm{c,min}}}{b_{\mathrm{scale}}\, J_{\mathrm{c0}}}$$

then the current density in the superconductor is expressed as:

$$J_{\mathrm{tot}} = wJ_{\mathrm{SC}} + (1-w)\sigma_{\mathrm{n}}E$$

This formulation provides a smooth transition between full superconductor behavior and regular conductor.

The above equation for w assumes that b_{scale} is positive. If a negative value is provided for b_{scale}, w is defined as:

$$w = \begin{cases} 1 & \text{if } J_{\text{SC}} > J_{\text{c,min}} \\ 0 & \text{otherwise} \end{cases},$$

causing a sharp transition instead of a smooth one.

*EM_OUTPUT

Purpose: Define the level of EM related output on the screen and in the messag file.

Card 1	1	2	3	4	5	6	7	8
Variable	MATS	MATF	SOLS	SOLF	MESH	MEM	TIMING	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	

VARIABLE	DESCRIPTION
MATS	Level of matrix assembly output to the screen: EQ.0: No output EQ.1: Basic assembly steps EQ.2: Basic assembly steps + percentage completed + final statistics EQ.3: Basic assembly steps + percentage completed + statistics at each percentage of completion
MATF	Level of matrix assembly output to the messag file: EQ.0: No output EQ.1: Basic assembly steps EQ.2: Basic assembly steps + percentage completed + final statistics EQ.3: Basic assembly steps + percentage completed + statistics at each percentage of completion
SOLS	Level of solver output on the screen: EQ.0: No output EQ.1: Global information at each FEM iteration EQ.2: Detailed information at each FEM iteration
SOLF	Level of solver output to the messag file: EQ.0: No output

VARIABLE	DESCRIPTION
	EQ.1: Global information at each FEM iteration
	EQ.2: Detailed information at each FEM iteration
MESH	Controls the output of the mesh data to the d3hsp file:
	EQ.0: No mesh output written.
	EQ.1: Mesh info written.
MEMORY	Controls the output of information about the memory used by the EM solve to the messag file:
	EQ.0: No memory information written.
	EQ.1: Memory information written.
TIMING	Controls the output of information about the time spent in the different parts of the EM solver to the messag file
	EQ.0: No timing information written.
	EQ.1: Timing information written.

*EM_OUTPUT_BINARY

Purpose: Specify binary output options for the EM solver.

Card 1	1	2	3	4	5	6	7	8
Variable	IBIN							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
IBIN	Type of output: EQ.0: All EM results are output in ASCII format. See em_[…].dat files. EQ.1: EM results are output in the binout file under the `em-print` directory. See Remark 1. EQ.2: Both ASCII and binary outputs are turned on.

Remarks:

1. **Binary output.** Using a binary file for the output of data saves computational time in models (for example, large battery models that feature a large number of Randles circuits and isopotential conditions) by avoiding excessive printing of ASCII files. The path to access the results in the binary file is `/em-print/d########`. The `########` indicates the 8-digit print cycles index. The following table overviews the supported output with `########` indicating the ID of the entity:

ASCII Family	BINOUT Equivalent
em_randlesCell_*	rdlCell########
em_randlesCellTot_*	rdlTot
em_isoPotOut*	IsoPot########
em_isoPotConnOut*	IsoPotC########
em_boundaryOut*	BoundPre########
em_magnet_*	Magnet########

ASCII Family	BINOUT Equivalent
em_partData*	Part########
em_circuit*	Circuit########
em_rogoCoil_*	Rogo########
em_circuitRes_*	CircRes########
em_pointout*	Point########
em_nodout*	Node########
em_elout*	Elem########
em_globEnergy*	GlobEnergy

The binout tree structure for the EM output is as follows:

```
binout
 emprint
     d00000001
         time
         rdlCell00000012/   {emtime,V,U,...}
         rdlTot/            {emtime,V,...,EJHR,...}
         IsoPot00000003/    {emtime,Volt,Curr[,F*]}
         IsoPotC00000001/   {emtime,Volt,...,Resistance}
         BoundPre00000005/  {emtime,Volt,Curr,Res,Area[,F]}
         Magnet00000002/    {emtime,Force*,Torq*,Pos*,Vel*,Acc*}
         Part00000007/      {emtime,LorF*,JouH|Power,...}
         Circuit00000001/   {emtime,Volt,...,Flux|FluxRe/Im}
         Rogo00000004/      {emtime,<Val>,<Ddt>[,Force*]}
         CircRes00000008/   {emtime,Curr,Res,JouH,Area}
         Point00000010/     {emtime,Xcoor,...,ScalPot[,*Phi]}
         Node00001234/
         Elem00005678/
         GlobEnergy/        {emtime,Ext*,Mesh*,AirMag,...,[Fem*]}
```

*EM_OUTPUT_FORCES

Purpose: Cause the output of a binary file (emforcout) containing the node IDs, the coordinates, and the EM forces at each node. Additionally, ASCII files in keyword format containing these forces can be output.

Card 1	1	2	3	4	5	6	7	8
Variable	IOUT	OUTDT	LCOFF	SF				
Type	I	F	I	F				
Default	0	0.0	0	1.				

VARIABLE	DESCRIPTION
IOUT	Type of the output: EQ.1: Outputs the binary file only. EQ.2: Outputs the binary file and generates a set of keyword files em_loadnid and em_loadlcid giving the EM forces at each node at the given output time. EQ.3: Same as 2 except the forces are output as function of time.
OUTDT	Time period at which the keyword files are generated. Setting OUTDT to 0.0 causes the output to be generated at each EM time step. The absolute value of a negative value refers to a time-dependent load curve. See Remark 2.
LCOFF	Optional offset in the load curve IDs that are associated with the forces at each node when generating the keyword files.
SF	Optional scaling factor that can be applied to the output forces when generating the keyword files.

Remarks:

1. **2-step EM coupling.** This feature can be used to perform a coupled analysis in two steps. The first step solves the magnetostatic EM problem and outputs the forces. The second step loads those forces in a transient mechanical analysis. This capability enables quickly gaining a good understanding of the model before turning to a fully coupled transient EM solid mechanics analysis.

2. **Memory cost.** The computational cost to generate the ASCII files can become high in large cases. When OUT=3 is used, there is also a memory cost associated to the output which can again become problematic in large cases. For this reason, it can recommended to use an OUTDT that is larger than the EM timestep to reduce the amount of times those ASCII files are output.

*EM_OUTPUT_VTK

Purpose: Cause the output of vtk files. It can be used in applications that require two material conductivities per EM node and whose electromagnetic conductivities are defined by a (3 × 3) tensor matrix. These applications include the Randles Batmac model and the Electrophysiology Bidomain model.

Card 1	1	2	3	4	5	6	7	8
Variable	VTKTYPE	VTKT						
Type	I	F						
Default	none	0.0						

VARIABLE	DESCRIPTION
VTKTYPE	Type of the vtk files output. EQ.1: A single .vtk file EQ.2: Parallel unstructured points data (.pvtu files), recommended in mpp executions.
VTKT	Time period at which vtk files are exported.

*EM_P2D_BATMAC

Purpose: Use a Battery Macro (BatMac) model for coupling the battery electrochemistry P2D circuits (see *BATTERY_ECHEM_CONTROL_SOLVER) with the finite element mesh. This BatMac model is a macro battery model where solid elements are retained for the solid mechanics and thermal solve, and each conducting node has its own P2D circuit associated with it. The P2D circuits receive the source current through *EM_ISOPOTENTIAL. See Figures 6-5 and 6-6.

Card Summary:

Card 1. This card is required.

CELLID	PSID		ATYPE	THK	SHRTID	SHRTMOD	

Card 1.1. Include this card if ATYPE = 2.

ATPS							

Card 1.2. Include this card if ATYPE = 2.

L1	L2	N					

Card 2. This card is required.

TEMP0	FRTERM						

Card 3. This card is optional.

DFIDLA	DFIDLS	DFIDLC	DFIDRA	DFIDRC			

Card 4. This card is optional.

VLCUT	VHCUT	TAPER	SOCMIN	SOCMAX			

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	CELLID	PSID		ATYPE	THK	SHRTID	SHRTMOD	
Type	I	I		I	F	I	I	
Default	optional	none		0	0.0	none	0	

VARIABLE	DESCRIPTION
CELLID	Optional P2D cell ID. If cell ID is unused on the *BATTERY_... keywords, then this field should be left blank. Otherwise, a cell ID must be provided to associate the nodes in the solid part set with a the correct P2D parameters.
PSID	Part set ID for the solid parts that have P2D circuits at each node
ATYPE	Circuit area, A_{icirc}, calculation type: EQ.-1: Same as 0 but THK is used for t_{cell} instead of the internally determined L_{cell} EQ.0: $A_{\text{icirc}} = \frac{t_{\text{node}}}{t_{\text{cell}}} A_{\text{node}}$, where A_{node} is the raw per-node face area internally determined from the 3D geometry, t_{node} is the local thickness associated to each circuit based on element size from the 3D geometry, and $t_{\text{cell}} = L_{\text{cell}}$. L_{cell} is the total length of a single unit cell in the circuit (see Remark 1 of *BATTERY_ECHEM_CELL_GEOMETRY). This is the default. EQ.1: $A_{\text{icirc}} = \frac{A_{\text{jroll}}}{A_{\text{cell}}} A_{\text{node}}$ where A_{cell} is the total geometric cell area (determined internally by the solver) and A_{jroll} is total area of the deployed jelly roll (active electrode area) set with THK. EQ.2: Same as 1, except A_{jroll} is determined with the method specified by ATPS on Card 1.1
THK	t_{cell} in the case of ATYPE = -1 and A_{jroll} in the case of ATYPE = 1. This field is ignored for all other values of ATYPE.

VARIABLE	DESCRIPTION
SHRTID	Function (*DEFINE_FUNCTION) ID for internal short handling. See Remark 2.
SHRTMOD	Type of internal short handling (see Remark 2): EQ.0: Circuits transition from the normal P2D electrochemical solve to a simple resistive short when the criteria in function SHRTID are met. EQ.1: The electrochemical solve continues while the circuit is shorted when meeting the criteria in SHRTID. However, the effective resistance is adjusted.

Jelly Roll Area Method Card. Include this card if ATYPE = 2.

Card 1.1	1	2	3	4	5	6	7	8
Variable	ATPS							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
ATPS	A_{jroll} computation method: EQ.0: A_{jroll} is specified with L1 on Card 1.2. EQ.1: $A_{\text{jroll}} = 2LWN$, where L is the length of a layer in the jelly roll set with L1 on Card 1.2, W is the width set with L2 on Card 1.2, and N is the number of layers set with N in Card 1.2. EQ.2: $A_{\text{jroll}} = \pi / t_{\text{cell}} \left(R_{\text{o}}^2 - R_{\text{i}}^2\right) H$,where $t_{\text{cell}} = L_{\text{cell}}$, R_{o} is the outer radius of the jelly roll set with L1 on Card 1.2, R_{i} is the inner radius set with L2 on Card 1.2, and H is the height set with N on Card 1.2. L_{cell} is the total length of a single unit cell in the circuit (see Remark 1 of *BATTERY_ECHEM_CELL_GEOMETRY).

Jelly Roll Area Card. Include this card if ATYPE = 2.

Card 1.2	1	2	3	4	5	6	7	8
Variable	L1	L2	N					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
L1	A_{jroll} for ATPS = 0, L for ATPS = 1, and R_o for ATPS = 2
L2	W for ATPS = 1 and R_i for ATPS = 2
N	Number of layer for ATPS = 1 and H for ATPS = 2

Card 2	1	2	3	4	5	6	7	8
Variable	TEMP0	FRTERM						
Type	F	I						
Default	0.0	0						

VARIABLE	DESCRIPTION
TEMP0	Initial temperature. Ignored if FRTERM ≠ 0.
FRTERM	Flag to determine how the initial temperature is specified: EQ.0: The initial temperature is taken from TEMP. EQ.1: The initial temperature is read from the thermal solver.

Optional Scaling Card. This card is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	DFIDLA	DFIDLS	DFIDLC	DFIDRA	DFIDRC			
Type	I	I	I	I	I			
Default	0	0	0	0	0			

VARIABLE	DESCRIPTION
DFIDLA	Function (*DEFINE_FUNCTION) ID giving a scale factor on the anode thickness. See Remark 4.
DFIDLS	Function (*DEFINE_FUNCTION) ID giving a scale factor on the separator thickness. See Remark 4.
DFIDLC	Function (*DEFINE_FUNCTION) ID giving a scale factor on the cathode thickness. See Remark 4.
DFIDRA	Function (*DEFINE_FUNCTION) ID giving a scale factor on the anode particle radius. See Remark 4.
DFIDRC	Function (*DEFINE_FUNCTION) ID giving a scale factor on the cathode particle radius. See Remark 4.

Optional Cutoff Values Card. This card is optional. See Remark 3.

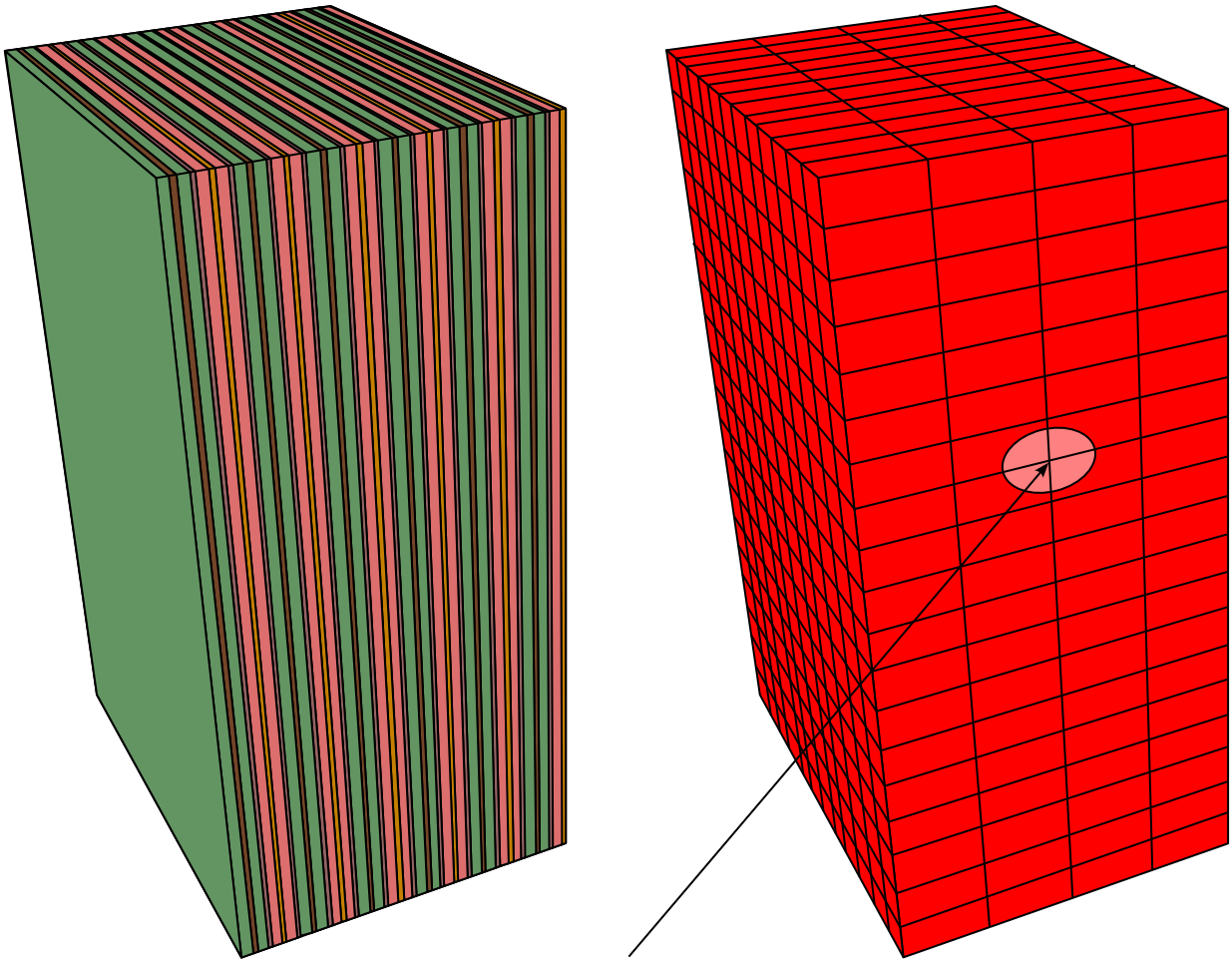
Card 4	1	2	3	4	5	6	7	8
Variable	VLCUT	VHCUT	TAPER	SOCMIN	SOCMAX			
Type	F	F	F	F	F			
Default	0.0	0.0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
VLCUT	Low voltage cutoff
VHCUT	High voltage cutoff. If it is left as the default value, this cutoff is not used.

VARIABLE	DESCRIPTION
TAPER	Taper current. If it is left as the default value, this cutoff is not used.
SOCMIN	Minimum state of charge value
SOCMAX	Maximum state of charge value. If it is left as the default value, this cutoff is not used.

Schematic of several layers of unit cells. The layers are scaled in the unit cells width direction for clarity

BatMac model:



At each node, a single P2D unit cell is solved, but it is scaled to model multiple unit cells in parallel. Thus, each node models multiple layers of unit cells but costs much less than modeling individually.

Figure 6-5. The left figure is a schematic of layers of unit cells. Modeling every layer is costly, so the BatMac model has a single circuit at each node in which a single unit cell is solved, but it is scaled to account for multiple unit cells in parallel.

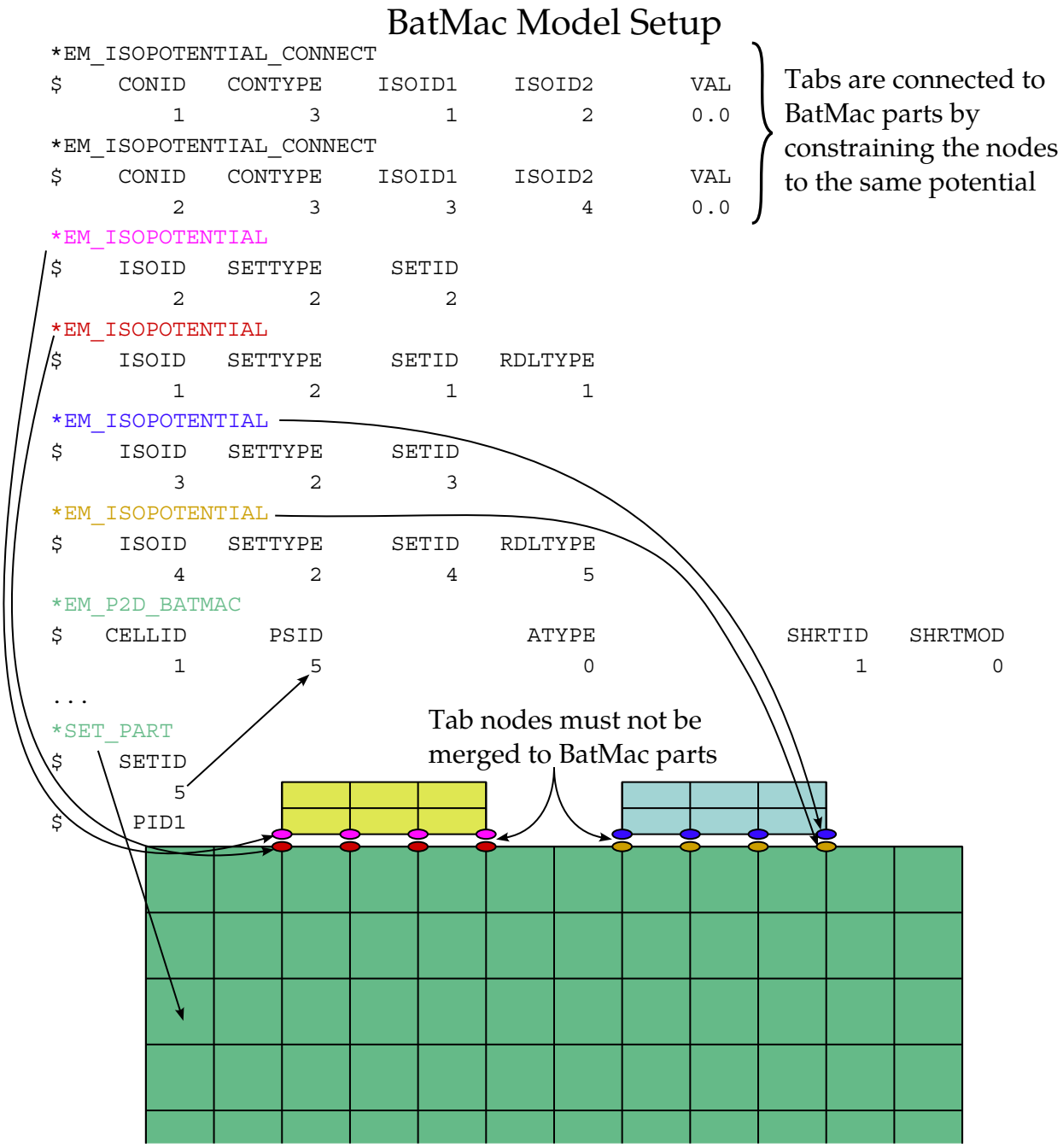


Figure 6-6. BatMac model setup

Remarks:

1. **BatMac model.** For the BatMac model, each finite element node in PSID has a circuit with two degrees of freedom representing the scalar potentials of the positive and negative current collectors. These potentials are connected with a Robin boundary condition. The idea of the BatMac model is to operate at a higher level than unit cells, meaning that a single node represents several layers

of a battery (see Figure 6-5). Thus, a single unit cell is solved at each node, but it is scaled to represent several unit cells in the circuit in parallel in that region of the battery (see ATYPE).

Figure 6-6 shows the BatMac model setup. In this setup, the EM solver is solved on the two tabs that are coupled to adjacent nodes on the BatMac part set through *EM_ISOPOTENTIAL_CONNECT. The nodes on the tabs must *not* be merged with the nodes on the BatMac parts. Through the isopotential (see *EM_ISOPOTENTIAL), the current from the EM solver can be applied to all the nodes in the BatMac parts as a boundary condition for the P2D solver.

Note that when this model is used the solver loops over all local circuits and solves the P2D system on each one. For large models with thousands of circuits this can be computationally expensive. Optional Card 3 of *EM_CONTROL_COUPLING_P2D exposes complementary mechanisms for reducing the number of P2D solves per EM step. See Remark 3 of *EM_CONTROL_COUPLING_P2D for a discussion of these mechanisms.

2. **Internal shorts.** For both internal short modes, the short criteria are controlled by a *DEFINE_FUNCTION with ID SHRTID. This function returns the short resistance, R_{short}, whose sign dictates how the short is treated. For the *DEFINE_FUNCTION, the return value can be interpreted as follows:

Return value	Meaning
$R_{\text{short}} = 0.$	No short
$R_{\text{short}} > 0.$	Reversible short with resistance R_{short} (units: Ωm^2). This short can be deactivated when the function returns to zero.
$R_{\text{short}} < 0.$	Permanent short with resistance $\lvert R_{\text{short}} \rvert$

The arguments for the function are: f(time, soc, temperature, current, voltage, x_circ, y_circ, z_circ, vmstress, efstrain, stress11, stress22, stress33, stress12, stress13, stress23, strainlocx, strainlocy, strainlocz, strainlocxy, strainlocxz, strainlocyz, la_cur, ls_cur, lc_cur, ra_cur, rc_cur, areacircuit). Here, soc is the state of charge; x_circ, y_circ, and z_circ are the x, y, and z coordinates; vmstress is the von Mises stress; efstrain is the effective strain; stressij are the components of the stress tensor; strainlocx, strainlocy, strainlocz, strainlocxy, strainlocxz, and strainlocyz are the components of the strain tensor; la_cur, ls_cur, and lc_cur are the anode, cathode, and separator thicknesses; ra_cur and rc_cur are the anode and cathode particle radii; and areacircuit is the circuit area. Note that *MAT_ADD_P2D_EXPANSION is required to obtain the stress and strain from the mechanical solver.

For SHRTMOD = 0, when a short occurs, the P2D electrochemical solve is skipped and the last computed P2D state is preserved. A simple short model is used in this case with R_{short} as the short resistance.

For SHRTMOD = 1, when a short occurs, the P2D electrochemical solve continues. The short resistance, R_{short}, acts as parallel path along the electrochemistry. At the cell level, the total current becomes the sum of the short current and the current flowing through the P2D:

$$I_{\text{tot}} = I_{\text{chem}} + I_{\text{short}} = I_{\text{chem}} + \frac{V_{\text{cell}}}{R_{\text{short}}}$$

In practice this means that during the coupling stage, when V_{cell} is returned and R_{eff} is calculated, R_{eff} is modified to consider the parallel path through the short:

$$R_{\text{eff}} = \frac{R_{\text{eff}}|R_{\text{short}}|}{R_{\text{eff}} + |R_{\text{short}}|}$$

and the contribution to the right-hand side is also modified accordingly:

$$F = F + \frac{V_{\text{cell}}}{|R_{\text{short}}|}$$

This approach allows the electrochemical quantities to continue to evolve during the short and the state of charge to be tracked. The heat contributions from the P2D solve are also kept (along with the Ohmic heating generated by the short).

3. **Cutoffs.** If any of the cutoffs are reached for a node, that circuit is removed from the model. In other words that part of the battery becomes inactive.

4. **Electrode or separator thinning.** By default, the unit cell dimensions are constant and defined in *BATTERY_ECHEM_CELL_GEOMETRY. However, optional scaling factors can be defined on Card 3 that use *DEFINE_FUNCTIONs to locally scale those values by macro quantities coming from the mechanical or thermal solve. This coupling between the macro scale and the P2D local electrochemistry solve enables capturing thickness changes due to compression, particle damage, or separator thinning. DFIDLA, DFIDLS, and DFIDLC control the thickness of each individual component (anode thickness L_a, separator thickness L_s, and cathode thickness L_c). DFIDRA and DFIDRS control the solid particle radius behavior for the anode and cathode (see *BATTERY_ECHEM_MAT_ANODE and *BATTERY_ECHEM_MAT_CATHODE). At each EM time step, before each circuit's P2D solve, the actual parameter is computed as:

$$p_{\text{actual}} = p_{\text{reference}} f$$

where f is the value returned by the *DEFINE_FUNCTION.

The arguments for the function are: f(time, soc, temperature, current, voltage, x_circ, y_circ, z_circ, vmstress, efstrain, stress11, stress22, stress33, stress12, stress13, stress23, strainlocx, strainlocy, strainlocz, strainlocxy, strainlocxz,

strainlocyz). Here, soc is the state of charge; x_circ, y_circ, and z_circ are the x, y, and z coordinates; vmstress is the von Mises stress; efstrain is the effective strain; stressij are the components of the stress tensor; and strainlocx, strainlocy, strainlocz, strainlocxy, strainlocxz, and strainlocyz are the components of the strain tensor. Note that *MAT_ADD_P2D_EXPANSION is required to obtain the stress and strain from the mechanical solver.

*EM_P2D_MESHLESS

Purpose: Associate a P2D circuit with the EM solver without a mesh for the battery. In this model, a single unit cell is modeled and area scaled to represent the battery. It is connected to the EM solvers through *EM_ISOPOTENTIAL_CONNECT. See Figure 6-7.

Card Summary:

Card 1. This card is required.

P2DMCID	ATPS	PSID					

Card 2. This card is required.

L1	L2	N					

Card 3. This card is optional.

VLCUT	VHCUT	TAPER	SOCMIN	SOCMAX			

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	P2DMCID	ATPS	PSID					
Type	I	I	I					
Default	none	0	0					

VARIABLE	DESCRIPTION
P2DMCID	ID for the meshless P2D circuit. This ID must match RDLID on *EM_ISOPOTENTIAL_CONNECT because it connects the meshless circuit to the EM system (positive and negative tabs) as shown in Figure 6-7. If CELLID is unused on the *BATTERY keywords, then this field can be any value as long as it matches RDLID. Otherwise, it must match the cell ID on the *BATTERY keywords to associate the correct P2D parameters with the circuit.
ATPS	A_{circ} computation method:

VARIABLE	DESCRIPTION
	EQ.0: A_{circ} is specified with L1 on Card 2.
	EQ.1: $A_{circ} = 2LWN$, where L is the length set with L1 on Card 2, W is the width set with L2 on Card 2, and N is the number of layers set with N in Card 2. This method is for modeling a pouch geometry.
	EQ.2: $A_{icirc} = \pi / L_{cell}(R_o^2 - R_i^2)H$,where L_{cell} is the total length of a single unit cell in the circuit (see Remark 1 of *BATTERY_ECHEM_CELL_GEOMETRY), R_o is the outer radius of the jelly roll set with L1 on Card 2, R_i is the inner radius set with L2 on Card 2, and H is the height set with N on Card 2. This method is for modeling jelly roll geometry.
PSID	Optional part set ID for temperature coupling only

Card 2	1	2	3	4	5	6	7	8
Variable	L1	L2	N					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
L1	A_{icirc} for ATPS = 0, L for ATPS = 1, and R_o for ATPS = 2
L2	W for ATPS = 1 and R_i for ATPS = 2
N	Number of layer for ATPS = 1 and H for ATPS = 2

Optional Cutoff Values Card. This card is optional. See Remark 2.

Card 3	1	2	3	4	5	6	7	8
Variable	VLCUT	VHCUT	TAPER	SOCMIN	SOCMAX			
Type	F	F	F	F	F			
Default	0.0	0.0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
VLCUT	Low voltage cutoff
VHCUT	High voltage cutoff. If it is left as the default value, this cutoff is not used.
TAPER	Taper current. If it is left as the default value, this cutoff is not used.
SOCMIN	Minimum state of charge value
SOCMAX	Maximum state of charge value. If it is left as the default value, this cutoff is not used.

Remarks:

1. **Meshless model.** In the meshless model a single P2D circuit is connected to two battery tabs through *EM_ISOPOTENTIAL_CONNECT. See Figure 6-7. The tabs are part of the EM solve. The P2D circuit is not part of the mesh, and the EM solver is not solved in the P2D circuit. The EM solver provides the current to the P2D circuit, and the P2D circuit gives the voltages. Note that the P2D circuit is comprised of a single unit cell that is area scaled (A_{icirc}) using the setting specified with ATYPE to model many unit cells in parallel.

   The following provides an example of how to set up coupling in the input deck:

```
$ isopotential for one of the tabs
*EM_ISOPOTENTIAL
$     ISOID    SETTYPE      SETID
          1          2          1
$ isopotential for the other tab
*EM_ISOPOTENTIAL
$     ISOID    SETTYPE      SETID
          2          2          2
*EM_ISOPOTENTIAL_CONNECT
$     CONID    CONTYPE     ISOID1     ISOID2                RDLID
          3          5          1          2                    7
*EM_P2D_MESHLESS
```

```
$  P2DMCID      ATPS
         7         0
$       L1
     0.066
$ P2D cell cards
*BATTERY_...
…
```

Note that RDLID and P2DMCID must match. If CELLID is used (nonzero) in the *BATTERY keywords, it must match RDLID and P2DMCID. CONTYPE must be set to 5 on *EM_ISOPOTENTIAL_CONNECT for this model.

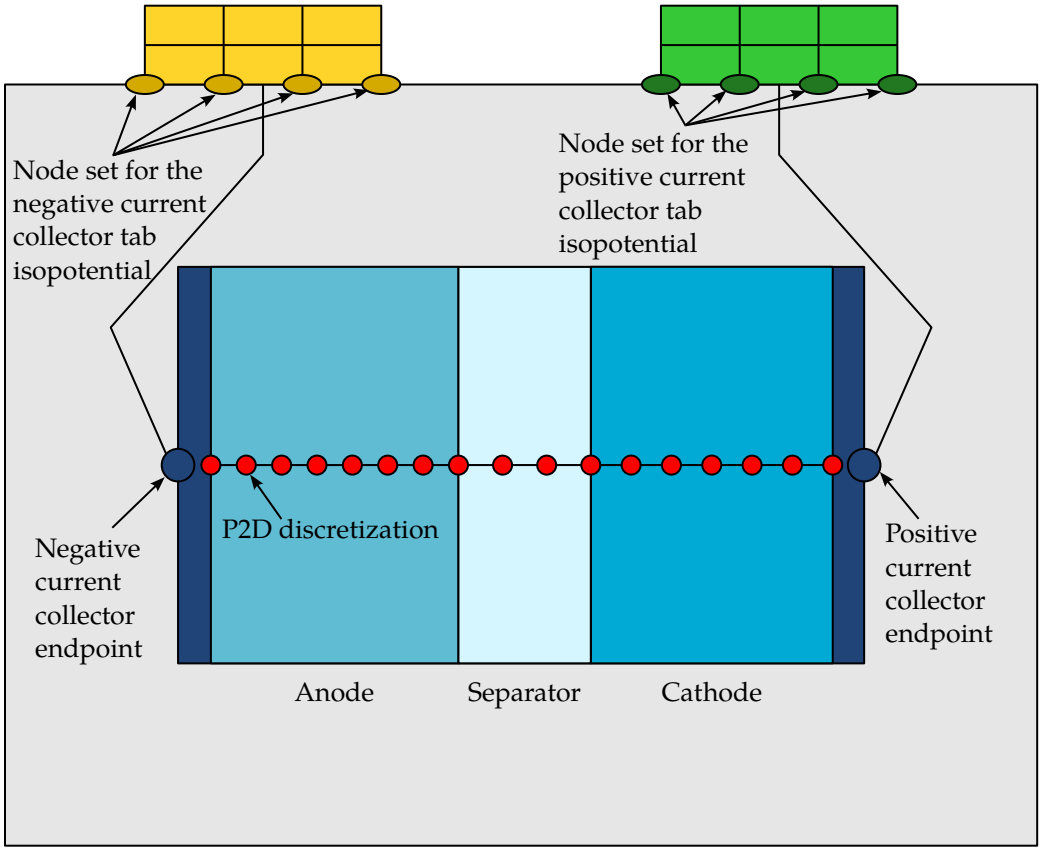


Figure 6-7. Schematic of the P2D meshless model setup. The P2D circuit is connected to the battery tabs through *EM_ISOPOTENTIAL_CONNECT. No EM is solved in the P2D cell. A single P2D unit cell is scaled by ATYPE to model the battery.

2. **Cutoffs.** If any of the cutoffs are reached, the circuit is removed from the model, meaning the battery becomes inactive.

3. **Thermal coupling.** For this model, the P2D cell is by default thermally isolated from the finite element model. An optional thermal coupling can be activated by specifying PSID. This part set is used to identify the solid elements that represent the cell volume. Before each P2D solve, the average temperature is gathered and used during the P2D solve.

If ITYPE = 1 on *BATTERY_ECHEM_CONTROL_THERMAL, the heat terms are locally consumed during the chemistry evolution, and the returned temperature is assigned to each node belonging to the part set. This allows a two-step temperature update where the thermal solver oversees the "macro" temperature update (including convection, thermal contact, external heat terms, etc.), while the chemistry solver operating on a small timeframe is in charge of the "micro" temperature update.

If ITYPE = 0, temperature remains static during the P2D update. However, the heat source terms are returned and evenly distributed for the thermal solver update. If THCPL = 2 in *EM_CONTROL_COUPLING (recommended for highly nonlinear thermal problems), the heat source terms are distributed at the element level while the default approach (THCPL = 0) distributes the terms at the nodal level. This approach allows a lumped approach for chemistry cell modelling with a complete finite element thermal model.

*EM_PERMANENT_MAGNET

Purpose: Defines a permanent magnet.

Card Summary:

Card 1. This card is required.

ID	PID	MTYPE	NORTH	SOUTH	HC		

Card 2a. This card is included if MTYPE = 3.

X	Y	Z					

Card 2b. This card is included if MTYPE = 4.

NID1	NID2						

Card 2c. This card is included if MTYPE = 5, 6, or 7.

X	Y	Z	NDIVIS	AXIS	DIR/X2	Y2	Z2

Card 3. This card is optional.

TTYPE	OX	OY	OZ	NX	NY	NZ	

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	ID	PID	MTYPE	NORTH	SOUTH	HC		
Type	I	I	I	I	I	I		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
ID	ID of the magnet
PID	Part ID

VARIABLE	DESCRIPTION
MTYPE	Magnet definition type: EQ.0: Magnet defined by two node sets for North and South Poles. EQ.1: Magnet defined by two segments sets for North and South Poles. EQ.3: Magnet defined by a global vector orientation. EQ.4: Magnet defined by a global vector orientation given by two node IDs EQ.5: Magnetic gear with NDIVIS magnets oriented around an axis given by AXIS and centered at point (X, Y, Z). The pole orientations alternate for each magnet in the gear, meaning they alternate at increments of 360/NDIVIS around the gear starting along the vector given by (X2, Y2, Z2). See Figure 6-8. EQ.6: Magnetic gear with NDIVIS magnets oriented around an axis given by AXIS and centered at point (X, Y, Z). The pole orientations alternate for each magnet in the gear, meaning they alternate at increments of 360/NDIVIS around the gear starting along the vector given by DIR. See Figure 6-8. EQ.7: Same as 5 except the North and South orientation of the magnets follows the gear rotation axis (see AXIS in Card 2c) rather than its radial direction. EQ.8: Same as 6 except the North and South orientation of the magnets follows the gear rotation axis (see AXIS in Card 2c) rather than its radial direction.
NORTH	Set ID of the magnet north face for MTYPE = 0 and 1
SOUTH	Set ID of the magnet south face for MTYPE = 0 and 1
HC	Coercive force, H_c. See Remark 1. LT.0.0: |HC| refers to a load curve ID giving the coercive force as a function of time.

MTYPE = 3 Card. This card is only included for MTYPE = 3.

Card 2a	1	2	3	4	5	6	7	8
Variable	X	Y	Z					
Type	F	F	F					
Default	0.	0.	0.					

VARIABLE	DESCRIPTION
X, Y, Z	Orientation of magnetization vector

MTYPE = 4 Card. This card is only included for MTYPE = 4.

Card 2b	1	2	3	4	5	6	7	8
Variable	NID1	NID2						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
NID1 / NID2	Two node IDs defining the magnetization vector

Magnetic Gear Card. This card is only included if MTYPE = 5, 6, or 7. See Figure 6-8.

Card 2c	1	2	3	4	5	6	7	8
Variable	X	Y	Z	NDIVIS	AXIS	DIR/X2	Y2	Z2
Type	F	F	F	I	I	I/F	F	F
Default	0.	0.	0.	none	none	none	none	none

VARIABLE	DESCRIPTION
X, Y, Z	Origin / center point of the magnetic gear
NDIVIS	Number of subdivisions, that is, number of magnets around the circle
AXIS	Normal vector to the magnets: EQ.1: Global *X* axis EQ.2: Global *Y* axis EQ.3: Global *Z* axis
DIR	Directional vector giving the location of the starting magnet / starting magnetic orientation if MTYPE = 6: EQ.1: Global *X* axis EQ.2: Global *Y* axis EQ.3: Global *Z* axis
X2, Y2, Z2	Directional vector coordinates giving the starting magnet / starting magnetic orientation if MTYPE = 5 or 7.

Optional Card. This card is only useful to change some magnet-related output quantities.

Card 3	1	2	3	4	5	6	7	8
Variable	TTYPE	OX	OY	OZ	NX	NY	NZ	
Type	I	F	F	F	I	I	I	
Default	0	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
TTYPE	Torque calculation type: EQ.0: Default. Magnet torque is output as a vector based on the cross product of the EM force and the relative position of a given node with respect to the center of gravity. EQ.1: The position vector is based on a user-defined origin point and the result of the aforementioned cross product

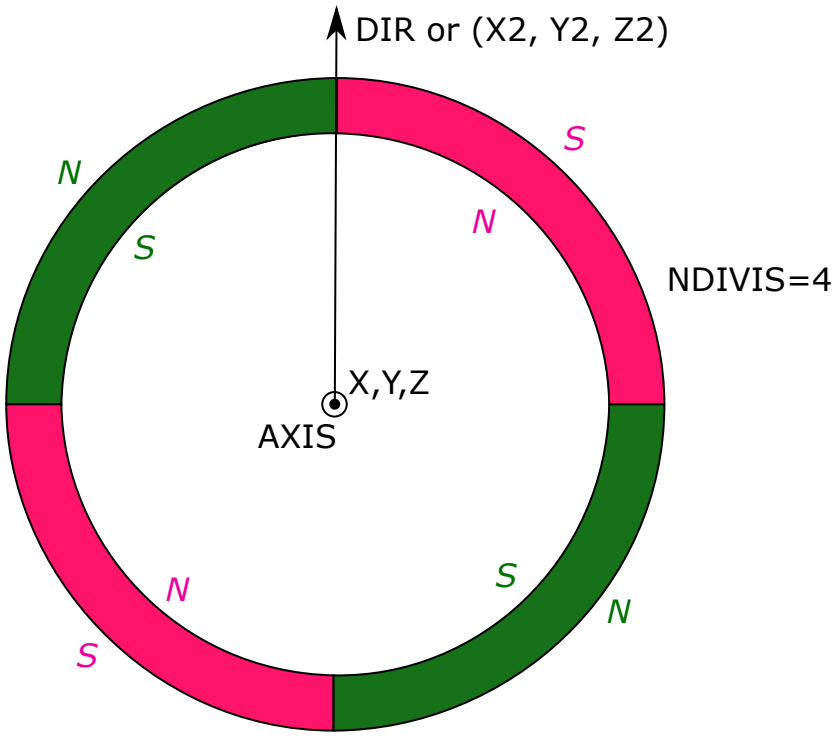


Figure 6-8. Example of Magnetic Gear (MTYPE = 5 or 6)

VARIABLE	DESCRIPTION
	operation is projected onto a user-defined axis. As a result, the torque output becomes a scalar quantity.
OX/OY/OZ	Origin point for the magnet torque calculation if TTYPE = 1.
NX/NY/NZ	Axis on which the torque is projected if TTYPE = 1.

Remark:

1. **Coercive Force.** The absolute value of coercive force H_c applied to the magnet (A/m) relates to the Residual induction B_r by the following relation:

$$H_c = B_r/\mu$$

Here, μ is the magnet's permeability defined in *EM_MAT_002 using a constant relative permeability $\mu = \mu_r\mu_0$.

The coercive force can also be expressed using the energy product $BH_{\max}$:

$$H_c = 2\sqrt{\frac{BH_{\max}}{\mu}}$$

*EM_POINT_SET

Purpose: Create a set of points that can be used by *EM_DATABASE_POINTOUT.

Card 1	1	2	3	4	5	6	7	8
Variable	PTSID	PTSTYPE	VX	VY	VZ			
Type	I	I	F	F	F			
Default	0	0	0.	0.	0.			

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	PTID	X	Y	Z	IPOS			
Type	I	F	F	F	I			
Default	none	none	none	none	0			

VARIABLE	DESCRIPTION
PSID	Point set ID
PTSTYPE	Point set type : EQ.0: Fixed points EQ.1: Tracer points using prescribed velocity
VX, VY, VZ	Constant velocities used when PTSTYPE = 1
PTID	Point ID
X, Y, Z	Point initial coordinates
IPOS	Position flag (for 2D, see Remark 1): EQ.0: The solver determines if the point is inside or outside of the conductors.

VARIABLE	DESCRIPTION
	EQ.1: The point is outside of the conductors during the entire simulation. The solver does not check; hence a gain in computation time.

Remarks:

1. **2D axisymmetric.** If using *EM_2DAXI, notice that the conductors represent the corresponding 3D conductors.

*EM_RANDLES

The keywords in this section facilitate defining Randles circuit parameters for Randles cells. The available keywords in alphabetical order are as follows:

*EM_RANDLES_BATMAC

*EM_RANDLES_EXOTHERMIC_REACTION

*EM_RANDLES_MESHLESS

*EM_RANDLES_SHORT

*EM_RANDLES_SOLID

*EM_RANDLES_TSHELL

We have implemented four models for the Randles cell: *EM_RANDLES_BATMAC, *EM_RANDLES_MESHLESS, *EM_RANDLES_SOLID, and *EM_RANDLES_TSHELL. *EM_RANDLES_BATMAC is for defining the parameters for the Randles cell when using the batmac model. *EM_RANDLES_MESHLESS, *EM_RANDLES_SOLID, and *EM_RANDLES_TSHELL are for defining the parameters when the cell is not associated with a mesh, the cell is associated with a solid mechanical model, and the cell is associated with a composite tshell mechanical model, respectively. *EM_RANDLES_EXOTHERMIC_REACTION and *EM_RANDLES_SHORT enable adding additional parameters to the Randles cell for modeling thermal runaway and shorts, respectively.

Using *DEFINE_FUNCTION for Randles Cell Parameters:

Table 6-2. Many of the parameters for these models can be specified with a *DEFINE_FUNCTION. This table indicates the available arguments, depending on the kind of model. The first column lists the arguments in single quotes with a description. The second column gives what is available for the Randles circuit parameters. The third column indicates what is available for R0CHA and R0DIS for user-defined Equivalent Circuit models, depending on the model chosen. The fourth and fifth columns give the availability for parameters in *EM_RANDLES_SHORT and *EM_RANDLES_EXOTHERMIC_REACTION.

	Parameters with *DEFINE_FUNCTIONs			
Variable names	Randles circuit parameters (r_0, r_{10}, c_{10}, etc.)	RDLTYPE = -1	Internal short	Exothermic reaction
'time': Current EM time	All models	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models

Variable names	Parameters with *DEFINE_FUNCTIONS: Randles circuit parameters (r_0, r_{10}, c_{10}, etc.)	RDLTYPE = -1	Internal short	Exothermic reaction
'emdt': Current EM time step	All models	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models
'x_sep, y_sep, z_sep': Positive electrode coordinates	Solid/tshell models	Solid/tshell models	Solid/tshell models	Solid/tshell models
'x_sen, y_sen, z_sen': Negative electrode coordinates	Solid/tshell models	Solid/tshell models	Solid/tshell models	Solid/tshell models
'x_ccp, y_ccp, z_ccp': Positive Current collector coordinates	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models
'x_ccn, y_ccn, z_ccn': Negative Current collector coordinates	Solid/tshell models	Solid/tshell models	Solid/tshell models	Solid/tshell models
'pres': Local pressure	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models
'rho': Local density	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models
'vmstress': Local von Mises stress	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models
'cond': Local electrical conductivity	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models
'temp': Local temperature	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models
'tempRand': Temperature associated to Randles circuit	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models
'efstrain': Local effective strain	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/batmac models	Solid/batmac models
'strainLocX/Y/Z': Local strain in the X/Y/Z directions	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/batmac models	Solid/batmac models
'soc,soceff': Local state of charge, effective state of charge	All models	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models
'current': Transverse Randles current	All models	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models
'ocv, vc, volt, r0': Open charge voltage, damping voltage, total voltage, r0 resistance.	All models	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models
'H_ex': Exothermal heating power integrated over time	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models

Variable names	Parameters with *DEFINE_FUNCTIONs: Randles circuit parameters (r_0, r_{10}, c_{10}, etc.)	RDLTYPE = -1	Internal short	Exothermic reaction
(= exothermal heating energy) when exothermic reaction keyword is present.				
'short': Short gives the state of the Randles circuit. If short = 0, then the circuit is not shorted, if short = 1, then it is shorted.	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models
'ero': Ero gives the state of the erosion. If ero = 1, then the circuit is adjacent to an element that has been eroded, ero = 0 otherwise.	Solid/batmac models	Solid/batmac models	Solid/batmac models	Solid/batmac models
'areaCircuit, areaCell, areashortGlob': Local Randles circuit area, total Randles cell area, total cell shorted area.	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models	Solid/tshell/batmac models
'save1, save2, save3…, save10': Ten local variables that you can define. Those additional parameters enable you to have trackable local variables that are saved during the run and associated with each local Randles circuit. Note that for the variable to be saved from the function, a "*" must proceed the variable name in the function. See *DEFINE_FUNCTION for an example.	No models	Solid/tshell/batmac models	No models	No models
'extvar': Value of an external variable (see *LOAD_EXTERNAL_VARIABLE)	No models	No models	Batmac models	No models

*EM_RANDLES_BATMAC

Purpose: Define the distributed Randles circuit parameters for a Randles cell when using the BatMac model. The BatMac model is a macro battery model where solid elements are retained for the solid mechanics and thermal solve, and each conducting node has its own Randles circuit associated with it. It must be used with *EM_MAT_005 or *EM_MAT_-006.

Card Summary:

Card 1. This card is required.

RDLID	RDLTYPE	RDLAREA	PSID				

Card 2. This card is required.

Q	CQ	SOCINIT	SOCTOU				

Card 3. This card is required.

R0CHA	R0DIS	R10CHA	R10DIS	C10CHA	C10DIS		

Card 4. This card is only read if RDLTYPE > 1. It is optional.

R20CHA	R20DIS	C20CHA	C20DIS	R30CHA	R30DIS	C30CHA	C30DIS

Card 5. This card is optional.

TEMP	FRTHER	R0TOTH	DUDT	TEMPU			

Card 6. This card is optional

USESOCS	TAU	FLCID					

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	RDLID	RDLTYPE	RDLAREA	PSID				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
RDLID	ID of the Randles cell
RDLTYPE	Type of Randles cell: EQ.-1: User-defined equivalent circuit model. See Remark 3. EQ.0: 0-order Randles Cell EQ.1: 1-order Randles Cell EQ.2: 2-order Randles Cell EQ.3: 3-order Randles Cell
RDLAREA	Randles area: EQ.1: The parameters are per unit area and are scaled in each Randle circuit by a factor depending on the local area of the circuit. Unit consistency in SI: Ohms times square meters. EQ.2: The parameters are defined for the whole cell and are scaled in each Randle circuit by a factor depending on the local area of the circuit and the global area of the cell (default). Unit consistency in SI: Ohms. EQ.3: The parameters are not scaled by area factors. Unit consistency in SI: Ohms. EQ.11: Same as 1, except the parameters are defined for the whole cell and are scaled in each Randles circuit by a factor depending on the local volume of the circuit and the global volume of the cell. EQ.22: Same as 2, except the parameters are defined for the whole cell and are scaled in each Randles circuit by a factor depending on the local volume of the circuit and the global volume of the cell.
PSID	Part set ID of all the parts composing the cell

Card 2	1	2	3	4	5	6	7	8
Variable	Q	CQ	SOCINIT	SOCTOU				
Type	F	F	F	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
Q	Cell capacity, Q
CQ	SOC conversion factor, c_q (%/s). It is known to be equal to 1/36 in SI units.
SOCINIT	Initial state of charge of the cell
SOCTOU	Equilibrium voltage (OCV): GE.0.0: Constant value LT.0.0: |SOCTOU| can refer to a load curve ID defining equilibrium voltage (OCV) as a function of the state of charge (SOC) or to a table ID with temperature indexing load curves that give OCV as a function of SOC.

Card 3	1	2	3	4	5	6	7	8
Variable	R0CHA	R0DIS	R10CHA	R10DIS	C10CHA	C10DIS		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
R0CHA/ R10CHA/ C10CHA	$r_0/r_{10}/c_{10}$ when the current flows in the charge direction: GE.0.0: Constant value LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit

VARIABLE	DESCRIPTION
	parameters can be made a function of the SOC and temperature.
R0DIS/ R10DIS/ C10DIS	$r_0/r_{10}/c_{10}$ when the current flows in the discharge direction: GE.0.0: Constant value LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit parameters can be made a function of the SOC and temperature.

Optional Card for RDLTYPE > 1. This card is only read for RDLTYPE > 1. It is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	R20CHA	R20DIS	C20CHA	C20DIS	R30CHA	R30DIS	C30CHA	C30DIS
Type	F	F	F	F	F	F	F	F
Default	0.	0.	0.	0.	0.	0.	0.	0.

VARIABLE	DESCRIPTION
R20CHA/ R30CHA/ C20CHA/ C30CHA	$r_{20}/r_{30}/c_{20}/c_{30}$ when the current flows in the charge direction: GE.0.0: Constant value LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit parameters can be made a function of the SOC and temperature.
R20DIS/ R30DIS/ C20DIS/ C30DIS	$r_{20}/r_{30}/c_{20}/c_{30}$ when the current flows in the discharge direction: GE.0.0: Constant value LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit parameters can be made a function of the SOC and temperature.

Optional Thermal Card.

Card 5	1	2	3	4	5	6	7	8
Variable	TEMP	FRTHER	R0TOTH	DUDT	TEMPU			
Type	F	I	I	F	I			
Default	0.	0	0	0.0	0			

VARIABLE	DESCRIPTION
TEMP	Constant temperature value used for the Randles circuit parameters when there is no coupling with the thermal solver (FRTHER = 0)
FRTHER	Flag setting where the temperature is coming from: EQ.0: The temperature used in the Randles circuit parameters is TEMP. EQ.1: The temperature used in the Randles circuit parameter is the temperature from the thermal solver.
R0TOTH	Flag for adding joule heating from r_0 to the thermal solve: EQ.0: The joule heating in the resistance r_0 is not added to the thermal solve. EQ.1: The joule heating in the resistance r_0 is added to the thermal solve.
DUDT	Reversible heat: GE.0.0: Constant value LT.0.0: |DUDT| is a load curve ID for the curve giving the reversible heat as a function of SOC.
TEMPU	Temperature unit: EQ.0: Celsius EQ.1: Kelvin

Optional SOC Correction Card.

Card 6	1	2	3	4	5	6	7	8
Variable	USESOCS	TAU	FLCID					
Type	I	F	F					
Default	0	0.0	0.0					

VARIABLE	DESCRIPTION
USESOCS	SOC correction method (see Remark 2): EQ.0: Off EQ.1: SOCshift method EQ.2: Capacity fade approach
TAU	Damping time, τ, in the SOCshift equation (see Remark 2)
FLCID	f for USESOCS = 1 or CFC for USESOCS = 2 (see Remark 2): GE.0.0: Constant value for f or 1. −CFC LT.0.0: |FLCID| is a load curve or table ID. For a load curve with USESOCS = 1, the curve gives f as a function of total current in the unit cell, i. For a table, temperature values index load curves giving $f(i)$ at each provided temperature. For a load curve with USESOCS = 2, the curve gives CFC as a function of i. For a table, temperature values index load curves giving CFC(i) at each provided temperature.

Remarks:

1. **Model combinations.** The batmac model cannot be mixed with the solid or thick shell Randles models. It can, however, be used in conjunction with the meshless model.

2. **Accounting for diffusion limitations.** With USESOC = 1, an extra term called SOCshift (or SocS) can be added to high-rate discharges (high C-rates) to account for diffusion limitations. The SOCshift is added to the SOC for the calculation of the OCV, $u(\text{SOC} + \text{SOCshift})$, and $r_0(\text{SOC} + \text{SOCshift})$. In this case, SOCshift satisfies the following equation:

$$\frac{\mathrm{d}(\mathrm{SOCshift})}{\mathrm{d}t} + \frac{\mathrm{SOCshift}}{\tau} = \frac{f(i(t), T)}{\tau}$$

with $\mathrm{SOCshift}(t = 0) = 0$. f is provided with FLCID. It can be constant, a function of total current in the unit cell (i), or a function of both i and the temperature (T).

As an alternative, SOC can be determined using a capacity fade method invoked by setting USESOC to 2:

$$\frac{\mathrm{d}(\mathrm{SOC})}{\mathrm{d}t} = -\frac{i(t)c_{\mathrm{q}}}{Q(1. - \mathrm{CFC}(i(t), T))}$$

In the above, Q is the cell capacity, and c_{q} is the SOC conversion factor. CFC is the capacity fade condition provided with FLCID. It can be constant, a function of i, or a function of both i and T. Note that in the constant case, FLCID specifies $1. - \mathrm{CFC}$.

3. **User-defined ECMs.** Randles circuits are based on the finite element Robin boundary condition, acting similarly to a convection boundary condition for the heat equation. As such, the circuit equation is decomposed into a term that enters the left-hand side of the stiffness matrix and a term that is added to the right-hand side. Changing the definition of those terms permits replacing the Randles circuits with any type of Equivalent Circuit model (ECM). This is the purpose of RDLTYPE = -1 where the term entering the left-hand side is defined by a negative integer referring to a *DEFINE_FUNCTION ID in R0CHA (unit consistency: resistance) while a negative integer in R0DIS associated with a *DEFINE_FUNCTION gives the term entering in the right-hand side (unit consistency: current).

*EM_RANDLES_EXOTHERMIC_REACTION

Purpose: Add an extra heat source term to the Randles circuit nodes to account for thermal runaway situations.

Card 1	1	2	3	4	5	6	7	8
Variable	AREATYPE	FUNCTID						
Type	I	I						
Default	2	none						

VARIABLE	DESCRIPTION
AREATYPE	Works the same way as RDLAREA in *EM_RANDLES_SOLID or *EM_RANDLES_TSHELL: EQ.1: The heat source in FUNCTID is per unit area so that, for each local Randles circuit, the result returned by FUNCTID is multiplied by a factor areaLoc. areaLoc is the local area associated with each Randles circuit while areaGlob is the area of the whole cell. Unit consistency in SI: W*m2. EQ.2: The heat source in FUNCTID is for the whole cell (the whole cell is shorted), so that, for each Randles circuit, the result returned by FUNCTID is multiplied by a factor areaLoc/areaGlob (default). Unit consistency in SI: W. EQ.3: The heat source returned by FUNCTID is taken as is in each Randles circuit. Unit consistency in SI: W. EQ.11: Same as 1, except the parameters are defined for the whole cell and are scaled in each Randles circuit by a factor depending on the local volume of the circuit and the global volume of the cell. EQ.22: Same as 2, except the parameters are defined for the whole cell and are scaled in each Randles circuit by a factor depending on the local volume of the circuit and the global volume of the cell.
FUNCTID	DEFINE_FUNCTION ID giving the local heat source as a function of local parameters for the local Randles circuit. See Table 6-2 in *EM_RANDLES.

*EM_RANDLES_MESHLESS

Purpose: Define the distributed Randles circuit parameters for a Randles cell that is not associated with a mesh (lumped Randles circuit).

Card Summary:

Card 1. This card is required.

RDLID	RDLTYPE						

Card 2. This card is required.

Q	CQ	SOCINIT	SOCTOU				

Card 3. This card is required.

R0CHA	R0DIS	R10CHA	R10DIS	C10CHA	C10DIS		

Card 4. This card is only read if RDLTYPE > 1. It is optional.

R20CHA	R20DIS	C20CHA	C20DIS	R30CHA	R30DIS	C30CHA	C30DIS

Card 5. This card is optional.

TEMP			DUDT	TEMPU			

Card 6. This card is optional.

USESOCS	TAU	FLCID					

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	RDLID	RDLTYPE						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
RDLID	ID of the Randles cell
RDLTYPE	Type of Randles cell: EQ.0: 0-order Randles cell EQ.1: 1-order Randles cell EQ.2: 2-order Randles cell EQ.3: 3-order Randles cell

Card 2	1	2	3	4	5	6	7	8
Variable	Q	CQ	SOCINIT	SOCTOU				
Type	F	F	F	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
Q	Cell capacity, Q
CQ	SOC conversion factor, c_q (%/s). It is known to be equal to 1/36 in SI units.
SOCINIT	Initial state of charge of the cell
SOCTOU	Equilibrium voltage (OCV): GE.0.0: Constant value LT.0.0: |SOCTOU| is a load curve ID defining the equilibrium voltage (OCV) as a function of the state of charge (SOC).

Card 3	1	2	3	4	5	6	7	8
Variable	R0CHA	R0DIS	R10CHA	R10DIS	C10CHA	C10DIS		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
R0CHA/ R10CHA/ C10CHA	$r_0/r_{10}/c_{10}$ when the current flows in the charge direction: GE.0.0: Constant value LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit parameters can be made a function of the SOC and temperature.
R0DIS/ R10DIS/ C10DIS	$r_0/r_{10}/c_{10}$ when the current flows in the discharge direction: GE.0.0: Constant value LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit parameters can be made a function of the SOC and temperature.

Optional Card for RDLTYPE > 1. This card is only read if RDLTYPE is greater than 1. It is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	R20CHA	R20DIS	C20CHA	C20DIS	R30CHA	R30DIS	C30CHA	C30DIS
Type	F	F	F	F	F	F	F	F
Default	0.	0.	0.	0.	0.	0.	0.	0.

VARIABLE	DESCRIPTION
R20CHA/ R30CHA/ C20CHA/ C30CHA	$r_{20}/r_{30}/c_{20}/c_{30}$ when the current flows in the charge direction: GE.0.0: Constant value LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit parameters can be made a function of the SOC and temperature.
R20DIS/ R30DIS/ C20DIS/ C30DIS	$r_{20}/r_{30}/c_{20}/c_{30}$ when the current flows in the discharge direction: GE.0.0: Constant value. LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit parameters can be made a function of the SOC and temperature.

Optional Thermal Card.

Card 5	1	2	3	4	5	6	7	8
Variable	TEMP			DUDT	TEMPU			
Type	F			F	I			
Default	0.			none	0			

VARIABLE	DESCRIPTION
TEMP	Constant temperature value used for the Randles circuit parameters
DUDT	Reversible heat: GE.0.0: Constant value LT.0.0: |DUDT| is a load curve ID for the curve giving the reversible heat as a function of SOC.
TEMPU	Temperature unit: EQ.0: Celsius

VARIABLE	DESCRIPTION
	EQ.1: Kelvin

Optional SOC Correction Card.

Card 6	1	2	3	4	5	6	7	8
Variable	USESOCS	TAU	FLCID					
Type	I	F	F					
Default	0	0.0	0.0					

VARIABLE	DESCRIPTION
USESOCS	SOC correction method (see Remark 1): EQ.0: Off EQ.1: SOCshift method EQ.2: Capacity fade approach
TAU	Damping time, τ, in the SOCshift equation. See Remark 1.
FLCID	f for USESOCS = 1 or CFC for USESOCS = 2 (see Remark 1): GE.0.0: Constant value for f or 1. −CFC LT.0.0: |FLCID| is a load curve or table ID. For a load curve with USESOCS = 1, the curve gives f as a function of total current in the unit cell, i. For a table, temperature values index load curves giving $f(i)$ at each provided temperature. For a load curve with USESOCS = 2, the curve gives CFC as a function of i. For a table, temperature values index load curves giving CFC(i) at each provided temperature.

Remarks:

1. **Accounting for diffusion limitations.** With USESOC = 1, an extra term called SOCshift (or SocS) can be added to high-rate discharges (high C-rates) to account for diffusion limitations. The SOCshift is added to the SOC for the calculation of the OCV, $u(\text{SOC} + \text{SOCshift})$, and $r_0(\text{SOC} + \text{SOCshift})$. In this case, SOCshift satisfies the following equation:

$$\frac{\mathrm{d}(\mathrm{SOCshift})}{\mathrm{d}t} + \frac{\mathrm{SOCshift}}{\tau} = \frac{f(i(t), T)}{\tau}$$

with $\mathrm{SOCshift}(t = 0) = 0$. f is provided with FLCID. It can be constant, a function of total current in the unit cell (i), or a function of both i and the temperature (T).

As an alternative, SOC can be determined using a capacity fade method invoked by setting USESOC to 2:

$$\frac{\mathrm{d}(\mathrm{SOC})}{\mathrm{d}t} = -\frac{i(t)c_{\mathrm{q}}}{Q(1. - \mathrm{CFC}(i(t), T))}$$

In the above, Q is the cell capacity, and c_{q} is the SOC conversion factor. CFC is the capacity fade condition provided with FLCID. It can be constant, a function of i, or a function of both i and T. Note that in the constant case, FLCID specifies $1. - \mathrm{CFC}$.

*EM_RANDLES_TSHELL

Purpose: Define the distributed Randles circuit parameters for a Randles cell when using a composite thick shell mechanical model.

Card Summary:

Card 1. This card is required.

RDLID	RDLTYPE	RDLAREA	PSID				

Card 2. This card is required.

Q	CQ	SOCINIT	SOCTOU				

Card 3. This card is required.

R0CHA	R0DIS	R10CHA	R10DIS	C10CHA	C10DIS		

Card 4. This is only read if RDLTYPE > 1. It is optional.

R20CHA	R20DIS	C20CHA	C20DIS	R30CHA	R30DIS	C30CHA	C30DIS

Card 5. This card is optional.

TEMP	FRTHER	R0TOTH	DUDT	TEMPU			

Card 6. This card is optional.

USESOCS	TAU	FLCID					

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	RDLID	RDLTYPE	RDLAREA	PSID				
Type	I	I	I	I				
Default	none	none	2	none				

VARIABLE	DESCRIPTION
RDLID	ID of the Randles cell
RDLTYPE	Type of Randles cell: EQ.-1: User-defined equivalent circuit model. See Remark 3. EQ.0: 0-order Randles cell EQ.1: 1-order Randles cell EQ.2: 2-order Randles cell EQ.3: 3-order Randles cell
RDLAREA	Randles Area: EQ.1: The parameters are per unit area and will be scaled in each Randles circuit by a factor depending on the local area of the circuit. Unit consistency in SI: Ohms times square meters. EQ.2: The parameters are defined for the whole cell and will be scaled in each Randles circuit by a factor depending on the local area of the circuit and the global area of the cell (default). Unit consistency in SI: Ohms. EQ.3: The parameters are not scaled by area factors. Unit consistency in SI: Ohms.
PSID	Part set ID of all the parts composing the cell (see Remark 1)

Card 2	1	2	3	4	5	6	7	8
Variable	Q	CQ	SOCINIT	SOCTOU				
Type	F	F	F	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
Q	Cell capacity, Q
CQ	SOC conversion factor, c_q (%/s). It is known to be equal to 1/36 in SI units.

VARIABLE	DESCRIPTION
SOCINIT	Initial state of charge of the cell
SOCTOU	Equilibrium voltage (OCV): GE.0.0: Constant value LT.0.0: |SOCTOU| is a load curve ID defining equilibrium voltage (OCV) as a function of the state of charge (SOC).

Card 3	1	2	3	4	5	6	7	8
Variable	R0CHA	R0DIS	R10CHA	R10DIS	C10CHA	C10DIS		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
R0CHA/ R10CHA/ C10CHA	$r_0/r_{10}/c_{10}$ when the current flows in the charge direction: GE.0.0: Constant value LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit parameters can be made a function of the SOC and temperature.
R0DIS/ R10DIS/ C10DIS	$r_0/r_{10}/c_{10}$ when the current flows in the discharge direction: GE.0.0: Constant value LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit parameters can be made a function of the SOC and temperature.

Optional Card for RDLTYPE > 1. This card is only read if RDLTYPE > 1. It is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	R20CHA	R20DIS	C20CHA	C20DIS	R30CHA	R30DIS	C30CHA	C30DIS
Type	F	F	F	F	F	F	F	F
Default	0.	0.	0.	0.	0.	0.	0.	0.

VARIABLE	DESCRIPTION
R20CHA/ R30CHA/ C20CHA/ C30CHA	$r_{20}/r_{30}/c_{20}/c_{30}$ when the current flows in the charge direction: GE.0.0: Constant value LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit parameters can be made a function of the SOC and temperature.
R20DIS/ R30DIS/ CDIS/ C30DIS	$r_{20}/r_{30}/c_{20}/c_{30}$ when the current flows in the discharge direction: GE.0.0: Constant value LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit parameters can be made a function of the SOC and temperature.

Optional Thermal Card.

Card 5	1	2	3	4	5	6	7	8
Variable	TEMP	FRTHER	ROTOTH	DUDT	TEMPU			
Type	F	I	I	F	I			
Default	0.	0	0	0.0	0			

VARIABLE	DESCRIPTION
TEMP	Constant temperature value used for the Randles circuit parameters when there is no coupling with the thermal solver (FRTHER = 0)
FRTHER	Flag about where the temperature is coming from: EQ.0: The temperature used in the Randles circuit parameters is TEMP. EQ.1: The temperature used in the Randle circuit parameter is the temperature from the thermal solver.
R0TOTH	Flag for adding joule heating from r_0 to the thermal solve: EQ.0: The joule heating in the resistance r_0 is not added to the thermal solve. EQ.1: The joule heating in the resistance r_0 is added to the thermal solve.
DUDT	Reversible heat: GE.0.0: Constant value LT.0.0: |DUDT| is a load curve ID for the curve givig the reversible heat as a function of SOC.
TEMPU	Temperature unit: EQ.0: Celsius EQ.1: Kelvin

Optional SOCShift Card.

Card 6	1	2	3	4	5	6	7	8
Variable	USESOCS	TAU	FLCID					
Type	I	F	F					
Default	0	0.0	0.0					

VARIABLE	DESCRIPTION
USESOCS	SOC correction method (see Remark 2): EQ.0: Off EQ.1: SOCshift method EQ.2: Capacity fade approach
TAU	Damping time in the SOCshift equation (see Remark 2)
FLCID	f for USESOCS = 1 or CFC for USESOCS = 2 (see Remark 2): GE.0.0: Constant value for f or 1. −CFC LT.0.0: |FLCID| is a load curve or table ID. For a load curve with USESOCS = 1, the curve gives f as a function of total current in the unit cell, i. For a table, temperature values index load curves giving $f(i)$ at each provided temperature. For a load curve with USESOCS = 2, the curve gives CFC as a function of i. For a table, temperature values index load curves giving CFC(i) at each provided temperature.

Remarks:

1. **Sectioning of a circuit.** *PART_COMPOSITE_TSHELL defines each part of PSID. With this keyword for defining the part, each layer of a part can serve a different function, namely, as a current collector positive, current collector negative, separator, negative electrode, or positive electrode. The RDLTYPE field of *EM_MAT_001 sets a given layer's function.

2. **Accounting for diffusion limitations.** With USESOC = 1, an extra term called SOCshift (or SocS) can be added to high-rate discharges (high C-rates) to account for diffusion limitations. The SOCshift is added to the SOC for the calculation of the OCV, $u(\text{SOC} + \text{SOCshift})$, and $r_0(\text{SOC} + \text{SOCshift})$. In this case, SOCshift satisfies the following equation:

$$\frac{\mathrm{d}(\text{SOCshift})}{\mathrm{d}t} + \frac{\text{SOCshift}}{\tau} = \frac{f(i(t), T)}{\tau}$$

with $\text{SOCshift}(t = 0) = 0$. f is provided with FLCID. It can be constant, a function of total current in the unit cell (i), or a function of both i and the temperature (T).

As an alternative, SOC can be determined using a capacity fade method invoked by setting USESOC to 2:

$$\frac{\mathrm{d}(\mathrm{SOC})}{\mathrm{d}t} = -\frac{i(t)c_{\mathrm{q}}}{Q(1.-\mathrm{CFC}(i(t),T))}$$

In the above, Q is the cell capacity, and c_{q} is the SOC conversion factor. CFC is the capacity fade condition provided with FLCID. It can be constant, a function of i, or a function of both i and T. Note that in the constant case, FLCID specifies $1.-\mathrm{CFC}$.

3. **User-defined ECMs.** Randles circuits are based on the finite element Robin boundary condition, acting similarly to a convection boundary condition for the heat equation. As such, the circuit equation is decomposed into a term that enters the left-hand side of the stiffness matrix and a term that is added to the right-hand side. Changing the definition of those terms permits replacing the Randles circuits with any type of Equivalent Circuit model (ECM). This is the purpose of RDLTYPE = -1 where the term entering the left-hand side is defined by a negative integer referring to a *DEFINE_FUNCTION ID in R0CHA (unit consistency: resistance) while a negative integer in R0DIS associated with a *DEFINE_FUNCTION gives the term entering in the right-hand side (unit consistency: current).

*EM_RANDLES_SHORT

Purpose: For a battery cell internal short, define conditions to turn on a Randles short (replace one or several Randles circuits by resistances). Also, define the value of the short resistance.

Card 1	1	2	3	4	5	6	7	8
Variable	AREATYPE	FUNCTID						
Type	I	I						
Default	2	none						

VARIABLE	DESCRIPTION
AREATYPE	Works the same way as RDLAREA in *EM_RANDLES_SOLID or *EM_RANDLES_TSHELL: EQ.1: The resistance inverse in FUNCTID is per unit area so that, for each local Randles circuit, the result returned by FUNCTID is multiplied by a factor 1./areaLoc (areaLoc is the local area associated with each Randles circuit, while areaGlob is the area of the whole cell). Unit consistency in SI: Ohms times square meters. EQ.2: The resistance in FUNCTID is for the whole cell (the whole cell is shorted), so that, for each Randles circuit, the result returned by FUNCTID is multiplied by a factor areaGlob/areaLoc (default). Unit consistency in SI: Ohms. EQ.3: The resistance returned by FUNCTID is taken as is for each Randles circuit. Unit consistency in SI:Ohms. EQ.11: Same as 1, except the parameters are defined for the whole cell and are scaled in each Randles circuit by a factor depending on the local volume of the circuit and the global volume of the cell. EQ.22: Same as 2, except the parameters are defined for the whole cell and are scaled in each Randles circuit by a factor depending on the local volume of the circuit and the global volume of the cell.
FUNCTID	Function ID (see *DEFINE_FUNCTION) for the function giving the local resistance function of local parameters for the local Randles

VARIABLE	DESCRIPTION
	circuit. See Remark 1. See Table 6-2 in *EM_RANDLES for the accepted function arguments. Additionally, external variable introduced with *LOAD_EXTERNAL_VARIABLE can be considered (see Remark 2).

Remarks:

1. **Short criterion.** If the return value of the function is zero, there is no short. The Randles circuit is maintained. A positive returned value causes replacing the Randles circuit with the returned short resistance. To ensure that the short is maintained even after the original criterion is no longer met, the default positive value may be replaced by a negative value. The solver then takes the absolute value returned and adopts it as the new short resistance in case the original short criterion is no longer met rather than reverting to a Randles circuit.

2. **External variable to trigger short circuits.** If an external variable is defined in the input for *LOAD_EXTERNAL_VARIABLE that influences the EM solver with property index 1 and a reference to the same function as referenced in FUNCTID, then the current and local value of the external variable can be considered as a function argument, `float extvar`.

3. ***Example *DEFINE_FUNCTION.** The following partial input provides an example of a function:

```
*DEFINE_FUNCTION
FID (Function Id)
Float resistance_short_randle(
float time,
float x_ccp,float y_ccp,float z_ccp,
float x_sep,float y_sep,float z_sep,
float x_sem,float y_sem,float z_sem,
float x_ccm,float y_ccm,float z_ccm)
{
  float seThick0;
  seThick0=1.e-5;
  seThick=(sqrt(x_sep-x_sem)^2+(y_sep-y_sem)^2+(z_sep-z_sem)^2);
  if (seThick >= seThick0) then
  return 1.e-2;
  else
  return -1.e-3;
  endif
}
```

   In this example, as long as seThick is smaller than seThick0, no short occurs. Once seThick becomes larger than seThick0, a short occurs, and the short resistance is 1.e-2. If during the run, seThick once again becomes smaller than seThick0, the short is maintained and the short resistance becomes 1.e-3. Replacing 1.e-3 by 0.0 causes the short to revert to the original Randles circuit.

4. **Erosion.** For the solid model (see *EM_RANDLES_SOLID), when an erosion-based criterion is defined, the erosion variable, ero, enables distinguishing which component of the battery has been eroded and adjusts the short resistance law accordingly:

   a) ero = 1 when the eroded element comes from the positive current collector.

   b) ero = 10 when the eroded element comes from the negative current collector.

   c) ero = 100 when the eroded element comes from the separator.

   d) ero = 1000 when the eroded element comes from the positive electrode.

   e) ero = 10000 when the eroded element comes from the negative electrode.

   When multiple components have eroded elements, ero is the sum of the indicators. For example, a returned value of 1001 indicates that both the elements coming from the positive electrode and the positive current collector have been eroded.

*EM_RANDLES_SOLID

Purpose: Define the distributed Randles circuit parameters for a Randles cell when using a solid mechanical model.

Card Summary:

Card 1. This card is required.

RDLID	RDLTYPE	RDLAREA	CCPPART	CCNPART	SEPPART	PELPART	NELPART

Card 2. This card is required.

Q	CQ	SOCINIT	SOCTOU				

Card 3. This card is required.

R0CHA	R0DIS	R10CHA	R10DIS	C10CHA	C10DIS		

Card 4. This card is only read for RDLTYPE > 1. It is optional.

R20CHA	R20DIS	C20CHA	C20DIS	R30CHA	R30DIS	C30CHA	C30DIS

Card 5. This card is optional.

TEMP	FRTHER	R0TOTH	DUDT	TEMPU			

Card 6. This card is optional.

USESOCS	TAU	FLCID					

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	RDLID	RDLTYPE	RDLAREA	CCPPART	CCNPART	SEPPART	PELPART	NELPART
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
RDLID	ID of the Randles cell
RDLTYPE	Type of Randles cell: EQ.-1: User-defined equivalent circuit model. See Remark 3. EQ.0: 0-order Randles cell EQ.1: 1-order Randles cell EQ.2: 2-order Randles cell EQ.3: 3-order Randles cell
RDLAREA	Randles Area: EQ.1: The parameters are per unit area and will be scaled in each Randles circuit by a factor depending on the local area of the circuit. Unit consistency in SI: Ohms times square meters. EQ.2: The parameters are defined for the whole cell and will be scaled in each Randles circuit by a factor depending on the local area of the circuit and the global area of the cell (default). Unit consistency in SI: Ohms. EQ.3: The parameters are not scaled by area factors. Unit consistency in SI: Ohms.
CCPPART	Current collector positive (CCP) part ID
CCNPART	Current collector negative (CCN) part ID
SEPPART	Separator part ID
PELPART	Positive electrode part ID
NELPART	Negative electrode part ID

Card 2	1	2	3	4	5	6	7	8
Variable	Q	CQ	SOCINIT	SOCTOU				
Type	F	F	F	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
Q	Cell capacity, Q
CQ	SOC conversion factor, c_q (%/s). It is known to be equal to 1/36 in SI units.
SOCINIT	Initial state of charge of the cell
SOCTOU	Equilibrium voltage (OCV): GE.0.0: Constant value LT.0.0: |SOCTOU| can refer to a load curve ID defining equilibrium voltage (OCV) as a function of the state of charge (SOC) or to a table ID with temperature indexing load curves that give OCV as a function of SOC.

Card 3	1	2	3	4	5	6	7	8
Variable	R0CHA	R0DIS	R10CHA	R10DIS	C10CHA	C10DIS		
Type	F	F	F	F	F	F		
Default	none	None	none	none	none	none		

VARIABLE	DESCRIPTION
R0CHA/ R10CHA/ C10CHA	$r_0/r_{10}/c_{10}$ when the current flows in the charge direction: GE.0.0: Constant value LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit parameters can be made a function of the SOC and temperature.
R0DIS/ R10DIS/ C10DIS	$r_0/r_{10}/c_{10}$ when the current flows in the discharge direction: GE.0.0: Constant value LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit

VARIABLE	DESCRIPTION
	parameters can be made a function of the SOC and temperature.

Optional Card for RDLTYPE > 1. This card is only read for RDLTYPE greater than 1. It is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	R20CHA	R20DIS	C20CHA	C20DIS	R30CHA	R30DIS	C30CHA	C30DIS
Type	F	F	F	F	F	F	F	F
Default	0.	0.	0.	0.	0.	0.	0.	0.

VARIABLE	DESCRIPTION
R20CHA/ R30CHA/ C20CHA/ C30CHA	$r_{20}/r_{30}/c_{20}/c_{30}$ when the current flows in the charge direction: GE.0.0: Constant value LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit parameters can be made a function of the SOC and temperature.
R20DIS/ R30DIS/ C20DIS/ C30DIS	$r_{20}/r_{30}/c_{20}/c_{30}$ when the current flows in the discharge direction: GE.0.0: Constant value LT.0.0: Absolute value is a function (see *DEFINE_FUNCTION) or table ID. See Table 6-2 in *EM_RANDLES for the accepted function arguments. For a table, the circuit parameters can be made a function of the SOC and temperature.

Optional Thermal Card.

Card 5	1	2	3	4	5	6	7	8
Variable	TEMP	FRTHER	R0TOTH	DUDT	TEMPU			
Type	F	I	I	F	I			
Default	0.	0	0	0.0	0			

VARIABLE	DESCRIPTION
TEMP	Constant temperature value used for the Randles circuit parameters when there is no coupling with the thermal solver (FRTHER = 0)
FRTHER	Flag about where the temperature is coming from: EQ.0: The temperature used in the Randles circuit parameters is TEMP. EQ.1: The temperature used in the Randle circuit parameter is the temperature from the thermal solver.
R0TOTH	Flag for adding joule heating from r_0 to the thermal solve: EQ.0: The joule heating in the resistance r_0 is not added to the thermal solve. EQ.1: The joule heating in the resistance r_0 is added to the thermal solve.
DUDT	Reversible heat: GE.0.0: Constant value LT.0.0: |DUDT| is a load curve ID for the curve giving the reversible heat as a function of SOC.
TEMPU	Temperature unit: EQ.0: Celsius EQ.1: Kelvin

Optional SOC Shift Card.

Card 6	1	2	3	4	5	6	7	8
Variable	USESOCS	TAU	FLCID					
Type	I	F	F					
Default	0	0.0	0.0					

VARIABLE	DESCRIPTION
USESOCS	SOC correction method (see Remark 2): EQ.0: Off EQ.1: SOCshift method EQ.2: Capacity fade approach
TAU	Damping time, τ, in the SOCshift equation (see Remark 2)
FLCID	f for USESOCS = 1 or CFC for USESOCS = 2 (see Remark 2): GE.0.0: Constant value for f or 1. −CFC LT.0.0: |FLCID| is a load curve or table ID. For a load curve with USESOCS = 1, the curve gives f as a function of total current in the unit cell, i. For a table, temperature values index load curves giving $f(i)$ at each provided temperature. For a load curve with USESOCS = 2, the curve gives CFC as a function of i. For a table, temperature values index load curves giving CFC(i) at each provided temperature.

Remarks:

1. **Element normal vector orientation.** The solid element normal vectors must all be oriented in the direction pointing from the positive current collector to the negative current collector to detect which current collector nodes are connected to one another. Furthermore, any number of layers can be modeled, but the meshes of the CCP, anode, separator, cathode, and CCN must be continuous and have merged nodes at the boundaries.

2. **Accounting for diffusion limitations.** With USESOC = 1, an extra term called SOCshift (or SocS) can be added to high-rate discharges (high C-rates) to account

for diffusion limitations. The SOCshift is added to the SOC for the calculation of the OCV, $u(\text{SOC} + \text{SOCshift})$, and $r_0(\text{SOC} + \text{SOCshift})$. In this case, SOCshift satisfies the following equation:

$$\frac{\mathrm{d}(\text{SOCshift})}{\mathrm{d}t} + \frac{\text{SOCshift}}{\tau} = \frac{f(i(t), T)}{\tau}$$

with $\text{SOCshift}(t = 0) = 0$. f is provided with FLCID. It can be constant, a function of total current in the unit cell (i), or a function of both i and the temperature (T).

As an alternative, SOC can be determined using a capacity fade method invoked by setting USESOC to 2:

$$\frac{\mathrm{d}(\text{SOC})}{\mathrm{d}t} = -\frac{i(t)c_{\mathrm{q}}}{Q(1. - \text{CFC}(i(t), T))}$$

In the above, Q is the cell capacity, and c_{q} is the SOC conversion factor. CFC is the capacity fade condition provided with FLCID. It can be constant, a function of i, or a function of both i and T. Note that in the constant case, FLCID specifies $1. - \text{CFC}$.

3. **User-defined ECMs.** Randles circuits are based on the finite element Robin boundary condition, acting similarly to a convection boundary condition for the heat equation. As such, the circuit equation is decomposed into a term that enters the left-hand side of the stiffness matrix and a term that is added to the right-hand side. Changing the definition of those terms permits replacing the Randles circuits with any type of Equivalent Circuit model (ECM). This is the purpose of RDLTYPE = -1 where the term entering the left-hand side is defined by a negative integer referring to a *DEFINE_FUNCTION ID in R0CHA (unit consistency: resistance) while a negative integer in R0DIS associated with a *DEFINE_FUNCTION gives the term entering in the right-hand side (unit consistency: current).

*EM_ROTATION_AXIS

Purpose: Define a rotation axis for the EM solver. This keyword is used with the 2D axisymmetric solver. The axis is defined by a point and a direction.

Card 1	1	2	3	4	5	6	7	8
Variable	XP	YP	ZP	XD	YD	ZD	NUMSEC	
Type	F	F	F	F	F	F	I	
Default	none	none	none	none	none	none	none	

Optional Card. Define an additional vector normal to the rotation vector that gives the rotation direction. See Remark 1.

Card 2	1	2	3	4	5	6	7	8
Variable	XT	YT	ZT	XH	YH	ZH		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
XP, YP, ZP	x, y, and z coordinates of the point
XD, YD, ZD	x, y, and z components of direction of the axis
NUMSEC	Number of sectors. This field gives the ratio of the full circle to the angular extension of the mesh. This has to be a power of two. For example, NUMSEC = 4 means that the mesh of the part represents one-fourth of the total circle. If NUMSEC = 0 for *EM_2DAXI, the solver replaces it with this value.
XT/YT/ZT and XH/YH/ZH	Tail and head coordinates of the additional vector orthogonal to the rotation axis that gives the direction of the rotation. See Remark 1.

Remarks:

1. **Simplified 2D axisymmetric input.** Starting with R15, the EM 2D axisymmetric solver enables specifying an additional vector orthogonal to the rotation axis that provides the rotation direction. When used, *EM_2DAXI is no longer needed. Also, SIDCUR, SIDVIN, and SIDVOUT in *EM_CIRCUIT are no longer needed, but PID becomes mandatory. This results in a simplified input deck (fewer segment sets and keywords need to be defined).

*EM_SOLVER_BEM

Purpose: Define the type of linear solver and pre-conditioner as well as tolerance for the EM_BEM solve.

Card 1	1	2	3	4	5	6	7	8
Variable	RELTOL	MAXITE	STYPE	PRECON	USELAST	NCYLBEM		
Type	I	I	I	I	I	I		
Default	10^{-6}	1000	2	2	1	5000		

VARIABLE	DESCRIPTION
RELTOL	Relative tolerance for the iterative solvers (PCG or GMRES). If the results are not accurate enough, the user should try to decrease this tolerance. More iterations will then be needed.
MAXITER	Maximum number of iterations for the iterative solvers
STYPE	Solver type: EQ.1: Direct solve. The matrices are considered dense. EQ.2: Pre-conditioned gradient method (PCG). This method allows for block matrices with low-rank blocks and thus reduces the memory used. EQ.3: GMRES method. This method allows for block matrices with low-rank blocks and thus reduces the memory used. The GMRES option only works in serial for now.
PRECON	Preconditioner type for PCG or GMRES iterative solves: EQ.-1: No preconditioner EQ.1: Diagonal line EQ.2: Diagonal block (default) EQ.3: Broad diagonal including all neighbor faces EQ.4: LLT factorization
USELAST	Only used for iterative solvers (PCG or GMRES). EQ.-1: Start from 0 as the initial guess for the solution of the

VARIABLE	DESCRIPTION
	linear system. EQ.1: Starts from the previous solution normalized by the RHS change. See Remark 1.
NCYLBEM	Number of electromagnetism cycles between the recalculation of BEM matrices. If negative, the absolute value refers to a load curve giving the number of cycles as a function of time. See Remarks 2 and 3.

Remarks:

1. **USELAST.** Using USELAST = 1 can save many iterations in the subsequent solves if the vector solution of the present solve can be assumed to be nearly parallel to the vector solution of the previous solve, as usually happens in time-domain eddy-current problems.

2. **Moving conductors.** Since the BEM matrices depend on (and only on) the surface node coordinates of the conductors, recalculating them when the conductors are moving is important. NCYLBEM controls the frequency with which they are updated. Note that very small values, for example, NCYLBEM = 1, should, generally, be avoided since this calculation involves a high computational cost. However, when two conductors are moving and in contact with each other, we recommend recalculating the matrices at *every* time step.

3. **Order of precedence for NCYLBEM.** You can set NCYLBEM with *EM_CONTROL_SOLUTION, *EM_SOLVER_BEM, and *EM_CONTROL. *EM_CONTROL_SOLUTION has the highest priority for setting this field. If NCYLBEM is the default value on *EM_CONTROL_SOLUTION, the solver looks at the setting of NCYLBEM on *EM_SOLVER_BEM. If left as default on this keyword, the solver looks at the setting of NCYLBEM on *EM_CONTROL.

*EM_SOLVER_BEM_REASSEMBLY

Purpose: Specify the domain decomposition method, and for certain domain decomposition methods, request storing portions of the BEM matrices to save calculation time during BEM system recomputations.

Card 1	1	2	3	4	5	6	7	8
Variable	IOCT							
Type	I							
Default	0							

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	PSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
IOCT	Octree domain decomposition method (see Remark 1): EQ.0: BEM faces are grouped in domains based on their proximity. EQ.1: Same as 0 except the BEM faces from a given domain must belong to the same part. In addition, portions of the BEM matrix can be stored by specifying the parts for storage in PSID on Card 2.
PSID	Part sets IDs for which BEM information is kept. Ignored for IOCT = 0. See Remark 2.

Remarks:

1. **IOCT.** The domain decomposition method of the BEM faces is typically handled internally and is not of direct practical use to the user. Switching between IOCT = 0 and IOCT = 1 may slightly impact the calculation times one way or another, depending on the case studied. However, it should not significantly affect any calculation result, as only the order of some operations will vary. Using IOCT = 1 allows for storing in memory BEM information that belongs to certain parts throughout the calculation, which in turn saves calculation time during subsequent BEM matrices reassemblies. The parts are selected with part set IDs specified on Card 2.

2. **Rigid parts.** The parts that belong to the various PSIDs should only belong to non-deformable structures (typically rigid bodies) that do not move relative to each other.

*EM_SOLVER_BEMMAT

Purpose: Define the type of BEM matrices as well as the way they are assembled.

Card 1	1	2	3	4	5	6	7	8
Variable	MATID							RELTOL
Type	I							F
Default	none							10^{-6}

VARIABLE	DESCRIPTION
MATID	Defines which BEM matrix the card refers to: EQ.1: **P** matrix EQ.2: **Q** matrix EQ.3: **W** matrix
RELTOL	Relative tolerance on the sub-blocks of the matrix when doing low rank approximations. The user should try to decrease these tolerances if the results are not accurate enough. More memory will then be needed.

Remarks:

1. The **W** matrix only exists when the monolithic solver is activated (see *EM_SOLVER_FEMBEM_MONOLITHIC).

*EM_SOLVER_FEM

Purpose: Define some parameters for the EM FEM solver.

Card 1	1	2	3	4	5	6	7	8
Variable	RELTOL	MAXITE	STYPE	PRECON	USELAST	NCYLFEM		
Type	I	I	I	I	I	I		
Default	10^{-3}	1000	1	1	1	5000		

VARIABLE	DESCRIPTION
RELTOL	Relative tolerance for the iterative solvers (PCG and Hybrid Direct-PCG). If the results are not accurate enough, try decreasing this tolerance. More iterations will then be needed.
MAXITER	Maximum number of iterations for iterative solvers (PCG and Hybrid Direct-PCG)
STYPE	Solver type: EQ.1: Direct solve EQ.2: Preconditioned Conjugate Gradient Method (PCG) EQ.3: Hybrid Direct-PCG method. See Remark 4.
PRECON	Preconditioner type for PCG. EQ.-1: No preconditioner EQ.1: Drop tolerance (default) EQ.2: Diagonal line EQ.5: MUMPS factorization
USELAST	This is used only for iterative solvers (PCG and Hybrid Direct-PCG). EQ.-1: Starts from 0 as the initial solution of the linear system. EQ.1: Starts from the previous solution, normalized by the right-hand-side change. See Remark 1.
NCYLFEM	Number of electromagnetism cycles between the recalculation of

VARIABLE	DESCRIPTION
	FEM matrices. If negative, the absolute value refers to a load curve giving NCYLFEM as a function of time. See Remarks 2 and 3.

Remarks:

1. **Starting from the previous solution.** Using USELAST = 1 can save many iterations in the subsequent solves if the vector solution of the present solve can be assumed to be nearly parallel to the vector solution of the previous solve, as usually happens in time-domain eddy-current problems.

2. **NCYLFEM.** When the conductor parts are deforming or undergoing changes in their EM material properties (conductivity for example), the FEM matrices should be recalculated more often, so NCYLFEM may need to be changed.

3. **Order of precedence for NCYLFEM.** You can set NCYLFEM with *EM_CONTROL_SOLUTION, *EM_SOLVER_FEM, and *EM_CONTROL. *EM_CONTROL_SOLUTION has the highest priority for setting this field. If NCYLFEM is the default value on *EM_CONTROL_SOLUTION, the solver looks at the setting of NCYLFEM on *EM_SOLVER_FEM. If left as default on this keyword, the solver looks at the setting on *EM_CONTROL.

4. **Hybrid Direct-PCG approach.** The Hybrid Direct-PCG approach has been developed for large problems, especially battery problems that involve small matrix changes between steps (due to temperature-dependent Randles circuits). It works as follows: Initially, a factorization is performed, followed by a direct solve. Then, for the subsequent steps, the code checks whether the stiffness matrix has changed. If not, then a follow-up direct solve is done, which is usually very fast. If the stiffness matrix has changed, rather than redo a complete factorization (which is costly), the solver automatically switches to a conjugate gradient solve using the original factorization as a preconditioner and RELTOL and MAXITER as convergence criteria. In cases with only small changes in the stiffness matrix, this allows for quick convergence in just a few iterations, significantly reducing the total solve time compared to either a pure direct approach or a pure iterative approach. If the conjugate gradient does not converge within the bounds set by RELTOL and MAXITER, the factorization is automatically discarded, and a new attempt is made using the direct solver.

*EM_SOLVER_FEMBEM

Purpose: Define some parameters for the standard coupling between the EM FEM and EM BEM solvers.

Card 1	1	2	3	4	5	6	7	8
Variable	RELTOL	MAXITE	FORCON					
Type	F	I	I					
Default	10^{-2}	50	0					

VARIABLE	DESCRIPTION
RELTOL	Relative tolerance for the FEM/BEM system solve. If the results are not accurate enough, try decreasing this tolerance. A smaller tolerance will, however, require more iterations.
MAXITER	Maximal number of iterations
FORCON	Force convergence: EQ.0: The code stops with an error if no convergence. EQ.1: The code continues to the next time step even if the RELTOL convergence criterion has not been reached.

Remarks:

This keyword couples the FEM and BEM systems with the Richardson method. At each time step, the solver iterates between the FEM and the BEM systems until convergence is reached (based on the choice of RELTOL and MAXITER). The cost for this solve is low. However, to ensure stability, we recommend imposing a limit on the timestep based on the characteristic diffusion time (See *EM_CONTROL_TIMESTEP). Furthermore, it can be unstable whenever magnetic materials are involved (conductor's permeability different than vacuum permeability). The monolithic solver invoked with *EM_SOLVER_FEMBEM_MONOLITHIC aims to remove those two limitations by solving both the FEM and BEM systems in one single monolithic block. For such cases, it is, therefore, the recommended choice (see *EM_SOLVER_FEMBEM_MONOLITHIC).

*EM_SOLVER_FEMBEM_MONOLITHIC

Purpose: Replaces *EM_SOLVER_FEMBEM and turns on the monolithic FEM-BEM solver.

Card 1	1	2	3	4	5	6	7	8
Variable	MTYPE	STYPE	ABSTOL	RELTOL	MAXIT			
Type	I	I	F	F	I			
Default	0	0	10^{-6}	10^{-4}	500			

Optional Card. Optional card for nonlinear magnetic cases.

Card 2	1	2	3	4	5	6	7	8
Variable		NTSTOL	NTATOL	NTRTOL	NTMIT			
Type		F	F	F	I			
Default		10^{-5}	10^{-12}	10^{-4}	20			

Optional Card. Optional card for nonlinear magnetic cases.

Card 3	1	2	3	4	5	6	7	8
Variable	LS_ON							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
MTYPE	Monolithic solver type: EQ.0: Direct symmetric solver

VARIABLE	DESCRIPTION
STYPE	Solver type: EQ.0: MINRES iterative solver EQ.1: GMRES iterative solver
ABSTOL	Absolute tolerance
RELTOL	Relative tolerance
MAXIT	Maximum number of iterations
NTSTOL/NTATOL/ NTROL	Solution change tolerance, absolute tolerance, and relative tolerance used during the nonlinear Newton step. These tolerances only apply when the EM model includes nonlinear magnetic materials.
NTMIT	Maximum nonlinear iterations
LS_ON	Line search: EQ.0: Off EQ.1: On (only used for nonlinear magnetic materials)

Remarks:

The monolithic solver aims to overcome the limitations of the classic Richardson iterative coupling between the FEM and BEM systems. The monolithic solver offers better stability for large time steps and for simulations involving ferromagnetic materials. We recommend this method whenever *EM_MAT_002 or *EM_MAT_007 is present.

*ICFD

The keyword *ICFD covers all the different options available in the implicit computational fluid dynamics solver . The keyword cards in this section are defined in alphabetical order:

*ICFD_BOUNDARY_CONJ_HEAT

*ICFD_BOUNDARY_CONVECTION_TEMP

*ICFD_BOUNDARY_FLUX_SPECIES

*ICFD_BOUNDARY_FLUX_TEMP

*ICFD_BOUNDARY_FREESLIP

*ICFD_BOUNDARY_FSI

*ICFD_BOUNDARY_FSI_EXCLUDE

*ICFD_BOUNDARY_FSI_FIXED

*ICFD_BOUNDARY_FSI_ONEWAY

*ICFD_BOUNDARY_FSWAVE

*ICFD_BOUNDARY_GROUND

*ICFD_BOUNDARY_IMMERSED_DYNAMIC

*ICFD_BOUNDARY_IMMERSED_STATIC

*ICFD_BOUNDARY_ISENTROPIC_NRBC

*ICFD_BOUNDARY_ISENTROPIC_NRBC0

*ICFD_BOUNDARY_NAVIERSLIP

*ICFD_BOUNDARY_NONSLIP

*ICFD_BOUNDARY_PERIODIC

*ICFD_BOUNDARY_PRESCRIBED_LEVELSET

*ICFD_BOUNDARY_PRESCRIBED_MOVEMESH

*ICFD_BOUNDARY_PRESCRIBED_PRE

*ICFD_BOUNDARY_PRESCRIBED_SPTRANSP_CONC

*ICFD_BOUNDARY_PRESCRIBED_TEMP

*ICFD_BOUNDARY_PRESCRIBED_TURBULENCE

*ICFD_BOUNDARY_PRESCRIBED_VEL

*ICFD_BOUNDARY_PRESCRIBED_VISCOELASTIC

*ICFD_BOUNDARY_PRESCRIBED_WATERVAPOR

*ICFD_BOUNDARY_WEAKVEL

*ICFD_BOUNDARY_WINDKESSEL

*ICFD_CONTROL_ADAPT

*ICFD_CONTROL_ADAPT_SIZE

*ICFD_CONTROL_ADVECTION

*ICFD_CONTROL_BACKFLOW

*ICFD_CONTROL_CONJ

*ICFD_CONTROL_DEM_COUPLING

*ICFD_CONTROL_EMBEDSHELL

*ICFD_CONTROL_FSI

*ICFD_CONTROL_GAP

*ICFD_CONTROL_GENERAL

*ICFD_CONTROL_IMMERSED

*ICFD_CONTROL_IMMERSED_FSI

*ICFD_CONTROL_IMPOSED_MOVE

*ICFD_CONTROL_ISENTROPIC

*ICFD_CONTROL_LEVELSET

*ICFD_CONTROL_LOAD

*ICFD_CONTROL_MESH

*ICFD_CONTROL_MESH_MOV

*ICFD_CONTROL_MONOLITHIC

*ICFD_CONTROL_OUTPUT

*ICFD_CONTROL_OUTPUT_SUBDOM

*ICFD_CONTROL_OUTPUT_VAR

*ICFD_CONTROL_PARTITION

*ICFD_CONTROL_POROUS

*ICFD_CONTROL_RANS

*ICFD_CONTROL_STEADY

*ICFD_CONTROL_SURFMESH

*ICFD_CONTROL_TAVERAGE

*ICFD_CONTROL_TIME

*ICFD_CONTROL_TRANSIENT

*ICFD_CONTROL_TURB_SYNTHESIS

*ICFD_CONTROL_TURBULENCE

*ICFD_DATABASE_AVERAGE

*ICFD_DATABASE_BIN

*ICFD_DATABASE_DRAG

*ICFD_DATABASE_FLUX

*ICFD_DATABASE_FLUXSURF

*ICFD_DATABASE_FORCE_DEM

*ICFD_DATABASE_GOA

*ICFD_DATABASE_HTC

*ICFD_DATABASE_NODEAVG

*ICFD_DATABASE_NODOUT

*ICFD_DATABASE_NTEMPOUT

*ICFD_DATABASE_POINTAVG

*ICFD_DATABASE_POINTOUT

*ICFD_DATABASE_RESIDUALS

*ICFD_DATABASE_SSOUT

*ICFD_DATABASE_SSOUT_EXCLUDE

*ICFD_DATABASE_TEMP

*ICFD_DATABASE_TIMESTEP

*ICFD_DATABASE_TPD

*ICFD_DATABASE_TWINBUILDER

*ICFD_DATABASE_UINDEX

*ICFD_DATABASE_VOL

*ICFD_DATABASE_WETNESS

*ICFD_DEFINE_GAPCLOSURE

*ICFD_DEFINE_GAPRESTRIC

*ICFD_DEFINE_HEATSOURCE

*ICFD_DEFINE_HEMOLYSIS_INDEX

*ICFD_DEFINE_NONINERTIAL

*ICFD_DEFINE_POINT

*ICFD_DEFINE_POROUS_REGION

*ICFD_DEFINE_RESIDENCETIMESOURCE

*ICFD_DEFINE_SOURCE

*ICFD_DEFINE_SPTRANSPSOURCE

*ICFD_DEFINE_TRANSFORM

*ICFD_DEFINE_TURBSOURCE

*ICFD_DEFINE_WAVE_DAMPING

*ICFD_INITIAL

*ICFD_INITIAL_BIN

*ICFD_INITIAL_LEVELSET

*ICFD_INITIAL_SPTRANSP

*ICFD_INITIAL_TEMPNODE

*ICFD_INITIAL_TURBULENCE

*ICFD_INITIAL_VISCOELASTIC

*ICFD_INITIAL_WATERVAPOR

*ICFD_MAT

*ICFD_MODEL_NONNEWT

*ICFD_MODEL_POROUS

*ICFD_MODEL_SPECIES_TRANSPORT

*ICFD_MODEL_VISCOELASTIC

*ICFD_PART

*ICFD_PART_VOL

*ICFD_SECTION

*ICFD_SET_NODE

*ICFD_SOLVER_SPLIT

*ICFD_SOLVER_TOL_FSI

*ICFD_SOLVER_TOL_LSET

*ICFD_SOLVER_TOL_MMOV

*ICFD_SOLVER_TOL_MOM

*ICFD_SOLVER_TOL_MONOLITHIC

*ICFD_SOLVER_TOL_PRE

*ICFD_SOLVER_TOL_TEMP

*ICFD_BOUNDARY_CONJ_HEAT

Purpose: Specify which boundary of the fluid domain exchanges heat with the structure.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	CTYPE	VAL	SFLCID				
Type	I	I	F	I				
Default	none	0	0.	0				

VARIABLE	DESCRIPTION
PID	PID of the fluid surface in contact with the structure
CTYPE	Contact type: EQ.0: Constraint approach EQ.1: Mortar contact
VAL	Optional temperature drop if CTYPE = 0 or interface heat transfer coefficient if CTYPE = 1 (high value by default to ensure perfect contact).
SFLCID	Load curve ID used to describe scale factor on VAL value as a function of time; see *DEFINE_CURVE, *DEFINE_CURVE_FUNCTION, or *DEFINE_FUNCTION. If a *DEFINE_FUNCTION is used, the following parameters are allowed: f(x, y, z, vx, vy, vz, temp, pres, time).

*ICFD_BOUNDARY_CONVECTION_TEMP

Purpose: Impose a heat transfer coefficient on the boundary expressed as $h = \frac{q}{T_s - T_b}$

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	HLCID	HSF	TBLCID	TBSF			
Type	I	I	F	I	F			
Default	none	none	1.	none	1.0			

VARIABLE	DESCRIPTION
PID	PID for a fluid surface.
HLCID	Load curve ID to describe the heat transfer coefficient value versus time, see *DEFINE_CURVE,*DEFINE_CURVE_FUNCTION or *DEFINE_FUNCTION. If a DEFINE_FUNCTION is used, the following parameters are allowed: $f(x, y, z, vx, vy, vz, temp, pres, time)$.
HSF	Load curve scale factor applied on the heat transfer coefficient value. (default = 1.0)
TBLCID	Load curve ID to describe the environment (i.e bulk) temperature value versus time, see *DEFINE_CURVE,*DEFINE_CURVE_FUNCTION or *DEFINE_FUNCTION. If a DEFINE_FUNCTION is used, the following parameters are allowed: $f(x, y, z, vx, vy, vz, temp, pres, time)$.
TBSF	Load curve scale factor applied on the environment value. (default = 1.0)

*ICFD_BOUNDARY_FLUX_SPECIES

Purpose: Specify the fluxes of concentration for the transported species at the boundaries.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	LCID1	LCID2	LCID3	LCID4	LCID5	LCID6	LCID7	LCID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
PID	Part ID of the boundary with the concentrations flux boundary conditions
LCID*i*	Load curve ID for the curve giving the flux of concentration of species *i* at the boundary as a function of time. Note that all the transported species must have a load curve defined. For instance, in the four-species case, LCID1, LCID2, LCID3, and LCID4 must be specified.

*ICFD_BOUNDARY_FLUX_TEMP

Purpose: Impose a heat flux on the boundary expressed as $q = -k\nabla T$

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	LCID	SF	DEATH	BIRTH			
Type	I	I	F	F	F			
Default	none	none	1.	1.E+28	0.0			

VARIABLE	DESCRIPTION
PID	PID for a fluid surface.
LCID	Load curve ID to describe the temperature flux value versus time, see *DEFINE_CURVE, *DEFINE_CURVE_FUNCTION or *DEFINE_FUNCTION. . If a DEFINE_FUNCTION is used, the following parameters are allowed: $f(x, y, z, vx, vy, vz, temp, pres, time)$.
SF	Load curve scale factor. (default = 1.0)
DEATH	Time at which the imposed motion/constraint is removed: EQ.0.0: default set to 10e28
BIRTH	Time at which the imposed pressure is activated starting from the initial abscissa value of the curve

*ICFD_BOUNDARY_FREESLIP

Purpose: Specify the fluid boundary with free-slip boundary condition.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
PID	Part ID of the fluid surface where a free-slip boundary condition is applied

*ICFD_BOUNDARY_FSI

Purpose: Define which fluid surfaces will be considered in contact with the structure surfaces for fluid-structure interaction (FSI) analysis. This keyword should not be defined if *ICFD_CONTROL_FSI is not defined.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
PID	Part ID of the fluid surface in contact with the structure domain

*ICFD_BOUNDARY_FSI_EXCLUDE

Purpose: Specify which solid part IDs are excluded from the FSI search. No forces coming from the fluid will be transmitted to those parts.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
PID	Part ID of a part from the solid mechanics problem that is to be excluded from the FSI analysis

*ICFD_BOUNDARY_FSI_FIXED

Purpose: Define fixed fluid surfaces that will be considered for contact with the solid surfaces for FSI. This keyword is similar to *ICFD_BOUNDARY_FSI, except the fluid surface cannot move. This restriction allows a solid surface to "slide" over the fluid and for the exchange of data, such as temperature in CHT applications. This keyword should not be defined if *ICFD_CONTROL_FSI is not defined.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
PID	Part ID of the fluid surface that can be considered for contact with the solid domain

*ICFD_BOUNDARY_FSI_ONEWAY

Purpose: Specify which solid part IDs experience one-way FSI coupling. These part IDs follow the structural motion to capture the structural deformation.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	IOWC						
Type	I	I						
Default	none	1						

VARIABLE	DESCRIPTION
PID	Part ID of a part from the solid mechanics problem for which only one-way FSI coupling occurs
IOWC	Indicates the coupling direction to the solver: EQ.1: The solid mechanics solver transfers displacements to the fluid solver. EQ.2: The fluid solver transfers stresses to the solid mechanics solver.

*ICFD_BOUNDARY_FSWAVE

Purpose: Impose a wave inflow boundary condition.

Card Summary:

Card Sets. Include as many sets of the following cards as needed. This input ends with the next keyword ("*") card.

Card 1. This card is required.

PID	WTYPE	H0	WAMP	WLENG	WMAX	SFLCID	WANG

Card 2. This card is included if WTYPE = 7.

WPEAK							

Data Card Definitions:

Include as many of this card and/or sets of this card with the next as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	WTYPE	H0	WAMP	WLENG	WMAX	SFLCID	WANG
Type	I	I	F	F	F	F	I	F
Default	none	none	none	none	none	none	↓	none

VARIABLE	DESCRIPTION
PID	PID for a fluid surface
WTYPE	Wave type: EQ.1: Stokes wave of first order EQ.2: Stokes wave of second order EQ.3: Stokes wave of fifth order EQ.4: Solitary wave EQ.5: Irregular waves using JONSWAP spectrum

VARIABLE	DESCRIPTION
	EQ.6: Irregular waves using One Parameter Pierson-Moskowitz spectrum EQ.7: Irregular waves using Two Parameter Pierson-Moskowitz spectrum
H0	Water level (from the bottom of the channel) for the unperturbed condition
WAMP	Wave amplitude or height for WTYPE = 1 and 4. Significant wave height for WTYPE = 5, 6, and 7.
WLENG	WTYPE.LE.2: Wave length WTYPE.EQ.3: Wave period WTYPE.EQ.4: Not used WTYPE.GE.5: Minimum wave frequency in spectrum (rad/sec)
WMAX	Maximum wave frequency in spectrum (rad/sec) for WTYPE = 5, 6, and 7. Angle between the boundary and the incident waves (in degrees) for WTYPE = 3.
SFLCID	Scale factor LCID on the wave amplitude for WTYPE = 1, 2 and 3. Number of wave modes (default = 1024) for WTYPE = 5, 6, and 7.
WANG	Angle between incoming wave direction and x-axis for z- and y-aligned gravity vector, or angle between incoming wave direction and y-axis for x-aligned gravity vector.

Card included for WTYPE = 7 only

Card 2	1	2	3	4	5	6	7	8
Variable	WPEAK							
Type	F							
Default	none							

VARIABLE	DESCRIPTION
WPEAK	Peak wave frequency in spectrum [rad/sec] for WTYPE = 7.

Remarks:

1. **Peak Wave Frequency for WTYPE = 6.** For the irregular waves using the One Parameter Pierson-Moskowitz spectrum, the peak wave frequency in the spectrum LS-DYNA calculates the peak wave frequency with:

$$0.4\sqrt{\frac{g}{H_s}}$$

Here g the gravity and H_s is the significant wave height input with WAMP.

*ICFD_BOUNDARY_GROUND

Purpose: Specify the fluid boundary with a ground boundary condition. The ground boundary condition is similar to the nonslip boundary condition except that it will keep V = 0 in all circumstances, even if the surface nodes are moving. This is useful in cases where the nodes are allowed to move or translate (using ICFD_BOUNDARY_PRESCRIBED_MOVEMESH for example) but those displacements are only to accommodate for mesh movement and do not correspond to a physical motion.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
PID	PID of the fluid surface where a ground boundary condition is applied.

*ICFD_BOUNDARY_IMMERSED_DYNAMIC

Purpose: List solid parts IDs for immersed solid parts that are moving (dynamic). All other immersed solid parts are treated as static. This keyword overrides ISSTA = 0 on *ICFD_CONTROL_IMMERSED. However, if ISSTA = 1 on *ICFD_CONTROL_IMMERSED, this keyword is ignored, and all solid parts are treated as static.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	PID2	PID3	PID4	PID5	PID6	PID7	PID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
PID	Solid part IDs to be treated as dynamic immersed boundaries

Remarks

1. **Supported immersed interface methods.** This keyword applies to immersed ICFD coupling activated through IMMTH = 4 on *ICFD_CONTROL_IMMERSED.

2. **Restrictions.** Do not use this keyword and *ICFD_BOUNDARY_IMMERSED_STATIC in the same input deck. The two keywords are mutually exclusive.

*ICFD_BOUNDARY_IMMERSED_STATIC

Purpose: List solid part IDs for immersed solid parts that are stationary. The immersed interface search can reuse previously computed information for these parts saving computational time. Immersed parts not listed remain treated as dynamic. However, if ISSTA = 1 on *ICFD_CONTROL_IMMERSED, this keyword is ignored and all solid parts are treated as static.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID1	PID2	PID3	PID4	PID5	PID6	PID7	PID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
PID	Part ID of the structural part that is static and not moving

Remarks

1. **Supported immersed interface methods.** This keyword applies to immersed ICFD coupling activated through IMMTH = 4 on *ICFD_CONTROL_IMMERSED.

2. **Restrictions.** Do not use this keyword and *ICFD_BOUNDARY_IMMERSED_DYNAMIC in the same input deck. The two keywords are mutually exclusive..

*ICFD_BOUNDARY_ISENTROPIC_NRBC_{OPTION}

Available options include:

<BLANK>

TITLE

Purpose: Define a non-reflecting boundary condition (NRBC) on surfaces bounding an isentropic compressible fluid (see FLG = 2 on *ICFD_MAT) with selectable inlet, outlet, and radiating slip-wall formulations.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	NW	BETASF	VXLCID	VYLCID	VZLCID	PREFLCID	KIND
Type	I	I	F	I	I	I	I	I
Default	none	100	10.0	0	0	0	0	0

VARIABLE	DESCRIPTION
PID	Part ID of the fluid surface where the boundary condition is applied
NW	Number of time steps defining the exponential averaging window at this boundary. Values less than 1 use a window of 100 time steps. See Remark 1.
BETASF	Dimensionless multiplier for the Nitsche penalty in the KIND = 0 formulation. The default value is 10.0. A negative value disables the Nitsche penalty. This parameter is unused for KIND = 1 through 5. See Remark 2.
VXLCID, VYLCID, VZLCID	Load curve IDs defining the reference velocity components in the global X-, Y-, and Z-directions as functions of time. A blank or nonpositive ID specifies no curve for that component. See Remark 1 for the distinction between the prescribed inlet target and the averaged radiation target.
PREFLCID	Load curve ID defining the far-field static (reference) gauge pressure, $\bar{p}_g$, as a function of time. When positive, this curve prescribes the reference pressure throughout the calculation. A blank or

VARIABLE	DESCRIPTION
	nonpositive ID specifies no pressure curve. See Remarks 1 and 3.
KIND	Boundary formulation: EQ.0: Inlet or lateral boundary with weak imposition of the mean velocity by a Nitsche penalty and acoustic radiation. See Remark 4. EQ.1: Outlet with Sommerfeld radiation and a strongly imposed pressure condition. See Remark 4. EQ.2: Outlet with Sommerfeld radiation, a weak pressure-fluctuation load, and a pressure absorber. See Remark 4. EQ.3: Radiating slip-wall formulation with Sommerfeld radiation and a pressure absorber. See Remark 4. EQ.4: Zeroth-order impedance traction outlet. See *ICFD_BOUNDARY_ISENTROPIC_NRBC0. See Remark 5. EQ.5: Same as KIND = 4 with additional stabilization when backflow occurs. See Remark 5. Values outside the range 0 through 5 are treated as 0.

Remarks:

1. **Reference fields.** At this boundary, the running means use the exponential update below, independent of NW on *ICFD_CONTROL_ISENTROPIC:

$$\bar{q}^{n+1} = (1-\alpha)\bar{q}^{n} + \alpha q^{n+1}, \quad \alpha = 2/(\mathrm{NW}+1) \ .$$

The initial mean velocity is taken from the initial velocity, except where a velocity curve supplies the initial value. For KIND = 0, each supplied velocity curve continues to prescribe the corresponding radiation target; components without curves retain their running mean. The Nitsche inlet target uses the supplied velocity curves and zero for any unspecified component. For KIND = 1 through 5, velocity curves seed the initial mean only; the mean velocity subsequently follows the running average.

If PREFLCID > 0, the reference gauge pressure, $\bar{p}_g$, follows the specified curve throughout the simulation for every value of KIND. The curve is evaluated at the current simulation time, and its value is not averaged using NW. If no curve is specified, the reference gauge pressure starts at the initial pressure. For KIND = 0 through 3, it then follows the boundary running average controlled by NW. For KIND = 4 and 5, it remains fixed at the initial pressure, so NW does not affect the reference gauge pressure.

2. **Penalty scaling.** For KIND = 0, BETASF multiplies the local viscous and convective scales to determine the penalty coefficient, β. The implemented penalty coefficient includes a small acoustic-impedance floor:

$$\beta = \text{BETASF} \times \max\left(\frac{\mu}{h} + \rho|\mathbf{u}|, 10^{-3}\rho c\right).$$

Here μ is the dynamic viscosity, h is the local mesh size, ρ is the density, $\mathbf{u}$ is the velocity, and c is the local sound speed. Note that BETASF is dimensionless while β is dimensional. The penalty controls the mean inlet velocity, while the radiation terms act on acoustic disturbances.

3. **Pressure convention.** The reference gauge pressure, $\bar{p}_g$, determined as discussed in Remark 1 is in the model's pressure units, not absolute pressure or acoustic pressure amplitude. The operating pressure, p_0, is obtained from the isentropic reference state:

$$p_0 = \rho_0 R\theta_0, \quad \bar{p}_g = p_{\text{abs}} - p_0, \quad p' = p_g - \bar{p}_g.$$

Here, ρ_0, R, and θ_0 are the reference density, specific gas constant, and reference temperature defined by *ICFD_MAT (see Card 4), p_{abs} is the absolute pressure, p_g is the unknown gauge pressure that is solved for, and p' is the pressure fluctuation relative to the reference gauge pressure. If PREFLCID is provided, a curve value of zero means the absolute pressure corresponds to the operating pressure.

4. **Outlet and wall options.** KIND = 1 imposes the outlet pressure strongly. KIND = 2 uses a weak pressure load and an acoustic pressure absorber. KIND = 3 is a radiating slip-wall formulation and permits the acoustic normal motion needed for radiation. It should not be interpreted as an acoustically rigid wall.

5. **Traction outlet and backflow.** KIND = 4 applies the same impedance traction formulation as *ICFD_BOUNDARY_ISENTROPIC_NRBC0, with NW available to control the mean velocity. KIND = 5 adds the following traction only where the fluid velocity relative to the mesh points into the domain:

$$\mathbf{t}_b = -\frac{1}{2}\rho|(\mathbf{u} - \mathbf{w}) \cdot \mathbf{n}|\mathbf{u} \quad \text{if } (\mathbf{u} - \mathbf{w}) \cdot \mathbf{n} < 0.$$

Here, $\mathbf{w}$ is the mesh velocity, and $\mathbf{n}$ is the outward unit normal. The additional traction is zero on outflow.

*ICFD_BOUNDARY_ISENTROPIC_NRBC0_{OPTION}

Available options include:

<BLANK>

TITLE

Purpose: Define a zeroth-order non-reflecting boundary condition (NRBC) with an impedance traction formulation on an outlet surface of an isentropic compressible fluid (see FLG = 2 on *ICFD_MAT).

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	PREFLCID						
Type	I	I						
Default	none	0						

VARIABLE	DESCRIPTION
PID	Part ID of the fluid surface where the impedance traction outlet is applied
PREFLCID	Load curve ID defining the reference static gauge pressure as a function of time. A blank or nonpositive ID fixes the reference pressure at the initial pressure on the boundary. A positive ID prescribes the reference pressure from the curve throughout the calculation. See Remark 2.

Remarks:

1. **Impedance traction.** The outlet applies an acoustic impedance traction of the form:

$$\sigma \cdot \mathbf{n} = -\left[\bar{p}_g + \rho c(\mathbf{u} - \bar{\mathbf{u}}) \cdot \mathbf{n}\right]\mathbf{n}.$$

Here σ is the fluid stress expressed using gauge pressure, $\mathbf{n}$ is the outward unit normal, ρ is the local density, c is the local sound speed, and $\bar{\mathbf{u}}$ is the mean velocity. The acoustic impedance is ρc. This local zeroth-order condition is intended to transmit outgoing acoustic waves, with exact matching in the ideal

plane-wave, normal-incidence limit. It does not guarantee zero reflection for arbitrary incidence or discretization.

2. **Reference pressure and velocity.** Without PREFLCID, the reference pressure remains fixed at its initial boundary value; it is not updated by time averaging. With a positive PREFLCID, the reference pressure follows the supplied curve. The curve's ordinate values are gauge pressures (p_g), relative to the operating pressure, p_0, obtained from *ICFD_MAT:

$$p_0 = \rho_0 R \theta_0, \quad p_g = p_{abs} - p_0.$$

A curve value of zero corresponds sets the absolute pressure equal to the operating pressure. The mean velocity is initialized from the initial velocity and updated by exponential smoothing with a fixed averaging window of 100 time steps:

$$\bar{\mathbf{u}}^{n+1} = (1-\alpha)\bar{\mathbf{u}}^{n} + \alpha \mathbf{u}^{n+1}, \quad \alpha = 2/101.$$

The global NW on *ICFD_CONTROL_ISENTROPIC does not change this averaging window.

3. **Relation to the general NRBC keyword.** This keyword is the compact form of *ICFD_BOUNDARY_ISENTROPIC_NRBC with KIND = 4, NW = 100, no velocity curves, and the same PREFLCID. Use the general keyword to change the velocity averaging window. Select KIND = 5 on that keyword when additional backflow stabilization is required; this keyword does not include that term.

*ICFD_BOUNDARY_NAVIERSLIP

Purpose: Specify the fluid boundary with the Navier slip boundary condition.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	LS						
Type	I	F						
Default	none	0.						

VARIABLE	DESCRIPTION
PID	PID of the fluid surface where a Navier boundary condition is applied
LS	Slip length. If a negative value is entered, it will refer to a load curve ID used to describe the slip length value as a function of time; see *DEFINE_CURVE, *DEFINE_CURVE_FUNCTION, or *DEFINE_FUNCTION. If a *DEFINE_FUNCTION is used, the following parameters are allowed: f(x, y, z, vx, vy, vz, temp, pres, time).

Remarks:

1. **About the Navier condition.** The Navier slip condition is a Robin-type boundary condition that relates the tangential wall shear, τ_w, to the tangential slip velocity:

$$\tau_w = \frac{\mu}{L_s} U_w$$

In tha above, μ is the viscosity, U_w is the tangential component of the relative velocity at the wall, and L_s is the slip length. For very large L_s values, free slip is recovered, while a very small L_s value trends towards nonslip behavior. If a turbulence law of the wall is used combined with a Navier slip condition, the law of the wall behavior will be modified to include the Navier slip length as:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}$$
$$\frac{U_w}{u_\tau} = u^+(y^+) + L_s^+$$
$$L_s^+ = \frac{u_\tau L_s}{\nu}$$

See *ICFD_CONTROL_TURBULENCE for a complete description of u^+ and y^+.

*ICFD_BOUNDARY_NONSLIP

Purpose: Specify the fluid boundary with a non-slip boundary condition.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
PID	PID of the fluid surface where a non-slip boundary condition is applied.

*ICFD_BOUNDARY_PERIODIC

Purpose: Impose various kinds of constraints between two fluid surfaces.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	PTYPE	PID2	PDLCID	AXE	PTID	ANGLE	
Type	I	I	I	I	I	I	F	
Default	none	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
PID	Part ID for a fluid surface
PTYPE	Boundary type: EQ.1: Periodic rotation boundary condition EQ.2: Periodic reflective boundary condition EQ.3: Sliding mesh boundary condition
PID2	Part ID for the second surface mesh. The boundary condition selected with PTYPE will be applied between PID and PID2. See Remark 1.
PDLCID	Optional load curve ID to describe the pressure drop value between PID and PID2 as a function of time. This curve can be specified with *DEFINE_CURVE,*DEFINE_CURVE_FUNCTION, or *DEFINE_FUNCTION. For *DEFINE_FUNCTION, the following parameters are allowed: f(x,y,z,vx,vy,temp,pres,time).
AXE	The meaning of AXE depends on PTYPE. It only applies for PTYPE = 1 and 3. For the periodic rotation boundary condition (PTYPE = 1): EQ.1: Rotation around *X*-axis EQ.2: Rotation around *Y*-axis EQ.3: Rotation around *Z*-axis For the sliding mesh boundary condition (PTYPE = 3): EQ.0: The contact distance between two faces of PID and PID2 is based on the characteristic local element size.

VARIABLE	DESCRIPTION
	EQ.1: The contact distance between two faces of PID and PID2 is based on the characteristic local element size scaled by a factor given by ANGLE.
	EQ.2: The contact distance between two faces of PID and PID2 is based on the length given by ANGLE.
PTID	Origin point ID for PTYPE = 1 and PTYPE = 2. See *ICFD_DEFINE_POINT.
ANGLE	Rotation angle for PTYPE = 1. Characterizes contact distance for PTYPE = 3 and AXE ≠ 0.

Remarks:

1. **Selection of PID and PID2.** When the two meshes are of different densities, we recommend selecting the finer mesh to be PID and the coarser mesh to be PID2.

*ICFD_BOUNDARY_PRESCRIBED_LEVELSET

Purpose: Prescribe the fluid height on a boundary.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	PTID	AXE					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
PID	PID of the fluid surface where a fluid height will be imposed.
PTID	Point ID specifying the origin of the fluid surface. See *ICFD_DEFINE_POINT.
AXE	Global axis specifying the direction of the fluid: EQ.1: X-axis EQ.2: Y-axis EQ.3: Z-axis

*ICFD_BOUNDARY_PRESCRIBED_MOVEMESH_{OPTION1}

Available options for *OPTION1* include:

DR

Purpose: Allows the fluid surface nodes to translate in specific directions using an ALE approach. This feature is helpful in piston-type applications or for avoiding large mesh deformation in certain cases.

Include as many cards as needed. This input ends at the next keyword ("*") card. If the DR keyword option is active, then only one occurrence of card 1 is allowed.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	DOFX	DOFY	DOFDZ				
Type	I	I	I	I				
Default	none	1	1	1				

Include this card for the DR keyword option. When the dynamic relaxation phase includes the ICFD solver, this second line allows the application of different parameters during that phase. See IDR of *ICFD_CONTROL_GENERAL.

Card 2	1	2	3	4	5	6	7	8
Variable		DRDOFX	DRDOFY	DRDOFZ				
Type		I	I	I				
Default		1	1	1				

VARIABLE	DESCRIPTION
PID	PID for a fluid surface
DOFX/DOFY/DOFZ	Degrees of freedom in the *X*, *Y*, and *Z* directions: EQ.0: Degree of freedom left free (surface nodes can translate in the chosen direction) EQ.1: Prescribed degree of freedom (surface nodes are restrained)

VARIABLE	DESCRIPTION
DRDOFX/DRDOFY/ DRDOFZ	Same as DOFX, DOFY, and DOFZ but applies to the mesh displacement during the dynamic relaxation phase.

*ICFD_BOUNDARY_PRESCRIBED_PRE

Purpose: Impose a fluid pressure on the boundary.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	LCID	SF	DEATH	BIRTH	ISO		
Type	I	I	F	F	F	I		
Default	none	none	1.	1.E+28	0.0	0		

VARIABLE	DESCRIPTION
PID	PID for a fluid surface
LCID	Load curve ID to describe the pressure value as a function of time. See *DEFINE_CURVE, *DEFINE_CURVE_FUNCTION, or *DEFINE_FUNCTION. If using a *DEFINE_FUNCTION, the following parameters are allowed: f(x, y, z, vx, vy, vz, temp, pres, time).
SF	Load curve scale factor. The default is 1.0.
DEATH	Time at which the imposed motion/constraint is removed: EQ.0.0: Default set to 1028
BIRTH	Time at which the imposed pressure is activated starting from the initial abscissa value of the curve
ISO	Adds an additional isovelocity condition to the system on top of the user-defined imposed pressure (see Remark 1): EQ.0: Off EQ.1: On

Remarks:

1. **Adding an isovelocity condition.** Flow instability is a well known numerical issue that can occur with imposed pressure conditions, particularly in complex FSI cases or when the global flow changes direction during the transient run (inlet transitions into outlet and vice versa). Using ISO = 1 adds an isovelocity

condition which makes the velocity uniform across that surface. The objective is an attempt to provide an elegant form of constraining the flow and preventing numerical issues in cases where this hypothesis can be considered physically valid. This feature imposes an additional numerical cost.

*ICFD_BOUNDARY_PRESCRIBED_SPTRANSP_CONC

Purpose: Specify the concentration of the transported species at the boundaries.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	LCID1	LCID2	LCID3	LCID4	LCID5	LCID6	LCID7	LCID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
PID	Part ID of the boundary with the concentrations boundary conditions
LCID*i*	Load curve ID for the curve giving the concentration of species *i* at the boundary as a function of time. Note that all the transported species must have a load curve defined. For instance, in the 4-species case, LCID1, LCID2, LCID3, and LCID4 must be specified.

*ICFD_BOUNDARY_PRESCRIBED_TEMP

Purpose: Impose a fluid temperature on the boundary.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	LCID	SF	DEATH	BIRTH			
Type	I	I	F	F	F			
Default	none	none	1.	1.E+28	0.0			

VARIABLE	DESCRIPTION
PID	PID for a fluid surface.
LCID	Load curve ID to describe the temperature value versus time; see *DEFINE_CURVE, *DEFINE_CURVE_FUNCTION or *DEFINE_-FUNCTION. . If a DEFINE_FUNCTION is used, the following parameters are allowed: $f(x,y,z,vx,vy,vz,temp,pres,time)$.
SF	Load curve scale factor. (default = 1.0)
DEATH	Time at which the imposed temperature is removed: EQ.0.0: default set to 10E28
BIRTH	Time at which the imposed temperature is activated starting from the initial abscissa value of the curve

*ICFD_BOUNDARY_PRESCRIBED_TURBULENCE

Purpose: Optional keyword for strongly imposing turbulence quantities when you select a RANS turbulence model. See *ICFD_CONTROL_TURBULENCE to turn on RANS turbulence models and for a description of the main equations and variables. This keyword is intended for modifying the default boundary conditions at the inlet, but can be used to experiment with different Dirichlet impositions at the wall.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	VTYPE	IMP	LCID	KS	CS		
Type	I	I	I	I	F	F		
Default	none	none	0	none	0.	0.		

VARIABLE	DESCRIPTION
PID	PID for a fluid surface
VTYPE	Variable type: EQ.1: Turbulence kinetic energy (see Remark 1) EQ.2: Turbulence dissipation rate (see Remark 2) EQ.3: Specific dissipation rate (see Remark 3) EQ.4: Modified turbulence viscosity (see Remark 4)
IMP	Imposition method: EQ.0: Direct imposition through value specified by LCID EQ.1: Using turbulence intensity specified by LCID if VTYPE = 1 (see Remark 1). Using turbulence length scale specified by LCID if VTYPE = 2, 3, or 4 (see Remarks 2, 3, and 4). EQ.2: Using turbulence viscosity ratio specified by LCID. Only available for VTYPE = 2, 3, and 4. See Remarks 2 and 3.
LCID	Load curve ID to describe the variable value as a function of time; see *DEFINE_CURVE, *DEFINE_CURVE_FUNCTION or *DEFINE_FUNCTION. If a *DEFINE_FUNCTION is used, the following parameters are allowed: f(x, y, z, vx, vy, vz, temp, pres, time, k,

VARIABLE	DESCRIPTION
	e, mut).
KS/CS	Roughness physical height and roughness constant. When defined, the global values of *ICFD_CONTROL_TURBULENCE are replaced for this surface part.

Remarks:

1. **Turbulence kinetic energy.** At the inlet, the relationship between the turbulence kinetic energy, k, and the turbulence intensity, I, is given by:

$$k = \frac{3}{2}\left(U^2 I^2\right) \ .$$

By default, the solver uses an inlet intensity of 0.05 (5%).

2. **Turbulence dissipation rate.** At the inlet, if you specify the turbulent dissipation rate using a length scale, l, the following relationship is used:

$$\epsilon = C_\mu{}^{3/4} \frac{k^{3/2}}{l} \ .$$

By default, the solver estimates a length scale based on the total height of the channel $l = 0.07 L_{\text{ref}}$.

If you specify the turbulent viscosity ratio, $r = \nu_t/\nu$, the following relationship is used:

$$\epsilon = C_\mu \frac{k^2}{\nu r} \ .$$

3. **Specific dissipation rate.** At the inlet, if you specify the specific dissipation rate using a length scale, l, the following relationship is used:

$$\omega = C_\mu{}^{-1/4} \frac{k^{1/2}}{l} \ .$$

By default, the solver estimates a length scale based on the total height of the channel.

If you specify the turbulence viscosity ratio, $r = \nu_t/\nu$, the following relationship is used:

$$\omega = \frac{k}{\nu r} \ .$$

4. **Modified turbulent viscosity.** At the inlet, the relationship between the modified turbulent viscosity, $\tilde{\nu}$, and the length scale, l, is given by:

$$\tilde{\nu} = 0.05\sqrt{\frac{3}{2}}(Ul) \ .$$

Or $\tilde{\nu} = r\nu$ if you specify the viscosity ratio.

*ICFD_BOUNDARY_PRESCRIBED_VEL

Purpose: Impose the fluid velocity on the boundary.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	DOF	VAD	LCID	SF	VID	DEATH	BIRTH
Type	I	I	I	I	F	I	F	F
Default	none	none	1	none	1.	0	10^{28}	0.0

VARIABLE	DESCRIPTION
PID	PID for a fluid surface.
DOF	Applicable degrees of freedom: EQ.1: x- degree of freedom EQ.2: y- degree of freedom EQ.3: z degree of freedom EQ.4: Normal direction degree of freedom
VAD	Velocity flag: EQ.1: Linear velocity EQ.2: Angular velocity EQ.3: Parabolic velocity profile EQ.4: Activates synthetic turbulent field on part. See *ICFD_CONTROL_TURB_SYNTHESIS.
LCID	Load curve ID used to describe motion value versus time, see *DEFINE_CURVE, *DEFINE_CURVE_FUNCTION, or *DEFINE_FUNCTION. If a DEFINE_FUNCTION is used, the following parameters are allowed: f(x, y, z, vx, vy, vz, temp, pres, time). For steady state the motion value is a function of the number of iterations instead of time.
SF	Load curve scale factor. (default = 1.0)

VARIABLE	DESCRIPTION
VID	Point ID for angular velocity application point, see *ICFD_DEFINE_POINT.
DEATH	Time at which the imposed motion/constraint is removed: EQ.0.0: default set to 10^{28}
BIRTH	Time at which the imposed motion/constraint is activated starting from the initial abscissa value of the curve

*ICFD_BOUNDARY_PRESCRIBED_VISCOELASTIC

Purpose: Specify the conformation tensor at the boundaries.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	LCIDXX	LCIDYY	LCIDZZ	LCIDXY	LCIDXZ	LCIDYZ	
Type	I	I	I	I	I	I	I	
Default	none	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
PID	Part ID of the boundary with the prescribed conformation tensor
LCID*ij*	Load curve IDs for the conformation tensor components

*ICFD_BOUNDARY_PRESCRIBED_WATERVAPOR

Purpose: Impose the water vapor mass fraction on a boundary.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	MFRAC	LCID	MFRACU				
Type	I	F	I	I				
Default	none	none	optional	0				

VARIABLE	DESCRIPTION
PID	Part ID of the fluid surface with the water vapor mass fraction condition
MFRAC	Water vapor mass fraction value. If MFRACU = 0, this value is the mass of water vapor in kg divided by the mass of dry air in kg. If MFRACU = 1, this value is the relative humidity and is in the range of [0,1]. Ignored if LCID is defined.
LCID	Optional load curve ID used to describe the water vapor mass fraction value as a function of time; see *DEFINE_CURVE, *DEFINE_CURVE_FUNCTION, or *DEFINE_FUNCTION. If a DEFINE_FUNCTION is used, the following parameters are allowed: f(x, y, z, vx, vy, vz, temp, pres, time).
MFRACU	Water vapor mass fraction type: EQ.0: The water vapor mass fraction value is the mass of the water vapor in kg divided by the mass of the dry air in kg. EQ.1: The water vapor mass fraction value is the relative humidity with a range of [0,1].

*ICFD_BOUNDARY_WEAKVEL

Purpose: Specify the fluid boundary with a non-slip boundary condition which is imposed in a weak form.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
PID	Part ID of the fluid surface where a weak non-slip boundary condition is applied

*ICFD_BOUNDARY_WINDKESSEL

Purpose: Impose the pressure function with circuit parameters where an analogy is made between pressure and scalar potential and between flow rate and current intensity. Such conditions are frequently encountered in hemodynamics.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	WTYPE	R1	C1	R2	L1/ QPLCID		
Type	I	I	F	F	F	F		
Default	none	none	0.	0.	0.	0.		

Optional Card. This card defines additional parameters for the Windkessel circuits depending on WTYPE.

Card 2	1	2	3	4	5	6	7	8
Variable	P2LCID	C2	R3	P0	P1			
Type	I	F	F	F	F			
Default	0	0.	0.	0.	0.			

VARIABLE	DESCRIPTION
PID	PID for a fluid surface
WTYPE	Circuit type. See remarks and figures.
R1/C1/L1/R 2/C2/R3	Parameters (resistances, inductances, capacities) for the different circuits. See remarks and figures.
P2LCID	ID of load curve describing the behavior of $P_2(t)$ as a function of time when applicable. See Remarks 2, 3, and 4 and Figures 7-2, 7-3, and 7-4.
QPLCID	ID of load curve describing the behavior of $Q_p(t)$ as a function of time when applicable. See Remark 4 and Figure 7-4.

VARIABLE	DESCRIPTION
P0/P1	Initial pressures at circuit junctions when applicable. See remarks and figures.

Remarks:

1. **WTYPE = 1 and 2.** See Figure 7-1 for a diagram of the circuit. The equation solved at the boundary is:

$$P(t) = P_0(t) + \text{SF} \times Q \times R_2$$
$$P_0'(t) = -\frac{1.0}{R_1 C_1} P_0(t) + \text{SF} \times \left[L_1 Q'' + \frac{Q}{C_1} + \frac{L_1}{R_1 C_1} Q' \right]$$

with Q, the flow rate calculated by the solve, and P, the pressure. Time derivatives are marked using Lagrange notation. By convention, the flow rate has a negative value if the fluid goes out of the fluid domain (outlet) and a positive value if it goes in (inlet). The scale factor, SF, applied on the circuit equation's flow-rate-dependent terms enables recovering the desired behavior of the Windkessel circuit. SF = -1.0 for WTYPE = 1, and SF = 1.0 for WTYPE = 2.

For WTYPE = 1 and 2, R_1, R_2 and L_1 are optional and can be left as 0.

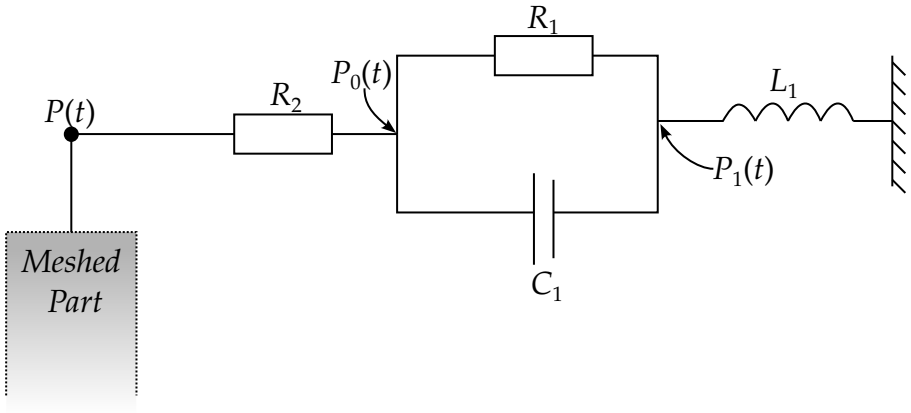


Figure 7-1. WTYPE = 1 and 2 circuit

2. **WTYPE = 3 and 4.** See Figure 7-2 for a diagram of the circuit. The equation solved at the boundary is:

$$P(t) = P_0(t) + \text{SF} \times Q \times R_1$$
$$P_0'(t) = -\frac{1.0}{R_2 C_1} P_0(t) + \frac{1.0}{R_2 C_1} P_1(t) + \text{SF} \times \frac{Q}{C_1}$$
$$P_1'(t) = \frac{1.0}{R_2 C_2} P_0(t) - \frac{R_3 + R_2}{R_3 R_2 C_2} P_1(t) + P_2'(t)$$

with Q and SF as described in Remark 1. SF = -1.0 for WTYPE = 3, and SF = 1.0 for WTYPE = 4. R_1 and R_3 are optional for this type of circuit. If R_3 is set to 0., that branch of the circuit is removed (open circuit). A negative value can be entered for R_3 which points to a *DEFINE_FUNCTION. The function accepts the following parameters: f(time, dt, flux , pre). In the function, flux is the current flow rate, Q, and pre is the current pressure, $P(t)$. With this method, R_3can be set to a specific value or left as 0.0 based on the sign of flux, thus modeling diode behavior.

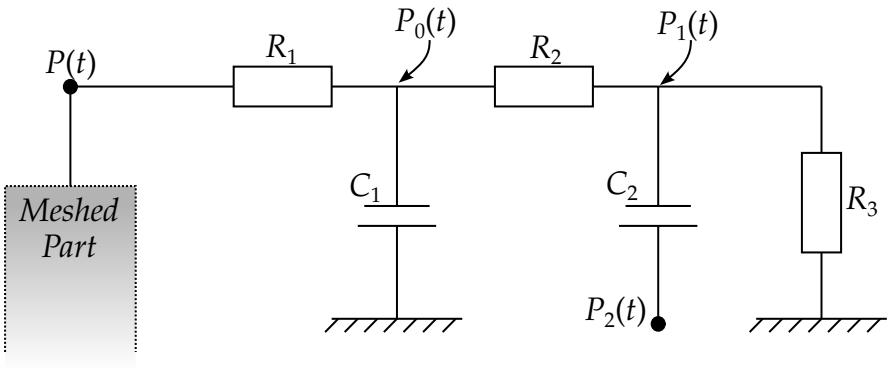


Figure 7-2. WTYPE = 3 and 4 circuit

3. **WTYPE = 7 and 8.** See Figure 7-3 for a diagram of the circuit. The equation solved at the boundary is:

$$P(t) = P_0(t) + \text{SF} \times Q \times R_1$$
$$P_0'(t) = -\frac{1.0}{R_2 C_1} P_0(t) + \text{SF} \times \frac{Q}{C_1} + \frac{1.0}{R_2 C_1} P_2(t)$$

with Q and SF as described in Remark 1. SF = -1.0 for WTYPE = 7, and SF = 1.0 for WTYPE = 8. For this type of circuit, R_1, R_2 , and $P_2(t)$ are optional.

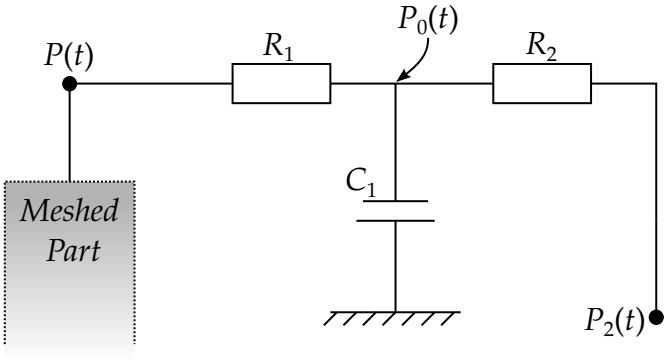


Figure 7-3. WTYPE = 7 and 8 circuit

4. **WTYPE = 9 and 10.** See Figure 7-4 for a diagram of the circuit. The equation solved at the boundary is:

$$P(t) = \frac{R_3}{R_2+R_3}P_0(t) + \mathrm{SF} \times Q \times \frac{R_3R_2}{R_2+R_3} + \frac{R_2}{R_2+R_3}P_2(t)$$

$$P_0'(t) = -\frac{R_1+R_2+R_3}{(R_2+R_3)R_1C_2}P_0(t) + \frac{1.0}{R_1C_2}P_1(t) + \frac{R_3}{(R_2+R_3)}\mathrm{SF} \times \frac{Q}{C_2} - \frac{P_2(t)}{C_2} \times \frac{1.0}{R_2+R_3}$$

$$P_1'(t) = \frac{1.0}{R_1C_1}P_0(t) - \frac{1.0}{R_1C_1}P_1(t) + \frac{Q_p(t)}{C_1}$$

with Q and SF as described in Remark 1. SF = -1.0 for WTYPE = 9, and SF = 1.0 for WTYPE = 10. For this type of circuit, R_2, $P_2(t)$, $Q_p(t)$, and R_3 are optional. If R_3 is set to 0.0, that branch of the circuit is removed (open circuit). A negative value can be entered for R_3 which points to a *DEFINE_FUNCTION. The function accepts the following parameters: f(time, dt, flux , pre). In the function, flux is the current flow rate, Q, and pre is the current pressure, $P(t)$. With this method, R_3can be set to a specific value or left as 0.0 based on the sign of flux, thus modeling diode behavior.

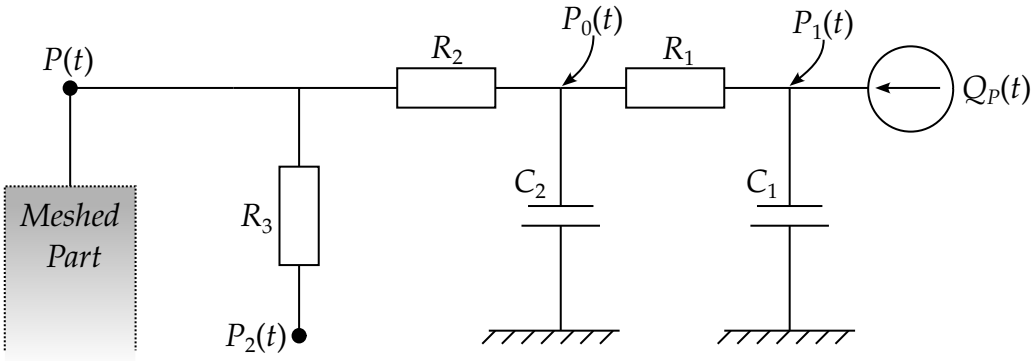


Figure 7-4. WTYPE = 9 and 10 circuit

5. **WTYPE = 6.** WTYPE = 6 enables implementing user-defined circuit laws by compiling the usermat versions of LS-DYNA. The routine that can be modified is called `icfd_usercircuit` and can be found in dyn21icfd.f.

*ICFD_CONTROL_ADAPT

Purpose: Activate the adaptive mesh refinement feature. The solver uses an a-posteriori error estimator to compute a new mesh size bounded by the user to satisfy a maximum perceptual global error.

Card 1	1	2	3	4	5	6	7	8
Variable	MINH	MAXH	ERR	MTH	NIT	VAR		KIS
Type	F	F	F	I	I	I		I
Default	none	none	1.	1	0	0		0

Optional Card.

Card 2	1	2	3	4	5	6	7	8
Variable	DT							
Type	F							
Default	0.0							

VARIABLE	DESCRIPTION
MINH	Minimum mesh size allowed to the mesh generator. The resulting mesh will not have an element smaller than MINH even if the minimum size does not satisfy the maximum error.
MAXH	Maximum mesh size
ERR	Maximum perceptual error allowed in the whole domain
MTH	Specify if the mesh size is computed based on function error or gradient error: EQ.1: Function error (default) EQ.2: Gradient error EQ.3: Relative error EQ.4: Gradient relative error

VARIABLE	DESCRIPTION
NIT	Number of iterations before a remeshing is forced: LT.0: |NIT| is a load curve ID giving the number of iterations before a remeshing as a function of time. EQ.0: Do not remesh. GT.0: Number of iterations before a forced remeshing.
VAR	Specify which variable is taken into account for the error calculation: LT.0: Adaptive meshing is based only on the level-set function. Elements of size MINH are targeted over a length |VAR| on each side of the interface. EQ.0: Velocity, pressure, and level-set function are taken into account. EQ.1: Remove the level-set function from the error calculation. EQ.2: Remove the pressure from the error calculation. EQ.3: Remove both the pressure and the level-set function from the error calculation. Only the fluid velocity will, therefore, remain. EQ.5: For immersed interface methods, adapt the mesh only at the interface of the immersed surfaces. EQ.10: For species transport problems, use species concentrations and the velocity field for the error calculation.
KIS	Keep initial mesh size: EQ.0: Turned off. The remeshing process will ignore the initial mesh size in the volume. EQ.1: Turned on. Whenever a remeshing occurs, the new local mesh size will not be allowed to be substantially coarser than the one from the previous mesh. The object is to diminish the excessive coarsening that can occur between two remeshings.
DT	Optional time step to control the remeshing frequency. A negative value points to a load curve ID giving the remeshing time step as a function of time.

*ICFD_CONTROL_ADAPT_SIZE

Purpose: Control the remeshing of elements. This keyword enables considering the element quality and distortion instead of only checking for inverted elements.

Card 1	1	2	3	4	5	6	7	8
Variable	ASIZE	NIT	KIS					
Type	I	I	I					
Default	0	none	0					

Optional Card. When including ICFD during the dynamic relaxation phase, this second line enables applying different parameters during that phase. See IDR of *ICFD_CONTROL_GENERAL.

Card 2	1	2	3	4	5	6	7	8
Variable	DRASIZE	DRNIT	DRKIS					
Type	I	I	I					
Default	0	optional	0					

VARIABLE	DESCRIPTION
ASIZE	Remesh criteria flag: EQ.0: Only remesh in cases where elements invert. EQ.1: Remesh if elements invert or if element quality deteriorates.
NIT	Number of iterations before a remeshing is forced. If a negative integer is entered, then a load curve function of time defines NIT.
KIS	Keep initial mesh size: EQ.0: Turned off. The remeshing process will ignore the initial mesh size in the volume. EQ.1: Turned on. Whenever a remeshing occurs, the new local mesh size will not be allowed to be substantially coarser

VARIABLE	DESCRIPTION
	than the one from the previous mesh. The object is to diminish the excessive coarsening that can occur between two remeshes.
DRASIZE	Same as ASIZE but for dynamic relaxation
DRNIT	Same as NIT but for dynamic relaxation
DRKIS	Same as KIS but for dynamic relaxation

*ICFD_CONTROL_ADVECTION

Purpose: Modify the default advection schemes for different parts of the solver like Navier Stokes, Level Set, thermal, etc.

Card 1	1	2	3	4	5	6	7	8
Variable				SLLS	SLNS	SLT	SLST	SLRT
Type				I	I	I	I	I
Default				0	0	0	0	0

VARIABLE	DESCRIPTION
SLLS	Advection scheme for the Level Set solver: EQ.0: Default advection scheme EQ.1: Changes the default advection scheme in the Level Set solver to use a linear semi-Lagrangian approach (Min and Gibou [2006]). EQ.2: Changes the default advection scheme in the Level Set solver to use a higher-order semi-Lagrangian approach.
SLNS	Advection scheme for the Navier-Stokes solver: EQ.0: Default advection scheme EQ.1: Changes the default advection scheme in the Navier-Stokes solver to use a semi-Lagrangian approach (Min and Gibou [2006]).
SLT	Advection scheme for the Thermal solver: EQ.0: Default advection scheme EQ.1: Changes the default advection scheme in the Thermal solver to use a semi-Lagrangian approach.
SLST	Advection scheme for the Species Transport solver: EQ.0: Default advection scheme EQ.1: Changes the default advection scheme in the Species Transport solver to use a semi-Lagrangian approach.

VARIABLE	DESCRIPTION
SLRT	Advection scheme for the Residence Time solver: EQ.0: Default advection scheme EQ.1: Changes the default advection scheme in the Residence Time solver to use a semi-Lagrangian approach.

References:

Min, Chohong, and Frédéric Gibou. "A second order accurate projection method for the incompressible Navier–Stokes equations on non-graded adaptive grids." Journal of Computational Physics 219.2 (2006): 912-929.

*ICFD_CONTROL_BACKFLOW

Purpose: Modify default values for backflow stabilization.

Card 1	1	2	3	4	5	6	7	8
Variable	BFOR	SF						
Type	I	F						
Default	0	1.0						

VARIABLE	DESCRIPTION
BFOR	Set the backflow stabilization formulation: EQ.0: Default stabilization dependent on spatial velocity gradients. EQ.1: The stabilization adds a temporal velocity gradient which could be necessary for added stabilization.
SF	Scale factor to increase the stabilization if needed. A very small value larger than zero minimizes the effect.

*ICFD_CONTROL_CONJ

Purpose: Pick between the different coupling methods for conjugate heat transfer applications.

Card 1	1	2	3	4	5	6	7	8
Variable	CTYPE							TSF
Type	I							F
Default	0							none

This card is optional and only used with CTYPE = 2.

Card 2	1	2	3	4	5	6	7	8
Variable	IMHTC	IMDMIN	IMHSF					
Type	F	F	F					
Default	0.	0.	1.					

VARIABLE	DESCRIPTION
CTYPE	Indicates the thermal coupling type: EQ.0: Robust and accurate monolithic coupling where the temperature fields are solved simultaneously between the fluid and the structure. EQ.1: Weak thermal coupling (obsolete) EQ.2: Immersed thermal coupling. See Remark 2.
TSF	Thermal Speedup Factor. This factor multiplies all thermal parameters present in the heat equation with units of time in the denominator (e.g., thermal conductivity, convection heat transfer coefficients). It is used to artificially time scale the thermal problem. A negative value refers to a time dependent load curve.
IMHTC	Heat transfer coefficient used when the minimum distance IMDMIN between the fluid node and immersed solid is reached. See

VARIABLE	DESCRIPTION
	Remark 2. A negative value refers to a *DEFINE_FUNCTION ID. If a *DEFINE_FUNCTION is used, the following parameters are allowed: f(x, y, z, vx, vy, vz, temp, pres, time).
IMDMIN	Minimum distance for IMHTC to be applied. This field is ignored if IMHTC is zero.
IMHSF	Scale factor applied to IMHTC to determine the effect heat transfer coefficient when the distance between fluid node and immersed body is larger than IMDMIN. Note that if a thermal law of the wall is used rather than a purely conductive based heat transfer, this field is ignored because the thermal law of the wall has its own scale factor, SF_{cond}, on the conductive portion of the wall heat transfer; see SFCOND on *ICFD_CONTROL_RANS. This field is ignored if IMHTC is zero.

Remarks:

1. **Keywords used in thermal coupling.** *ICFD_BOUNDARY_CONJ_HEAT is ignored if CTYPE = 1. *ICFD_BOUNDARY_FSI is required in all thermal coupling cases.

2. **Immersed FSI with thermal coupling.** *ICFD_CONTROL_IMMERSED with IMMTH = 4 is required to be active to add thermal heat exchange between the immersed body and the fluid volume. The thermal extension adds a Robin-type coupling on both the fluid and the structural thermal equations, driving them toward temperature equilibrium at the immersed interface. The strength of that coupling is set by an effective heat transfer coefficient, h_{eff}. It determines the heat flux q_T:

$$q_T = h_{eff}(T_s - T_f)$$

In the above, T_s is the temperature of the structure and T_f is the temperature of the fluid.

If a RANS model is selected in combination with the thermal law of the wall (see TWLAW in *ICFD_CONTROL_TURBULENCE), h_{eff} is automatically computed by the thermal law of the wall. Otherwise, a purely conductive h_{eff} is estimated based on the fluid thermal conductivity and the distance between the fluid nodes that see the immersed structure. Note that you can also specify a minimum distance, IMDMIN, under which IMHTC (if nonzero) is used to determine the effective heat transfer coefficient instead of either of these methods. In that case,

$$h_{eff} = \text{SF} \times \text{IMHTC}$$

where SF is IMHSF unless a thermal law of the wall is used. If a thermal law of the wall is used, it is SFCOND on *ICFD_CONTROL_RANS.

*ICFD_CONTROL_DEM_COUPLING

Purpose: Activate coupling between the ICFD and DEM solvers.

Card 1	1	2	3	4	5	6	7	8
Variable	CTYPE	BT	DT	SF	MAXVEL	DTYPE	SFF	FORM
Type	I	F	F	F	F	I	F	I
Default	0	0.	10^{28}	1.	none	0	1.0	0

Card 2	1	2	3	4	5	6	7	8
Variable	NITBS							
Type	I							
Default	↓							

VARIABLE	DESCRIPTION
CTYPE	Indicates the coupling direction of the solvers: EQ.0: Two-way coupling between the fluid and the solid particles. EQ.1: One-way coupling. The DEM particles transfer their location to the fluid solver. EQ.2: One-way coupling. The fluid solver transfers forces to the DEM particles.
BT	Birth time for the DEM coupling
DT	Death time for the DEM coupling
SF	Scale factor applied to the force transmitted by the fluid to the structure
MAXVEL	Maximal fluid velocity that can be used for the calculation of the fluid force passed on to the DEM particle. This is to avoid having spurious velocities in the fluid causing very high and unrealistic

VARIABLE	DESCRIPTION
	forces on the DEM particles which may lead to a crash.
DTYPE	Drag calculation type: EQ.0: Constant C_d value 0.5 scaled by SF EQ.1: Formula for C_d calculation from Cheng [2009] based on the local Reynolds number value scaled by SF. See Remark 1.
SFF	Scale factor applied to the force transmitted by the structure to the fluid
FORM	Type of formulation used in the coupling: EQ.0: The force at the particle is based on a velocity drag value. EQ.2: The force is computed using the fluid pressure gradient.
NITBS	Number of time steps between particle searches. The default is 50 in explicit analysis and 1 in implicit. Increasing this number can reduce the computational time at the expense of possible less accurate particle tracking.

Remarks:

1. **Coefficient of drag by Cheng [2009].** The calculation for C_d is:

$$C_d = \frac{24}{\mathrm{Re}} \times (1 + 0.27 \times \mathrm{Re})^{0.43} + 0.47 \times \left(1 - \exp\left(-0.04 \times \mathrm{Re}^{0.38}\right)\right)$$

See Cheng [2009] for details.

References:

[1] Cheng, N.-S., “Comparison of formulas for drag coefficient and settling velocity of spherical particles,” Powder Technology, 189, 395-398 (2009).

*ICFD_CONTROL_EMBEDSHELL

Purpose: This keyword allows the user to control specific options related to the use of the keyword MESH_EMBEDSHELL.

Card 1	1	2	3	4	5	6	7	8
Variable	GTYPE	DIST	TPS					
Type	I	F	I					
Default	0	0.1	0					

VARIABLE	DESCRIPTION
GTYPE	Gap type. Defines the criteria for selecting a distance to build the gap between the embedded nodes and the newly generated : EQ.0: Automatic and based on the surface mesh size multiplied by a scale factor given by DIST. Default method. EQ.1: Specific gap size given by the user and defined by DIST.
DIST	Distance value if GTYPE = 1 or scale factor value if GTYPE = 0.
TPS	Triple Point Seal. Allows to control the fluid escape through triple points EQ.0: Off. EQ.1: On. The triple points of embedded shells in contact to walls or among each other are sealed and no flow goes through them.

*ICFD_CONTROL_FSI

Purpose: This keyword modifies default values for the fluid-structure interaction coupling algorithm.

Card 1	1	2	3	4	5	6	7	8
Variable	OWC	BT	DT	IDC	LDICSF	XPROJ		
Type	I	F	F	F	I	I		
Default	0	0	10^{28}	0.25	0	0		

Card 2	1	2	3	4	5	6	7	8
Variable	NSUB			IDETEC	VFORC			
Type	I			I	I			
Default	none			0	0			

VARIABLE	DESCRIPTION
OWC	Indicates the coupling direction to the solver: EQ.0: Two-way coupling. Loads and displacements are transferred across the FSI interface, and the full non-linear problem is solved. It gives weak FSI coupling when coupled to explicit mechanical solver and strong FSI coupling when coupled to implicit mechanical solver. EQ.1: One-way coupling. The solid mechanics solver transfers displacements to the fluid solver. EQ.2: One-way coupling. The fluid solver transfers stresses to the solid mechanics solver. EQ.3: Two-way coupling. It causes weak coupling (no sub-stepping) with the implicit mechanical solver.
BT	Birth time for the FSI coupling. Before BT the fluid solver will not pass any loads to the structure, but it will receive displacements

VARIABLE	DESCRIPTION
	from the solid mechanics solver.
DT	Death time for the FSI coupling. After DT the fluid solver will not transfer any loads to the solid mechanics solver, but the fluid will continue to deform with the solid.
IDC	Interaction detection coefficient. See Remark 1.
LCIDSF	Optional load curve ID to apply a scaling factor on the forces transferred to the solid: GT.0: Load curve ID for scale factor as a function of iterations LT.0: |LCIDSF| is a load curve ID for scale factor as a function of time.
XPROJ	Projection of the nodes of the CFD domain that are at the FSI interface onto the structural mesh (see Remark 2): EQ.0: No projection EQ.1: Projection
NSUB	Optional limit on the number of FSI fluid subiterations. This avoids the sometimes unneeded excessive number of FSI subiterations when the fluid and very light structures (like parachutes) develop a resonance-like mode inside the FSI subiterations (coupling iterations).
IDETEC	Fluid-structure interface detection frequency (see Remark 3): LT.0: |IDETEC| refers to a load curve ID for the curve that gives the detection frequency as a function of time. EQ.0: The interface detection is performed only once at the beginning of the simulation. GT.0: The interface is detected every IDETEC time steps.
VFORC	Add viscous tangential forces to the FSI coupling: EQ.0: Only normal pressure forces are used. EQ.1: Viscous forces are added to normal forces. EQ.2: Viscous forces are recomputed at every sub-step during the nonlinear iterations and added to normal forces.

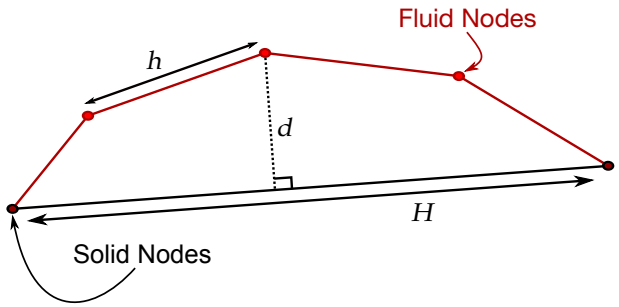


Figure 0-1. Geometry of FSI contact.

Remarks:

1. **Detecting fluid-solid interaction.** One of the criteria to automatically detect the fluid and solid surfaces that will interact in FSI problems is the distance d between a fluid (solid) node and a solid (fluid) element, respectively:

$$d \leq \text{IDC} \times \min(h, H) \ ,$$

where h is the size of the fluid mesh, H is the size of the solid mechanics mesh, and IDC is a detection coefficient criteria with IDC $= 0.25$ by default. In the majority of cases, this default value is sufficient to ensure FSI interaction. However, it can happen in special cases that the fluid and solid geometries have curvatures that differ too much (such as pipe flows in conjugate heat transfer applications). In such cases, a bigger IDC value may be needed. This flag should be handled with care.

2. **Rotation and projection of nodes.** XPROJ = 1 is recommended for cases with rotation.

3. **Fluid-solid interaction detection frequency.** By default, the detection of the fluid and structure surfaces that will interact in FSI or conjugate heat problems is performed only once at the beginning of the analysis. Finding the interface only once is adequate for most cases because the fluid mesh usually follows and is bound to the structure's displacements. One notable exception is for applications using *ICFD_BOUNDARY_FSI_FIXED. Typically, this boundary condition is used for conjugate heat transfer problems where a solid surface 'slides' over a fluid domain and temperature is exchanged. Thus, the fluid-structure detection must be periodically recomputed with the use of the IDETEC flag.

*ICFD_CONTROL_GAP

Purpose: Activate and modify the type of gap closure treatment that deals with flow blockage when surfaces come into contact. In ICFD, surface meshes are not allowed to collide or penetrate each other. Rather, activation of the gap closure feature triggers a flow blockage between two surfaces in close proximity based on a specified contact distance. There are two types of flow treatment available. The default is a strong exclusion of the fluid elements within the gap. The second option is a porous media description for the fluid nodes inside the contact gap. See *ICFD_DEFINE_GAPCLOSURE for defining the gap closure criteria and the surfaces involved in the gap closure.

Card 1	1	2	3	4	5	6	7	8
Variable	HGAP	PGAP	PERM					
Type	F	I	F					
Default	optional	0	1.e-5					

This card is optional. We recommend using *ICFD_DEFINE_GAPCLOSURE instead to specify the surfaces. See Remark 1.

Card 2	1	2	3	4	5	6	7	8
Variable	PID1	PID2	PID3	PID4	PID5	PID6	PID7	PID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
HGAP	Threshold distance. See Remark 1 of *ICFD_DEFINE_GAPCLOSURE for more information. If Card 2 is included, then HGAP must be set on this keyword. However, we recommend using *ICFD_DEFINE_GAPCLOSURE instead for defining the threshold distance and surfaces involved. See Remark 1.
PGAP	Flag for flow treatment of fluid elements in the gap: EQ.0: Element exclusion contact treatment EQ.1: Porous media description
PERM	Permeability coefficient. Its value determines the ratio between the

VARIABLE	DESCRIPTION
	fluid and porous representation in the contact region. Low values induce a Darcy description.
PID*n*	Part IDs of the surfaces involved in the gap closure treatment.

Remarks:

1. **Change in the input.** Prior to R18, this keyword was required to activate the gap closure feature. The gap closure feature could only be active on one set of surfaces specified with Card 2 with a gap closure threshold of HGAP. As of R18, this keyword does *not* need to be in the input deck to activate the gap closure feature. *ICFD_DEFINE_GAPCLOSURE can activate the feature. We recommend only including this keyword to change PGAP and/or PERM as those are global parameters for all gap closures and cannot be set with *ICFD_DEFINE_GAPCLOSURE. With R18, HGAP and PID*n* are no longer required, and we recommend setting these fields for each flow blockage using *ICFD_DEFINE_GAPCLOSURE instead. HGAP and PID*n* remain valid input to maintain backward compatibility but should be considered obsolete. *ICFD_DEFINE_GAPCLOSURE has two main advantages. It can be included in the input deck multiple times, so multiple flow blockages can be defined with different threshold distances between different sets of surfaces. It also includes an additional criterion based on a threshold angle (see Remark 2 of *ICFD_DEFINE_GAPCLOSURE).

2. **Restricting the surfaces for gap search.** To apply gap detection to smaller regions on the surfaces instead of the entire surfaces, use *ICFD_DEFINE_GAPRESTRIC.

*ICFD_CONTROL_GENERAL

Purpose: Specify the type of ICFD analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	ATYPE	MTYPE	DVCL	RDVCL	SOLCL		IDR	
Type	I	I	I	I	I		I	
Default	0	0	0	0	0		0	

VARIABLE	DESCRIPTION
ATYPE	Analysis type: EQ.-2: Turn off the ICFD solver but leave level set, thermal, etc. Velocities can be set using the define function. It is useful for debugging models or for simple thermal / level set cases that do not need a full ICFD solution. EQ.-1: Turn off the ICFD solver after the initial keyword reading EQ.0: Transient analysis (default) EQ.1: Steady-state analysis
MTYPE	Solving method type: EQ.0: Fractional step method EQ.1: Monolithic solve EQ.2: Potential flow solve
DVCL	Divergence cleaning flag: EQ.0: Initialize the solution with divergence cleaning (default) EQ.1: No divergence cleaning EQ.2: Initial divergence cleaning using potential flow EQ.4: Initial divergence cleaning using steady-state solver
RDVCL	Remeshing divergence cleaning: EQ.0: No divergence cleaning after remesh (default) EQ.1: Divergence cleaning after each remeshing step

VARIABLE	DESCRIPTION
SOLCL	Solver control: EQ.0: The solver automatically detects if the analysis is 2D or 3D based on element connectivity (default). EQ.1: Turns on the 2D axisymmetric solver. EQ.2: 2D axisymmetric solver for FSI cases that use section shell type 15 for solid parts.
IDR	Flag to include ICFD during the structural dynamic relaxation phase: EQ.0: Off EQ.1: Include the ICFD solver during the dynamic relaxation phase. The solver only computes the mesh displacement. It does not find pressure/velocity, but the mesh adapts and follows the displacement of the FSI parts.

*ICFD_CONTROL_IMMERSED

Purpose: Change the default behavior of the immersed interface options.

Card 1	1	2	3	4	5	6	7	8
Variable	IMMTH	ISSTA		NITBS				
Type	I	I		I				
Default	0	0		↓				

Optional card. This card is optional. Include as many instantiations of this card as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							

VARIABLE	DESCRIPTION
IMMTH	Method used to represent the immersed interface. By default, the solver assumes that all interfaces of interest are body-fitted. EQ.0: Immersed methods are not used in the model. EQ.1: Method based on a discontinuous FEM approximation at the interface EQ.2: Method based on the Resistive Immersed Implicit Surface (RIIS) method. This option needs DEM particles on the surface of immersed structural parts. See *DEFINE_DE_MESH_SURFACE. Note that prior to R17, using this method required *ICFD_CONTROL_DEM_COUPLING to specify the coupling between the DEM particles and fluid. With R17, *ICFD_CONTROL_IMMERSED_FSI controls the coupling for all immersed interfaces, and *ICFD_CONTROL_DEM_COUPLING is no longer needed. EQ.4: Method based on the Resistive Immersed Implicit Surface (RIIS) method. The fluid solver detects structural parts

VARIABLE	DESCRIPTION
	and computes the interfaces automatically.
ISSTA	Flag for specifying whether the immersed interface remains unchanged (static) during the simulation: EQ.0: The immersed interface is dynamic. Frequent interface computations are needed. Note that *ICFD_BOUNDARY_IMMERSED_STATIC and *ICFD_BOUNDARY_IMMERSED_DYNAMIC take precedence over this option. EQ.1: The immersed interface is static. The interface is computed at the first time step or after re-meshing steps only. This option saves computation time. Note that this option takes precedence over *ICFD_BOUNDARY_IMMERSED_STATIC and *ICFD_BOUNDARY_IMMERSED_DYNAMIC.
NITBS	Indicate how often the interface is computed. The default value depends on the mechanical solver (explicit or implicit) for the structure. For the explicit mechanical solver, the default is every 50 time steps, while for implict, the default is every time step.
PID	Part ID of the structural element that is used to compute the immersed interface. If no PID is defined in this card, the solver uses all the structural parts. This field is only available for IMMTH = 1 and 4.

*ICFD_CONTROL_IMMERSED_FSI

Purpose: Modify default values for the fluid-structure interaction coupling algorithm when immersed interfaces are used for FSI.

Card 1	1	2	3	4	5	6	7	8
Variable	OWC	BT	DT					
Type	I	I	I					
Default	0	0	10^{28}					

VARIABLE	DESCRIPTION
OWC	Indicates the coupling direction to the solver: EQ.0: Two-way coupling. Loads and displacements are transferred across the FSI interface, and the full nonlinear problem is solved. It gives weak FSI coupling when coupled to the explicit structural solver and strong FSI coupling when coupled to the implicit structural solver. EQ.1: One-way coupling. The structural solver transfers displacements to the fluid solver. EQ.2: One-way coupling. The fluid solver transfers stresses to the structural solver. EQ.3: Two-way coupling. It causes weak coupling (no sub-stepping) with the structural solver (for both implicit and explicit).
BT	Birth time for the FSI coupling. Before BT, the fluid solver does not pass any loads to the structure, but it receives displacements from the structural solver.
DT	Death time for the FSI coupling. After DT, the fluid solver does not transfer any loads to the structural solver, but the fluid continues to deform with the structure.

*ICFD_CONTROL_IMPOSED_MOVE

Purpose: Impose a velocity on specific ICFD parts or on the whole volume mesh. Global translation, global rotation, and local rotation components can be defined and combined. This keyword can be used to save calculation time in certain applications, such as sloshing, where the modeling of the whole fluid box and the solving of the consequent FSI problem are not necessarily needed.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	LCVX	LCVY	LCVZ	VADT	IDR		
Type	I	I	I	I	I	I		
Default	none	none	none	none	0	0		

Optional Card. Rotational velocity components using Euler or Tait-Bryan angles (see Remark 1).

Card 2	1	2	3	4	5	6	7	8
Variable	ALPHAL	BETAL	GAMMAL	ALPHAG	BETAG	GAMMAG	VADR	IANG
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

Optional Card. Local reference frame definition if ALPHAL, BETAL or GAMMAL used.

Card 3	1	2	3	4	5	6	7	8
Variable	PTID	X1	Y1	Z1	X2	Y2	Z2	
Type	I	F	F	F	F	F	F	
Default	0	1.	0.	0.	0.	1.	0.	

Optional Card. When defined, Cards 2 and 3 are ignored. With this card, rotation is imposed around a point using the velocity of a second point.

Card 4	1	2	3	4	5	6	7	8
Variable	PTID0	AXE	NID					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
PID	Part ID. This can be any part ID referenced in *ICFD_PART or *ICFD_PART_VOL. If PID = 0, then the whole volume mesh will be used.
LCVX, LCVY, LCVZ	Load curve IDs for the velocity/displacements in the three global directions (X, Y, Z). To use a *DEFINE_FUNCTION, see Remark 4.
VADT	Velocity/displacements flag for translation components: EQ.0: Prescribe velocity EQ.1: Prescribe displacements
IDR	Flag determining whether to only impose mesh displacement during dynamic relaxation phase (see IDR of *ICFD_CONTROL_GENERAL): EQ.0: Off. EQ.1: On. Apply imposed mesh movement only during the dynamic relaxation phase.
ALPHAL, BETAL, GAMMAL	Load curves IDs for the three angle rotational velocities/displacements in the local reference frame (see Remarks 1 and 2). To use a *DEFINE_FUNCTION, see Remark 4.
ALPHAG, BETAG, GAMMAG	Load curve IDs for the three angle rotational velocities/displacements in the global reference frame (see Remarks 1 and 2). To use a *DEFINE_FUNCTION, see Remark 4.

VARIABLE	DESCRIPTION
VADR	Velocity/displacements flag for rotation components: EQ.0: Prescribe velocity EQ.1: Prescribe displacements
IANG	Rotation matrix type: EQ.0: Euler angles using $\mathbf{Z}(\alpha)\mathbf{X}(\beta)\mathbf{Z}(\gamma)$ intrinsic rotations. EQ.1: Tait-Bryan angles using $\mathbf{Z}(\alpha)\mathbf{X}(\beta)\mathbf{Y}(\gamma)$ intrinsic rotations.
PTID	Point ID for the origin of the local reference frame. If not defined, the barycenter of the volume mesh will be used
X1, Y1, Z1	Three components of the local reference X1 axis. If not defined, the global X axis will be used. See Remark 2.
X2, Y2, Z2	Three components of the local reference X2 axis. If not defined, the global Y axis will be used. See Remark 2.
PTIDO	Point ID (See *ICFD_DEFINE_POINT) for the center of rotation.
AXE	Rotation axis: EQ.1: X-axis EQ.2: Y-axis EQ.3: Z-axis
NID	ICFD surface node ID for the rotational velocity. If the node is static, no rotation will occur. See Remark 3.

Remarks:

1. **Rotations.** Any target orientation can be reached starting from a known reference orientation using a specific sequence of intrinsic rotations whose magnitudes are the Euler or Tait Bryan angles (α, β, γ). Equivalently, any rotation matrix $\mathbf{R}$ can be decomposed as a product of three elemental rotation matrices. For instance:

$$\mathbf{R} = \mathbf{X}(\alpha)\mathbf{Y}(\beta)\mathbf{Z}(\gamma)$$

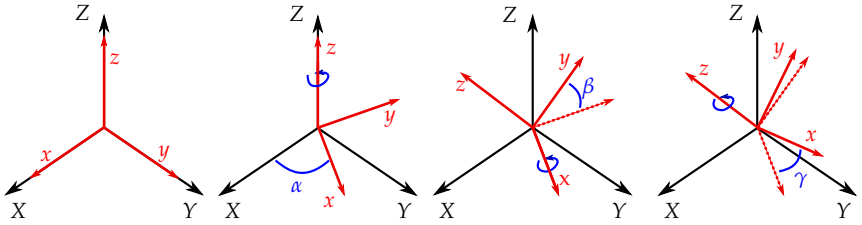


Figure 7-5. A rotation represented by Euler angles (α, β, γ) using $\mathbf{Z}(\alpha)\mathbf{X}(\beta)\mathbf{Z}(\gamma)$ intrinsic rotations.

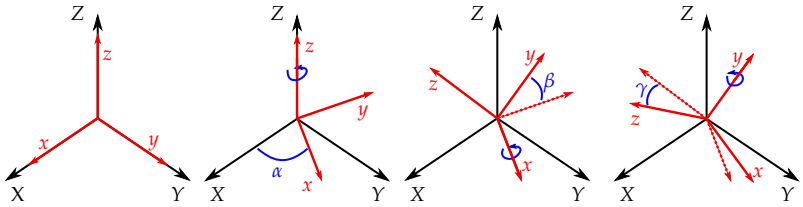


Figure 7-6. A rotation represented by Tait-Bryan angles (α, β, γ) using $\mathbf{Z}(\alpha)\mathbf{X}(\beta)\mathbf{Y}(\gamma)$ intrinsic rotations

However, a different definition of the elemental rotation matrices and their multiplication order can be adopted. The ICFD solver uses the following approach and rotation matrix for the Euler angles:

$$\mathbf{Z}(\alpha)\mathbf{X}(\beta)\mathbf{Z}(\gamma) = \begin{bmatrix} c_\alpha c_\gamma - c_\beta s_\alpha s_\gamma & -c_\beta c_\gamma s_\alpha - c_\alpha s_\gamma & s_\alpha s_\beta \\ c_\gamma s_\alpha + c_\alpha c_\beta s_\gamma & c_\alpha c_\beta c_\gamma - s_\alpha s_\gamma & -c_\alpha s_\beta \\ s_\beta s_\gamma & c_\gamma s_\beta & c_\beta \end{bmatrix}$$

Here, $s_\alpha = \sin(\alpha)$, and $c_\beta = \cos(\beta)$.

For the Tait-Bryan approach, the following rotation matrix is used:

$$\mathbf{Z}(\alpha)\mathbf{X}(\beta)\mathbf{Y}(\gamma) = \begin{bmatrix} c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma & -c_\beta s_\alpha & c_\alpha s_\gamma + c_\gamma s_\alpha s_\beta \\ c_\gamma s_\alpha + c_\alpha s_\beta s_\gamma & c_\alpha c_\beta & s_\alpha s_\gamma - c_\alpha c_\gamma s_\beta \\ -c_\beta s_\gamma & s_\beta & c_\beta c_\gamma \end{bmatrix}$$

2. **Local coordinate systems.** It is possible to have the ICFD parts or ICFD_PART_VOLs rotate around the global reference frame but also to define and use a local reference frame by defining its point of origin and two of its vectors $\mathbf{v}_1 = (\text{X1}, \text{Y1}, \text{Z1})$ and $\mathbf{v}_2 = (\text{X2}, \text{Y2}, \text{Z2})$ (note that $\mathbf{v}_1$ and $\mathbf{v}_2$ should be orthogonal). The third vector is, then, in the direction of $\mathbf{v}_1 \times \mathbf{v}_2$. See Figure 7-5.

3. **Purpose of NID.** We developed this feature for rotating problems involving FSI and sliding mesh. For example, the airflow can push the blades of a wind turbine and the rotation of the sliding mesh can be prescribed as function of the blade rotation speed.

4. ***DEFINE_FUNCTION.** For each of the load curves, a *DEFINE_FUNCTION can be used. If a *DEFINE_FUNCTION is used, the following parameters are allowed: f(x, y, z, vx, vy, vz, temp, visc, pres, time, dt).

*ICFD_CONTROL_ISENTROPIC

Purpose: Define the averaging window and number of fractional-step Picard iterations for the isentropic compressible ICFD solver.

Card 1	1	2	3	4	5	6	7	8
Variable	NW	NPICARD						
Type	I	I						
Default	100	1						

VARIABLE	DESCRIPTION
NW	Number of time steps defining the global exponential averaging window for the mean velocity and pressure. Values less than 1 are reset to 100. See Remark 2.
NPICARD	Number of fractional-step Picard nonlinear iterations per time step. Each iteration consists of a momentum predictor, a pressure solve, and a velocity correction. Values less than 1 are reset to 1. See Remark 3.

Remarks:

1. **Activating the isentropic compressible solver.** *ICFD_MAT with FLG = 2 activates the isentropic compresible solver. If this control keyword is omitted, NW = 100 and NPICARD = 1.

2. **Time averaging.** The mean fields are initialized from the initial solution and updated after the velocity correction by exponential smoothing:

$$\bar{q}^{n+1} = (1 - \alpha)\bar{q}^{n} + \alpha q^{n+1}, \quad \alpha = 2/(\mathrm{NW} + 1).$$

Here q denotes pressure or a velocity component. NW is a number of time steps, not a physical duration or a finite moving-average buffer. NW = 1 gives the current value; larger values give a slower response. These mean fields are also used to separate acoustic pressure from the mean pressure. At an isentropic non-reflecting boundary condition (NRBC), the boundary keyword supplies its own averaging window and reference-field rules; the global NW does not override them.

3. **Picard iterations.** NPICARD controls the nonlinear/coupling iterations within a time step.

*ICFD_CONTROL_LEVELSET

Purpose: Modify default values for the level set solver.

Card 1	1	2	3	4	5	6	7	8
Variable	LSRST	LSINL		LSMTH			SGC	AST
Type	I	I		F			I	I
Default	20	0		0.0			0	0

Card 2	1	2	3	4	5	6	7	8
Variable	SRL					LSHREF	LSW	
Type	I					F	I	
Default	0					0.0	0	

VARIABLE	DESCRIPTION
LSRST	Frequency the level set distance function is re-initialized. The default value is every 20 time steps.
LSINL	Set the level set to be positive at a velocity inlet: EQ.0: Let the level set algorithm compute the value (default). EQ.1: Force a positive level set value at the inlet.
LSMTH	Scale factor for level set smoothness. GT.0.0: Add smoothness to the free surface. Small values like 0.1 are reasonable. The optimal value could be problem-dependent.
SGC	Smoother approximation of gradient and curvature: EQ.0: No smoothing added. EQ.1: Linear gradient and curvature. EQ.-1: Least-squares-based gradient and curvature.

VARIABLE	DESCRIPTION
	EQ.2: Least-squares gradient smoothing.
AST	Advanced formulation for surface tension: EQ.0: Smoothed Dirac EQ.1: Smoothed Heaviside EQ.2: Laplace-Beltrami
SRL	Smoother reinitialization method for the level set: EQ.0: Standard, geometric-based EQ.1: Linear closest point algorithm EQ.2: Higher-order closest point algorithm
LSHREF	Reference size for the narrow-band approach. Level-set transport is done in a standard way up to distances to interfaces that are smaller than 4 times LSHREF (Peng et al. [1999]). This field is relevant only if semi-Lagrangian advection of the level set is enabled. See SLLS on *ICFD_CONTROL_ADVECTION. EQ.0.0: Disable narrow-band approach.
LSW	Weighted Essentially Non-Oscillatory scheme for advection and reinitialization of the level set. This field is relevant only if higher-order advection (see SLLS = 2 on *ICFD_CONTROL_ADVECTION) and higher-order reinitialization (SRL = 2) of the level set are enabled. EQ.0: Off EQ.1: On

References:

Peng, Danping, Barry Merriman, Stanley Osher, Hongkai Zhao, and Myungjoo Kang. "A PDE-based fast local level set method." Journal of Computational Physics 155, no. 2 (1999): 410-438.

*ICFD_CONTROL_LOAD

Purpose: Reset the body load in the ICFD solver to zero, while leaving the body load unchanged for the solid mechanics solver. It is useful in problems where the gravitational acceleration may be neglected for the fluid problem, but not for the solid mechanics problem. Additionally, the constant gravity value in the fluid can be defined in any global direction with this keyword.

Card 1	1	2	3	4	5	6	7	8
Variable	ABL	DIR	VAL					
Type	I	I	F					
Default	1	0	0.					

VARIABLE	DESCRIPTION
ABL	Flag to reset the body load in the fluid: EQ.0: The body load provided in *LOAD_BODY is reset to zero only for the fluid analysis. In addition, by specifying DIR and VAL, the body load on the fluid can be specified. EQ.1: Do not reset the body load provided in *LOAD_BODY.
DIR	Global direction in which gravity is applied to the fluid. See Remark 1. DIR is available if ABL = 0. EQ.0: No body load specified for the fluid EQ.1: X-axis EQ.2: Y-axis EQ.3: Z-axis
VAL	Constant value of gravity (for example, $g = 9.81\ \mathrm{m/s^2}$). VAL is available if ABL = 0.

Remarks:

1. **Hydrostatic pressure.** The usual approach to define gravity with the ICFD solver is to simply define *LOAD_BODY, which by default applies gravity to both the fluid and the structure. For the fluid, and contrary to some practices

with explicit methods, we recommend directly setting gravity to its final value rather than using a slow ramp in time, especially for strong FSI-coupling problems; otherwise, the pressure might converge to an incorrect value. However, there may be valid reasons for using a ramp on the structural side, making the use of a common keyword (*LOAD_BODY) to define gravity on both the fluid and solid problematic. By defining DIR, VAL, and ABL, the hydrostatic pressure can be defined for the fluid independently of gravity on the structure side. In any case, for strong FSI problems, we strongly recommend only starting the coupling (see birth time of *ICFD_CONTROL_FSI) once gravity has been established on both the fluid and structure.

2. **Applying other loads and accelerations.** If other loads in addition to gravity are present in the analysis, one approach is to use the method described in Remark 1 to include gravity and *ICFD_DEFINE_SOURCE to include the accelerations applied in the other directions. *LOAD_BODY can still be used to apply accelerations on the structure side. In contrast to gravity, we recommend that other additional accelerations start at 0 when the FSI analysis begins.

*ICFD_CONTROL_MESH

Purpose: Modify default values for automatic volume mesh generation.

Card 1	1	2	3	4	5	6	7	8
Variable	MGSF		MSTRAT	2DSTRUC	NRMSH			
Type	F		I	I	I			
Default	1.41		0	0	0			

Optional card. This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	AVER	SFBL						
Type	I	F						
Default	14	1.0						

VARIABLE	DESCRIPTION
MGSF	Mesh Growth Scale Factor. It specifies the maximum mesh size that the volume mesher is allowed to use when generating the volume mesh based on the mesh surface element sizes defined in *MESH_SURFACE_ELEMENT. See Remark 1.
MSTRAT	Mesh generation strategy (see Remark 2): EQ.0: Mesh generation based on Delaunay criteria EQ.1: Mesh generation based on octree
2DSTRUC	Flag to decide between an unstructured mesh generation strategy in 2D or a structured mesh strategy: EQ.0: Structured mesh EQ.1: Unstructured mesh

VARIABLE	DESCRIPTION
NRMSH	Flag to turn off any remeshing (see Remark 3): EQ.0: Remeshing possible EQ.1: Remeshing not allowed
AVER	Automatic Volume Mesher version (see Remark 4): EQ.14: Version 14 EQ.16: Version 16
SFBL	Scale factor that controls the speed of boundary layer inflation. If the boundary layer is much larger in size than the surface mesh, we recommend a value less than one.

Remarks:

1. **MGSF.** For MGSF, values between 1 and 2 are allowed. Values closer to 1 will result in a finer volume mesh (1 means the volume mesh is not allowed to be coarser than the element size from the closest surface meshes) and values closer to 2 will result in a coarser volume mesh (2 means the volume can use elements as much as twice as coarse as those from the closest surface mesh). MGSF has a fixed value of 1 in 2D.

2. **Mesh Generation Strategy.** The default mesh generation strategy (based on Delaunay criteria) yields a linear interpolation of the mesh size between two surfaces facing each other whereas the octree-based generation strategy allows for the sizes of the elements to remain close to the element surface mesh size over a longer distance. This octree strategy can be useful for creating a smoother transition in configurations where two surface meshes facing each other have very distinct sizes.

3. **NRMSH.** If you know in advance that no remeshing will occur during the analysis, then setting NRMSH to 1 may be useful as it will free up space used to back up the mesh and consequently lower memory consumption.

4. **Version.** Version 14 is the default version used for the ICFD solver automatic volume mesher. Version 16 is now supported and available as option. In some cases, it can yield an approximatively 20% mesh generation speed gain.

*ICFD_CONTROL_MESH_MOV

Purpose: Choose the type of algorithm for mesh movement.

Card 1	1	2	3	4	5	6	7	8
Variable	MMSH	LIM_ITER	RELTOL					
Type	I	I	F					
Default	2	100	10^{-3}					

VARIABLE	DESCRIPTION
MMSH	Mesh motion selector: EQ.-1: Completely shuts off any mesh movement EQ.1: Mesh moves based on the distance to moving walls. EQ.2: Mesh moves by solving a linear elasticity problem using the element sizes as stiffness (default). EQ.3: Mesh uses a Laplacian smoothing with stiffness on edges and from node to opposite faces. Very robust, but costly. EQ.4: Full Lagrangian. The mesh moves with the velocity of the flow. EQ.11: Mesh moves using an implicit ball-vertex spring method. EQ.20: Mesh moves by solving a linear elasticity problem using a constant size. This can be useful to avoid large distortions in rotating problems that involve large discrepancies in mesh sizes (typically in cases involving boundary layer mesh).
LIM_ITER	Maximum number of linear solver iterations for the ball-vertex linear system
RELTOL	Relative tolerance to use as a stopping criterion for the ball-vertex method iterative linear solver (conjugate gradient solver with diagonal scaling preconditioner)

*ICFD_CONTROL_MONOLITHIC

Purpose: This keyword allows to choose between the Fractional Step Solver and the Monolithic Solver.

Card 1	1	2	3	4	5	6	7	8
Variable	SID							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
SID	Solver ID : EQ.0: Fractional Step Solver. Default. EQ.1: Monolithic Solver.

*ICFD_CONTROL_OUTPUT

Purpose: This keyword modifies default values for screen and file outputs related to this fluid solver only.

Card 1	1	2	3	4	5	6	7	8
Variable	MSGL	OUTL	DTOUT	LSPPOUT		ITOUT		
Type	I	I	F	I		I		
Default	0	0	0	0		0		

Optional Card.

Card 2	1	2	3	4	5	6	7	8
Variable	PITOUT							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MSGL	Message level. EQ.0: only time step information is output. EQ.1: first level solver information. EQ.2: full output information with details about linear algebra and convergence steps. EQ.4: full output information is also copied to the messag file.

VARIABLE	DESCRIPTION
OUTL	Output the fluid results in other file formats apart from d3plot. EQ.0: only d3plot output EQ.2: output a file with mesh statistics and the fluid results in OpenDX format. A directory named dx will be created in the work directory where the output files will be written. EQ.6: output a file with mesh statistics and the fluid results in VTK format readable by Paraview. A directory named vtk will be created in the work directory where the output files will be written. EQ.7: output a file with mesh statistic and the fluid results in VTU format readable by Paraview. A directory named vtk will be created in the work directory where the output files will be written. EQ.10: output a file with mesh statistic and the fluid results in Fieldview ASCII format. . A directory named fv will be created in the work directory where the output files will be written. Only available in 3D. EQ.11: output a file with mesh statistic and the fluid results in Fieldview binary format. . A directory named fv will be created in the work directory where the output files will be written. Only available in 3D.
DTOUT	Time interval to print the output when OUTL is different than 0.
LSPPOUT	EQ.0: no LSPP output is produced. EQ.1: outputs a file with the automatically created fluid volume mesh in a format compatible for LSPP at each remesh. Also outputs the fluid volume mesh in a format compatible with a subsequent ICFD analysis. EQ.3: Outputs the fluid volume mesh in a format compatible with a subsequent ICFD analysis at each DTOUT.
ITOUT	Iteration interval to print the output, including the d3plot files when the steady state solver is selected (See ICFD_CONTROL_GENERAL).

VARIABLE	DESCRIPTION
PITOUT	Pressure iteration limit output. If the number of pressure iterations in the fractional step solve goes above PITOUT, an extra d3plot will be dumped. This is mainly a debugging feature which can help the user identify problematic areas in the model which often precede a divergence.

.

*ICFD_CONTROL_OUTPUT_SUBDOM

Purpose: Defines a specific zone that should be output in the format specified by the ICFD_CONTROL_OUTPUT card rather than the whole domain.

Shape Control. First card specifies the shape of the output sub domain.

Card 1	1	2	3	4	5	6	7	8
Variable	SNAME							
Type	A							
Default	none							

Box Case. Card 2 for Sname = box

Cards 2	1	2	3	4	5	6	7	8
Variable	PMINX	PMINY	PMINZ	PMAXX	PMAXY	PMAXZ		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

Sphere Case. Card 2 for Sname = sphere

Cards 3	1	2	3	4	5	6	7	8
Variable	RADIUS	CENTERX	CENTERY	CENTERZ				
Type	F	F	F	F				
Default	none	none	none	none				

Cylinder Case. Card 2 for Sname = cylinder

Cards 4	1	2	3	4	5	6	7	8
Variable	Radius	PMINX	PMINY	PMAXZ	PMAXX	PMAXY	PMAXZ	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
SNAME	Shape name. Possibilities include 'box', 'cylinder' and 'sphere'
PMINX, Y, Z]	X, Y, Z for the point of minimum coordinates
PMAX[X, Y, Z]	X, Y, Z for the point of maximum coordinates
CEN-TER[X, Y, Z]	Coordinates of the sphere center in cases where Sname is Sphere
RADIUS	Radius of the sphere if SNAME is *sphere* or of the cross section disk if SNAME is *cylinder*.

*ICFD_CONTROL_OUTPUT_VAR

Purpose: This keyword allows the user to turn off the output of certain CFD variables to reduce the size of the d3plot files.

Card 1	1	2	3	4	5	6	7	8
Variable	VEL	AVGVEL	VORT					
Type	I	I	I					
Default	0	0	0					

Card 2	1	2	3	4	5	6	7	8
Variable	PRE	PREAVG	LSET	QC	CFL			
Type	I	I	I	I	I			
Default	0	0	0	0	0			

Card 3	1	2	3	4	5	6	7	8
Variable	TEMP	TEMPAVG						
Type	I	I						
Default	0	0						

Card 4	1	2	3	4	5	6	7	8
Variable	KP	EP	MUT	INT	CMU			
Type	I	I	I	I	I			
Default	0	0	0	0	0			

VARIABLE	DESCRIPTION
VEL/AVGVEL/ VORT	Velocity, average velocity, vorticity : EQ.0: Is output. EQ.1: Is not output.
PRE/PREAVG/ LSET/QC/CFL	Pressure, average pressure, levelset, Q criterion, CFL number : EQ.0: Is output. EQ.1: Is not output.
TEMP/ TEMPAVG	Temperature, average temperature : EQ.0: Is output. EQ.1: Is not output.
KP/EP/MUT /INT/CMU	RANS output variables, kinetic energy, diffusion, turbulent viscosity, turbulent intensity, Cmu variable : EQ.0: Is output. EQ.1: Is not output.

.

*ICFD_CONTROL_PARTITION

Purpose: This keyword changes the default option for the partition in MPP, thus it is only valid in MPP.

Card 1	1	2	3	4	5	6	7	8
Variable	PTECH							
Type	I							
Default	1							

VARIABLE	DESCRIPTION
PTECH	Indicates the type of partition: EQ.1: the library Metis is used. EQ.2: partition along the axis with higher aspect ratio EQ.3: partition along *X*-axis EQ.4: partition along *Y*-axis EQ.5: partition along *Z*-axis

*ICFD_CONTROL_POROUS

Purpose: This keyword modifies the porous media solve.

Card 1	1	2	3	4	5	6	7	8
Variable	PMSTYPE	VELMETH						
Type	I	I						
Default	0	0						

VARIABLE	DESCRIPTION
PMSTYPE	Indicates the porous media solve type. EQ.0: Anisotropic Generalized Navier-Stokes model for porous media (See *ICFD_MODEL_POROUS) using Fractional step method. EQ.1: Anisotropic Darcy-Forcheimer model using a Monolithic approach for the solve. This method is better suited for very low Reynolds flows through porous media (frequently encountered in Resin Transfer Molding (RTM) applications). See Remark 1.
VELMETH	Method for determining advection velocity: EQ.0: Uses FEM approximation for advection velocity EQ.1: Uses PFEM2 for advection velocity

Remarks:

1. **Anisotropic Darcy-Forcheimer Model.** When using the Anisotropic Darcy-Forcheimer model, the convective term in the Navier Stokes formulation is neglected.

*ICFD_CONTROL_RANS

Purpose: Change the default values related to RANS turbulence models and trigger additional options. See *ICFD_CONTROL_TURBULENCE to turn on RANS turbulence models and for a description of the main equations and variables.

Card 1	1	2	3	4	5	6	7	8
Variable	KATO	PKLIM	CLIM	LRC				
Type	I	I	F	I				
Default	0	0	20.0	0				

This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	KROB	EROB						
Type	I	I						
Default	0	0						

This card is optional

Card 3	1	2	3	4	5	6	7	8
Variable	ISBUO	ISC3	C3VAL	SFCOND				
Type	I	I	F	F				
Default	0	0	0.	1.				

VARIABLE	DESCRIPTION
KATO	Kato-Launder turbulent production term modification. See Remark 1. EQ.0: Default. Same as 2.

VARIABLE	DESCRIPTION
	EQ.1: Modify the term. EQ.2: Do not modify the term.
PKLIM	Turbulent production term limiter flag. See Remark 2. EQ.0: Default. Same as 1. EQ.1: Turn on. EQ.2: Turn off.
CLIM	Production term limiter scaling coefficient
LRC	Low Reynolds correction flag that is applicable to $k-\omega$ turbulence models. See Remark 3. EQ.0: Default. Same as 2. EQ.1: Apply the correction. EQ.2: Do not apply the correction.
KROB	Option to choose between applying Dirichlet boundary conditions at the wall or Robin boundary conditions for the k or $\tilde{\nu}$ solve. See Remark 4. EQ.0: Default. Same as 1. EQ.1: Dirichlet boundary condition EQ.2: Robin boundary condition
EROB	Option to choose between applying Dirichlet boundary conditions at the wall or Robin boundary conditions for the ε/ω solve. See Remark 4. EQ.0: Default. Same as 1. EQ.1: Dirichlet boundary condition EQ.2: Robin boundary condition
ISBUO	Flag to add free convection terms to the $k-\varepsilon$ and $k-\omega$ models. See Remark 5. EQ.0: Default. Same as 2. EQ.1: Include free convection terms. EQ.2: Do not include free convection terms.
ISC3	Flag to set a specific $C_{3\varepsilon}$ with C3VAL or have the solver

VARIABLE	DESCRIPTION
	automatically calculate this quantity (see Remark 5): EQ.0: Set $C_{3\varepsilon}$ to 0. EQ.1: Automatically estimate $C_{3\varepsilon}$ EQ.2: Set the $C_{3\varepsilon}$ estimate with C3VAL
C3VAL	$C_{3\varepsilon}$ if ISC3 = 2 (see Remark 5)
SFCOND	Scale factor, SF_{cond}, used by the thermal law of the wall (see TWLAW on *ICFD_CONTROL_TURBULENCE) when determining the conductive thermal resistance and the convective thermal resistance. See Remark 9 of *ICFD_CONTROL_TURBULENCE. Setting this scale factor to a small value causes the conductive contribution to the thermal law of the wall to be effectively disabled and for the heat exchange between the solid/fluid to be dominated by the convective contribution. This scale factor also has priority over IMHSF on *ICFD_CONTROL_CONJ when the thermal law of the wall is active.

Remarks:

1. **Kato-Launder modification.** When KATO = 1, the term $G_k = \nu_T S^2$ is replaced by:

$$G_k = \nu_T \Omega^2$$

whenever applicable.

2. **Production term limiter.** When the production term limiter, PKLIM, is active, we have:

$$G_k = \min|G_k, C_{lim}\varepsilon|$$

for $k-\varepsilon$ models. And:

$$G_k = \min|G_k, 0.09C_{lim}\omega k|$$

for $k-\omega$ models.

3. **Low Reynolds correction.** The low Reynolds correction flag, LRC, applies only to $k-\omega$ models. When active, the terms $\alpha^* = \alpha^*_\infty$, $\alpha = \alpha_{\infty,3}$, and $\beta^*_i = \beta^*_\infty$ become:

$$\alpha^* = \alpha^*_\infty \left(\frac{\alpha^*_0 + R_t/R_k}{1. + R_t/R_k}\right)$$
$$\alpha = \frac{\alpha_{\infty,3}}{\alpha^*}\left(\frac{\alpha_0 + R_t/R_\omega}{1. + R_t/R_\omega}\right)$$

$$\beta_i^* = \beta_\infty^* \left(\frac{4./15. + \left(R_t/R_\beta\right)^4}{1. + \left(R_t/R_\beta\right)^4} \right)$$

where $R_\beta = 8.0$, $R_k = 6.0$, $R_\omega = 2.95$, and $R_t = \nu k/\omega$.

4. **Dirichlet/Robin boundary conditions.** At the wall, when Dirichlet boundary conditions are selected for k, the following conditions are imposed:

$$k_{\text{lam}} = 0$$
$$k_{\text{turb}} = \frac{U_\tau^2}{\sqrt{0.09}}$$
$$k_{\text{wall}} = \begin{cases} k_{\text{lam}} & y^+ \le 5.0 \\ \left(1.0 - F(y^+)\right)k_{\text{lam}} + F(y^+)k_{\text{turb}} & 5.0 < y^+ < 30.0 \\ k_{\text{turb}} & y^+ \ge 30.0 \end{cases}$$

Similarly, for ε and ω when EROB = 0 or 1:

$$\varepsilon_{\text{lam}} = \frac{2.0\nu k}{(y + K_s)^2}$$
$$\varepsilon_{\text{turb}} = \frac{U_\tau^3}{\kappa(y + K_s)}$$
$$\varepsilon_{\text{wall}} = \begin{cases} \varepsilon_{\text{lam}} & y^+ \le 5.0 \\ \left(1.0 - F(y^+)\right)\varepsilon_{\text{lam}} + F(y^+)\varepsilon_{\text{turb}} & 5.0 < y^+ < 30.0 \\ \varepsilon_{\text{turb}} & y^+ \ge 30.0 \end{cases}$$
$$\omega_{\text{lam}} = \frac{6.0\nu}{0.075(y + K_s)^2}$$
$$\omega_{\text{turb}} = \frac{U_\tau}{\kappa(y + K_s)}$$
$$\omega_{\text{wall}} = \begin{cases} \omega_{\text{lam}} & y^+ \le 5.0 \\ \left(1.0 - F(y^+)\right)\omega_{\text{lam}} + F(y^+)\omega_{\text{turb}} & 5.0 < y^+ < 30.0 \\ \omega_{\text{turb}} & y^+ \ge 30.0 \end{cases}$$

When Robin boundary conditions are requested, if y^+ is lower than 1.0, the solver reverts to the Dirichlet conditions as described above. Otherwise, for k, it applies the following:

$$\nu_k \frac{\partial k}{\partial n} = \beta_k (k + k_{\text{wall}})$$

where

$$\beta_k = C_k \frac{\nu_k}{y + K_s}$$
$$\nu_k = \nu + \frac{\nu_T}{\sigma_k}$$
$$C_k = 2.0$$

and similarly for ε and ω:

$$\nu_\varepsilon \frac{\partial \varepsilon}{\partial n} = \beta_\varepsilon(\varepsilon + \varepsilon_{\text{wall}})$$
$$\nu_\omega \frac{\partial \omega}{\partial n} = \beta_\omega(\omega + \omega_{\text{wall}})$$

where

$$\beta_\varepsilon = C_\varepsilon \frac{\nu_\varepsilon}{y + K_s}$$
$$\nu_\varepsilon = \nu + \frac{\nu_T}{\sigma_\varepsilon}$$
$$C_\varepsilon = 1.0$$
$$\beta_\omega = C_\omega \frac{\nu_\omega}{y + K_s}$$
$$\nu_\omega = \nu + \frac{\nu_T}{\sigma_\omega}$$
$$C_\omega = 1.0$$

For the Spalart-Allmaras model, $\tilde{\nu}_{\text{wall}} = 0.0$ at the wall if Dirichlet (KROB = 1) and:

$$\nu_{\tilde{\nu}} \frac{\partial \tilde{\nu}}{\partial n} = \beta_{\tilde{\nu}}(\tilde{\nu} + \tilde{\nu}_{\text{wall}})$$

with

$$\beta_{\tilde{\nu}} = \frac{\nu_{\tilde{\nu}}}{y + K_s}$$
$$\nu_{\tilde{\nu}} = \frac{\nu}{\sigma_{\tilde{\nu}}} + \frac{\tilde{\nu}}{\sigma_{\tilde{\nu}}}$$

if Robin (KROB = 2).

5. **Effects of buoyancy on turbulence.** If enabled (ISBUO = 1), the following G_b term is added to the turbulent production in the k equation:

$$G_b = -\beta \frac{\nu_T}{\text{Pr}_T} \mathbf{g} \cdot \nabla T$$

where $\mathbf{g}$ is the gravity vector, ∇T is the temperature gradient, and β is the thermal expansion coefficient. For the ε/ω equations:

$$G_{b\varepsilon} = C_{1\varepsilon} \frac{\varepsilon}{k} C_{3\varepsilon} G_b$$
$$G_{b\omega} = \frac{\omega}{k}\left[(1.0 + \text{sf}_{G_\omega})C_{3\varepsilon} G_b - G_b\right]$$

The value of $C_{3\varepsilon}$ can be set with C3VAL or automatically calculated by the solver as:

$$C_{3\varepsilon} = \tanh\left(\frac{u_t}{u_n}\right)$$
$$u_t = \sqrt{|\mathbf{u}|^2 - (\mathbf{u} \cdot \mathbf{g})^2}$$

$$u_n = \sqrt{(\mathbf{u} \cdot \mathbf{g})^2}$$

with $\mathbf{u}$ as the velocity vector.

*ICFD_CONTROL_STEADY

Purpose: Specify convergence options for the steady state solver.

Card 1	1	2	3	4	5	6	7	8
Variable	ITS	TOL1	TOL2	TOL3	REL1	REL2	UREL	ORDER
Type	I	F	F	F	F	F	F	I
Default	1e6	1.e-3	1.e-3	1.e-3	0.3	0.7	1.	0

VARIABLE	DESCRIPTION
ITS	Maximum number of iterations to reach convergence
TOL1/2/3	Tolerance limits for the momentum, pressure, and temperature equations, respectively
REL1/2	Relaxation parameters for the pressure and velocity, respectfully. Decreasing these values may add stability, but more iterations may be needed to reach convergence.
UREL	Under-relaxation parameter. Lowering this value may improve the final accuracy of the solution, but more iterations may be needed to achieve convergence.
ORDER	Analysis order: EQ.0: Second order. More accurate but more time consuming. EQ.1: First order. More stable and faster but may be less accurate.

*ICFD_CONTROL_SURFMESH

Purpose: Enable automatic surface remeshing. The objective of the remeshing is to improve the mesh quality on the boundaries. It should not be used on a regular basis.

Card 1	1	2	3	4	5	6	7	8
Variable	RSRF	SADAPT						
Type	I	I						
Default	0	0						

VARIABLE	DESCRIPTION
RSRF	Indicates whether or not to perform a surface remeshing: EQ.0: Do not perform surface remeshing. EQ.1: Perform Laplacian smoothing surface remeshing. EQ.2: Perform curvature-preserving surface remeshing.
SADAPT	Indicates whether or not to trigger adaptive surface remeshing: EQ.0: Do not apply adaptive surface remeshing. EQ.1: Apply automatic surface remeshing when quality deteriorates. EQ.2: Apply automatic surface remeshing when quality deteriorates. Keep the initial element size in the case of adaptive mesh refinement using *ICFD_CONTROL_ADAPT throughout the computation.

.

*ICFD_CONTROL_TAVERAGE

Purpose: This keyword controls the restarting time for computing the time average values. By default, there is no restarting and the average quantities are given starting from $t = 0$. This keyword can be useful in turbulent problems that admit a steady state.

Card 1	1	2	3	4	5	6	7	8
Variable	DT							
Type	F							
Default	none							

VARIABLE	DESCRIPTION
DT	Over each DT time interval, the average quantities are reset.

*ICFD_CONTROL_TIME

Purpose: Change the default values related to time parameters for the fluid problem.

Card 1	1	2	3	4	5	6	7	8
Variable	TTM	DT	CFL	LCIDSF	DTMIN	DTMAX	DTINIT	TDEATH
Type	F	F	F	I	F	F	F	F
Default	10^{28}	0.0	1.0	0	10^{-9}	10^{28}	↓	10^{28}

Optional card

Card 2	1	2	3	4	5	6	7	8
Variable	DTT	LCIDTH	CHTCTR					
Type	F	I	I					
Default	Rem 2	0	0					

Optional card

Card 3	1	2	3	4	5	6	7	8
Variable	DTBL	DTST	DTVISC					
Type	I	I	I					
Default	0	0	0					

Optional card

Card 4	1	2	3	4	5	6	7	8
Variable	IDR	DTDR	CFLDR	LCIDSFDR	DTMINDR	DTMAXDR	DTINITDR	
Type	I	F	F	I	F	F	F	
Default	0	0.0	1.0	0	10^{-9}	10^{28}	0.0	

VARIABLE	DESCRIPTION
TTM	Total time of simulation for the fluid problem
DT	Time step for the fluid problem. If nonzero, the time step will be constant and equal to this value. If set to 0.0, then the time step is automatically computed based on the fluid advection CFL condition. See Remark 1.
CFL	CFL number for DT = 0.0. In general, CFL specifies a scale factor that is applied to the time step. When DT = 0.0, the time step is set to the maximum value satisfying the CFL condition, in which case this scale factor is equal to the *CFL number*.
LCIDSF	Load curve ID specifying the CFL number when DT = 0.0 as a function of time, and more generally LCIDSF specifies the time step scale factor as a function of time.
DTMIN	Minimum time step. When an automatic time step is used and DTMIN is defined, the time step cannot drop below DTMIN. A negative value will refer to a time dependent load curve.
DTMAX	Maximum time step. When an automatic time step is used and DTMAX is defined, the time step cannot increase beyond DTMAX. A negative value will refer to a time dependent load curve.
DTINIT	Initial time step. If not defined, the solver will automatically determine an initial time step based on the flow velocity or dimensions of the problem in cases where there is no inflow.
TDEATH	Death time for the Navier Stokes solve. After TDEATH, velocity and pressure will no longer be updated; however, temperature and mesh movements remain active. GT.0.0: Death time

VARIABLE	DESCRIPTION
	LT.0.0: |TDEATH| is a load curve ID for dynamically turning the Navier Stokes on and off during a transient solve. Positive values returned by the load curve indicate on, while negative values indicate off.
DTT	Thermal time step. A negative value causes an automatic thermal time step calculation based on the heat equation's CFL number. See Remark 2.
LCIDTH	Optional load curve ID for a curve that scales the thermal time step DTT
CHTCTR	Conjugate heat transfer (CHT) time step control. It is only applicable when solving a CHT problem. EQ.0: The minimum time step between the thermal solver and ICFD solver is picked for the CHT analysis (default). EQ.1: The thermal solver's time step is always picked for CHT analysis. EQ.2: The ICFD solver's time step is always picked for CHT analysis. EQ.3: The maximum time step between the thermal solver and the ICFD solver is picked for CHT analysis.
DTBL	Flag to include boundary layer elements in the automatic time step calculation. EQ.0: Exclude boundary layer elements (default). EQ.1: Include boundary layer elements.
DTST	Flag to include surface tension effects (when present) in the automatic time step calculation: EQ.0: Do not take it into account (default). EQ.1: Add surface tension effects. See Remark 3.
DTVISC	Flag to include viscous effects in the automatic time step calculation: EQ.0: Do not take it into account (default). EQ.1: Add viscous term effects. See Remark 4.

VARIABLE	DESCRIPTION
IDR	Temporarily overwrite time step settings during dynamic relaxation (DR) phase (see field IDR of *ICFD_CONTROL_GENERAL): EQ.0: Off EQ.1: Overwrite timesteps settings during DR phase
DTDR/[..]/ DTINITDR	Time step parameters equivalent to the fields 2 through 7 on Card 1 used during DR phase.

Remarks:

1. **Courant Friedrichs Lewy (CFL) condition.** When an automatic time step is selected, the default behavior is to use the following Courant condition:

$$\Delta t \leq \Delta t_{\text{adv}} \leq \frac{l}{V}$$

Here, V is the fluid velocity, and l is the characteristic mesh size. When using an automatic time step, it is recommended to define a minimum and maximum acceptable value. It is also recommended to give an initial value, especially in cases where no velocity boundary condition exists.

2. **Thermal time step.** By default, the heat equation is solved using the same time step as that for the velocity/pressure system. To assign a specific time step for the heat equation solve, set DTT to a positive value. This feature can be useful in cases where the time scales are very different between the two domains. The heat equation time step DTT can be larger than the Navier-Stokes time step but not smaller, so define the parameters on Card 1 accordingly (minimum and maximum bounds are also respected). Setting DTT to a negative value causes the thermal time step to be estimated automatically using the heat equation's CFL number:

$$\mathrm{d}t_{\text{th}} = \left(\frac{V}{l} + 2\frac{\alpha}{l^2}\right)^{-1}$$

In the above, V is the fluid velocity, l is the characteristic mesh size, and α is the thermal diffusion coefficient.

3. **Surface tension effects.** For free surface or bi-phasic cases where surface tension is present and is the dominant force, using DTST = 1 in combination with an automatic time step may be convenient. In such cases, the automatic time step is limited by the following Courant condition:

$$\Delta t \leq \Delta t_{\text{st}} \leq \sqrt{\frac{\tilde{\rho} l^3}{2\pi\sigma}}$$

In this condition, $\tilde{\rho}$ is the average density between the two fluids, l is the mesh size, and σ isthe surface tension value.

4. **Diffusive effects.** In advection-driven flows, the default criteria to estimate an automatic time step is sufficient. However, in applications where the advection term is small compared to the viscous forces (usually cases that combine low Reynolds number with high viscosity value), adding the diffusion Courant condition as an additional constraint may be justified:

$$\Delta t \leq \Delta t_{\text{diff}} \leq \frac{l^2}{\nu}$$

Here, l is the mesh size, and ν is the kinematic viscosity.

*ICFD_CONTROL_TRANSIENT

Purpose: This keyword allows to specify different integration scheme options for the transient solver.

Card 1	1	2	3	4	5	6	7	8
Variable	TORD	FSORD						
Type	I	I						
Default	0	0						

VARIABLE	DESCRIPTION
TORD	Time integration order : EQ.0: Second order. EQ.1: First order.
FSORD	Fractional step integration order : EQ.0: Second order. EQ.1: First order.

*ICFD_CONTROL_TURBULENCE

Purpose: Modify the default values for the turbulence model.

Card Summary:

Card 1. This card is required.

TMOD	SUBM	WLAW	KS	CS		TWLAW	

Card 2. This card is read if TMOD = 2 or 3. It is optional.

CS							

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	TMOD	SUBM	WLAW	KS	CS		TWLAW	
Type	I	I	I	F	F		I	
Default	0	1	1	0.	0.		none	

VARIABLE	DESCRIPTION
TMOD	Indicates which turbulence model to use: EQ.0: Turbulence model based on a variational multiscale approach (default) EQ.1: RANS k - ε approach (see Remark 1) EQ.2: LES Smagorinsky or dynamic sub-grid scale model EQ.3: LES Wall adapting local eddy-viscosity (WALE) model EQ.4: RANS k - ω approach (see Remark 2) EQ.5: RANS Spalart-Allmaras approach (see Remark 3)
SUBM	Turbulence submodel. For RANS k - ε approach (TMOD = 1): EQ.1: Standard model

VARIABLE	DESCRIPTION
	EQ.2: Realizable model For LES Smagorinsky or dynamic sub-grid model (TMOD = 2): EQ.1: Smagorinsky model (see Remark 7) EQ.2: Dynamic model (see Remark 8). For RANS k - ω approach (TMOD = 4): EQ.1: Standard model EQ.2: Baseline (BSL) model EQ.3: SST model
WLAW	Law of the wall ID if a RANS turbulence model is selected (see Remark 3): EQ.1: Standard classic law of the wall (default for TMOD = 1) with linear blending between the log and linear regions. Applies a Neumann boundary condition at the wall. EQ.8: Blended law of the wall that uses a Lambert-W function estimator for the log region and directly applies the wall velocity using a Dirichlet condition.
KS	Roughness physical height, only used for RANS turbulence models.
CS	Roughness constant, only used for RANS turbulence models.
TWLAW	Thermal law of the wall flag (see Remark 9): EQ.0: No thermal law of the wall activated. EQ.8: Thermal law of the wall. Only valid if WLAW = 8.

LES Card. Optional card read if TMOD = 2 or 3.

Card 2	1	2	3	4	5	6	7	8
Variable	CS							
Type	F							
Default	0.18							

VARIABLE	DESCRIPTION
CS	Smagorinsky constant if TMOD = 2 and SUBM = 1 or WALE constant if TMOD = 3

Remarks:

1. ***k* -ε model (TMOD = 1) equations.** For this model, the following equations are solved to obtain the turbulent kinetic energy, k, and turbulent dissipation rate, ε:

$$\frac{\partial k}{\partial t} + \nabla \cdot \left(k\mathbf{u} - \left(\nu + \frac{\nu_\mathrm{T}}{\sigma_\mathrm{k}} \right) \nabla k \right) + Y_\mathrm{k} = G_\mathrm{k}$$
$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot \left(\varepsilon\mathbf{u} - \left(\nu + \frac{\nu_\mathrm{T}}{\sigma_\varepsilon} \right) \nabla \varepsilon \right) + Y_\varepsilon = G_\varepsilon$$

where

$$Y_\mathrm{k} = \mathrm{sf}_{Y_\mathrm{k}} \frac{\varepsilon}{k} k$$
$$Y_\varepsilon = \mathrm{sf}_{Y_\varepsilon} \frac{\varepsilon}{k} \varepsilon$$
$$G_\varepsilon = \begin{cases} \mathrm{sf}_{G_\varepsilon} G_\mathrm{k} \dfrac{\varepsilon}{k} & \text{for the standard model} \\ \mathrm{sf}_{G_\varepsilon} S\varepsilon & \text{for the realizable model} \end{cases}$$
$$G_\mathrm{k} = \nu_\mathrm{T} S^2$$
$$\mathrm{sf}_{G_\varepsilon} = \begin{cases} C_{1\varepsilon} & \text{for the standard model} \\ \max\left[0.43, \dfrac{\eta}{5.0 + \eta} \right] & \text{for the realizable model} \end{cases}$$
$$\eta = S \frac{k}{\varepsilon}$$
$$\mathrm{sf}_{Y_\mathrm{k}} = 1.0$$
$$\mathrm{sf}_{Y_\varepsilon} = \begin{cases} C_2 & \text{for the standard model} \\ C_2 \dfrac{k}{k + \sqrt{\nu\varepsilon}} & \text{for the realizable model} \end{cases}$$

Additionally, $\mathbf{u}$ is the velocity, ν is the viscosity, and S is the magnitude of the rate of strain tensor (see Remark 4). The following parameters are constants in the above equations:

$$C_{1\varepsilon} = 1.44, \qquad C_2 = 1.92$$
$$\sigma_\mathrm{k} = 1.0, \qquad \sigma_\varepsilon = \begin{cases} 1.2 & \text{for the standard model} \\ 1.3 & \text{for the realizable model} \end{cases}$$

From k and ε, the turbulent viscosity is updated and injected into the momentum equation in the next iteration:

$$\nu_\mathrm{T} = \mathrm{sf}_\nu \frac{k^2}{\varepsilon}$$

where

$$\mathrm{sf}_\nu = \begin{cases} 0.09 & \text{for the standard model} \\ \dfrac{1}{4.04 + A_s \dfrac{kU^*}{\varepsilon}} & \text{for the realizable model} \end{cases}$$
$$U^* = \sqrt{0.5S^2 + 0.5\Omega^2}$$
$$A_s = \sqrt{6}\cos\varphi$$
$$\varphi = \frac{1}{3}\arccos(W)$$
$$W = \min\left|1.0, \max\left[\left(\sqrt{6.0}\frac{S_{ij}S_{jk}S_{ki}}{\sqrt{0.5S^2}}\right), 1.0\right]\right|$$

In the above, S is the magnitude of the rate of strain tensor, Ω is the magnitude of the vorticity tensor, and S_{ij}, S_{jk}, and S_{ki} are components of the rate of strain tensor in Einstein notation. See Remark 4.

2. **k - ω model (TMOD = 4) equations.** For this model, the following equations are solved to obtain the turbulent kinetic energy, k, and specific turbulent dissipation rate, ω:

$$\frac{\partial k}{\partial t} + \nabla\cdot\left(k\mathbf{u} - \left(\nu + \frac{\nu_\mathrm{T}}{\sigma_\mathrm{k}}\right)\nabla k\right) + Y_\mathrm{k} = G_\mathrm{k}$$
$$\frac{\partial \omega}{\partial t} + \nabla\cdot\left(\omega\mathbf{u} - \left(\nu + \frac{\nu_\mathrm{T}}{\sigma_\omega}\right)\nabla\omega\right) + Y_\omega = G_\omega + D_\omega$$

where

$$Y_\omega = \mathrm{sf}_{Y_\omega}\omega^2$$
$$Y_\mathrm{k} = \mathrm{sf}_{Y_\mathrm{k}}\omega k$$
$$G_\omega = \begin{cases} \mathrm{sf}_{G_\omega} G_\mathrm{k}\dfrac{\omega}{k} & \text{for the standard model} \\ \mathrm{sf}_{G_\omega}\dfrac{G_\mathrm{k}}{\nu_\mathrm{T}} & \text{for the BSL and SST models} \end{cases}$$
$$G_\mathrm{k} = \nu_\mathrm{T}S^2$$
$$D_\omega = \mathrm{sign}\left(\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}\right)\min[|(1.0 - F_1)d_\omega|, |\mathrm{CD}_\mathrm{lim}|]$$
$$\mathrm{CD}_\mathrm{lim} = \frac{(1.0 - F_1)d_\omega}{1. + \left|\dfrac{(1.0 - F_1)d_\omega}{\mathrm{sf}_{Y_\omega}\omega}\right|}\mathrm{sf}_{Y_\omega}\omega$$
$$\mathrm{sf}_{G_\omega} = \begin{cases} \alpha & \text{for the standard model} \\ \alpha\alpha^* & \text{for the BSL and SST models} \end{cases}$$
$$\mathrm{sf}_{Y_\mathrm{k}} = \beta_\mathrm{i}^* f_{\beta^*}$$
$$\mathrm{sf}_{Y_\omega} = \begin{cases} \beta_\mathrm{i} f_\beta & \text{for the standard model} \\ F_1\beta_{\mathrm{i},1} + (1. - F_1)\beta_{\mathrm{i},2} & \text{for the BSL and SST models} \end{cases}$$
$$f_\beta = \frac{1.0 + 70.0\,\chi_\omega}{1.0 + 80.0\chi_\omega}$$

$$\chi_\omega = \left| \frac{\Omega_{ij}\Omega_{jk}S_{ki}}{(\beta^*_\infty \omega)^3} \right|$$

$$f_{\beta^*} = \begin{cases} 1.0 & \text{for the standard model if } \chi_k \le 0.0 \\ \dfrac{1.0 + 680.0\chi_k^2}{1.0 + 400.0\chi_k^2} & \text{for the standard model if } \chi_k > 0.0 \\ 1.0 & \text{for the BSL and SST models} \end{cases}$$

$$\beta^*_i = \beta^*_\infty$$

$$\alpha = \alpha_{\infty,3}$$

$$\alpha_{\infty,3} = \begin{cases} \alpha_\infty & \text{for the standard model} \\ F_1\alpha_{\infty,1} + (1. - F_1)\alpha_{\infty,2} & \text{for the BSL and SST models} \end{cases}$$

$$\sigma_\omega = \begin{cases} \sigma_{\omega,1} & \text{for the standard model} \\ \dfrac{1.0}{\dfrac{F_1}{\sigma_{\omega,1}} + \left(1.0 - \dfrac{F_1}{\sigma_{\omega,2}}\right)} & \text{for the BSL and SST models} \end{cases}$$

$$\sigma_k = \begin{cases} \sigma_{k,1} & \text{for the standard model} \\ \dfrac{1.0}{\dfrac{F_1}{\sigma_{k,1}} + \left(1.0 - \dfrac{F_1}{\sigma_{k,2}}\right)} & \text{for the BSL and SST models} \end{cases}$$

$$F_1 = \begin{cases} 1.0 \\ \tanh\left(\Phi_1^4\right) \end{cases}$$

$$\Phi_1 = \min\left[\max\left[\frac{\sqrt{k}}{0.09\omega d}, \frac{500.0\nu}{d^2\omega}\right], \frac{4.0\rho k}{\sigma_{\omega,2} D^+_\omega d^2}\right]$$

$$D^+_\omega = \max\left[d_\omega, 1.e^{-20}\right]$$

$$d_\omega = \frac{2.0}{\omega\sigma_{\omega,2}} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j}$$

In the above, ρ is the density, d is the distance to the closest wall, and ν is the kinematic linear viscosity (see Remark 4). The following parameters are constants:

$$\sigma_{k,1} = \begin{cases} 2.0 & \text{for the standard and BSL models} \\ 1.176 & \text{for the SST model} \end{cases}, \qquad \sigma_{k,2} = 1.0$$

$$\sigma_{\omega,1} = 2.0, \qquad \sigma_{\omega,2} = 1.168$$

$$\beta^*_\infty = 0.09, \qquad \beta_i = 0.072, \qquad \beta_{i,1} = 0.075, \qquad \beta_{i,2} = 0.0828$$

$$\alpha_{\infty,1} = \frac{\beta_{i,1}}{\beta^*_\infty} - \frac{\kappa^2}{\sigma_{\omega,1}\sqrt{\beta^*_\infty}}, \qquad \alpha_{\infty,2} = \frac{\beta_{i,2}}{\beta^*_\infty} - \frac{\kappa^2}{\sigma_{\omega,2}\sqrt{\beta^*_\infty}}$$

$$\alpha^*_\infty = 1.0, \qquad \alpha_\infty = 0.52, \qquad \alpha_0 = \frac{1.0}{9.0}, \qquad \alpha^*_0 = \frac{\beta_i}{3.0}$$

From k and ω, the turbulent viscosity is updated and injected into the momentum equation in the next iteration:

$$\nu_T = \mathrm{sf}_\nu \frac{k}{\omega}$$

$$\mathrm{sf}_\nu = \begin{cases} \dfrac{1.0}{\max\left[\dfrac{1}{\alpha^*}, S\dfrac{2.92}{\omega}\right]} & \text{for the standard and BSL models} \\ \dfrac{1.0}{\max\left[\dfrac{1}{\alpha^*}, S\dfrac{F_2}{0.31\omega}\right]} & \text{for the SST model} \end{cases}$$

In the above, S is the magnitude of the rate of strain tensor (see Remark 4).

3. **Spalart-Allmaras (TMOD = 5) model equations.** For this model, the following equations are solved to obtain the modified turbulent kinetic viscosity, $\tilde{\nu}$:

$$\frac{\partial \tilde{\nu}}{\partial t} + \nabla \cdot \left(\tilde{\nu} \boldsymbol{u} - \left(\frac{\nu}{\sigma_{\tilde{\nu}}} + \frac{\tilde{\nu}}{\sigma_{\tilde{\nu}}} \right) \nabla \tilde{\nu} \right) + Y_{\tilde{\nu}} = G_{\tilde{\nu}} + D_{\tilde{\nu}}$$

where

$$\begin{aligned} D_{\tilde{\nu}} &= \frac{C_{b2}}{\sigma_{\tilde{\nu}}} \left(\frac{\partial \tilde{\nu}}{\partial x_j} \right)^2 \\ Y_{\tilde{\nu}} &= \mathrm{sf}_{Y_{\tilde{\nu}}} \tilde{\nu}^2 \\ G_{\tilde{\nu}} &= \mathrm{sf}_{G_{\tilde{\nu}}} \tilde{\nu} \\ \mathrm{sf}_{G_{\tilde{\nu}}} &= C_{b1}(1.0 - f_{t2})\tilde{S} \\ \mathrm{sf}_{Y_{\tilde{\nu}}} &= \frac{C_{\omega 1} f_\omega}{d^2} - \frac{C_{b1} f_{t2}}{\kappa^2 d^2} \\ f_{t2} &= C_{t3} e^{-C_{t3}\chi^2} \\ f_\omega &= g\left[\frac{1.0 + C_{\omega 3}^6}{g^6 + C_{\omega 3}^6} \right] \\ g &= r + C_{\omega 2}\left(r^6 - r\right) \\ r &= \min\left[10.0, \frac{\tilde{\nu}}{\tilde{S}(\kappa d)^2} \right] \\ \tilde{S} &= \begin{cases} S + \dfrac{S\left({C_{\nu 2}}^2 S + C_{\nu 3}\bar{S}\right)}{(C_{\nu 3} - 2C_{\nu 2})S - \bar{S}} & \text{for } \bar{S} < -C_{\nu 2} S \\ S + \bar{S} & \text{for } \bar{S} \geq -C_{\nu 2} S \end{cases} \\ \bar{S} &= \frac{\tilde{\nu}}{(\kappa d)^2} f_{\nu 2} \\ f_{\nu 2} &= 1.0 - \frac{\chi}{1.0 + \chi \mathrm{sf}_\nu} \\ \chi &= \frac{\tilde{\nu}}{\nu} \\ \mathrm{sf}_\nu &= \frac{\chi^3}{\chi^3 + C_{\nu 1}^3} \end{aligned}$$

In the above, d is the distance to the closest wall, ν is the kinematic linear viscosity, κ is the von Kármán constant, and S is the magnitude of the rate of strain tensor (see Remark 4). The following parameters are constants:

$$C_{t1} = 1.0, \quad C_{t2} = 2.0, \quad C_{t3} = 0.0, \quad C_{t4} = 0.5$$

$$C_{\omega1} = \frac{C_{b1}}{\kappa^2} + \frac{1.0 + C_{b2}}{\sigma_{\tilde{\nu}}}, \quad C_{\omega2} = 0.3, \quad C_{\omega3} = 2.0$$
$$C_{\nu1} = 7.1, \quad C_{\nu2} = 0.7, \quad C_{\nu3} = 0.9$$
$$C_{b1} = 0.1355, \quad C_{b2} = 0.622$$
$$\sigma_{\tilde{\nu}} = 0.66$$

From $\tilde{\nu}$, the turbulent viscosity is updated and injected into the momentum equation in the next iteration:

$$\nu_T = \mathrm{sf}_\nu \tilde{\nu}$$

4. **Additional variables and model limits.** In the previously described turbulence models, d represents the distance to the closest wall, κ is the von Kármán constant (0.4187), ν is the kinematic laminar viscosity, and ρ is the fluid density. S and Ω are the magnitudes of the rate of strain and vorticity tensors with S_{ij} and Ω_{ij} as the components. For instance in 3D,

$$S = \sqrt{2S_{ij}S_{ij}} = \sqrt{2[S_{11}^2 + S_{22}^2 + S_{33}^2 + 2(S_{12}^2 + S_{13}^2 + S_{23}^2)]}$$
$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right)$$

with u_i as the i^{th} component of velocity. At the wall, S_{ij} is expressed as:

$$S_{ij} = \frac{\rho\tau}{2\nu}(t_i n_j + t_j n_i)$$

with τ the wall shear, **t** the tangent vector to the wall, and **n** the normal vector to the wall.

Additionally, note that the turbulent viscosity is bounded by:

$$\nu_{T,\min} < \nu_T < \nu_{T,\max}$$
$$\nu_{T,\min} = 1.e^{-2}\nu$$
$$\nu_{T,\max} = \min\left(1.e^{4}\nu, \sqrt{k}L_{\text{ref}}0.07\right)$$

Here L_{ref} is a reference length automatically estimated by the solver based on inlet length or model dimensions (the latter condition is not present for the Spalart-Allmaras models).

5. **Laws of the wall for RANS models.** For RANS models, the following laws of the wall are available:

   a) *Standard Classic (WLAW = 1) with linear blending when* $5 \le y^+ \le 60$.

$$U^+_{\text{blend}}(y^+) = \left(1 - \alpha(y^+)\right)y^+ + \alpha(y^+)(1/\kappa \ln(Ey^+))$$
$$\alpha(y^+) = \frac{y^+ - 5}{60 - 5}$$
$$y^+ = \frac{yU_\tau}{\nu}$$

$$U^+_{\text{blend}} = \frac{U}{U_\tau}$$
$$U_\tau = \sqrt{\frac{\tau_w}{\rho}}$$

This is the default for TMOD = 1. In the above, U is the mean streamwise velocity at a distance y from the wall, τ_w is the wall shear stress, U_τ is the friction velocity, ρ is the density, E is a constant that has the value 9.793, ν is the kinematic viscosity, and κ is the von Kármán constant.

b) *Lambert W-function blended law of the wall (WLAW = 8)*

$$z = \kappa U e^{\kappa(B-\Delta B)} \frac{y}{\nu}$$
$$W = \text{LambertW}(z)$$
$$U_\tau = \frac{\kappa U}{W}$$
$$U^+_{\text{lam}} = y^+$$
$$U^+_{\text{turb}} = \frac{1.0}{\kappa}\ln(y^+) + B - \Delta B$$
$$U_w = U_\tau \left(\left(1.0 - F(y^+)\right) U^+_{\text{lam}} + F(y^+) U^+_{\text{turb}} \right)$$

In the above, $F(y^+)$ is a blending function, B is $\ln E$, and U_w is the velocity at the wall.

6. **RANS turbulence model with roughness included.** When a RANS turbulence model is selected, it is possible to define extra parameters to account for roughness effects. In such cases, an extra term will be added to the logarithmic part of the different laws of the wall:

$$U^+ = \frac{1}{\kappa}\ln(EY^+) - \Delta B$$

If we introduce the non-dimensional roughness height:

$$K^+ = \frac{K_s {C_\mu}^{1/4} k^{1/2}}{\nu} \ ,$$

we have:

$$\Delta B = \begin{cases} 0 & \text{for } K^+ \le 2.25 \\ \frac{1}{\kappa}\ln\left(\frac{K^{\pm 2.25}}{87.75} + C_s K^+\right) \times \sin\left(0.4258(\ln K^+ - 0.811)\right) & \text{for } 2.25 < K^+ \le 90.0 \\ \frac{1}{\kappa}\ln(1 + C_s K^+) & \text{for } 90. < K^+ \end{cases}$$

7. **LES Smagorinksy.** The LES Smagorinsky turbulence model uses the Van Driest damping function close to the wall:

$$f_v = 1 - e^{-\frac{y^+}{A^+}}$$

8. **LES dynamic model.** The LES dynamic model is based on the model originally proposed by Germano et. al. (1991) and improved by Lilly (1992), with localization on Cs by Piomelli and Liu (1995).

9. **Thermal law of the wall.** When the thermal law of the wall is activated, a wall heat transfer coefficient h_{wall} is calculated. This heat transfer coefficient is applied as a Robin boundary condition at walls where you have prescribed a temperature value, therefore replacing the Dirichlet imposition. In conjugate heat transfer problems (CHT), this h_{wall} is added to the user-defined heat transfer coefficient value at thermal-solid-fluid interfaces when Mortar contact is selected. See *ICFD_CONTROL_CONJ and *ICFD_BOUNDARY_CONJ_HEAT. This heat transfer coefficient is calculated as:

$$h_{\text{wall}} = \frac{1.}{Tr_{\text{cond}} + Tr_{\text{conv}}}$$

where

$$Tr_{\text{conv}} = \min\left[0., \frac{T^+ - \text{SF}_{\text{cond}} y_{\text{T}}^+}{U_\tau \rho C_{\text{p}}}\right]$$

$$Tr_{\text{cond}} = \text{SF}_{\text{cond}} \frac{y}{k}$$

$$T^+ = \begin{cases} T_{\text{lam}}^+ & y \le 5. \\ (1. - F(y^+) T_{\text{lam}}^+ + F(y^+) T_{\text{turb}}^+ & 5. < y^+ < 30. \\ T_{\text{turb}}^+ & y \ge 30. \end{cases}$$

$$T_{\text{turb}}^+ = \frac{1.0}{\kappa_{\text{T}}} \ln(y_{\text{T}}^+) + B_{\text{T}}(\text{Pr})$$

$$T_{\text{lam}}^+ = y_{\text{T}}^+$$

$$B_{\text{T}}(\text{Pr}) = \left(3.85 \text{Pr}^{1./3.} - 1.3\right)^2$$

$$y_{\text{T}}{}^+ = \text{Pr} y^+$$

$$\text{Pr} = \nu \frac{\rho C_{\text{p}}}{k}$$

$$\kappa_{\text{T}} = \frac{\kappa}{\text{Pr}_{\text{T}}}$$

In the above, r_{cond} is the conductive thermal resistance, r_{conv} is the convective thermal resistance, SF_{cond} is a scale factor specified with SFCOND on *ICFD_CONTROL_RANS, k is the thermal conductivity, C_{p} is the heat capacity, y is the distance to the first internal node, Pr is the Prandtl number, and Pr_{T} is the turbulent Prandtl number defined with PRT on *ICFD_MAT or automatically calculated as $\text{Pr}_{\text{T}} = 0.85 + 0.012/\text{Pr}$.

When a thermal law of the wall is present, the thermal conductivity in the solve of the heat equation will be modified as:

$$k_{\text{eff}} = k + k_{\text{T}}$$

$$k_T = \frac{\nu_T \rho C_p}{Pr_T}$$

*ICFD_CONTROL_TURB_SYNTHESIS

Purpose: This keyword enables the user impose a divergence-free turbulent field on inlets.

Card must be used jointly with VAD = 4 of keyword *ICFD_BOUNDARY_PRESCRIBED_VEL.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	IU	IV	IW	LS			
Type	I	F	F	F	F			
Default	0	10^{-3}	10^{-3}	10^{-3}	h_{min}			

VARIABLE	DESCRIPTION
PID	Part ID of the surface with the turbulent velocity inlet condition.
IU, IV, IW	Intensity of field fluctuations over x, y, and z directions, $\text{IU} = \frac{u'}{u_{\text{avg}}}$.
LS	Integral length scale of turbulence

Remarks:

1. If this card is not defined but a turbulent field inlet has been activated. See VAD = 4 of *ICFD_BOUNDARY_PRESCRIBED_VEL, the default parameters will be used.

*ICFD_DATABASE_AVERAGE

Purpose: This keyword enables the computation of time average variables at given time intervals.

Card 1	1	2	3	4	5	6	7	8
Variable	DT							
Type	F							
Default	none							

VARIABLE	DESCRIPTION
DT	Over each DT time interval, an average of the different fluid variables will be calculated and then reset when moving to the next DT interval.

Remarks:

1. The file name for this database is icfdavg.*.dat with the different averaged variable values copied in a ASCII format.

*ICFD_DATABASE_BIN

Purpose: Print out ICFD solver results in a binary file that can be used for a subsequent ICFD simulation. See Remark 1.

Card 1	1	2	3	4	5	6	7	8
Variable	IOUTLV	DTOUT	IAVG	IMESH				
Type	I	F	I	I				
Default	none	0.	0	0				

File Name for the Binary File Card. Include this card if IOUTLV = 2.

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A							
Default	none							

VARIABLE	DESCRIPTION
IOUTLV	Flag to specify if the binary file should be output: EQ.0: No output file is generated. EQ.1: The output file is generated with the default file name icfdbintot. EQ.2: The output file is generated with a file name specified on Card 2. See Remark 1.
DTOUT	Time interval to print the output. If DTOUT is equal to 0.0, then the ICFD time step is used.
IAVG	Flag to specify whether the output quantities are instantaneous or average: EQ.0: Instantaneous fluid quantities are printed. EQ.1: Time averaged fluid quantities are printed.

VARIABLE	DESCRIPTION
IMESH	Flag to specify whether the fluid mesh should also be output in the binary file: EQ.0: Off. The fluid mesh is not part of the binary file output. EQ.1: Output the fluid mesh at the start and whenever a remesh occurs.
FILENAME	Name of the binary file if IOUTLV = 2

Remarks:

1. **ICFD initialization.** This keyword is meant to be used with *ICFD_INITIAL_-BIN for two step simulation workflows. For example, a first ICFD analysis that uses the steady state solver can output results in a binary file using this keyword. The results are then used as an initial condition in the next simulation which is often transient, FSI or conjugate heat transfer. One advantage of this approach is that the meshes between the two simulations need not be exactly the same because *ICFD_INITIAL_BIN interpolates results from the binary file and applies them on its own current mesh. The default name for this binary output is icfdbintot but can be changed with FILENAME if IOUTLV = 2.

***ICFD_DATABASE_DRAG**_*{OPTION}*

Available options include:

<BLANK>

CSYS

VOL

Purpose: Enable the computation of drag forces over given surface parts of the model. If multiple keywords are given, the forces over the PID surfaces are output in separate files and are also added and output in a separate file.

For the VOL option, the drag calculation can also be applied to a volume defined by *ICFD_PART_VOL. This feature is mostly useful in porous media applications to output the pressure drag of the porous media domain.

The CSYS option enables defining a local reference frame in which the forces will be calculated rather than the global X, Y, and Z components. Defining the reference frame requires a second line. Consequently, the use of the CSYS option restricts the number of lines to two for that version of the keyword.

Surface Drag Cards. Include one card for each surface (or volume in the case of the VOL keyword option) on which drag is applied unless the CSYS keyword option is used. For the CSYS, keyword option, each instantiation of the keyword can only include two data cards, namely, Cards 1 and 2, without repeated cards. When not using the CSYS keyword option, this input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	CPTID	DTOUT	PEROUT	DIVI	ELOUT		
Type	I	I	F	I	I	I		
Default	none	none	0.	0	10	0		

CSYS option. Only include this card if the CSYS keyword option is selected.

Card 2	1	2	3	4	5	6	7	8
Variable	IREFTY	PTID1	PTID2	PTID3				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
PID	Part ID of the surface (see *ICFD_PART) or fluid volume (see *ICFD_PART_VOL) in the case of the VOL keyword option where the drag force will be computed
CPTID	Center point ID (see *ICFD_DEFINE_POINT) used for the calculation of the force's moment. By default, the reference frame center is used, that is, $(0, 0, 0)$.
DTOUT	Time interval between outputs. If DTOUT is equal to 0.0, then the ICFD timestep is used.
PEROUT	Flag for outputting the contribution of the different elements to the total drag in fractions of the total drag in d3plot: EQ.0: Off GE.1: On
DIVI	Number of drag divisions for PEROUT. The default is 10 which means the contributions will be grouped in 10 deciles.
ELOUT	Flag for outputting the drag value of each element in the d3plot: EQ.0: Off GE.1: On
IREFTY	Local reference frame definition type: EQ.0: Using node IDs coming from a mechanical model (when present, see *NODE) EQ.1: Using ICFD points defined by *ICFD_DEFINE_POINT.

VARIABLE	DESCRIPTION
PTID1, PTID2, PTID3	Local reference frame built by building an orthogonal base based on the direction vectors created by the coordinates of PTID2 minus PTID1 and PTID3 minus PTID1. The coordinates of PTID1 act as the origin point of the local reference frame. PTID1, PTID2, and PTID3 can either be point IDs (*ICFD_DEFINE_PONT) or node IDs, depending on the setting of IREFTY. Note that the optional translation of the application point for the force's moment calculation is still done with CPTID and not PTID1.

Remarks:

1. **Output file names.** The file name for this database is icfdragi for instantaneous drag and icfdraga for the drag computed using average values of pressure and velocities.

2. **Output information.** The output is described as follows:

   a) Fpx, Fpy, and Fpz refer to the three components of the pressure drag force:

   $$\mathbf{F}_p = \int P d\mathbf{A} \ ,$$

   where P is the pressure and A is the surface area.

   b) Fvx, Fxy, and Fvz refer to the three components of the viscous drag force:

   $$\mathbf{F}_v = \int \mu \frac{\partial \mathrm{u}}{\partial \mathrm{y}} \mathrm{d}\mathbf{A} \ ,$$

   where $\frac{\partial \mathrm{u}}{\partial \mathrm{y}}$ is the shear velocity at the wall, μ is the viscosity and A is the surface area.

   c) Mpx, Mpy, Mpz, Mvx, Mvy, and Mvz refer to the three components of the pressure and viscous force moments.

*ICFD_DATABASE_FLUX

Purpose: This keyword enables the computation of the flow rate and average pressure over given parts of the model. If multiple keywords are given, separate files are output.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	DTOUT						
Type	I	F						
Default	none	↓						

VARIABLE	DESCRIPTION
PID	Part ID of the surface where the flow rates will be computed
DTOUT	Output frequency. Default is at every fluid timestep.

Remarks:

1. **Database Name.** The file name for this database is icfd_flux.dat.

2. **Database Components.** The flux database contains the flow rate through a section, called "output flux,"

$$\Phi = \sum_i (\mathbf{V}_i \cdot \mathbf{n}_i) A_i \ ,$$

the average pressure, called "Pre-avg,"

$$P_{\text{avg}} = \frac{\sum_i P_i A_i}{\sum_i A_i} \ ,$$

and the total area, called "Areatot."

*ICFD_DATABASE_FLUX_SURF

Purpose: Enable the computation of the flow rate and average pressure over a given surface (icfd part) of the model which is not necessary part of the analysis but lies in the fluid volume by mapping and interpolating volume results on the given surface.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	DTOUT						
Type	I	F						
Default	none	↓						

VARIABLE	DESCRIPTION
PID	Part ID of the surface where the flow rates will be computed
DTOUT	Output frequency. Default is at every fluid timestep.

Remarks:

1. **Database Name.** The file name for this database is icfd_fluxsurf.dat.

*ICFD_DATABASE_FORCE_DEM

Purpose: Enable the computation of the total fluid force that is transferred to the DEM particles over time. The output is in icfd_force_dem.dat.

Card 1	1	2	3	4	5	6	7	8
Variable	OUT							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
OUT	Flag to enable computing the fluid forces and generating output: EQ.0: No output is generated. EQ.1: Output is generated.

*ICFD_DATABASE_GOA

Purpose: Enable the computation of the geometric orifice area (GOA), a quantity of interest used to evaluate valve performance. The GOA surface connects the free edges of the leaflets. It is defined in the fluid mesh as surface mesh. If multiple keywords exist in the input deck, separate files are output. It must be used with *MESH_SURFACE_NULL which defines parts where no physics is computed. These parts are only used for geometrical computations or post-processing.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	DTOUT						
Type	I	F						
Default	none	↓						

VARIABLE	DESCRIPTION
PID	Fluid part ID of the fluid surface mesh where the GOA will be computed (see *MESH_SURFACE_NULL)
DTOUT	Output frequency. The default is at every fluid time step.

Remarks:

1. **Database name.** The file name for this database is icfd_goa.dat.

2. **GOA surface.** Define the GOA surface in the fluid mechanics problem with *MESH_SURFACE_NULL. It is unnecessary to define *ICFD_PART for any *MESH_SURFACE_NULL part IDs. The surface is automatically excluded from the FSI coupling.

*ICFD_DATABASE_HTC

Purpose: This keyword allows the user to trigger the calculation of the Heat transfer coefficient using different methods and to control the output options.

Card 1	1	2	3	4	5	6	7	8
Variable	OUT	HTC	TB					OUTDT
Type	I	I	F					F
Default	0	0.	0.					0.

VARIABLE	DESCRIPTION
OUT	Determines if the solver should calculate the heat transfer coefficient and how to output it : EQ.0: No HTC calculation EQ.1: HTC calculated and output in LSPP as a surface variable. EQ.2: The solver will also look for FSI boundaries and output the HTC value at the solid nodes in an ASCII file called icfdhtci.dat. EQ.3: The solver will also look for FSI boundaries that are part of SEGMENT_SETS and output the HTC for those segments in an ASCII file called icfd_convseg.****.key in a format that can directly read by LS-DYNA for a subsequent pure structural thermal analysis.
HTC	Determines how the HTC is calculated. EQ.0: Automatically calculated by the solver based on the average temperature flowing through the pipe section (See Remark 1). EQ.1: User imposed value (See Remark 2).
TB	Value of the bulk temperature if HTC = 1.
OUTDT	Output frequency of the HTC in the various ASCII files. If left to 0., the solver will output the HTC at every timestep.

Remarks:

1. The heat transfer coefficient is frequently used in thermal applications to estimate the effect of the fluid cooling and it derived from a CFD calculation.

2. The heat transfer coefficient is defined as follows:

$$h = \frac{q}{T_s - T_b}$$

with q the heat flux, T_s the surface temperature and T_b the so called "bulk" temperature. For external aerodynamic applications, this bulk temperature is often defined as a constant (ambient or far field conditions, HTC = 1). However, for internal aerodynamic application, this temperature is often defined as an average temperature flowing through the pipe section with the flow velocity being used as a weighting factor (HTC = 0).

*ICFD_DATABASE_NODEAVG

Purpose: This keyword enables the computation of the average quantities on surface nodes defined in *ICFD_DATABASE_NODOUT.

Card 1	1	2	3	4	5	6	7	8
Variable	ON							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
ON	If equal to 1, the average quantities will be computed.

Remarks:

1. The file name for this database is icfd_nodeavg.dat.

*ICFD_DATABASE_NODOUT

Purpose: This keyword enables the output of ICFD data on surface nodes. For data in the fluid volume, it is advised to use points or tracers (See *ICFD_DATABASE_-POINTOUT).

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	OUTLV	DTOUT						
Type	I	F						
Default	0	0.						

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	NID1	NID2	NID3	NID4	NID5	NID6	NID7	NID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
OUTLV	Determines if the output file should be dumped. EQ.0: No output file is generated. EQ.1: The output file is generated.
DTOUT	Time interval to print the output. If DTOUT is equal to 0.0, then the ICFD timestep will be used.
NID..	Node IDs.

Remarks:

1. The file name for this database is icfd_nodout.dat.

*ICFD_DATABASE_NTEMPOUT

Purpose: Output the temperature at individual nodes in a format consistent with *ICFD_-INITIAL_TEMPNODE to initialize a subsequent ICFD problem.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	NID	DTOUT						
Type	I	F						
Default	none	0.0						

VARIABLE	DESCRIPTION
NID	Internal ICFD node ID
DTOUT	Output frequency EQ.0.0: The ICFD timestep will be used.

*ICFD_DATABASE_POINTAVG

Purpose: This keyword enables the computation of the average quantities on point sets using the parameters defined in *ICFD_DATABASE_POINTOUT.

Card 1	1	2	3	4	5	6	7	8
Variable	ON							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
ON	If equal to 1, the average quantities will be computed.

Remarks:

1. The file name for this database is icfd_psavg.dat.

*ICFD_DATABASE_POINTOUT

Purpose: This keyword enables the output of ICFD data on points.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID	DTOUT	PSTYPE	VX	VY	VZ		
Type	I	F	I	F	F	F		
Default	0	0.	0	0.	0.	0.		

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	PID	X	Y	Z				
Type	I	F	F	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
PSID	Point Set ID.
DTOUT	Time interval to print the output. If DTOUT is equal to 0.0, then the ICFD timestep will be used.
PSTYPE	Point Set type : EQ.0: Fixed points. EQ.1: Tracer points using prescribed velocity. EQ.2: Tracer points using fluid velocity. EQ.3: Tracer points using mesh velocity..
VX, VY, VZ	Constant velocities to be used when PSTYPE = 1
PID	Point ID

VARIABLE	DESCRIPTION
X, Y, Z	Point initial coordinates

Remarks:

1. The file name for this database is icfd_pointout.dat.

*ICFD_DATABASE_RESIDUALS

Purpose: This keyword allows the user to output the residuals of the various systems.

Card 1	1	2	3	4	5	6	7	8
Variable	RLVL							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
RLVL	Residual output level : EQ.0: No output. EQ.1: Only outputs the number of iterations needed for solving the pressure Poisson equation. EQ.2: Outputs the number of iterations for the momentum, pressure, mesh movement and temperature equations. EQ.3: Also gives the residual for each iteration during the solve of the momentum, pressure, mesh movement and temperature equations.

Remarks:

1. The file names for the momentum, pressure, mesh movement and temperature equations are called icfd_residuals.moms.dat, icfd_residuals.pres.dat, icfd_residuals.mmov.dat, and icfd_residuals.temp.dat respectively.

*ICFD_DATABASE_SSOUT

Purpose: Output the pressure load on a structure from the fluid. It can be useful for linear FSI applications, where the structure is made static, and the loads applied by the fluid are retrieved.

Card 1	1	2	3	4	5	6	7	8
Variable	OUT	OUTDT	LCIDSF					POFF
Type	I	I	I					F
Default	0	0.	0					0.

VARIABLE	DESCRIPTION
OUT	Determines if the solver should retrieve the pressure loads and how to output it: EQ.0: Inactive EQ.1: The fluid solver will collect the segment sets (see *SET_SEGMENT) that are part of a FSI boundary and retrieve the pressure for subsequent print out in an icfd_presseg and icfd_lcsegid pair of files. EQ.2: Same as 1 except the results are collected in a single icfd_presseg / icfd_plcsegid pair of files and the load curves associated with each segment are made functions of time, thus taking the transient nature of the CFD analysis into account. EQ.3: Same as 2 except uses a more memory efficient way of handling and outputting the data.
OUTDT	Frequency of the pressure extraction. If left as 0., the solver will extract the pressure of the fluid on the FSI boundary at every time step. This can lead to additional memory and calculation cost.
LCIDSF	Optional load curve ID to apply a scale factor on the fluid pressure output
POFF	Optional pressure offset on the fluid pressure output

*ICFD_DATABASE_SSOUT_EXCLUDE

Purpose: This keyword defines which segment set IDs are excluded from the SSOUT search. No forces coming from the fluid will be transmitted on those segment sets for output (see ICFD_DATABASE_SSOUT).

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SSOUTID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
SSOUTID	Segment Set ID of the solid mechanics problem which is to be excluded from the output of the fluid forces on the solid boundaries.

*ICFD_DATABASE_TEMP

Purpose: This keyword enables the computation of the average temperature and the heat flux over given parts of the model. If multiple keywords are given, separate files are output.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	DTOUT						
Type	I	F						
Default	none	none						

VARIABLE	DESCRIPTION
PID	Part ID of the surface where the average temperature and heat flux will be computed.
DTOUT	Output frequency. Default is at every fluid timestep.

Remarks:

1. The file name for this database is icfd_thermal.dat.

2. Two average temperature are given in the icfd_thermal.dat file: "Temp-avg" and "Temp-sum". The average temperature is calculated using the local node area as weighting factor,

$$T_{\text{avg}} = \frac{\sum_i^N T_i A_i}{\sum_i^N A_i},$$

whereas, the sum is not weighted by area

$$T_{\text{sum}} = \frac{\sum_i^N T_i}{N}$$

If the mesh is regular, the two values will be of similar value. The icfd_thermal.dat output file also includes the average heat flux, the total surface area, and the average heat transfer coefficients (See *ICFD_DATABASE_HTC).

*ICFD_DATABASE_TIMESTEP

Purpose: This keyword enables the output of ICFD data regarding the ICFD timestep.

Output Options Card.

Card 1	1	2	3	4	5	6	7	8
Variable	OUTLV							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
OUTLV	Determines if the output file should be dumped. EQ.0: No output file is generated. EQ.1: The output file is generated.

Remarks:

1. The file name for this database is icfd_tsout.dat.

2. Outputs the run's ICFD timestep versus the timestep calculated using the ICFD CFL condition as criteria (autotimestep). This can be useful in cases using a fixed timestep where big mesh deformations and/or big fluid velocity changes occur in order to track how that fixed timestep value compares to the reference auto-timestep.

*ICFD_DATABASE_TPD

Purpose: Enable the output of the transvalvular pressure difference (TPD) data. The TPD is computed by averaging the fluid pressure at two spheres upstream and downstream of a valve and computing the difference between them.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	X1	Y1	Z1	X2	Y2	Z2	
Type	I	F	F	F	F	F	F	
Default	none	0.	0.	0.	0.	0.	0.	

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	DTOUT	RAD1	RAD2					
Type	F	F	F					
Default	↓	0.	0.					

VARIABLE	DESCRIPTION
PID	Part ID used to plot multiple files if multiple *ICFD_DATABASE_-TPD keywords are defined.
X1, X2, X3	Coordinates of the center of the upstream sphere where the upstream average pressure is computed.
Y1, Y2, Y3	Coordinates of the center of the downstream sphere where the downstream average pressure is computed.
DTOUT	Time interval to print the output. If DTOUT is equal to 0.0, then the ICFD time step is used.
R1	Radius of the upstream sphere
R2	Radius of the downstream sphere

Remarks:

1. **Output filename.** The filename for this database is icfd_tpd.dat.

2. **Computing TPD.** The TPD is computed as:

$$\mathrm{TPD}(t^n) = P_{\mathrm{up}}(t^n) - P_{\mathrm{down}}(t^n)$$

*ICFD_DATABASE_TWINBUILDER

Purpose: Enable the output of ICFD data to be consumed by the Ansys Twin Builder static or dynamic ROM module. If this keyword is active, the solver automatically creates a directory tree compatible with the ROM module. If the model is transient, the dynamic ROM format is used. If the model is steady, the static ROM format is used. In the dynamic case, DTOUT in *ICFD_CONTROL_OUTPUT determines the output frequency. The output scenarios are identified using the job ID for each run. The job IDs should go from 1 to the number of scenarios used to train the ROM. See Remark 1.

Include as many *cards* as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	OUTRP	INEX1	INEX2	INEX3	INEX4	INEX5	INEX6	INEX7
Type	A	A	A	A	A	A	A	A
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
OUTRP	Name of the output variable or response. All the fields are nodal. EQ.pres: Pressure EQ.apres: Time average pressure EQ.vel: Velocity EQ.avel: Time average velocity EQ.lset: Level set EQ.temp: Temperature EQ.atemp: Time average temperature
INEX1-7	Name of the input variables or excitations. Field values like pressure, velocity, and temperature are nodal arrays of boundary conditions. In the case of material properties, the values written to the excitation's files are scalars. A combination of up to seven excitations can be used for each output. EQ.pres: Pressure values at imposed pressure boundary conditions. EQ.vel: Velocity values at inflow boundary conditions.

VARIABLE	DESCRIPTION
	EQ.temp: Temperature values at imposed temperature boundary conditions.
	EQ.rho: Fluid density
	EQ.mu: Fluid viscosity
	EQ.hc: Heat capacity
	EQ.tc: Thermal conductivity

Remarks:

1. **Output structure.** This database creates a tree structure of directories that contain the different scenarios to be used to train the ROMs. The root directory name is `tb_s` for a static ROM and `tb_d` for a dynamic ROM. They are created in the location where the executable file is. The scenarios are numbered according to the job ID of the process. The user is responsible for providing numerical job IDs which are recommended to start from 1 and increase to the number of scenarios.

*ICFD_DATABASE_UINDEX

Purpose: This keyword allows the user to have the solver calculate the uniformity index (See Remark 1).

Card 1	1	2	3	4	5	6	7	8
Variable	OUT							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
OUT	Determines if the solver should calculate the uniformity index. EQ.0: Off. EQ.1: On.

Remarks:

1. **Uniformity Index.** The uniformity index is a post treatment quantity which measures how uniform the flow is through a given section. It is especially useful in internal aerodynamics cases. It is expressed as :

$$\gamma = 1 - \frac{1}{2nA}\sum_{i=1}^{n}\left[\frac{\sqrt{(u_i - \bar{u})^2}}{\bar{u}}A_i\right]$$

with A_i, the local cell area, A the total section area, u_i the local velocity, $\bar{u}$ the average velocity through the section, and n the number of cells.

Values close to 0 means that the flow is very unevenly distributed. This can be used to identify bends, corners or turbulent effects. Values close to 1 imply smooth or equally distributed flow through the surface.

*ICFD_DATABASE_VOL

Purpose: Print out general geometry information regarding the fluid domain, such as total volume and center of gravity. This information is output to an ASCII file beginning with icfd_volout.

Card 1	1	2	3	4	5	6	7	8
Variable	IOUTLV							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
IOUTLV	Determines if the output file should be dumped. EQ.0: No output file is generated. EQ.1: The output file is generated.

*ICFD_DATABASE_WETNESS

Purpose: For free surface problems, output the wetted area as a function of time for a given surface part. It is output to an ASCII file beginning with icfd_wetness.

Card 1	1	2	3	4	5	6	7	8
Variable	SPID	DTOUT						
Type	I	F						
Default	none	0.						

VARIABLE	DESCRIPTION
SPID	Part ID of the surface where the wetted area will be computed
DTOUT	Time interval to print the output. If DTOUT is equal to 0.0, then the ICFD time step will be used.

Remarks:

1. **Wetted Area.** The ICFD solver identifies a surface as wet if its level set value is positive.

*ICFD_DEFINE_GAPCLOSURE

Purpose: Activate a gap closure treatment that deals with flow blockage when surfaces come into contact. In ICFD, surface meshes are not allowed to collide or penetrate each other. The gap closure feature activated with this keyword triggers a flow blockage between two surfaces in close proximity based on a contact distance given in the input here. More than one instance of this keyword can be used in an input deck to define different sets of surfaces with different threshold values.

Note that the flow treatment in the gap and the permeability can be changed with *ICFD_CONTROL_GAP.

Card 1	1	2	3	4	5	6	7	8
Variable	HGAP	ANGLE						
Type	F	F						
Default	none	0.						

Card 2	1	2	3	4	5	6	7	8
Variable	PID1	PID2	PID3	PID4	PID5	PID6	PID7	PID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
HGAP	Threshold distance. See Remark 1 for a detailed description.
ANGLE	An additional condition of threshold angle if a nonzero value is specified. This field takes values between 0 and 180 degrees. See Remark 2.
PID*n*	Part IDs of the surfaces involved in the gap closure treatment. More than one part ID must be defined because gap closure treatment on a single surface is currently not available. See Remark 3.

Remarks:

1. **Gap closure detection considering only threshold distance.** Gap closure detection is described as follows. Consider the space between surfaces A and B. For a fluid node near these two surfaces, find its closest distances to surfaces A (d_A) and B (d_B), respectively, as shown in Figure 7-7. If the sum of d_A and d_B is less than HGAP, then the node is identified as in the gap closure region. If a fluid element has all its vertices in the gap closure region, the element is identified as a gap closure element. Note that for FSI cases, the gap closure detection alghorithm works independently from the mechanical solver's contact algorithms (see *CONTACT). The onset of contact between two surfaces in the mechanical solver does not automatically activate gap closure treatment in ICFD.

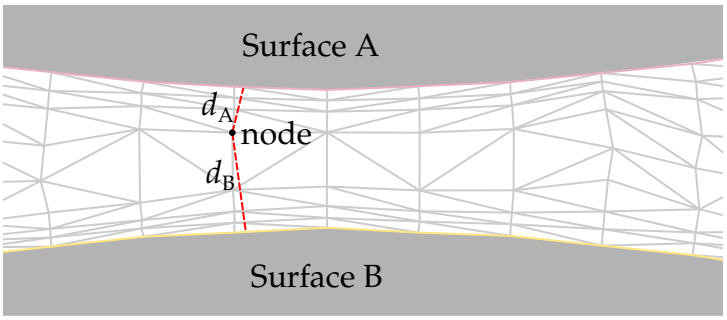


Figure 7-7. Threshold distance condition for gap closure detection

2. **Gap closure detection additionally considering threshold angle.** If ANGLE is nonzero, the gap closure detection algorithm addition considers a threshold angle condition. Consider a fluid node. Form a vector $\mathbf{V}_A$ from the closest point on surface A to the node. Similarly, form a vector $\mathbf{V}_B$ from the closest point on surface B to the node. See Figure 7-8 for illustration. For a node to be identified as in the gap closure region, not only does the threshold distance condition (see Remark 1) need to be satisfied, but also the angle θ between $\mathbf{V}_A$ and $\mathbf{V}_B$ must be larger than the threshold angle. The threshold angle offers additional control over the gap closure region near corners.

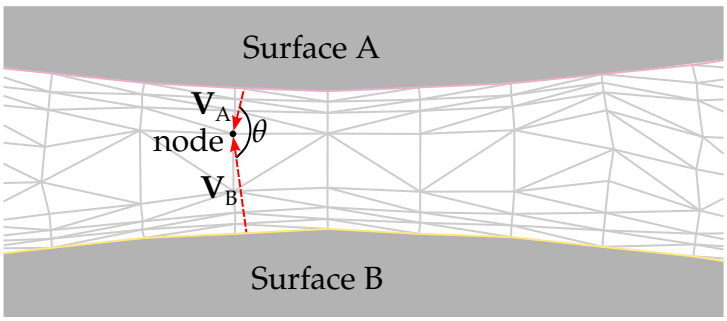


Figure 7-8. Threshold angle condition for gap closure detection

3. **Surface part IDs.** There is no gap self-detection on a given surface part ID. Thus, you must make sure to separate your surfaces into appropriate surface parts to ensure that gap detection is triggered as desired.

4. **Restricting the surfaces for gap search.** To apply gap detection to smaller regions on the surfaces instead of the entire surfaces, use *ICFD_DEFINE_-GAPRESTRIC.

*ICFD_DEFINE_GAPRESTRIC

Purpose: Define a geometric restriction on the surfaces for the gap search when using the gap closure feature (see *ICFD_DEFINE_GAPCLOSURE and *ICFD_CONTROL_GAP). When this keyword is present, gap search is restricted to regions on the surfaces defined by the keyword instead of the entire surfaces defined by *ICFD_DEFINE_GAPCLOSURE or *ICFD_CONTROL_GAP.

Card 1	1	2	3	4	5	6	7	8
Variable	RID				ISHAPE	R	PTID1	PTID2
Type	I				I	F	I	I
Default	none				none	none	none	none

VARIABLE	DESCRIPTION
RID	Restriction zone ID
ISHAPE	Shape to which the gap search is applied: EQ.1: Box EQ.2: Cylinder EQ.3: Sphere
R	Radius of the cylinder or sphere if ISHAPE = 2 or 3. This field is ignored if ISHAPE = 1.
PTID1	Point ID (see *ICFD_DEFINE_POINT) whose meaning depends on ISHAPE: ISHAPE.EQ.1: Minimum coordinates of the box ISHAPE.EQ.2: Tail point of the cylinder ISHAPE.EQ.3: Origin of the sphere
PTID2	Point ID (see *ICFD_DEFINE_POINT) whose meaning depends on ISHAPE: ISHAPE.EQ.1: Maximum coordinates of the box ISHAPE.EQ.2: Head point of the cylinder

*ICFD_DEFINE_HEATSOURCE

Purpose: Define a volumetric heat source for evaluating the heat equation.

Card 1	1	2	3	4	5	6	7	8
Variable	HSID	LCID	ISHAPE	R	PTID1	PTID2		
Type	I	I	I	F	I	I		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
HSID	Heat source ID
LCID	Load curve ID specifying the evolution of the heat source term function of time for the *X*, *Y*, and *Z* degrees-of-freedom (see *DEFINE_CURVE,*DEFINE_CURVE_FUNCTION, or *DEFINE_FUNCTION). If using a *DEFINE_FUNCTION, the following arguments are allowed: f(x, y, z, vx, vy, vz, temp, pres, time).
ISHAPE	Shape of the volumetric heat source: EQ.1: Box shape EQ.2: Cylinder shape EQ.3: Sphere shape
R	Radius of the cylinder if ISHAPE = 2 or radius of the sphere if ISHAPE = 3
PTID1	Point ID (see *ICFD_DEFINE_POINT) whose meaning depends on ISHAPE: ISHAPE.EQ.1: Minimum coordinates of the box ISHAPE.EQ.2: Point in the center of one base of the cylinder ISHAPE.EQ.3: Center of the sphere
PTID2	Point ID (see *ICFD_DEFINE_POINT) whose meaning depends on ISHAPE:

VARIABLE	DESCRIPTION
	ISHAPE.EQ.1: Maximum coordinates of the box
	ISHAPE.EQ.2: Point in the center of the other base of the cylinder

.

*ICFD_DEFINE_HEMOLYSIS_INDEX

Purpose: Compute and track the hemolysis index and scalar shear stresses.

Card 1	1	2	3	4	5	6	7	8
Variable	MASSDIF	C	ALPHA	BETA	PRESSTO			
Type	F	F	F	F	I			
Default	none	3.458E-6	0.2777	2.0639	0			

VARIABLE	DESCRIPTION
MASSDIFF	Mass diffusion for the hemolysis index transport problem:
C	Constant in the hemolysis index model, C
ALPHA	Exponent in the hemolysis index model, α
BETA	Exponent in the hemolysis index model, β
PRESSTO	Flag to impose a hemolysis index of 0 on boundaries with pressure boundary conditions: EQ.0: Interpolate the hemolysis index on the boundaries from the interior of the domain EQ.1: Set the hemolysis index to 0 on boundaries with pressure boundary conditions

*ICFD_DEFINE_POROUS_REGION

Purpose: Define moving isotropic porous regions. The permeability can be a constant value or defined through a time-dependent load curve or *DEFINE_FUNCTION. Alternatively (using SENSFLG ≠ 0), this keyword optionally enables pressure sensors to specify the permeability as a function of the pressure drop across the porous domain length (see Remark 1). *ICFD_DEFINE_POINTs determine the locations of the sensors. The first sensor is upstream of the region, and the second sensor downstream. The primary application of this feature is modeling the closing and opening of pressure-driven valves without the need for more complex moving parts (like leaflets). If points PTID1 and PTID2 (and sensors SENSID1 and SENSID2) are attached to a moving part, the points move accordingly.

Card 1	1	2	3	4	5	6	7	8
Variable	PRID	ISHAPE	R	PTID1	PTID2	PERM	LCID	SENSFLG
Type	I	I	F	I	I	F	I	I
Default	none	none	none	none	none	none	0	0

Sensor Card. Define only if SENSFLG ≠ 0 on Card 1.

Card 2	1	2	3	4	5	6	7	8
Variable	SENSID1	SENSID2	KMIN	KMAX	DLTAPREF	SLOPE		
Type	I	I	F	F	F	F		
Default	none	none	none	none	0	0		

VARIABLE	DESCRIPTION
PRID	Porous region ID
ISHAPE	Shape of the volumetric porous region: EQ.1: Box shape, EQ.2: Cylinder shape, EQ.3: Sphere shape,

VARIABLE	DESCRIPTION
R	Radius of the cylinder if ISHAPE = 2. Radius of the sphere if ISHAPE = 3.
PTID1	Point ID (see *ICFD_DEFINE_POINT) with meaning depending on ISHAPE: ISHAPE.EQ.1: Minimum coordinates of the box ISHAPE.EQ.2: Tail point of the cylinder ISHAPE.EQ.3: Center of the sphere
PTID2	Point ID with meaning depending on ISHAPE (ignored for ISHAPE = 3): ISHAPE.EQ.1: Maximum coordinates of the box ISHAPE.EQ.2: Head point of the cylinder.
PERM	Isotropic permeability, k, of the region. PERM is ignored if LCID ≠ 0 or SENSFLG = 1.
LCID	Load curve ID specifying the evolution of the permeability as a function of time for the X, Y, and Z degrees of freedom; see *DEFINE_CURVE, *DEFINE_CURVE_FUNCTION, or *DEFINE_FUNCTION. If a using a *DEFINE_FUNCTION, the following parameters are allowed: f(x, y, z, vx, vy, vz, temp, pres, time). Ignored if SENSFLG = 1.
SENSFLG	Sensor flag determining whether the permeability is a function of the pressure drop computed between the sensors: EQ.0: Not a function of pressure drop EQ.1: Function of a pressure drop between sensors. See Remark 1.
SENSID1	Sensor (*ICFD_DEFINE_POINT ID) upstream of the porous region
SENSID2	Sensor (ICFD_DEFINE_POINT ID) downstream of the porous region
KMIN	Minimum permeability for the valve (closed position), k_{min}
KMAX	Maximum permeability for the valve (open position), k_{max}
DLTAPREF	Pressure drop that opens/closes the valve, Δp_{ref}

VARIABLE	DESCRIPTION
SLOPE	Slope of the transition from k_{max} to k_{min}, m. See Remark 1.

Remarks:

1. **Permeability as a function of pressure drop.** When SENSFLG = 1, the pressure drop, Δp, between two sensors determines the permeability, k:

$$k = \frac{(k_{max} - k_{min})}{2} \times \left(1 + \tanh\left(\frac{(\Delta p - \Delta p_{ref})}{m}\right)\right) + k_{min}$$

This feature is intended for using the porous region to model the opening and closing of a valve. At a given Δp_{ref}, the permeability switches from k_{min} to k_{max} as Δp increases, modeling opening the valve. The valve closes when the opposite occurs. See Figure 7-9.

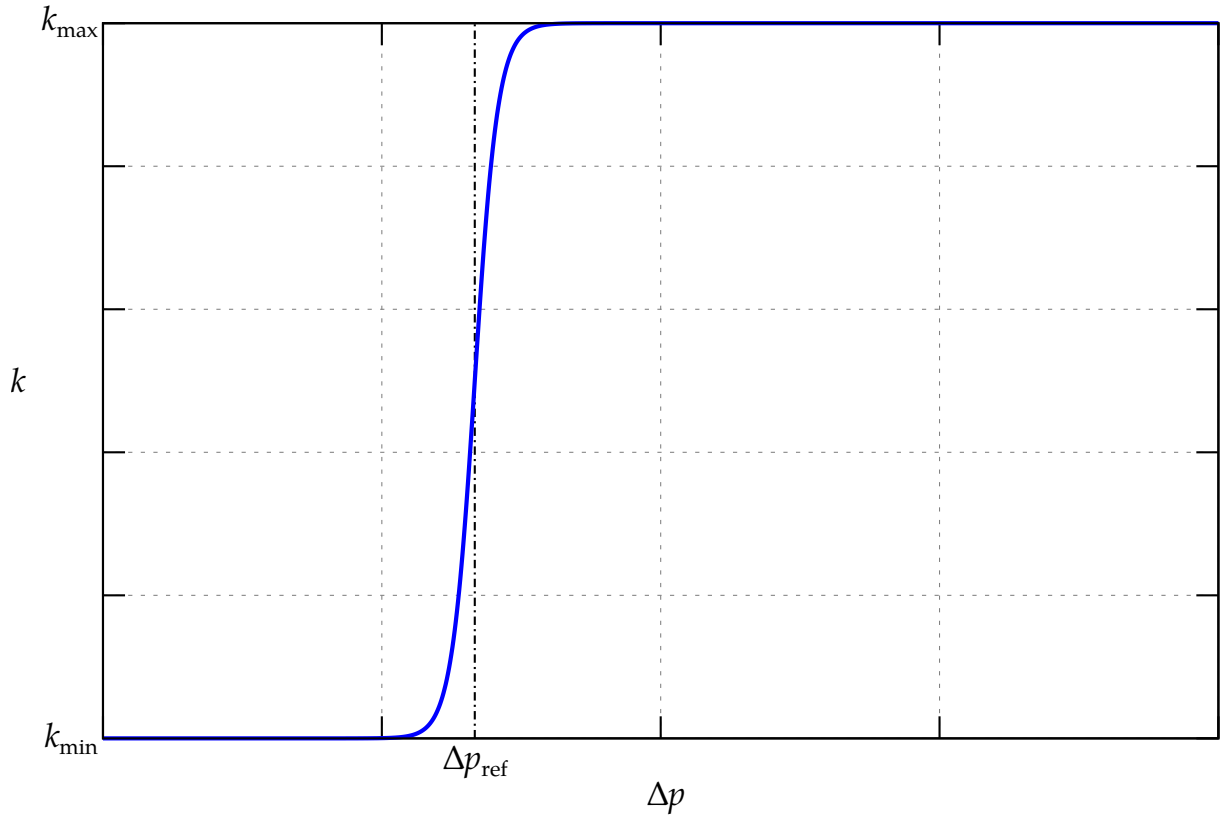


Figure 7-9. Pressure-driven valve law

*ICFD_DEFINE_RESIDENCETIMESOURCE

Purpose: Specify a volumetric residence time (RT) source.

Card 1	1	2	3	4	5	6	7	8
Variable	RTSID	ISHAPE	R	PTID1	PTID2	MASSDIFF	DEATHT	IRTOPBC
Type	I	I	F	I	I	F	F	I
Default	none	none	none	none	none	10^{-6}	↓	0

N-Planes RT Source Card. If ISHAPE = 4, include as many of this card as desired, one for each plane. The next keyword ("*") card terminates this input.

Card 2	1	2	3	4	5	6	7	8
Variable	N-PTID	N-NORID						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
RTSID	RT source ID
ISHAPE	Shape of the volumetric RT source: EQ.1: Box EQ.2: Cylinder EQ.3: Sphere EQ.4: N-planes (see Remark 1)
R	Radius of the cylinder if ISHAPE = 2 or radius of the sphere if ISHAPE = 3
PTID1	ID of a point (see *ICFD_DEFINE_POINT) giving the minimum coordinate of the box if ISHAPE = 1, the tail point for the cylinder if ISHAPE = 2, or the center of the sphere if ISHAPE = 3

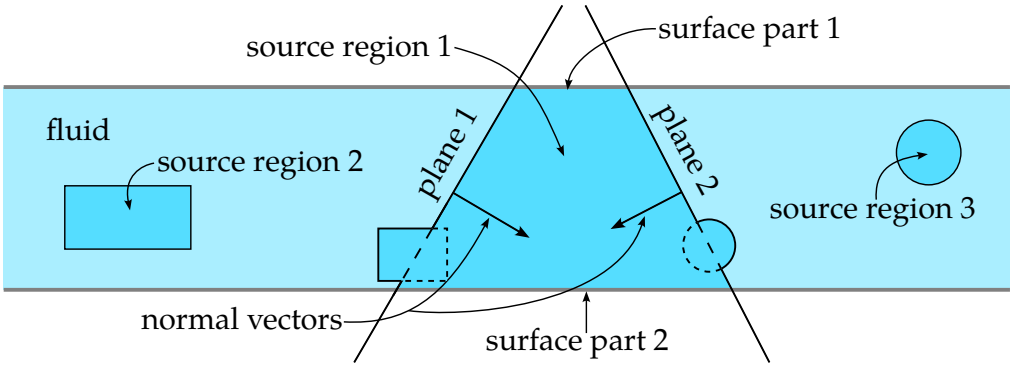


Figure 7-10. Illustration of different RT source regions. The combination of N-planes (two planes), two other RT sources (a box and sphere), and domain boundaries defines source region 1. Specifying source region 1 requires three instantiations of *ICFD_DEFINE_RESIDENCETIMESOURCE: one for the two planes, one for the box, and one for the sphere. The surface parts are automatically considered; thus, planes do not need to be specified on those surfaces. Defining source region 2 and source region 3 requires an instantiation of *ICFD_DEFINE_RESIDENCETIMESOURCE for each of those regions.

VARIABLE	DESCRIPTION
PTID2	ID of a point giving the maximum coordinate of the box if ISHAPE = 1 or the head point of the cylinder if ISHAPE = 2
MASSDIFF	Mass diffusion for the transport equation. See Remark 2.
DEATHT	End time for the source EQ.0.0: End time of the simulation.
IRT0PBC	Flag for which prescribed boundaries RT = 0 is imposed: EQ.0: Imposed only on boundaries with prescribed velocity EQ.1: Imposed on boundaries with either prescribed velocity or prescribed pressure.
N-PTID	ID of a point lying on the source plane for ISHAPE = 4
N-NORID	ID of a point whose coordinate components specify the normal vector of the plane for ISHAPE = 4

Remarks:

1. **N-planes RT source shape.** If ISHAPE = 4, the residence time source (seed) region is defined through a set of one or more infinite planes (N-planes). The

resulting region must be a concave region inside the fluid domain. The N-plane shape can be combined with other source shapes, and it works in the sense of nesting them and/or combining them through set union operations (set intersection operations are not supported). The domain boundary parts are natural surfaces defining these injection regions, and they do not need to be added to the list of planes. The source shapes can have different birth and end times, so the combined region can change in time. See Figure 7-10 for an illustration.

Each instantiation of Card 2 defines a plane using two point IDs (see *ICFD_DEFINE_POINT). The first point is a point lying on the i^{th} plane. The coordinate components of the second point are treated as a vector giving the normal vector of the plane. The normal vector must point to the interior of the region.

Figure 7-10 illustrates using N-planes with other sources and domain boundaries. This situation requires five instantiations of *ICFD_DEFINE_RESIDENCETIMESOURCE. Three of the instantiations are for source region 1 (for the planes, box, and sphere), one for source region 2, and one for source region 3. The surface parts making up source region 1 are automatically considered; thus, planes do not need to be specified on those surfaces. Keyword input for this scenario could look as follows:

```
*ICFD_DEFINE_RESIDENCETIMESOURCE
$ These are planes 1 and 2 in source region 1
$    RTSID    ISHAPE         R     PTID1     PTID2  MASSDIFF    DEATHT    IRT0BC
         1         4
$ plane 1
$   N-PTID   N-NORID
        10        11
$ plane 2
$   N-PTID   N-NORID
        20        21
*ICFD_DEFINE_RESIDENCETIMESOURCE
$ This is the box in source region 1
$    RTSID    ISHAPE         R     PTID1     PTID2  MASSDIFF    DEATHT    IRT0BC
         2         1                30        31
*ICFD_DEFINE_RESIDENCETIMESOURCE
$ This is the sphere in source region 1
$    RTSID    ISHAPE         R     PTID1     PTID2  MASSDIFF    DEATHT    IRT0BC
         3         3       5.0        20
*ICFD_DEFINE_RESIDENCETIMESOURCE
$ This is the box specifying source region 2
$    RTSID    ISHAPE         R     PTID1     PTID2  MASSDIFF    DEATHT    IRT0BC
         4         1                40        41
*ICFD_DEFINE_RESIDENCETIMESOURCE
$ This is the cylinder specifying source region 3
$    RTSID    ISHAPE         R     PTID1     PTID2  MASSDIFF    DEATHT    IRT0BC
         5         2       7.0        50        51
```

2. **Mass diffusion.** The same MASSDIFF parameter must be used when defining multiple *ICFD_DEFINE_RESIDENCETIMESOURCE instantiations in an input deck.

*ICFD_DEFINE_SOURCE

Purpose: Define a volumetric external force for the momentum equation solve.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	LCIDX	LCIDY	LCIDZ	SHAPE	R	PTID1	PTID2
Type	I	I	I	I	I	F	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
SID	Source ID
LCIDX/Y/Z	Load curve IDs specifying the evolution of the volumetric force as a function of time for the three global components
SHAPE	Shape to which the volumetric force is applied: EQ.1: Box EQ.2: Cylinder EQ.3: Sphere
R	Radius of the cylinder or sphere if SHAPE = 2 or 3
PTID1	Point ID (see *ICFD_DEFINE_POINT) whose meaning depends on SHAPE: SHAPE.EQ.1: Minimum coordinates of the box SHAPE.EQ.2: Tail point of the cylinder SHAPE.EQ.3: Origin of the sphere
PTID2	Point ID (see *ICFD_DEFINE_POINT) whose meaning depends on SHAPE: SHAPE.EQ.1: Maximum coordinates of the box SHAPE.EQ.2: Head point of the cylinder

*ICFD_DEFINE_SPTRANSPSOURCE

Purpose: Specify a volumetric species source for the species transport solver.

Card 1	1	2	3	4	5	6	7	8
Variable	SPTRSID	LCID	ISHAPE	R	PTID1	PTID2		SPIND
Type	I	I	I	F	I	I		I
Default	none	none	none	none	none	none		none

VARIABLE	DESCRIPTION
SPTRSID	Species transport source ID
LCID	Load curve ID specifying the evolution of the species source term as a function of time for the X, Y, and Z degrees of freedom (see *DEFINE_CURVE, *DEFINE_CURVE_FUNCTION, and *DEFINE_FUNCTION). If a *DEFINE_FUNCTION is used, the following parameters are allowed: f(x, y,z, vx, vy, vz, temp, pres, time).
ISHAPE	Shape of the volumetric species source: EQ.1: Box EQ.2: Cylinder EQ.3: Sphere.
R	Radius of the cylinder if ISHAPE = 2 or radius of the sphere if ISHAPE = 3.
PTID1	ID of a point (see *ICFD_DEFINE_POINT) giving the minimum coordinate of the box if ISHAPE = 1, the tail point for the cylinder if ISHAPE = 2, or the origin of the sphere if ISHAPE = 3.
PTID2	ID of a point giving the maximum coordinate of the box if ISHAPE = 1 or the head point of the cylinder if ISHAPE = 2.
SPIND	Index of the species for which the source term applies. For instance, if the source term is for the second species in a multi-species transport problem, this index is 2.

*ICFD_DEFINE_TRANSFORM

Purpose: Enable applying translations, rotations, and scaling on the initial surface mesh before generating the volume mesh. This feature makes possible quick changes (such as when switching between unit systems) on the geometry without having to manually edit the case.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	TRX	TRY	TRZ	SF			
Type	I	F	F	F	F			
Default	none	0.0	0.0	0.0	1.0			

Optional card

Card 2	1	2	3	4	5	6	7	8
Variable	ORX	ORY	ORZ	NX	NY	NZ	ANGLE	
Type	F	F	F	F	F	F	F	
Default	0.	0.	0.	0.	0.	0.	0.	

VARIABLE	DESCRIPTION
PID	Surface PID on which to apply the transformation. EQ.0: Apply the transformation to the entire model.
TRX/TRY/TRZ	Translation offset in the global X, Y, and Z-directions
SF	Scale factor to be applied on the X, Y, and Z coordinates
ORX/ORY/ORZ	Origin point of the rotation
NX/NY/NZ	Normal for the rotation
ANGLE	Angle of rotation (in degrees)

*ICFD_DEFINE_TURBSOURCE

Purpose: Define an external source for the RANS turbulence equations.

Card 1	1	2	3	4	5	6	7	8
Variable	TSID	LCIDK	LCIDEP	LCIDNU	ISHAPE	R	PTID1	PTID2
Type	I	I	I	I	I	F	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
TSID	Turbulent external source ID
LCIDK	Load curve ID specifying the evolution of the external source term function of time for the turbulent kinetic energy, k, equation (see *DEFINE_CURVE,*DEFINE_CURVE_FUNCTION, or *DEFINE_FUNCTION). If using a *DEFINE_FUNCTION, the following arguments are allowed: f(x, y, z, vx, vy, vz, temp, pres, time).
LCIDEP	Load curve ID specifying the evolution of the external source term function of time for the turbulent diffusion, ε, or specific rate of dissipation, w, equation (see *DEFINE_CURVE,*DEFINE_CURVE_FUNCTION, or *DEFINE_FUNCTION). If using a *DEFINE_FUNCTION, the following arguments are allowed: f(x, y, z, vx, vy, vz, temp, pres, time).
LCIDNU	Load curve ID specifying the evolution of the external source term function of time for the kinematic eddy turbulent viscosity equation used in the Spalart-Allmaras model (see *DEFINE_CURVE,*DEFINE_CURVE_FUNCTION, or *DEFINE_FUNCTION). If using a *DEFINE_FUNCTION, the following arguments are allowed: f(x, y, z, vx, vy, vz, temp, pres, time).
ISHAPE	Shape of the external source: EQ.1: Box shape EQ.2: Cylinder shape

VARIABLE	DESCRIPTION
	EQ.3: Sphere shape
R	Radius of the cylinder if ISHAPE = 2 or radius of the sphere if ISHAPE = 3
PTID1	Point ID (see *ICFD_DEFINE_POINT) whose meaning depends on ISHAPE: ISHAPE.EQ.1: Minimum coordinates of the box ISHAPE.EQ.2: Point in the center of one base of the cylinder ISHAPE.EQ.3: Center of the sphere
PTID2	Point ID (see *ICFD_DEFINE_POINT) whose meaning depends on ISHAPE: ISHAPE.EQ.1: Maximum coordinates of the box ISHAPE.EQ.2: Point in the center of the other base of the cylinder

.

*ICFD_DEFINE_POINT

Purpose: This keyword defines a point in space that could be used for multiple purposes.

Card 1	1	2	3	4	5	6	7	8
Variable	POID	X	Y	Z	CONSTPID			
Type	I	F	F	F	I			
Default	none	none	none	none	none			

Tranlational Velocity Card. This card is optional. It is for defining a translational velocity for the point.

Card 2	1	2	3	4	5	6	7	8
Variable	LCIDX	LCIDY	LCIDZ					
Type	I	I	I					
Default	0	0	0					

Angular Velocity Card. This card is optional. It is for defining an angular velocity for the point and the rotation axis.

Card 2	1	2	3	4	5	6	7	8
Variable	LCIDW	XT	YT	ZT	XH	YH	ZH	
Type	I	F	F	F	F	F	F	
Default	0	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
POID	Point ID
X, Y, Z	x, y , z coordinates of the point

VARIABLE	DESCRIPTION
CONSTPID	Surface part ID to which the point is constrained. If the selected surface moves, the localization of the point also updates accordingly.
LCIDX, LCIDY, LCIDZ	Load curve IDs giving the components of translational velocity for the point as a function of time. This translational velocity is optional.
LCIDW	Load curve ID giving the angular velocity of the point as a function of time. This angular velocity is optional.
XT, YT, ZT	Coordinates of the tail point for specifying the rotation axis. Only define if LCIDW is nonzero.
XH, YH, ZH	Coordinates of the head point for specifying the rotation axis. Only define if LCIDW is nonzero.

.

*ICFD_DEFINE_NONINERTIAL

Purpose: Define a non-inertial reference frame to avoid heavy mesh distortions and remeshing associated with large-scale rotations. This frame helps when modeling spinning cylinders, wind turbines, and turbomachinery.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	W1	W2	W3	R	PTID	L	LCID	RELV
Type	F	F	F	F	I	F	I	I
Default	none	none	none	none	none	none	none	0

VARIABLE	DESCRIPTION
W1, W2, W3	Rotational velocity, ω, along the X, Y, and Z-axes
R	Radius of the rotating reference frame. GT.0.0: Constant value LT.0.0: |R| refers to a *DEFINE_FUNCTION ID. The function supports the following arguments: f(x, y, z, vx, vy, vz, temp, pres, time).

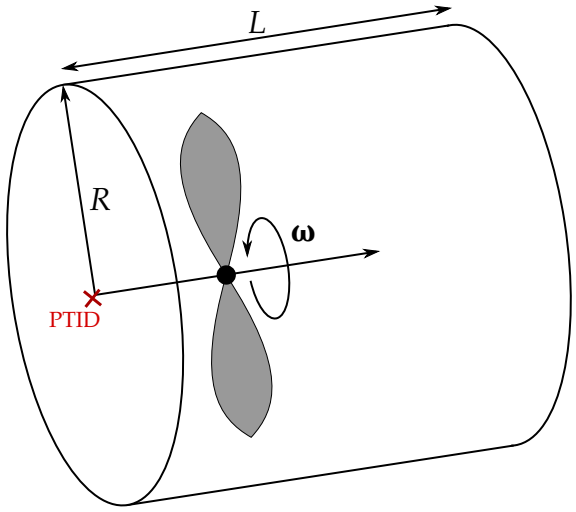


Figure 7-11. Example of using a non-inertial reference frame

VARIABLE	DESCRIPTION
PTID	Starting point ID for the reference frame (See *ICFD_DEFINE_-POINT)
L	Length of the rotating reference frame
LCID	Load curve for scaling factor of ω.
	GT.0: Load curve ID (see *DEFINE_CURVE) for the curve giving the scale factor as a function of time
	LT.0: |LCID| refers to a *DEFINE_FUNCTION ID. The function supports the following arguments: f(x, y, z, vx, vy, vz, temp, pres, time).
RELV	Velocities computed and displayed:
	EQ.0: Relative velocity, only the non-rotating components of the velocity are used and displayed.
	EQ.1: Absolute velocity. All the components of the velocity are used. Useful in cases where several or at least one non-inertial reference frame is combined with an inertial "classical" reference frame.

*ICFD_DEFINE_WAVE_DAMPING

Purpose: This keyword defines a damping zone for free surface waves.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	NID	L	F1	F2	N	LCID	
Type	I	I	F	F	F	I	I	
Default	none	none		10	10	1	none	

VARIABLE	DESCRIPTION
PID	Point ID defining the start of the damping layer.
NID	Normal ID defined using ICFD_DEFINE_POINT and pointing to the outgoing direction of the damping layer.
L	Length of damping layer. If no is value specified, the damping layer will have a length corresponding to five element lengths.
F1/F2	Linear and quadratic damping factor terms.
N	Damping term factor.
LCID	Load curve ID acting as temporal scale factor on damping term.

Remarks:

1. The damping is achieved by adding a source term to the momentum equations :

$$s^d = w\ (f_1 + f_2|u|)\ u$$

with w the weight function :

$$w = \frac{e^\gamma - 1}{e - 1}$$

and γ the blending function which allows a smooth insertion of the source term in the damping layer :

$$\gamma = \left(\frac{x - x_{sd}}{x_{ed} - x_{sd}}\right)^n$$

x_{sd} and x_{ed} representing the start and end coordinates of the damping zone.

*ICFD_INITIAL

Purpose: Simple initialization of velocity and temperature within a volume.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	Vx	Vy	Vz	T	P		DFUNC
Type	I	F	F	F	F	F		I
Default	none	none	none	none	none	none		0

VARIABLE	DESCRIPTION
PID	Part ID for the volume elements or the surface elements where the values are initialized (see *ICFD_PART_VOL and *ICFD_PART). PID = 0 to assign the initial condition to all nodes at once.
Vx	x coordinate for the velocity.
Vy	y coordinate for the velocity.
Vz	z coordinate for the velocity.
T	Initial temperature.
P	Initial Pressure.
DFUNC	Option to define initial conditions using *DEFINE_FUNCTION EQ.0: Turned off. EQ.1: Turned on. All previous flags for initial velocity, pressure and temperature now refer to *DEFINE_FUNCTION IDs. The following parameters are allowed : $f(x,y,z)$, allowing to define initial profiles function of coordinates.

*ICFD_INITIAL_BIN

Purpose: Initialize the current simulation with binary file results from a previous ICFD simulation. See Remark 1.

Card 1	1	2	3	4	5	6	7	8
Variable	IREADLV	IFRAME	ITEMP	IMESH				
Type	I	I	I	I				
Default	none	0	0	0				

File Name of the Binary File Card. Include this card if IREADLV = 2.

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A							
Default	none							

VARIABLE	DESCRIPTION
IREADLV	Flag to determine if the binary file should be read. EQ.0: Off. This keyword is ignored. EQ.1: On. The simulation is initialized by reading binary file icfdbintot. EQ.2: The simulation is initialized by reading the binary file with the name given on the second line.
IFRAME	Frame from the binary file that is used for initialization. If no specific frame is provided, the last frame from the binary file is used.
ITEMP	Flag to determine if the temperature field should be read from a binary file and used at initialization: EQ.0: On. If a temperature field is found in the binary file, it will be used for temperature initialization.

VARIABLE	DESCRIPTION
	EQ.1: Off. Even if a temperature field is present in the binary file, it will be ignored.
IMESH	Flag to determine if the fluid mesh should also be read from the binary file: EQ.0: Off. The usual *MESH keywords are used to generate or read the mesh. GE.1: Frame ID in the binary file from which the mesh should be read. Typically, you will set IMESH to 1 to read the initial mesh from the binary file. All *MESH keywords must be removed from the input deck.
FILENAME	Name of the binary file if IREADLV = 2.

Remarks:

1. **ICFD initialization.** This keyword is meant to be used with *ICFD_DATABASE_BIN for two step simulation workflows. For example, a first ICFD analysis that uses the steady state solver can output results in a binary file using *ICFD_DATABASE_BIN. The results are then used as an initial condition in the next simulation which is often transient, FSI or conjugate heat transfer by using this keyword. One advantage of this approach is that the meshes between the two simulations need not be the same because *ICFD_INITIAL_BIN interpolates results from the binary file and applies them on its own current mesh. The default name for this binary output is icfdbintot but the name can be specified with FILENAME if IREADLV = 2.

*ICFD_INITIAL_LEVELSET

Purpose: Define an initial level set surface instead of a multi-fluid domain (replaces the need for *MESH_INTERF).

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	STYPE	NX	NY	NZ	X	Y	Z	INVERT
Type	I	F	F	F	F	F	F	I
Default	none	none	none	none	none	none	none	0

VARIABLE	DESCRIPTION
STYPE	Initial surface type: EQ.0/1: Generated by a section plane. EQ.2: Generated by a box. See Remark 1. EQ.3: Generated by a sphere. EQ.4: Generated by a cylinder. EQ.5: Given by a function provided with *DEFINE_FUNCTION
NX, NY, NZ	X, Y. and Z components of the section plane normal if STYPE = 1. Minimum coordinates of the box, P_{min}, if STYPE = 2. NX is the sphere/cylinder radius if STYPE = 3 and STYPE = 4. NY is the cylinder length if STYPE = 4. NZ is the global axis if STYPE = 4 (NZ = 1, 2, or 3 means X-axis, Y-axis, or Z-axis, respectively). IF STYPE = 5, NX is the ID of a *DEFINE_FUNCTION with the following allowed arguments: f(x, y, z).
X, Y, Z	X, Y, and Z components of the section plane origin point if STYPE = 1 and 4. Maximum coordinates of the box, P_{max}, if STYPE = 2. Coordinates of the sphere origin point if STYPE = 3.
INVERT	Inversion of initial level set: EQ.0: No inversion. Positive level set values are assigned to nodes contained within the volume defined by STYPE. EQ.1: The sign of the initial level set values is reversed.

Remarks:

1. **Box adjacent to fluid boundaries.** When STYPE = 2 is used and the box is adjacent to the fluid boundaries, such as during a dam break simulation, the distance from any point in the fluid to the fluid boundary must remain smaller than the distance to the defined box. Therefore, the P_{min} coordinates need to be defined far outside the initial fluid domain.

2. **Multiple keyword definitions.** Multiple definitions of this keyword are possible, but the different shapes generated must not intersect for a correct initialization.

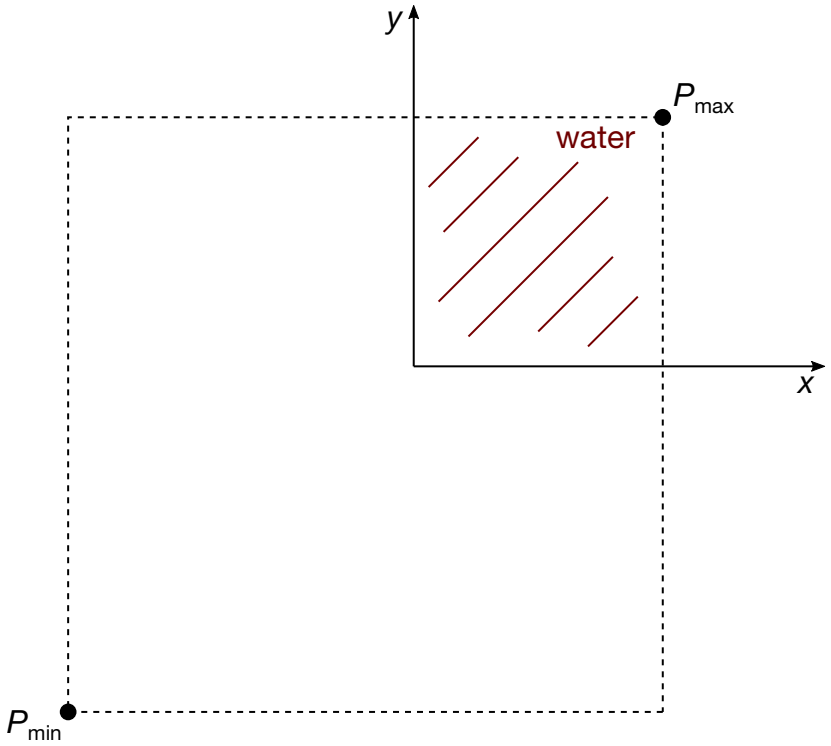


Figure 7-12. 2D Dam breaking example with initial levelset surface defined using STYPE = 2. P_{min} is defined sufficiently far away from the fluid surface boundaries.

*ICFD_INITIAL_SPTRANSP

Purpose: Initialize the concentration of the species being transported within a volume.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	0							

Card 2	1	2	3	4	5	6	7	8
Variable	DFID-1	DFID-2	DFID-3	DFID-4	DFID-5	DFID-6	DFID-7	DFID-8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
PID	Part ID for the volume elements or surface elements where the values are initialized. EQ.0: Assign the initial condition to all nodes at once
DFID-*i*	*DEFINE_FUNCTION ID defining the initialization for the concentration of the i^{th} species at $t = 0$. The following parameters are allowed: f(x, y, z).

*ICFD_INITIAL_TEMPNODE

Purpose: Allow the solver to initialize the temperature at individual nodes.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	NID	TEMP						
Type	I	F						
Default	none	none						

VARIABLE	DESCRIPTION
NID	Internal ICFD node ID
TEMP	Initial temperature value

*ICFD_INITIAL_TURBULENCE

Purpose: Modify the default initial values of the turbulence quantities for a RANS turbulence model.

Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	I	R	K	EW			
Type	I	F	F	F	F			
Default	none	none	none	optional	optional			

VARIABLE	DESCRIPTION
PID	Part ID for the volume elements or the surface elements where the values are initialized (see *ICFD_PART_VOL and *ICFD_PART). Set PID to 0 to assign the initial condition to all nodes at once.
I	Initial turbulence intensity, I
R	Initial turbulence viscosity to laminar viscosity ratio ($r = \mu_{\text{turb}}/\mu$).
K	Initial kinetic energy. When defined, it replaces the choice of I. LT.0.0: |K| refers to a *DEFINE_FUNCTION ID. The following parameters are allowed: $f(x,y,z)$, allowing you to define initial profiles as a function of coordinates.
EW	Initial turbulence specific dissipation rate or dissipation rate depending on the choice of turbulence model. When defined, it replaces the choice of R. LT.0.0: |EW| refers to a *DEFINE_FUNCTION ID. The following parameters are allowed: $f(x,y,z)$, allowing you to define initial profiles as a function of coordinates.

Remarks:

1. **Default Initial Conditions.** If no initial conditions have been assigned to a specific PID, the solver will automatically pick I = 0.05 (5%) and R = 10000.

*ICFD_INITIAL_VISCOELASTIC

Purpose: Initialize the conformation tensor components.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	FIDXX	FIDYY	FIDZZ	FIDXY	FIDXZ	FIDYZ	
Type	I	I	I	I	I	I	I	
Default	0	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
PID	Part ID for the volume elements or the surface elements where the conformation tensor components are initialized (see *ICFD_PART_VOL and *ICFD_PART). PID set to 0 means assigning the values to all nodes at once.
FID*ij*	Function IDs (see *DEFINE_FUNCTION) for the conformation tensor components. The components can spatially vary. The following parameters are allowed: f(x, y, z).

*ICFD_INITIAL_WATERVAPOR

Purpose: Initialize the water vapor mass fraction on a part.

Include as many sets of Cards 1 and 2 as needed. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	MFRAC	MFRACU						
Type	F	I						
Default	none	0						

VARIABLE	DESCRIPTION
PID	Part ID or part volume ID with the water vapor mass fraction condition
MFRAC	Water vapor mass fraction value. If MFRACU = 0, this value is the mass of water vapor in kg divided by the mass of dry air in kg. If MFRACU = 1, this value is the relative humidity and is in the range of [0,1].
MFRACU	Water vapor mass fraction type: EQ.0: MFRAC is the mass of the water vapor in kg divided by the mass of the dry air in kg. EQ.1: MFRAC is the relative humidity with a range of [0,1].

***ICFD_MAT**_*{OPTION}*

Available options include:

<BLANK>

TITLE

Purpose: Specify physical properties for the fluid material.

Fluid Material Card Sets:

The Material Fluid Parameters Card is required. If a second card is given, it must be a Thermal Fluid Parameters Card. If the fluid thermal properties are not needed, the second card can be a blank card. With the third card, you can associate the fluid material to a non-Newtonian model, to a porous media model, to a species transport model, and/or to a viscoelastic model (see *ICFD_MODEL_NONNEWT, *ICFD_MODEL_POROUS, *ICFD_MODEL_SPECIES_TRANSPORT, and *ICFD_MODEL_VISCOELASTIC). If FLG = 2, the Isentropic Fluid Parameters Card (Card 4) is required. Cards 2 and 3 must precede Card 4; include blank or zero-filled cards in these positions when their properties and models are not used. Omit Card 4 for other values of FLG.

Material Fluid Parameters Card.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	FLG	RO	VIS	ST	STSFLCID	CA	
Type	I	I	F	F	F	I	F	
Default	none	none	0.	0.	0.	none	0.	

Thermal Fluid Parameters Card. Only to be defined if the thermal problem is solved. Otherwise include a blank card.

Card 2	1	2	3	4	5	6	7	8
Variable	HC	TC	BETA	PRT	HCSFLCID	TCSFLCID		
Type	F	F	F	F	I	I		
Default	0.	0.	0.	0.85	none	none		

Additional fluid models. Only to be defined if the fluid is non-Newtonian, there is a porous media, and/or a species is being transported.

Card 3	1	2	3	4	5	6	7	8
Variable	NNMOID	PMMOID		SPTRID		VID		
Type	I	I		I		I		
Default	optional	optional		optional		optional		

Compressible Isentropic Fluid Parameters Card. Include this card if FLG = 2. Cards 2 and 3 must precede this card.

Card 4	1	2	3	4	5	6	7	8
Variable	GAMMA	THETA0	RGAS					
Type	F	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
MID	Material ID
FLG	Flag selecting the fluid material model: EQ.0: Vacuum (free surface problems only) EQ.1: Fully incompressible fluid. EQ.2: Isentropic compressible fluid. Card 4 is required. See Remarks 1 through 3.
RO	Flow density if FLG ≠ 2. If FLG = 2, RO is the reference density at the operating pressure and reference temperature THETA0. A positive value is required. The local density is updated from the isentropic equation of state; see Remark 2.
VIS	Dynamic viscosity
ST	Surface tension coefficient

VARIABLE	DESCRIPTION
STSFLCID	Load curve ID for scale factor applied on ST as a function of time. See *DEFINE_CURVE, *DEFINE_CURVE_FUNCTION, or *DEFINE_FUNCTION. If a *DEFINE_FUNCTION is used, the following parameters are allowed: f(x, y, z, vx, vy, vz, temp, pres, time).
CA	Contact angle
HC	Heat capacity
TC	Thermal conductivity
BETA	Thermal expansion coefficient used in the Boussinesq approximation for buoyancy
PRT	Turbulent Prandtl number. Only used if k-ε turbulence model selected.
HCSFLCID	Load curve ID for scale factor applied on HC function of time. See *DEFINE_CURVE, *DEFINE_CURVE_FUNCTION, or *DEFINE_FUNCTION. If a *DEFINE_FUNCTION is used, the following parameters are allowed: f(x, y, z, vx, vy, vz, temp, pres, time).
TCSFLCID	Load curve ID for scale factor applied on TC function of time. See *DEFINE_CURVE, *DEFINE_CURVE_FUNCTION, or *DEFINE_FUNCTION. If a *DEFINE_FUNCTION is used, the following parameters are allowed: f(x, y, z, vx, vy, vz, temp, pres, time).
NNMOID	Non-Newtonian model ID. This refers to a non-Newtonian fluid model defined using *ICFD_MODEL_NONNEWT.
PMMOID	Porous media model ID. This refers to a porous media model defined using *ICFD_MODEL_POROUS.
SPTRID	Species transport model ID. This refers to a species transport model defined using *ICFD_MODEL_SPECIES_TRANSPORT.
VID	Viscoelastic model ID. This refers to a viscoelastic model defined using *ICFD_MODEL_VISCOELASTIC.
GAMMA	Ratio of specific heats at constant pressure and constant volume for the isentropic gas. This parameter is dimensionless. Specify a value greater than 1.0. This parameter is required for FLG = 2.
THETA0	Reference absolute temperature of the isentropic gas. Specify a positive value using an absolute temperature scale consistent with

VARIABLE	DESCRIPTION
	RGAS. This parameter is required for FLG = 2.
RGAS	Specific gas constant, expressed as energy per unit mass per unit absolute temperature. Specify a positive value in a unit system consistent with RO, pressure, and THETA0. This is the gas constant per unit mass, not the universal molar gas constant. This parameter is required for FLG = 2.

Remarks:

1. **Required parameters for the isentropic compressible model.** FLG = 2 activates the isentropic velocity-pressure formulation. RO and VIS on Card 1 and GAMMA, THETA0, and RGAS in the first three fields of Card 4 must be given. There are no default values for the three isentropic parameters. Cards 2 and 3 retain their existing meanings and cannot be skipped when Card 4 is supplied. A comment line does not replace a data card. Thermal properties on Card 2 are not required to define the isentropic equation of state.

2. **Reference state and equation of state for the isentropic compressible model.** The operating pressure (p_0) and reference sound speed (c_0) are determined from the material data:

$$p_0 = \rho_0 R \theta_0, \quad c_0 = \sqrt{\gamma R \theta_0}.$$

Here, ρ_0 is RO, γ is GAMMA, θ_0 is the THETA0, and R is RGAS. The solved pressure is gauge pressure, p_g, relative to p_0. The isentropic density and local sound speed are calculated from the absolute pressure:

$$p_{\text{abs}} = p_0 + p_g, \quad \rho = \rho_0 \left(\frac{p_{\text{abs}}}{p_0} \right)^{\frac{1}{\gamma}}, \quad c = \sqrt{\frac{\gamma p_{\text{abs}}}{\rho}}.$$

Thus, zero gauge pressure gives the reference density and sound speed. Gauge pressure is distinct from the acoustic pressure fluctuation about the running mean. During the incompressible initialization stage, the equation of state returns the reference density and sound speed.

3. **Solver and boundary controls for isentropic compressible fluids.** Use *ICFD_CONTROL_ISENTROPIC to define the global averaging window and the fractional-step Picard iteration count. Use *ICFD_BOUNDARY_ISENTROPIC_NRBC or *ICFD_BOUNDARY_ISENTROPIC_NRBC0 to define non-reflecting boundaries. Their boundary averaging and reference-pressure rules are described in those entries.

*ICFD_MODEL_NONNEWT

Purpose: Specify a non-newtonian model or a viscosity law that can associated to a fluid material.

Non-Newtonian Model ID and type.

Card 1	1	2	3	4	5	6	7	8
Variable	NNMOID	NNID						
Type	I	I						
Default	none	none						

Non-Newtonian Fluid Parameters Card.

Card 2	1	2	3	4	5	6	7	8
Variable	K	N	MUMIN	LAMBDA	ALPHA	TALPHA		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	1.e30	0.0	0.0		

VARIABLE	DESCRIPTION
NNMOID	Non-Newtonian Model ID.
NNID	Non-Newtonian fluid model type : EQ.1 : Power-Law model EQ.2 : Carreau model EQ.3 : Cross model EQ.4 : Herschel-Bulkley model EQ.5 : Cross II model EQ.6 : Sutherland formula for temperature dependent viscosity EQ.7 : Power-Law for temperature dependent viscosity EQ.8 : Viscosity defined by Load Curve ID or Function ID

VARIABLE	DESCRIPTION
K	Consistency index if NNID = 1 and 4. Zero shear Viscosity if NNID = 2,3 and 5.Reference viscosity if NNID = 6 and NNID = 7. Load curve ID or function ID if NNID = 8.
N	Measure of the deviation of the fluid from Newtonian (Power Law index) for NNID = 1,2,3,4,5,7. Not used for NNID = 6 and 8.
MUMIN	Minimum acceptable viscosity value if NNID = 1. Infinite Shear Viscosity if NNID = 2,5.Yielding viscosity if NNID = 4.Not used if NNID = 3,6,7,8.
LAMBDA	Maximum acceptable viscosity value if NNID = 1. Time constant if NNID = 2, 3, 5. Yield Stress Threshold if NNID = 4.Sutherland constant if NNID = 6. Not used if NNID = 7,8.
ALPHA	Activation energy if NNID = 1, 2. Not used if NNID = 3,4,5,6,7,8.
TALPHA	Reference temperature if NNID = 2. Not used if NNID = 1,3,4,5,6,7,8

Remarks:

1. For the Non-Newtonian models, the viscosity is expressed as :

   a) POWER-LAW :

$$\mu = k\dot{\gamma}^{n-1}e^{\alpha T_0/T}$$

$$\mu_{min} < \mu < \mu_{max}$$

   With k the consistency index, n the power law index, α the activation energy, T_0 the initial temperature, T the temperature at any given time t, μ_{min} the minimum acceptable viscosity and μ_{max} the maximum acceptable viscosity.

   b) CARREAU :

$$\mu = \mu_\infty + (\mu_0 - \mu_\infty)\left[1 + (H(T)\dot{\gamma}\lambda)^2\right]^{(n-1)/2}$$

$$H(T) = exp\left[\alpha\left(\frac{1}{T - T_0} - \frac{1}{T_\alpha - T_0}\right)\right]$$

   With μ_∞ the infinite shear viscosity, μ_0 the zero shear viscosity, n the power law index, λ a time constant, α the activation energy, T_0 the initial temperature, T

the temperature at any given time t and T_α the reference temperature at which $H(T) = 1$.

c) CROSS :

$$\mu = \frac{\mu_0}{1 + (\lambda\dot{\gamma})^{1-n}}$$

With μ_0 the zero shear viscosity, n the power law index and λ a time constant.

d) HERSCHEL-BULKLEY :

$$\mu = \mu_0 \; if \; (\dot{\gamma} < \tau_0/\mu_0)$$

$$\mu = \frac{\tau_0 + k[\dot{\gamma}^n - (\tau_0/\mu_0)^n]}{\dot{\gamma}}$$

With k the consistency index, τ_0 the Yield stress threshold, μ_0 the yielding viscosity and n the power law index.

e) CROSS II :

$$\mu = \mu_\infty + \frac{\mu_0 - \mu_\infty}{1 + (\lambda\dot{\gamma})^n}$$

With μ_0 the zero shear viscosity, μ_∞ the infinite shear viscosity, n the power law index and λ a time constant.

2. For the temperature dependent viscosity models, the viscosity is expressed as :

a) SUTHERLAND's LAW :

$$\mu = \mu_0 (\frac{T}{T_0})^{3/2} \frac{T_0 + S}{T + S}$$

With μ_0 a reference viscosity, T_0 the initial temperature (which therefore must not be 0.), T the temperature at any given time t and S Sutherland's constant.

b) POWER LAW :

$$\mu = \mu_0 (\frac{T}{T_0})^n$$

With μ_0 a reference viscosity, T_0 the initial temperature (which therefore must not be 0.), T the temperature at any given time t and n the power law index.

3. For NNID = 8, a load curve function of time, a curve function or a function can be used. If it references a DEFINE_FUNCTION, the following arguments are allowed $f(x, y, z, vx, vy, vz, temp, pres, shear, time)$.

*ICFD_MODEL_POROUS

Purpose: Specify a porous media model.

Card Summary:

Card 1. This card is required.

PMMOID	PMMTYPE	FORM	RHOCP	KAPPA			

Card 2a. This card is included if PMMTYPE = 1, 2, or 8.

POR	PER	FF		PSFLCID			

Card 2b. This card is included if PMMTYPE = 3 or 10.

POR	TH		FABTH	PVLCID			

Card 2c. This card is included if PMMTYPE = 4, 6, or 7.

POR							

Card 2d. This card is included if PMMTYPE = 5.

POR	THX	THY	THZ	PVLCIDX	PVLCIDY	PVLCIDZ	

Card 2e. This card is included if PMMTYPE = 11.

POR	ALPHA	BETA					

Card 3. This card is included if PMMTYPE = 4, 5, 6, or 7

KXP	KYP	KZP					

Card 4a. This card is included if PMMTYPE = 4 or 6.

PROJXPX	PROJXPY	PROJXPZ	PROJYPX	PROJYPY	PROJYPZ		

Card 4b. This card is included if PMMTYPE = 5 or 7.

PID1REF	PID2REF						

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	PMMOID	PMMTYPE	FORM	RHOCP	KAPPA			
Type	I	I	I	F	F			
Default	none	none	0	0.0	0.0			

VARIABLE	DESCRIPTION
PMMOID	Porous media model ID
PMMTYPE	Porous media model type: EQ.1: Isotropic porous media - Ergun correlation EQ.2: Isotropic porous media - Darcy-Forchheimer model EQ.3: Isotropic porous media - permeability defined through pressure-velocity data EQ.4: Anisotropic porous media. Fixed local reference frame (see Figure 7-13). EQ.5: Anisotropic porous media model - moving local reference frame and permeability vector in local reference frame (x', y', z') defined by three pressure-velocity curves. EQ.6: Anisotropic porous media model - moving local reference frame and permeability vector constant. EQ.7: Anisotropic porous media model - moving local reference frame and permeability vector constant. This model differs from PMMTYPE = 6 in the way the local reference frame is moved. EQ.8: Main parachute model to be used jointly with *MESH_EMBEDSHELL for the parachute surface. Similar to PMMTYPE = 2. EQ.10: Parachute model to be used jointly with *MESH_EMBEDSHELL where the fabric permeability and Forchheimer factor are computed from the pressure-velocity curves of experimental data given by a *LOAD_CURVE. Similar to PMMTYPE = 3.

VARIABLE	DESCRIPTION
	EQ.11: Parachute model similar to PMMTYPE = 8, but pressure gradient is directly defined by coefficients α and β as: $\frac{\Delta P(u_x)}{\Delta x} = \alpha u_x + \beta {u_x}^2$.
FORM	Porous media formulation: EQ.0: Classical (default) EQ.2: Interstitial velocity
RHOCP	Density of the structure multiplied by the specific heat of the structure
KAPPA	Thermal conductivity of the structure

Porous Media Parameters Card (PMMTYPE = 1, 2, and 8). This card is included PMMTYPE = 1, 2, or 8.

Card 2a	1	2	3	4	5	6	7	8
Variable	POR	PER	FF		PSFLCID			
Type	F	F	F		I			
Default	0.	0.	0.		optional			

VARIABLE	DESCRIPTION
POR	Porosity, ε
PER	Permeability, κ
FF	Forchheimer factor to be defined if PMMTYPE = 2 or 8.
PSFLCID	Optional permeability scale factor load curve ID, *DEFINE_CURVE_FUNCTION ID or *DEFINE_FUNCTION ID. If a *DEFINE_FUNCTION is used, the following parameters are allowed: f(x, y, z, vx, vy, vz, temp, pres, time).

Porous Media Parameters Card (PMMTYPE = 3 and 10). This card is included if PMMTYPE = 3 or 10.

Card 2b	1	2	3	4	5	6	7	8
Variable	POR	TH		FABTH	PVLCID			
Type	F	F		F	I			
Default	0.	0.		0.	none			

VARIABLE	DESCRIPTION
POR	Porosity, ε
TH	Probe thickness if PMMTYPE = 3
FABTH	Fabric thickness if PMMTYPE = 10
PVLCID	Pressure as a function of velocity load curve ID

Porous Media Parameters Card (PMMTYPE = 4, 6, and 7). This card is included if PMMTYPE = 4, 6, or 7.

Card 2c	1	2	3	4	5	6	7	8
Variable	POR							
Type	F							
Default	0.							

VARIABLE	DESCRIPTION
POR	Porosity, ε

Porous Media Parameters Card (PMMTYPE = 5). This card is included if PMMTYPE = 5.

Card 2d	1	2	3	4	5	6	7	8
Variable	POR	THX	THY	THZ	PVLCIDX	PVLCIDY	PVLCIDZ	
Type	F	F	F	F	I	I	I	
Default	0.	0.	0.	0.	none	none	none	

VARIABLE	DESCRIPTION
POR	Porosity, ε
THX	Probe thickness, Δx
THY	Probe thickness, Δy
THZ	Probe thickness, Δz
PVLCIDX	Load curve ID for pressure as a function of velocity in the global X-direction
PVLCIDY	Load curve ID for pressure as a function of velocity in the global Y-direction
PVLCIDZ	Load curve ID for pressure as a function of velocity in the global Z-direction

Porous Media Parameters Card (PMMTYPE = 11). This card is included if PMMTYPE = 11.

Card 2e	1	2	3	4	5	6	7	8
Variable	POR	ALPHA	BETA					
Type	F	F	F					
Default	0.	0.	0.					

VARIABLE	DESCRIPTION
POR	Porosity, ε
ALPHA	Coefficient, α
BETA	Coefficient, β

Permeability Vector Card in local reference frame. Only to be defined if the porous media is anisotropic (PMMTYPE = 4, 5, 6, 7).

Card 3	1	2	3	4	5	6	7	8
Variable	KXP	KYP	KZP					
Type	F	F	F					
Default	0.	0.	0.					

VARIABLE	DESCRIPTION
KXP, KYP, KZP	Permeability vector in local reference frame (x', y', z'). Those values become scale factors if PMMTYPE = 5.

Projection of Local Vectors in Global Reference Frame. This card is defined if PMMTYPE = 4 or 6.

Card 4a	1	2	3	4	5	6	7	8
Variable	PROJXPX	PROJXPY	PROJXPZ	PROJYPX	PROJYPY	PROJYPZ		
Type	F/I	F/I	F/I	F/I	F/I	F/I		
Default	0./0	0./0	0./0	0./0	0./0	0./0		

VARIABLE	DESCRIPTION
PROJXPX, PROJXPY, PROJXPZ	Projection of the local permeability vector, **x**′, in the global reference frame, (x, y, z). If PMMTYPE = 6, PROJXPX, PROJXPY, and PRPJXPZ become load curve IDs, so the coordinates of the local **x**′ vector can change in time.

VARIABLE	DESCRIPTION
PROJYPX, PROJYPY, PROJYPZ	Projection of the local permeability vector, **y**′, in the global reference frame, (x,y,z). If PMMTYPE = 6, PROJYPX, PROJYPY, and PRPJYPZ become load curve IDs, so the coordinates of the local **y**′ vector can change in time.

Local Reference Frame Vectors. This card is defined if PMMTYPE = 5 or 7.

Card 4b	1	2	3	4	5	6	7	8
Variable	PID1REF	PID2REF						
Type	I	I						
Default	0	0.						

VARIABLE	DESCRIPTION
PID1REF, PID2REF	Two local reference frame vectors are defined by the coordinates of the two-point IDs defined by PID1REF and PID2REF. (See *ICFD_DEFINE_POINT). Since those points can be made to move, it is therefore possible to define a moving reference frame for the anisotropic porous media domain.

Remarks:

1. **Generalized Flow Equations in a Porous Media.** Let ε be the porosity and κ be the permeability of the porous media. Then,

$$\varepsilon = \frac{\text{void volume}}{\text{total volume}} \ .$$

u_i, the volume averaged velocity field, can then be defined in terms of the fluid velocity field, u_{if}, as:

$$u_i = \varepsilon u_{if} \ .$$

The generalized flow equations of momentum and mass conservation can be expressed as:

$$\frac{\partial u_i}{\partial x_i} = 0$$

$$\frac{\rho}{\varepsilon}\left[\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}\left(\frac{u_i u_j}{\varepsilon}\right)\right] = -\frac{1}{\varepsilon}\frac{\partial (P\varepsilon)}{\partial x_i} + \frac{\mu}{\varepsilon}\left(\frac{\partial^2 u_i}{\partial x_j \partial x_j}\right) + \rho g_i - D_i$$

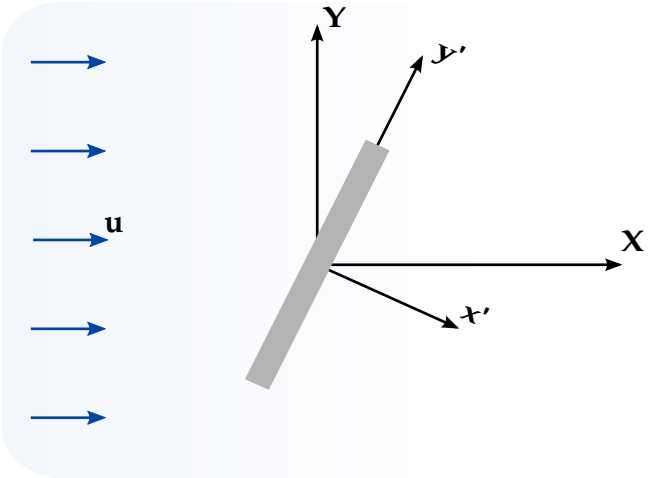


Figure 7-13. Anisotropic porous media vectors definition (PMMTYPE = 4, 5, 6, and 7). The vectors **X** and **Y** are the global axes; $\mathbf{x}'$ and $\mathbf{y}'$ define the system for the primed coordinate(x', y', z').

where D_i are the forces exerted on the fluid by the porous matrix (see Remarks 2 and 3).

2. **Porous Forces for Isotropic Models.** For the isotropic model, the porous forces are a function of the matrix porosity and its permeability. For the isotropic case, three models are available:

   a) *Model 1 (Ergun correlation).*

$$D_i = \frac{\mu u_i}{\kappa} + \frac{1.75\rho|U|}{\sqrt{150}\sqrt{\kappa}\varepsilon^{3/2}} u_i$$

   b) *Model 2 (Darcy-Forcheimer).*

$$D_i = \frac{\mu u_i}{\kappa} + \frac{F\varepsilon\rho|U|}{\sqrt{\kappa}} u_i$$

   c) *Model 3.* Using the $\Delta P - V$ experimental data. In this case, it is assumed that the pressure-velocity curve was obtained by applying a pressure difference or pressure drop on both ends of a porous slab of thickness Δx with porous properties κ and ε. It then becomes possible for the solver to fit that experimental curve with a quadratic polynomial of the form $\Delta P(u_x) = \alpha {u_x}^2 + \beta u_x$. Once α and β are known, it is possible to estimate D_i.

3. **Anisotropic Darcy-Forcheimer Term.** The anisotropic (see Figure 7-13) version of the Darcy-Forcheimer term can be written as:

$$D_i = \mu B_{ij} \mu_j + F\varepsilon|U|C_{ij}u_j$$
$$B_{ij} = \left(K_{ij}\right)^{-1}$$
$$C_{ij} = \left(K_{ij}\right)^{-1/2}$$

Here K_{ij} is the anisotropic permeability tensor.

*ICFD_MODEL_SPECIES_TRANSPORT

Purpose: Specify a passive or reactive multi-species transport model that can be associated with a fluid material.

Card 1	1	2	3	4	5	6	7	8
Variable	SPTRID	SPNMBR	DENSRCT	DENSFID	HVAC			
Type	I	I	I	I	I			
Default	none	none	0	none	0			

Card 2	1	2	3	4	5	6	7	8
Variable	MASSDIF1	MASSDIF2	MASSDIF3	MASSDIF4	MASSDIF5	MASSDIF6	MASSDIF7	MASSDIF8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Reaction Rate Card. Include this card if DENSRCT = 2.

Card 3	1	2	3	4	5	6	7	8
Variable	FID1	FID2	FID3	FID4	FID5	FID6	FID7	FID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

HVAC Card. Include this card if HVAC = 1. See Remark 1.

Card 4	1	2	3	4	5	6	7	8
Variable	WVMDIF	WVY0	WVTHCND	RWV	WVHTCAP	RDAIR	TOTATMP	
Type	F	F	F	F	F	F	F	
Default	2.5e-5	0.0099	0.019	461.0	1860.0	287.0	101325.0	

VARIABLE	DESCRIPTION
SPTRID	Species transport model ID
SPNMBR	Number of species: EQ.1: One (passive) species EQ.2: Two species ⋮ EQ.8: Eight species
DENSRCT	Flag for kind of species transport: EQ.0: Passive species transport EQ.1: Passive species transport where the fluid density depends on the concentrations of the species. DENSFID gives the fluid density as a function of species concentrations. EQ.2: Reactive species transport. Card 3 gives the reaction rates for each species.
DENSFID	Function ID for DENSRCT = 1 giving the fluid density as a function of species concentrations. The following parameters are allowed: f(x,y,z,vx,vy,vz,temp,pres,time,conc1,conc2,…,conc8). Here, x, y, and z are the coordinates, vx, vy, and vz are the components of the velocity, time is the time, pres is the pressure, conc*i* is the concentration of the i^{th} species.
HVAC	Heating, ventilation, and air conditioning model flag: EQ.0: Off EQ.1: On (see Remark 1)
MASSDIF*i*	Mass diffusion for the i^{th} species

VARIABLE	DESCRIPTION
FID*i*	Function ID giving the reaction rate for the i^{th} species. The following parameters are allowed: f(x, y, z, time, temp, pres, conc1, conc2, …, conc8). Here, x, y, and z are the coordinates, time is the time, pres is the pressure, and conc*i* is the concentration of the i^{th} species.
WVMDIF	Water vapor mass diffusion
WCY0	Water vapor reference concentration, Y_0
WVTHCND	Water vapor thermal conductivity
RWV	Water vapor gas constant *R*
WVHTCAP	Water vapor heat capacity
RDAIR	Dry air gas constant *R*
TOTATMP	Total atmospheric pressure

Remarks:

1. **HVAC models.** In general, HVAC models depend on experimentally determined parameters. To ensure these values are meaningful, you must set up the simulation using the SI unit system (m, s, kg, N, Pa, K, W, J). The default values defined in Card 4 use these units.

*ICFD_MODEL_VISCOELASTIC

Purpose: Specify a viscoelastic model that can be associated with a fluid material.

Card 1	1	2	3	4	5	6	7	8
Variable	VID	VTYPE						
Type	I	I						
Default	none	none						

Card 2	1	2	3	4	5	6	7	8
Variable	RTIME	PVISC	MOBF					
Type	F	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
VID	Viscoelastic model ID
VTYPE	Viscoelastic model type: EQ.1: Oldroyd-B model EQ.2: Giesekus model
RTIME	Relaxation time
PVISC	Polymeric viscosity
MOBF	Mobility factor (for VTYPE = 2 only)

***ICFD_PART**_*{OPTION}*

Available options include

TITLE

Purpose: Define parts for the implicit computational fluid dynamics solver.

The TITLE option allows the user to define an additional optional line with a HEADING in order to associate a name to the part.

Card 1	1	2	3	4	5	6	7	8
Variable	HEADING							
Type	A							
Default	none							

Part Material Card. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	PID	SECID	MID					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
PID	Part identifier for fluid surfaces.
SECID	Section identifier defined with the *ICFD_SECTION card.
MID	Material identifier defined with the *ICFD_MAT card.

*ICFD_PART_VOL_{OPTION}

Available options include

TITLE

Purpose: Assign material properties to the nodes enclosed by surface ICFD parts.

The TITLE option allows the user to define an additional optional line with a HEADING in order to associate a name to the part.

Title	1	2	3	4	5	6	7	8
Variable	HEADING							
Type	A							
Default	none							

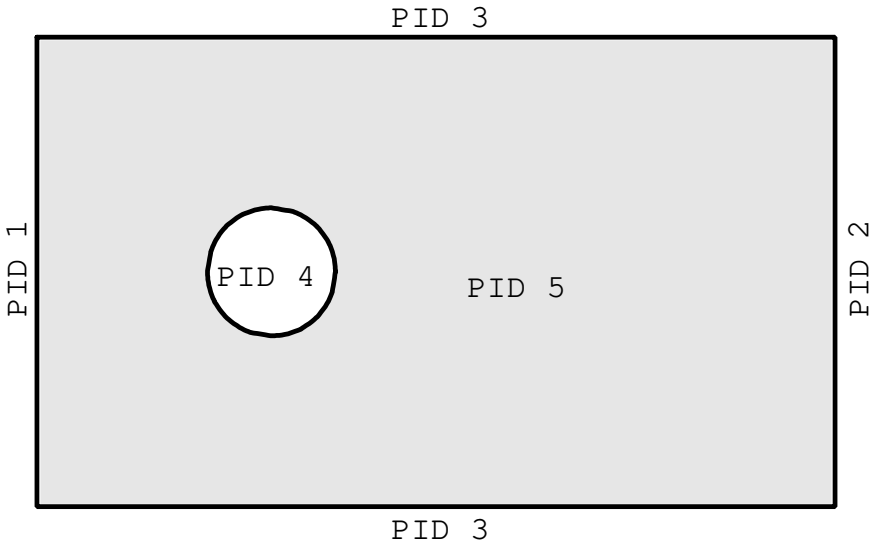
Card 1	1	2	3	4	5	6	7	8
Variable	PID	SECID	MID	MID2				
Type	I	I	I	I				
Default	none	none	none	0				

Provide as many cards as necessary. This input ends at the next keyword ("*") card

Card 2	1	2	3	4	5	6	7	8
Variable	SPID1	SPID2	SPID3	SPID4	SPID5	SPID6	SPID7	SPID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
PID	Part identifier for fluid volumes.

VARIABLE	DESCRIPTION
SECID	Section identifier defined by the *ICFD_SECTION card.
MID	Material ID
MID2	Optional material ID to specify a second material in the volume in the case of level sets. The first material (MID) fills the positive level set values.
SPID1, …	Part IDs for the surface elements that define the volume mesh.




```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$   *ICFD_PART_VOL
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ PART ID 5 is defined by the surfaces that enclose it.
$
*ICFD_PART_VOL
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      pid       secid       mid
         5           1         1
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      pid1      pid2      pid3      pid4      pid5      pid6      pid7      pid8
         1         2         3         4
```

*ICFD_SECTION

Purpose: Define a section for the implicit computational fluid dynamics solver.

Card 1	1	2	3	4	5	6	7	8
Variable	SID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
SID	Section identifier

*ICFD_SET_NODE_LIST

Purpose: Only used in cases where the mesh is specified by the user (See *MESH_VOLUME_ELEMENT). Defines a set of nodes associated with a part ID on which boundary conditions can be applied.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	PID						
Type	I	I						
Default	none	none						

Node List Card. Provide as many cards as necessary. This input ends at the next keyword ("*") card

Card 2	1	2	3	4	5	6	7	8
Variable	NID1	NID2	NID3	NID4	NID5	NID6	NID7	NID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
SID	Set ID
PID	Associated Part ID.
NID1, …	Node IDs

Remarks:

1. The convention is the similar to the one used by the keyword *SET_NODE_LIST and serves a similar purpose.

*ICFD_SOLVER_SPLIT

Purpose: This keyword provides an option to trigger an iterative procedure on the fluid system. This procedure aims to bring more precision to the final pressure and velocity values but is often very time consuming. It must therefore be used with caution. It is intended only for special cases. For stability purposes, this method is automatically used for the first ICFD time step.

Card 1	1	2	3	4	5	6	7	8
Variable	NIT	TOL						
Type	I	F						
Default	1	10^{-3}						

VARIABLE	DESCRIPTION
NIT	Maximum Number of iterations of the system for each fluid time step. If TOL criteria is not reached after NIT iterations, the run will proceed.
TOL	Tolerance Criteria for the pressure residual during the fluid system solve.

*ICFD_SOLVER_TOL_FSI

Purpose: This keyword allows the user to change the default tolerance values for the Newton Raphson loop in the strong FSI analysis. *Care should be taken when deviating from the default values*.

Card 1	1	2	3	4	5	6	7	8
Variable	ATOL	RTOL		MAXIT				
Type	F	F		I				
Default	10^{-5}	10^{-5}		1000				

VARIABLE	DESCRIPTION
ATOL	Absolute convergence criteria. Convergence is achieved when $\text{Residual}_{i+1} - \text{Residual}_i \leq \text{ATOL}$. If a negative integer is entered, then that value will be used as a load curve ID for ATOL.
RTOL	Relative convergence criteria. Convergence is achieved when $(\text{Residual}_{i+1} - \text{Residual}_i)/\text{Residual}_{\text{initial}} \leq RTOL$. If a negative integer is entered, then that value will be used as a load curve ID for RTOL.
MAXIT	Maximum number of iterations allowed to achieve convergence. If a negative integer is entered, then that value will be used as a load curve ID for MAXIT.

*ICFD_SOLVER_TOL_LSET

Purpose: This keyword allows the user to change the default tolerance values for the advection equation for levelset. *Care should be taken when deviating from the default values*.

Card 1	1	2	3	4	5	6	7	8
Variable	ATOL	RTOL		MAXIT				
Type	F	F		I				
Default	10^{-8}	10^{-8}		1000				

VARIABLE	DESCRIPTION
ATOL	Absolute convergence criteria. Convergence is achieved when $\text{Residual}_{i+1} - \text{Residual}_i \leq \text{ATOL}$. If a negative integer is entered, then that value will be used as a load curve ID for ATOL.
RTOL	Relative convergence criteria. Convergence is achieved when $(\text{Residual}_{i+1} - \text{Residual}_i)/\text{Residual}_{\text{initial}} \leq RTOL$. If a negative integer is entered, then that value will be used as a load curve ID for RTOL.
MAXIT	Maximum number of iterations allowed to achieve convergence. If a negative integer is entered, then that value will be used as a load curve ID for MAXIT.

*ICFD_SOLVER_TOL_MMOV

Purpose: This keyword allows the user to change the default tolerance values for the mesh movement algorithm. *Care should be taken when deviating from the default values*.

Card 1	1	2	3	4	5	6	7	8
Variable	ATOL	RTOL		MAXIT		DISPTOL		
Type	F	F		I		F		
Default	1e-8	1e-8		1000		0.		

VARIABLE	DESCRIPTION
ATOL	Absolute convergence criteria. Convergence is achieved when $\text{Residual}_{i+1} - \text{Residual}_i \leq \text{ATOL}$. If a negative integer is entered, then that value will be used as a load curve ID for ATOL.
RTOL	Relative convergence criteria. Convergence is achieved when $(\text{Residual}_{i+1} - \text{Residual}_i)/\text{Residual}_{\text{initial}} \leq \text{RTOL}$. If a negative integer is entered, then that value will be used as a load curve ID for RTOL.
MAXIT	Maximum number of iterations allowed to achieve convergence. If a negative integer is entered, then that value will be used as a load curve ID for MAXIT.
DISPTOL	Element deformation tolerance before a matrix reassembly is triggered. Default is 0. which means any element deformation detected will automatically trigger a matrix reassembly. Higher values will potentially save calculation times at the expense of accuracy.

*ICFD_SOLVER_TOL_MOM

Purpose: This keyword allows the user to change the default tolerance values for the momentum equation solve. *Care should be taken when deviating from the default values*.

Card 1	1	2	3	4	5	6	7	8
Variable	ATOL	RTOL		MAXIT				
Type	F	F		I				
Default	10^{-8}	10^{-8}		1000				

VARIABLE	DESCRIPTION
ATOL	Absolute convergence criteria. Convergence is achieved when $\text{Residual}_{i+1} - \text{Residual}_i \leq \text{ATOL}$. If a negative integer is entered, then that value will be used as a load curve ID for ATOL.
RTOL	Relative convergence criteria. Convergence is achieved when $(\text{Residual}_{i+1} - \text{Residual}_i)/\text{Residual}_{\text{initial}} \leq \text{RTOL}$. If a negative integer is entered, then that value will be used as a load curve ID for RTOL.
MAXIT	Maximum number of iterations allowed to achieve convergence. If a negative integer is entered, then that value will be used as a load curve ID for MAXIT.

*ICFD_SOLVER_TOL_MONOLITHIC

Purpose: This keyword allows the user to change the default tolerance values for the monolithic solver. *Care should be taken when deviating from the default values*.

Card 1	1	2	3	4	5	6	7	8
Variable	ATOL	RTOL		MAXIT				
Type	F	F		I				
Default	10^{-8}	10^{-8}		1000				

VARIABLE	DESCRIPTION
ATOL	Absolute convergence criteria. Convergence is achieved when $\text{Residual}_{i+1} - \text{Residual}_i \leq \text{ATOL}$. If a negative integer is entered, then that value will be used as a load curve ID for ATOL.
RTOL	Relative convergence criteria. Convergence is achieved when $(\text{Residual}_{i+1} - \text{Residual}_i)/\text{Residual}_{\text{initial}} \leq \text{RTOL}$. If a negative integer is entered, then that value will be used as a load curve ID for RTOL.
MAXIT	Maximum number of iterations allowed to achieve convergence. If a negative integer is entered, then that value will be used as a load curve ID for MAXIT.

*ICFD_SOLVER_TOL_PRE

Purpose: Change the default tolerance values for the Poisson equation for pressure. *Care should be taken when deviating from the default values*.

Card 1	1	2	3	4	5	6	7	8
Variable	ATOL	RTOL		MAXIT	PREID	PTOL		
Type	F	F		I	I	F		
Default	10^{-8}	10^{-8}		1000	2	10^{-3}		

VARIABLE	DESCRIPTION
ATOL	Absolute convergence criteria. Convergence is achieved when $\text{Residual}_{i+1} - \text{Residual}_i \leq \text{ATOL}$. If a negative integer is entered, then that value will be used as a load curve ID for ATOL.
RTOL	Relative convergence criteria. Convergence is achieved when $(\text{Residual}_{i+1} - \text{Residual}_i)/\text{Residual}_{\text{initial}} \leq \text{RTOL}$. If a negative integer is entered, then that value will be used as a load curve ID for RTOL.
MAXIT	Maximum number of iterations allowed to achieve convergence. If a negative integer is entered, then that value will be used as a load curve ID for MAXIT.
PREID	Choice of Preconditioner for the Conjugate Gradient Solve: EQ.1: Diagonal Preconditioner EQ.2: Incomplete LU factorization EQ.5: Global MUMPS factorization
PTOL	Preconditioner tolerance (a.k.a Drop Tolerance if PREID = 2)

*ICFD_SOLVER_TOL_TEMP

Purpose: This keyword allows the user to change the default tolerance values for the heat equation. To be handled with great care.

Card 1	1	2	3	4	5	6	7	8
Variable	ATOL	RTOL		MAXIT				
Type	F	F		I				
Default	1e-8	1e-8		1000				

VARIABLE	DESCRIPTION
ATOL	Absolute convergence criteria. Convergence is achieved when $Residual_{i+1} - Residual_i \leq ATOL$. If a negative integer is entered, then that value will be used as a load curve ID for ATOL.
RTOL	Relative convergence criteria. Convergence is achieved when $(Residual_{i+1} - Residual_i)/Residual_{initial} \leq RTOL$. If a negative integer is entered, then that value will be used as a load curve ID for RTOL.
MAXIT	Maximum number of iterations allowed to achieve convergence. If a negative integer is entered, then that value will be used as a load curve ID for MAXIT.

*MESH

The keyword *MESH is used to create a mesh that will be used in the analysis. So far only tetrahedral (or triangular in 2D) elements can be generated. The keyword cards in this section are defined in alphabetical order:

*MESH_BL

*MESH_BL_SYM

*MESH_EMBEDSHELL

*MESH_INTERF

*MESH_NODE

*MESH_SIZE

*MESH_SIZE_SHAPE

*MESH_SURFACE_ELEMENT

*MESH_SURFACE_NODE

*MESH_SURFACE_NULL

*MESH_VOLUME

*MESH_VOLUME_ELEMENT

*MESH_VOLUME_NODE

*MESH_VOLUME_PART

*MESH_BL

Purpose: Define a boundary-layer mesh as a refinement of the volume mesh. The boundary layer mesh is constructed by subdividing elements near the surface.

Boundary Layer Cards. Define as many cards as are necessary. The next keyword ("*") card terminates this input.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	NELTH	BLTH	BLFE	BLST	BLDR	ISIDE	
Type	I	I	F	F	I	I	I	
Default	none	none	0.	0.	0	0	0	

VARIABLE	DESCRIPTION
PID	Part identifier for the surface element
NELTH	Number of elements normal to the surface (in the boundary layer) is NELTH + 1. See Remark 5.
BLTH	Boundary layer mesh thickness if BLST = 1 or 2. Growth scale factor if BLST = 3. Ignored if BLST = 0. See Remark 5.
BLFE	Distance between the wall and the first volume mesh node if BLST = 3. Scaling coefficient if BLST = 1 or 2. Ignored if BLST =0. See Remark 5.
BLST	Boundary layer mesh generation strategy: EQ.0: $2^{NELTH+1}$ subdivisions based on surface mesh size (default). See Remark 1. EQ.1: Power law using BLTH and NELTH with BLFE as a scale factor. See Remark 2 and Figure 8-1. EQ.2: Geometric series based on BLTH and BLFE. See Remark 3 and Figure 8-2. EQ.3: Repartition following a growth scale factor (BLTH). See Remark 4 and Figure 8-3.
BLDR	Boundary layer mesh generation during the dynamic relaxation phase when present (see *ICFD_CONTROL_GENERAL):

VARIABLE	DESCRIPTION
	EQ.0: On. No distinction for boundary layer mesh generation between the dynamic relaxation and transient phases.
	EQ.1: Off. Boundary layer mesh is not generated during the dynamic relaxation phase.
ISIDE	For embedded shells only, flag to define the boundary layer on one side of the embedded surface:
	EQ.0: Generates the boundary layer mesh on both sides of *MESH_EMBEDSHELL.
	EQ.1: Generates the boundary layer mesh on one side only – on the faces that are originally present in the embedded surface.
	EQ.2: Generates the boundary layer mesh on one side only – on the additional faces that are generated by the embedded surface.

Remarks:

1. **BLST = 0.** For BLST = 0, for every additional NELTH, the automatic volume mesher will divide the elements closest to the surface by two so that the smallest element in the boundary layer mesh will have an aspect ratio of $2^{\text{NELTH}+1}$. A default boundary layer mesh thickness based on the surface mesh size will be chosen.

2. **BLST = 1.** For a constant repartition of the nodes in the boundary layer, use BLST = 1 with BLFE = 1. For BLST = 1, starting from the wall, the position of node n in the normal direction is given by:

$$X_n = \left(\frac{n}{\text{NELTH}+1}\right)^{[5\times(1-\text{BLFE})]} \frac{\text{BLTH}}{\sum_{i=1}^{\text{NELTH}+1}\left[i/(\text{NELTH}+1)\right]^{[5\times(1-\text{BLFE})]}}$$

3. **BLST = 2.** Setting BLFE = 1 makes BLST = 2 equivalent to BLST = 0, except that BLST = 2 allows you to specify the boundary layer thickness instead of automatically using the local surface mesh size. For BLST = 2, starting from BLTH from the wall, each newly inserted node will have its location closer to the wall, following this law:

$$X_n = (0.5 \times \text{BLFE})^n \times \text{BLTH}(1 - 0.5 \times \text{BLFE})$$

4. **BLST = 3.** For BLST = 3, starting from the wall, the position of node n in the normal direction is given by:

$$X_n = \sum_{i=0}^{n} \text{BLFE} \times \text{BLTH}^i \text{ with } 0 \leq n \leq \text{NELTH}$$

5. **Dynamically changing boundary layer.** For NELTH, BLTH, and BLFE, setting a negative value will point to a time-dependent load curve, which enables dynamically changing the boundary layer settings in cases involving remeshing (including returning a 0 value for the number of elements which effectively disables the boundary layer). NELTH and BLTH also accept a *DEFINE_FUNCTION as input with time and the x, y, and z coordinates accepted as arguments.

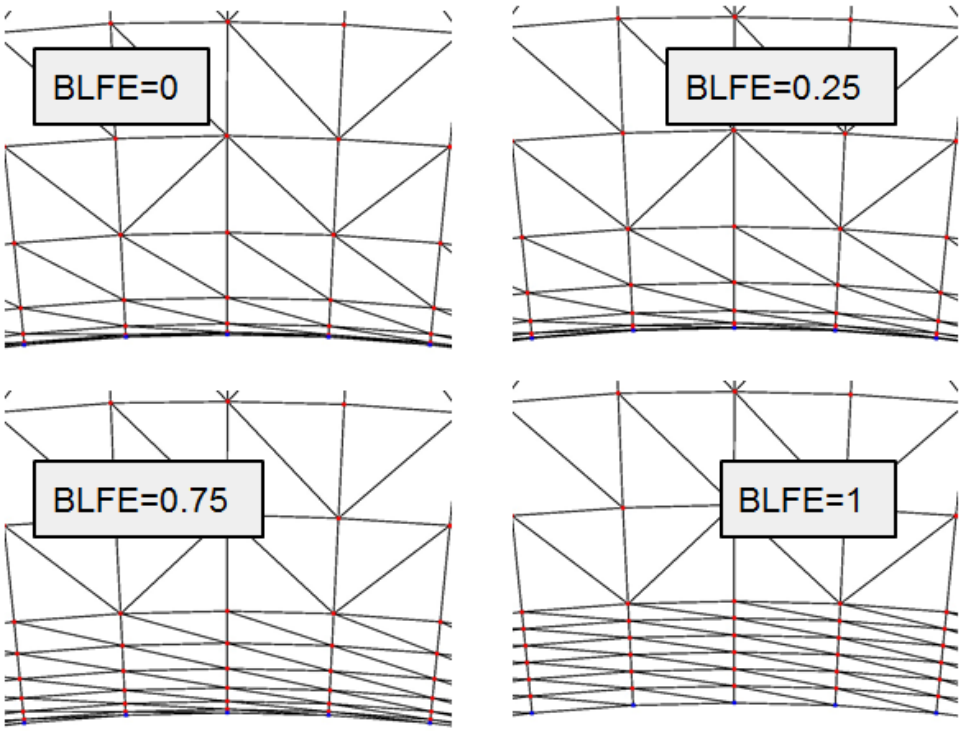


Figure 8-1. BLST = 1 example

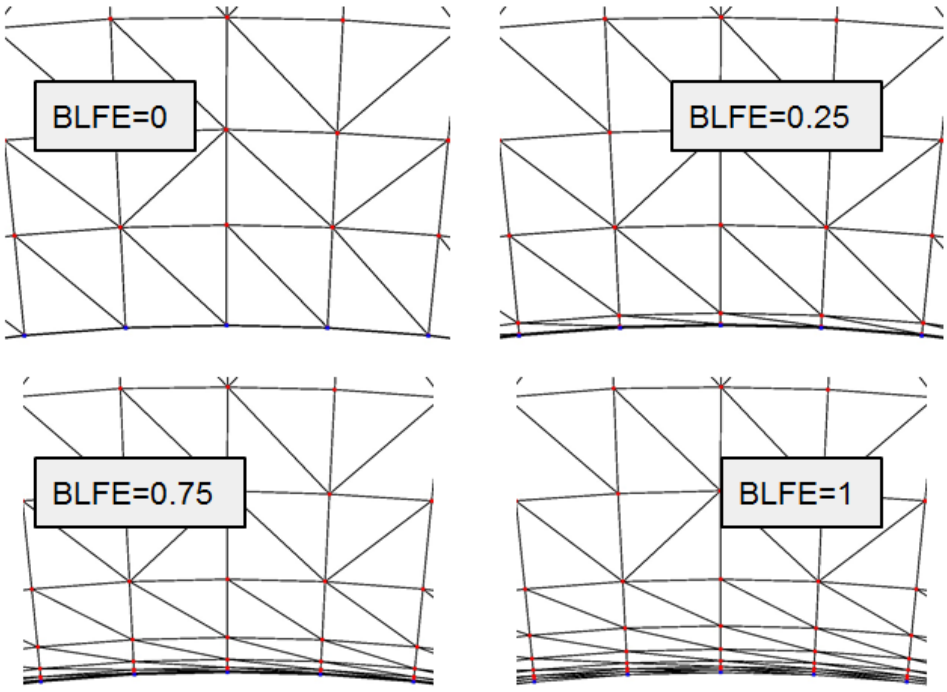


Figure 8-2. BLST = 2 example

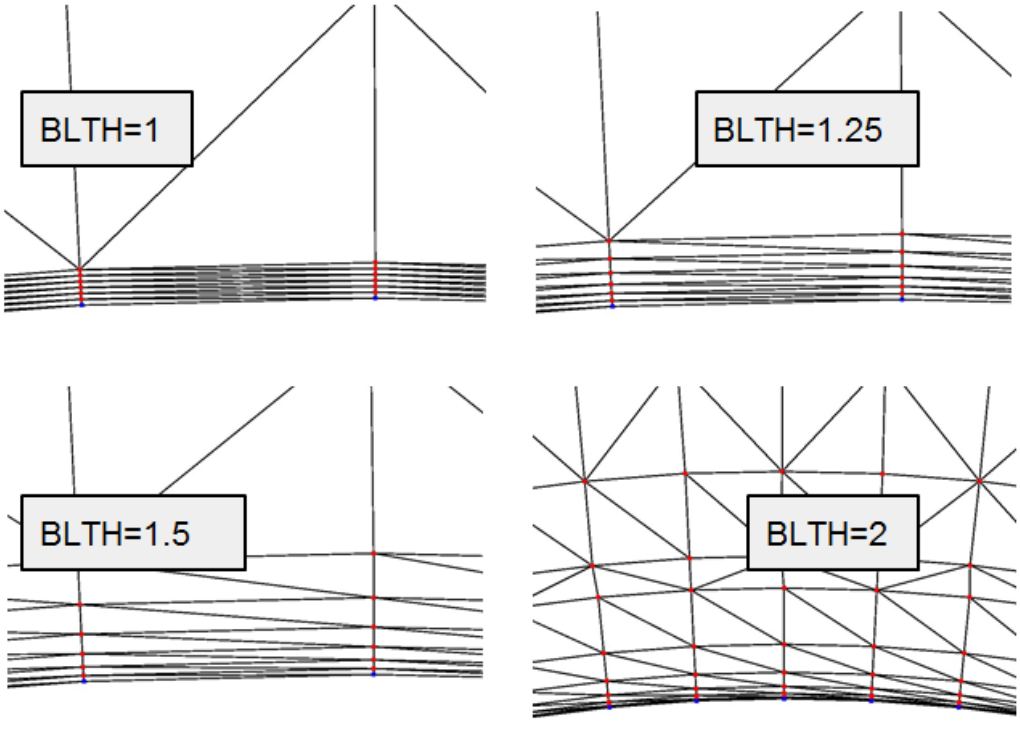


Figure 8-3. BLST = 3 example

*MESH_BL_SYM

Purpose: Specify the part IDs that will have symmetry conditions for the boundary layer. On these surfaces, the boundary layer mesh follows the surface tangent.

Boundary Layer with Symmetry Condition Cards. Define as many cards as necessary. The next "*" card terminates the input.

Card 1	1	2	3	4	5	6	7	8
Variable	PID1	PID2	PID3	PID4	PID5	PID6	PID7	PID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
PID1, …	Part identifiers for the surface element. This is the surface with symmetry.

*MESH_EMBEDSHELL

Purpose: Define surfaces that the mesher will embed inside the volume mesh. These surfaces have no thickness and conform to the rest of the volume mesh, having matching nodes on the interface.

Card 1	1	2	3	4	5	6	7	8
Variable	VOLID							
Type	I							
Default	none							

Define as many cards as are necessary based on the number of PIDs (the next "*" card terminates the input.)

Card 2	1	2	3	4	5	6	7	8
Variable	PID1	PID2	PID3	PID4	PID5	PID6	PID7	PID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
VOLID	ID assigned to the generated volume by the keyword *MESH_VOLUME. The surface mesh size will be applied to this volume.
PID*n*	Part IDs for the surface elements that will be embedded in the volume mesh

*MESH_INTERF

Purpose: Define the surfaces the mesher will use to specify fluid interfaces in multi-fluid simulations.

Card 1	1	2	3	4	5	6	7	8
Variable	VOLID							
Type	I							
Default	none							

Define as many cards as are necessary based on the number of PIDs. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	PID1	PID2	PID3	PID4	PID5	PID6	PID7	PID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
VOLID	ID assigned to the new volume in the keyword *MESH_VOLUME. The interface meshes will be applied to this volume.
PID*n*	Part IDs for the surface elements

*MESH_NODE

Purpose: Define a fluid node and its coordinates. These nodes are used in the mesh generation process by *MESH_SURFACE_ELEMENT or as user-defined volume nodes by *MESH_VOLUME_ELEMENT.

Node Cards. Include one additional card for each node. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	NID	X		Y		Z				
Type	I	F		F		F				
Default	none	0.0		0.0		0.0				

VARIABLE	DESCRIPTION
NID	Node ID. A unique number with respect to the other surface nodes.
X	*x* coordinate
Y	*y* coordinate
Z	*z* coordinate

Remarks:

1. **Data card format.** The data card format for the *MESH_NODE keyword is identical to *NODE.

2. **Related keywords.** The *MESH_NODE keyword supersedes *MESH_SURFACE_NODE, which was for surface nodes, and *MESH_VOLUME_NODE, which was for user-defined volume nodes.

*MESH_SIZE

Purpose: Define the surfaces that will be used by the mesher to specify a local mesh size inside the volume. If no internal mesh is used to specify the size, the mesher will use a linear interpolation of the surface sizes that define the volume enclosure.

Card 1	1	2	3	4	5	6	7	8
Variable	VOLID							
Type	I							
Default	none							

Define as many cards as are necessary based on the number of PIDs (the next "*" card terminates the input.).

Card 2	1	2	3	4	5	6	7	8
Variable	PID1	PID2	PID3	PID4	PID5	PID6	PID7	PID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
VOLID	ID assigned to the new volume in the keyword *MESH_VOLUME. The mesh sizing will be applied to this volume.
PID*n*	Part IDs for the surface elements that are used to define the mesh size next to the surface mesh.

*MESH_SIZE_SHAPE

Purpose: Defines a local mesh size in specific zones corresponding to given geometrical shapes (box, sphere, cylinder, and polynomial). The solver automatically applies the conditions specified during the generation of the volume mesh. This zone does not need to be entirely defined in the volume mesh. In the polynomial case, we recommend defining several zones for better mesh size control.

Card Summary:

Card Sets. Add as many sets of these cards as needed. This input ends with the next keyword ("*") card.

Card 1. This card is required.

SNAME	FORCE	METHOD	BT	DT			

Card 2a. Include this card if SNAME = box and METHOD = 0.

MSIZE	PMINX	PMINY	PMINZ	PMAXX	PMAXY	PMAXZ	

Card 2b. Include this card if SNAME = sphere and METHOD = 0.

MSIZE	RADIUS	CENTERX	CENTERY	CENTERZ			

Card 2c. Include this card if SNAME = cylinder and METHOD = 0.

MSIZE	RADIUS	PMINX	PMINY	PMINZ	PMAXX	PMAXY	PMAXZ

Card 2d. Include this card if SNAME = pol and METHOD = 0.

MSIZE	X	Y	Z	NX	NY	NZ	

Card 2e. Include this card if METHOD = 1.

MSIZE	RADIUS	PTID1	PTID2		RMIN		

Card 2f. Include this card if METHOD = 2.

MSIZE	RADIUS	NID1	NID2				

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	SNAME	FORCE	METHOD	BT	DT			
Type	A	I	I	F	F			
Default	none	0	0	0.	10^{12}			

VARIABLE	DESCRIPTION
SNAME	Shape name. Possibilities include box, cylinder, pol, and sphere.
FORCE	Flag to keep the mesh size criteria even after performing remeshing: EQ.0: Off. The mesh size in the shape will be lost if a remeshing occurs. EQ.1: On.
METHOD	Specifies which method to use when defining the second card: EQ.0: Default, directly input the coordinates. EQ.1: Define the coordinates with *ICFD_DEFINE_POINT IDs. See Remark 1. EQ.2: Define the coordinates by using user node IDs from solid mechanics. This method is only available when a solid mechanics problem is present. See Remark 1.
BT/DT	Birth and death time of the mesh size area in cases where remeshing occurs.

Box Case. Card 2 for SNAME = box and METHOD = 0

Cards 2a	1	2	3	4	5	6	7	8
Variable	MSIZE	PMINX	PMINY	PMINZ	PMAXX	PMAXY	PMAXZ	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
MSIZE	Mesh size that needs to be applied in the zone of the shape. A negative value points to a *DEFINE_FUNCTION. If using a *DEFINE_FUNCTION, the following parameters are allowed: f(x, y, z, time).
PMIN*i*	*x*, *y*, or *z* value for the point of minimum coordinates
PMAX*i*	*x*, *y*, or *z* value for the point of maximum coordinates

Sphere Case. Card 2 for SNAME = sphere and METHOD = 0

Cards 2b	1	2	3	4	5	6	7	8
Variable	MSIZE	RADIUS	CENTERX	CENTERY	CENTERZ			
Type	F	F	F	F	F			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
MSIZE	Mesh size that needs to be applied in the zone of the shape
RADIUS	Radius of the sphere
CENTER*i*	*x*, *y*, and *z* coordinates of the sphere's center

Cylinder Case. Card 2 for SNAME = cylinder and METHOD = 0

Cards 2c	1	2	3	4	5	6	7	8
Variable	MSIZE	RADIUS	PMINX	PMINY	PMINZ	PMAXX	PMAXY	PMAXZ
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
MSIZE	Mesh size that needs to be applied in the zone of the shape
RADIUS	Radius of the cylinder
PMIN*i*	x, y, or z value for the point of minimum coordinates
PMAX*i*	x, y, or z value for the point of maximum coordinates

Polynomial Case. Card 2 for SNAME = pol and METHOD = 0

Cards 2d	1	2	3	4	5	6	7	8
Variable	MSIZE	X	Y	Z	NX	NY	NZ	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
MSIZE	Mesh size that needs to be applied in the zone of the shape
X/Y/Z	Coordinates of starting point
NX/NY/NZ	Direction in which the mesh size will be applied

METHOD = 1 Case. Card 2 for METHOD = 1.

Cards 2e	1	2	3	4	5	6	7	8
Variable	MSIZE	RADIUS	PTID1	PTID2		RMIN		
Type	F	F	I	I		F		
Default	none	none	none	none		0.		

VARIABLE	DESCRIPTION
MSIZE	Mesh size that needs to be applied in the zone of the shape defined by SNAME
RADIUS	Radius of the sphere if SNAME is sphere or radius of the cylinder if SNAME is cylinder. Ignored otherwise.
PTID1	Point ID 1, referring to a *ICFD_DEFINE_POINT. It replaces PMIN*i* for box and cylinder, X/Y/Z for pol, and CENTER for sphere.
PTID2	Point ID 2. It is ignored if SNAME is sphere. It replaces PMAX*i* for box and cylinder and NX/NY/NZ for pol.
RMIN	Optional radius available if SNAME is sphere or cylinder to define an internal sphere/cylinder where the *MESH_SIZE_SHAPE does not apply. When defined, RMIN should be smaller than RADIUS.

METHOD = 2 Case. Card 2 for METHOD=2

Cards 2f	1	2	3	4	5	6	7	8
Variable	MSIZE	RADIUS	NID1	NID2				
Type	F	F	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
MSIZE	Mesh size that needs to be applied in the zone of the shape defined by SNAME
RADIUS	Radius of the sphere if SNAME is sphere or radius of the cylinder if SNAME is cylinder. Ignored otherwise.
NID1/NID2	User node IDs that come from the solid mechanics problem. Equivalent usage to PTID1 and PTID2. NID1 replaces PMIN*i* for box and cylinder, X/Y/Z for pol, and CENTER for sphere. NID2 is ignored if SNAME is sphere. It replaces PMAX*i* for box and cylinder and NX/NY/NZ for pol.

Remarks:

1. **Moving coordinates.** METHOD = 2 and 3 have the advantage of allowing the region to move since the regions move with *ICFD_DEFINE_POINTs and nodes from the solid mechanics' problem, respectively. Thus, these methods provide control over how the mesh size area evolves as a function of time (in cases where remeshing occurs).

*MESH_SURFACE_ELEMENT

Purpose: Specify a set of surface elements (quadrilateral or triangular in 3D and linear segments in 2D) that will be used by the mesher to construct a volume mesh. These surface elements may define the enclosed volume to be meshed, or alternatively they can be used to specify different mesh sizes inside the volume (see card *MESH_SIZE**).**

Each solver that supports the *MESH volume mesher for building its volume mesh uses the PID given for each surface element specified with this keyword differently.

1. For the *ICFD solver, the *ICFD_PART keyword references the surface mesh PIDs.

2. For the *DUALCESE solver, the MSPIDs in *DUALCESE_BOUNDARY_... keywords cards reference the surface element PIDs. Note that when the dual CESE solver uses *MESH_SURFACE_ELEMENT, this card defines which boundary faces belong to each mesh surface PID. In this case, no other mechanism exists for defining these PIDs. Also, when the dual CESE solver is defined with *MESH cards, there should not be any *DUALCESE_SEGMENTSET cards related to the dual CESE mesh.

Surface Element Card. Define as many cards as necessary. The next keyword ("*") card terminates this input.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	EID	PID	N1	N2	N3	N4				
Type	I	I	I	I	I	I				
Default	none	none	none	none	none	none				

VARIABLE	DESCRIPTION
EID	Element ID. A unique number with respect to all *MESH_SURFACE_ELEMENTS cards.
PID	Mesh surface part ID. A unique identifier for the surface to which this mesh surface element belongs.
N1	Nodal point 1
N2	Nodal point 2

VARIABLE	DESCRIPTION
N3	Nodal point 3
N4	Nodal point 4

Remarks:

1. **Defining Surface Elements.** The convention for defining surface elements is the same as for *ELEMENT_SHELL. In the case of a triangular face, N3 = N4. In 2D N2 = N3 = N4. Note that the accepted card format is 6i8 (not 6i10).

*MESH_SURFACE_NODE

Purpose: Define a node and its coordinates. These nodes will be used in the mesh generation process by the *MESH_SURFACE_ELEMENT keyword.

*MESH_NODE supersedes this card, so use *MESH_NODE instead of this card.

Surface Node Cards. Include one card for each node. Include as many cards a necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	NID	X		Y		Z				
Type	I	F		F		F				
Default	none	0.0		0.0		0.0				

VARIABLE	DESCRIPTION
NID	Node ID. This NID must be unique within the set of surface nodes.
X	*x* coordinate
Y	*y* coordinate
Z	*z* coordinate

*MESH_SURFACE_NULL

Purpose: Specify a set of surface part IDs to be used during post-processing. The part IDs are defined with the *MESH_SURFACE_ELEMENT keyword. Elements from *MESH_SURFACE_ELEMENT that belong to a *MESH_SURFACE_NULL part are not used by the mesher to construct a volume mesh. These surface elements are not involved in any physics computation.

For the *ICFD solver, the *ICFD_PART keyword referencing the null surface mesh PIDs is not necessary.

Null Surface Element Card. Define as many cards as necessary. The next keyword ("*") card terminates this input.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	PID1	PID2	PID3	PID4	PID5	PID6				
Type	I	I	I	I	I	I				
Default	none	none	none	none	none	none				

VARIABLE	DESCRIPTION
PID*i*	Mesh surface part ID *i*

*MESH_VOLUME

Purpose: Specify the volume space that will be meshed. The boundaries of the volume are the surfaces defined by *MESH_SURFACE_ELEMENT. The listed surfaces must be non-overlapping and should not leave any gaps or open spaces between their boundaries. On the boundary between two neighboring surfaces, nodes must be in common (no duplicate nodes) and should match exactly at the interface. They are defined by the keyword *MESH_NODE. This card will be ignored if the volume mesh is specified by the user and not generated automatically.

Card 1	1	2	3	4	5	6	7	8
Variable	VOLID							
Type	I							
Default	none							

Define as many cards as are necessary based on the number of PIDs (the next "*" card terminates the input.)

Card 2	1	2	3	4	5	6	7	8
Variable	PID1	PID2	PID3	PID4	PID5	PID6	PID7	PID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
VOLID	ID assigned to the new volume
PID*n*	Part IDs for the surface elements that are used to define the volume

*MESH_VOLUME_ELEMENT

Purpose: Specify a set of volume elements for the fluid volume mesh in cases where the volume mesh is specified by the user and not generated automatically. The nodal points are specified with *MESH_NODE. Only tetrahedral elements are supported (triangles in 2D).

Volume Element Card. Define as many cards as necessary. The next keyword ("*") card terminates the input.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	EID	PID	N1	N2	N3	N4				
Type	I	I	I	I	I	I				
Default	none	none	none	none	none	none				

VARIABLE	DESCRIPTION
EID	Element ID. A unique number with respect to all *MESH_VOLUME_ELEMENTS cards.
PID	Part ID. A unique part identification number.
N1	Nodal point 1
N2	Nodal point 2
N3	Nodal point 3
N4	Nodal point 4

Remarks:

1. **Data card input.** The data card input convention is the same used by the keyword *ELEMENT_SOLID.

*MESH_VOLUME_NODE

Purpose: Define a node and its coordinates. This keyword is only used in cases where the user provides the fluid volume mesh, and it is not automatically generated. It serves the same purpose as the *NODE keyword for solid mechanics. Only tetrahedral elements are supported.

*MESH_NODE supersedes this card, so use *MESH_NODE instead of this card.

Volume Node Cards. Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	NID	X		Y		Z				
Type	I	F		F		F				
Default	none	0.0		0.0		0.0				

VARIABLE	DESCRIPTION
NID	Node ID. A unique number with respect to the other volume nodes.
X	*x* coordinate
Y	*y* coordinate
Z	*z* coordinate

*MESH_VOLUME_PART

Purpose: Associate a volume part number created by a *MESH_VOLUME card with the part number of a part card from a selected solver (designated by the SOLVER field).

Mesh Volume Part Card. Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	VOLPRT	SOLPRT	SOLVER					
Type	I	I	A					
Default								

VARIABLE	DESCRIPTION
VOLPRT	Part ID of a volume part created by a *MESH_VOLUME card.
SOLPRT	Part ID of a part created using SOLVER's part card.
SOLVER	Name of a solver using a mesh created with *MESH cards.

*STOCHASTIC

The keyword *STOCHASTIC is used to describe the particles and numerical details for solving a set of stochastic PDEs. Currently, there are two types of stochastic PDE models in the code: a spray model and a model of embedded particles in TBX explosives. The keyword cards for using these models are:

*STOCHASTIC_SPRAY_PARTICLES

*STOCHASTIC_TBX_PARTICLES

An additional option "_TITLE" may be appended to all *STOCHASTIC keywords. If this option is used, then an 80 character string is read as a title from the first card of that keyword's input. At present, LS-DYNA does not make use of the title. Inclusion of titles gives greater clarity to input decks.

*STOCHASTIC_SPRAY_PARTICLES

Purpose: Specify particle and other model details for spray modeling using stochastic PDEs that approximate such processes. A pair of cards is required to specify the characteristics of each nozzle (cards 3 and 4 describe the first nozzle).

Card 1	1	2	3	4	5	6	7	8
Variable	INJDIST	IBRKUP	ICOLLDE	IEVAP	IPULSE	LIMPR	IDFUEL	
Type	I	I	I	I	I	I	I	
Default	1	none	none	0	none	none	1	

Card 2	1	2	3	4	5	6	7	8
Variable	RHOP	TIP	PMASS	PRTRTE	STRINJ	DURINJ		
Type	F	F	F	F	F	F		

Nozzle card 1: Provide as many pairs of nozzle cards 1 and 2 as necessary. This input ends at the next keyword ("*") card (following a nozzle card 2).

Card 3	1	2	3	4	5	6	7	8
Variable	XORIG	YORIG	ZORIG	SMR	VELINJ	DRNOZ	DTHNOZ	
Type	F	F	F	F	F	F	F	

Nozzle card 2: Provide as many pairs of nozzle cards 1 and 2 as necessary. This input ends at the next keyword ("*") card.

Card 4	1	2	3	4	5	6	7	8
Variable	TILTXY	TILTXZ	CONE	DCONE	ANOZ	AMP0		
Type	F	F	F	F	F	F		

VARIABLE	DESCRIPTION
INJDIST	Spray particle size distribution: EQ.1: uniform EQ.2: Rosin-Rammler (default) EQ.3: Chi-squared degree of 2 EQ.4: Chi-squared degree of 6
IBRKUP	Type of particle breakup model: EQ.0: off (no breakup) EQ.1: TAB EQ.2: KHRT
ICOLLDE	Turn collision modeling on or off
IEVAP	Evaporation flag: EQ.0: off (no evaporation) EQ.1: Turn evaporation on (see Remark 1)
IPULSE	Type of injection: EQ.0: continuous injection EQ.1: sine wave EQ.2: square wave
LIMPRT	Upper limit on the number of parent particles modeled in this spray. This is not used with the continuous injection case (IPULSE = 0).

VARIABLE	DESCRIPTION
IDFUEL	Selected spray liquid fuels: EQ.1: (Default), H_2O EQ.2: Benzene, C_6H_6 EQ.3: Diesel # 2, $C_{12}H_{26}$ EQ.4: Diesel # 2, $C_{13}H_{13}$ EQ.5: Ethanol, C_2H_5OH EQ.6: Gasoline, C_8H_{18} EQ.7: Jet-A, $C_{12}H_{23}$ EQ.8: Kerosene, $C_{12}H_{23}$ EQ.9: Methanol, CH_3OH EQ.10: N-dodecane, $C_{12}H_{26}$
RHOP	Particle density
TIP	Initial particle temperature.
PMASS	Total particle mass
PRTRTE	Number of particles injected per second for continuous injection.
STRINJ	Start of injection(s)
DURINJ	Duration of injection(s)
XORIG	X-coordinate of center of a nozzle's exit plane
YORIG	Y-coordinate of center of a nozzle's exit plane
ZORIG	Z-coordinate of center of a nozzle's exit plane
SMR	Sauter mean radius
VELINJ	Injection velocity
DRNOZ	Nozzle radius
DTHNOZ	Azimuthal angle (in degrees measured counterclockwise) of the injector nozzle from the j = 1 plane.

VARIABLE	DESCRIPTION
TILTXY	Rotation angle (in degrees) of the injector in the x-y plane, where 0.0 points towards the 3 o'clock position (j = 1 line), and the angle increases counterclockwise from there.
TILTXZ	Inclination angle (in degrees) of the injection in the x-z plane, where 0.0 points straight down, x > 0.0 points in the positive x direction, and x < 0.0 points in the negative x direction.
CONE	Spray mean cone angle (in degrees) for hollow cone spray; spray cone angle (in degrees) for solid cone spray.
DCONE	Injection liquid jet thickness in degrees.
ANOZ	Area of injector
AMP0	Initial amplitude of droplet oscillation at injector

Remarks:

1. When IEVAP = 1, the keyword input file must be modified in a fashion similar to a chemistry problem. This is illustrated in a portion of an example keyword file below. That is, the following keywords need to be used, along with the inclusion of other chemistry-related files (i.e. evap.inp and the corresponding thermodynamics data file):

   *CHEMISTRY_MODEL

   *CHEMISTRY_COMPOSITION

   *CHEMISTRY_CONTROL_FULL

   *CESE_INITIAL_CHEMISTRY

```
$ Setup stochastic particles
$
*STOCHASTIC_SPRAY_PARTICLES
$  injdist    ibrkup  icollide     ievap    ipulse    limprt    fuelid
         3         1         0         1         0    100000         1
$     rhop       tip pmass[Kg]    prtrte   str_inj   dur_inj
    1000.0      300.      0.01     1.0e7       0.0      10.0
$  the next card is needed for fireball position and max.  particle velocity:
$    XORIG     YORIG     ZORIG       SMR   Velinj      Drnoz    Dthnoz
     0.005     0.005    1.0e-5    5.0e-6    200.0     9.0e-5
$   TILTXY    TILTXZ      CONE     DCONE      ANOZ       AMP0
       0.0       0.0      15.0      15.0   2.5e-8        0.0
$
*CHEMISTRY_MODEL
$ model_id    jacsel    errlim
```

```
        10         1       0.0
  evap.inp
 therm.dat
  tran.dat
$
*CHEMISTRY_COMPOSITION
$  comp_id  model_id
        11        10
$  molefra   Species
       1.0        O2
      3.76        N2
$
*CHEMISTRY_CONTROL_FULL
$   sol_id    errlim
         5
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Set global initial conditions for fluid
$
*CESE_INITIAL_CHEMISTRY
$   sol_id   comp_id
         5        11
$INITIAL CONDITIONS
$      uic       vic       wic       ric       pic       tic       hic
       0.0       0.0       0.0       1.2   101325.     300.0       0.0
```

*STOCHASTIC_TBX_PARTICLES

Purpose: Specify particle and other model details for stochastic PDEs that model embedded particles in TBX explosives. Note that the components listed on the corresponding *CHEMISTRY_COMPOSITION card are in terms of molar concentrations of the species (in units of moles/[length]^3, where [length] is the user's length unit).

For further information on the theory of the TBX model that has been implemented, a document on this topic can be found at this URL:

https://lsdyna.ansys.com/cese-cfd-documentation/

Card 1	1	2	3	4	5	6	7	8
Variable	PCOMB	NPRTCL	MXCNT	PMASS	SMR	RHOP	TICP	T_IGNIT
Type	I	I	I	F	F	F	F	F
Default	0	none	none	none	none	none	none	none

Card 2	1	2	3	4	5	6	7	8
Variable	INITDST	AZIMTH	ALTITD	CPS/CVS	HVAP	EMISS	BOLTZ	
Type	I	F	F	F	F	F	F	
Default	1	none	none	none	none	optional	optional	
Remarks						1	1	

Card 3	1	2	3	4	5	6	7	8
Variable	XORIG	YORIG	ZORIG	XVEL	YVEL	ZVEL	FRADIUS	
Type	F	F	F	F	F	F	F	
Default	none	none	none	0.0	0.0	0.0	none	

VARIABLE	DESCRIPTION
PCOMB	Particle combustion model: EQ.0: No burning EQ.1: K-model
NPRTCL	Initial total number of parent particles (discrete particles for calculation)
MXCNT	Maximum allowed number of parent particles (during the simulation)
PMASS	Total particle mass
SMR	Sort mean particle radius
RHOP	Particle density
TICP	Initial particle temperature
T_IGNIT	Particle ignition temperature
INITDST	Initial particle distribution: EQ.1: Spatially uniform EQ.2: Rosin-Rammler EQ.3: Chi-squared
AZIMTH	Angle in degrees from the x-axis in the xy-plane of the reference frame of the TBX explosive (0 < AZMITH < 360)
ALTITD	Angle in degrees from the z-axis of the reference frame of the TBX explosive (0 < ALTITD < 180)
CPS/CVS	Heat coefficient
HVAP	Latent heat of vaporization
EMISS	Particle emissivity
BOLTZ	Boltzmann coefficient
XORIG	x-coordinate of the origin of the initial reference frame of the TBX explosive

VARIABLE	DESCRIPTION
YORIG	*y*-coordinate of the origin of the initial reference frame of the TBX explosive
ZORIG	*z*-coordinate of the origin of the initial reference frame of the TBX explosive
XVEL	*x*-component of the initial particle velocity of the TBX explosive
YVEL	*y*-component of the initial particle velocity of the TBX explosive
ZVEL	*z*-component of the initial particle velocity of the TBX explosive
FRADIUS	Radius of the explosive area

Remarks:

1. **Radiation heat transfer.** If radiation heat transfer is being modeled, then EMISS and BOLTZ are required.

*LSO

These cards provide a general data output mechanism, causing the creation of a sequence of LSDA files. This facility is intended to allow several different time sequences of data to be output in the same simulation. In addition, any number of domains (and any number of variables on those domains) may be specified within each time sequence. The keyword cards in this section are defined in alphabetical order:

*LSO_DOMAIN

*LSO_ID_SET (*not available in the single-precision version of LS-DYNA*)

*LSO_POINT_SET

*LSO_TIME_SEQUENCE

*LSO_VARIABLE_GROUP

Note that only the mechanics solver is available in the single-precision version of LS-DYNA, and therefore, only LSO mechanics variables are available for output from single precision LS-DYNA. These mechanics variables are listed by domain type in a separate document. This document (LSO_VARIABLES.TXT) is created by running the command: LS-DYNA print_lso_doc. Contrary to LSO_VARIABLES.TXT, element quantities such as stress are not available for output from the mechanics solver to the "lso" database.

An additional option "_TITLE" may be appended to all *LSO keywords. If this option is used, then an 80 character string is read as a title from the first card of that keyword's input. At present, LS-DYNA does not make use of the title. Inclusion of titles gives greater clarity to input decks.

*LSO_DOMAIN

Purpose: This command provides a way to specify variables on a subset of the domain for a given solver. This domain can be a subset of the mesh used by that solver, a set of output points created with *LSO_POINT_SET, or a set of objects created with *LSO_ID_SET. The frequency and duration of the output for any given domain is determined by each *LSO_TIME_SEQUENCE card that references this *LSO_DOMAIN card. Note that for the single-precision version of LS-DYNA, the only allowed value of SOLVER_NAME = MECH.

Card 1	1	2	3	4	5	6	7	8
Variable	DOMAIN_TYPE							
Type	A							

Card 2	1	2	3	4	5	6	7	8
Variable	SOLVER_NAME							
Type	A							

Special Domains Card. Card 3 when DOMAIN_TYPE is one of ROGO, CIRCUIT, THIST_POINT or TRACER_POINT.

Card 3	1	2	3	4	5	6	7	8
Variable	OUTID	REFID		REDUCT				
Type	I	I		I				
Default	none	none		none				

Miscellaneous Domain Card. Card 3 when DOMAIN_TYPE is one of NODE, PART, SEGMENT, SURFACE_NODE, SURFACE_ELEMENT, VOLUME_ELEMENT, SURFACE_PART, VOLUME_PART.

Card 3	1	2	3	4	5	6	7	8
Variable	OUTID	REFID	OVERRIDE	REDUCT				
Type	I	I	I	I				
Default	none	0	0	none				

Variable Name Card. Provide as many cards as necessary. This input ends at the next keyword ("*") card

Card 4	1	2	3	4	5	6	7	8
Variable	VARIABLE_NAME							
Type	A							

VARIABLE	DESCRIPTION
DOMAIN_TYPE	The type of domain for which LSO output may be generated.
SOLVER_NAME	Selects the solver from which data is output on this domain. Accepted entries so far are "MECH", "EM", "CESE", and "ICFD".
OUTID	LSO domain ID associated with this domain, and used by *LSO_TIME_SEQUENCE cards.
REFID	Support set ID. This can be a set defined by a *SET card, a *LSO_ID_SET, card, or a *LSO_POINT_SET card. Unless OVERRIDE is specified, this set must be of the same type as DOMAIN_TYPE.
OVERRIDE	If non-zero, then REFID is interpreted as: EQ.1: a PART set for SOLVER_NAME EQ.2: a PART set of volume parts created with a *LSO_ID_SET card (volume parts are defined with *MESH_VOLUME cards).

VARIABLE	DESCRIPTION
	EQ.3: a PART set of surface parts created with a *LSO_ID_SET card (surface parts are defined with *MESH_SURFACE_ELEMENT cards).
	EQ.4: a set of segment sets created with a *LSO_ID_SET card.
REDUCT	A function that operates on the entire domain and returns a single value for scalar variables, three values for vector variables, or 6 values for symmetric tensor variables. For REDUCT="range", the number of returned values doubles. The following are the supported functions: EQ.BLANK: no reduction (default) EQ."none": Same as above EQ."avg": the average by component EQ."average": Same as above EQ."min": the minimum by component EQ."minimum": Same as above EQ."max": the maximum by component EQ."maximum": Same as above EQ."sum": the sum by component EQ."range": the minimum by component followed by the maximum by component
VARIABLE_NAME	Either the name of a single output variable or a variable group. See remarks.

	EM
VECTORS	magneticF
	electricFie
	vecpotFie
	currentDens
SCALARS	ScalarPoter

Remarks:

1. Supported choices for VARIABLE_NAME are listed by DOMAIN_TYPE for each SOLVER_NAME in a separate document. This document (LSO_VARIABLES.TXT) is created by running the command: LS-DYNA print_lso_doc. The following table shows a sample of the point output variables available when DOMAIN_TYPE = THIST_POINT:

*LSO_ID_SET

Purpose: Provides a way to create a set of existing sets (segment sets), or to define a set that is not available with other set-related keyword cards. These are then used in other *LSO cards to specify LSO output. This card is not available in the single precision version of LS-DYNA.

Card 1	1	2	3	4	5	6	7	8
Variable	SETID	TYPE	SOLVER					
Type	I	A	A					
Default	none	none	MECH					

Referenced IDs. Provide as many cards as necessary. This input ends at the next keyword ("*") card

Card	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
SETID	Identifier for this ID set.

VARIABLE	DESCRIPTION
TYPE	The kind of IDs in this set: EQ.'SEG_SETS': Each ID is a segment set connected with SOLVER. EQ.'CIRCUIT': Each ID is a circuit ID (from *EM cards) EQ.'SURF_PARTS': Each ID is a surface part number (See *MESH_SURFACE_ELEMENT) EQ.'VOL_PARTS': Each ID is a volume part number (See *MESH_VOLUME) EQ.'SURF_ELES': Each ID is a surface element number (See *MESH_SURFACE_ELEMENT)
SOLVER	Name of the solver (MECH, ICFD, CESE, EM, …)
ID1, …	IDs of the TYPE kind.

*LSO_POINT_SET

Purpose: Define a list of points used to sample variables in time. Of the different sampling methods, the most common one is to specify points for time history output.

Card 1	1	2	3	4	5	6	7	8
Variable	SETID	USE						
Type	I	I						
Default	none	1						
Remarks		1						

Point Cards. Provide as many cards as necessary. This input ends at the next keyword ("*") card

Card 2	1	2	3	4	5	6	7	8
Variable	X	Y	Z					
Type	F	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
SETID	Identifier for this point set which is used by *LSO_DOMAIN
USE	Points in this set are used as: EQ.1: fixed time history points (default) EQ.2: positions of tracer particles
X, Y, Z	Coordinates of a point. As many points as desired can be specified.

Remarks:

1. **USE.** For USE = 1, with the ICFD and CESE solvers, the fixed points must remain inside the fluid mesh or a zero result is returned, while for the EM solver, the points can be defined inside the conductors or in the air. In the latter case, the fields will be computed using a Biot-Savart type integration. For USE = 2, a massless tracer particle is tracked for the ICFD and CESE solvers using their local velocity field to integrate the position of each particle in time.

*LSO_TIME_SEQUENCE

Purpose: This command provides users with maximum flexibility in specifying exactly what they want to have appear in the output LSO binary database. Each instance of the *LSO_TIME_SEQUENCE command creates a new time sequence with an independent output frequency and duration. Furthermore, while the default domain for each output variable will be the entire mesh on which that variable is defined, at all selected snapshot times, the *LSO_DOMAIN keyword commands can be used to specify that output will only occur on a portion of SOLVER_NAME's mesh, and for a limited time interval, or that it will occur at a set of points (see *LSO_POINT_SET), or over a set of object IDs (see *LSO_ID_SET). Note that for the single-precision version of LS-DYNA, the only allowed value of SOLVER_NAME = MECH.

Card 1	1	2	3	4	5	6	7	8
Variable	SOLVER_NAME							
Type	A							

Card 2	1	2	3	4	5	6	7	8
Variable	DT	LCDT	LCOPT	NPLTC	TBEG	TEND		
Type	F	I	I	I	F	F		
Default	0.0	0	1	0	0.0	0.0		
Remarks	1	1	1	1				

Domain IDs. Provide as many cards as necessary. This input ends at the next keyword ("*") card, or when a global variable name card appears

Card 3	1	2	3	4	5	6	7	8
Variable	DOMID1	DOMID2	DOMID3	DOMID4	DOMID5	DOMID6	DOMID7	DOMID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

Global variable names. Provide as many cards as necessary. This input ends at the next keyword ("*") card

Card 4	1	2	3	4	5	6	7	8
Variable	GLOBAL_VAR							
Type	A							

VARIABLE	DESCRIPTION
SOLVER_NAME	Selects the solver from which data is output in this time sequence. Accepted entries so far are 'MECH', 'EM', 'CESE' and 'ICFD'
DT	Time interval between outputs.
LCDT	Optional load curve ID specifying the time interval between dumps.
LCOPT	Flag to govern behavior of plot frequency load curve: EQ.1: At the time each plot is generated, the load curve value is added to the current time to determine the next plot time (this is the default behavior). EQ.2: At the time each plot is generated, the next plot time T is computed so that T = the current time plus the load curve value at the time T. EQ.3: A plot is generated for each ordinate point in the load curve definition. The actual value of the load curve is ignored.

VARIABLE	DESCRIPTION
NPLTC	DT = ENDTIM/NPLTC overrides DT specified in the first field.
TBEG	The problem time at which to begin writing output to this time sequence
TEND	The problem time at which to terminate writing output to this time sequence
DOMID1, …	Output set ID defining the domain over which variable output is to be performed in this time sequence. Each DOMID refers to the domain identifier in an *LSO_DOMAIN keyword card.
GLOBAL_VAR	The name of a global output variable computed by SOLVER_NAME. This variable must have a single value (scalar, vector, or tensor), and therefore does not depend upon any DOMID. Any number of such variables may be specified with a given time sequence. These variables are listed as having "global" domain for SOLVER_NAME in a separate document. This document (LSO_VARIABLES.TXT) is created by running the command: LS-DYNA print_lso_doc.

Remarks:

1. If LCDT is nonzero, then it is used and DT and NPLTC are ignored. If LCDT is zero and NPLTC is non-zero, then NPLTC determines the snapshot time increment. If LCDT and NPLTC are both zero, then the minimum non-zero time increment specified by DT is used to determine the snapshot times.

*LSO_VARIABLE_GROUP

Purpose: To provide a means of defining a shorthand name for a group of variables. That is, wherever the given group name is used, it is replaced by the list of variables given in this command. Note that for the single-precision version of LS-DYNA, the only allowed value of SOLVER_NAME = MECH.

Card 1	1	2	3	4	5	6	7	8
Variable	SOLVER_NAME							
Type	A							

Card 2	1	2	3	4	5	6	7	8
Variable	DOMAIN_TYPE							
Type	A							

Card 3	1	2	3	4	5	6	7	8
Variable	GROUP_NAME							
Type	A							

List Of Variables In Group. Provide as many cards as necessary. This input ends at the next keyword ("*") card

Card 4	1	2	3	4	5	6	7	8
Variable	VAR_NAME							
Type	A							

VARIABLE	DESCRIPTION
SOLVER_NAME	Selects the solver for which data is output in a time sequence.

VARIABLE	DESCRIPTION
DOMAIN_TYPE	Name of the type of domain on which each VAR_NAME is defined.
GROUP_NAME	Name of (or alias for) the group of names given by the listed VAR_NAMEs
VAR_NAME	The name of an output variable computed by SOLVER_NAME

Remarks:

1. Valid VAR_NAMEs depend both upon the SOLVER_NAME and the DOMAIN_TYPE. These variables are listed by DOMAIN_TYPE for each SOLVER_NAME in a separate document. This document (LSO_VARIABLES.TXT) is created by running the command: LS-DYNA print_lso_doc.