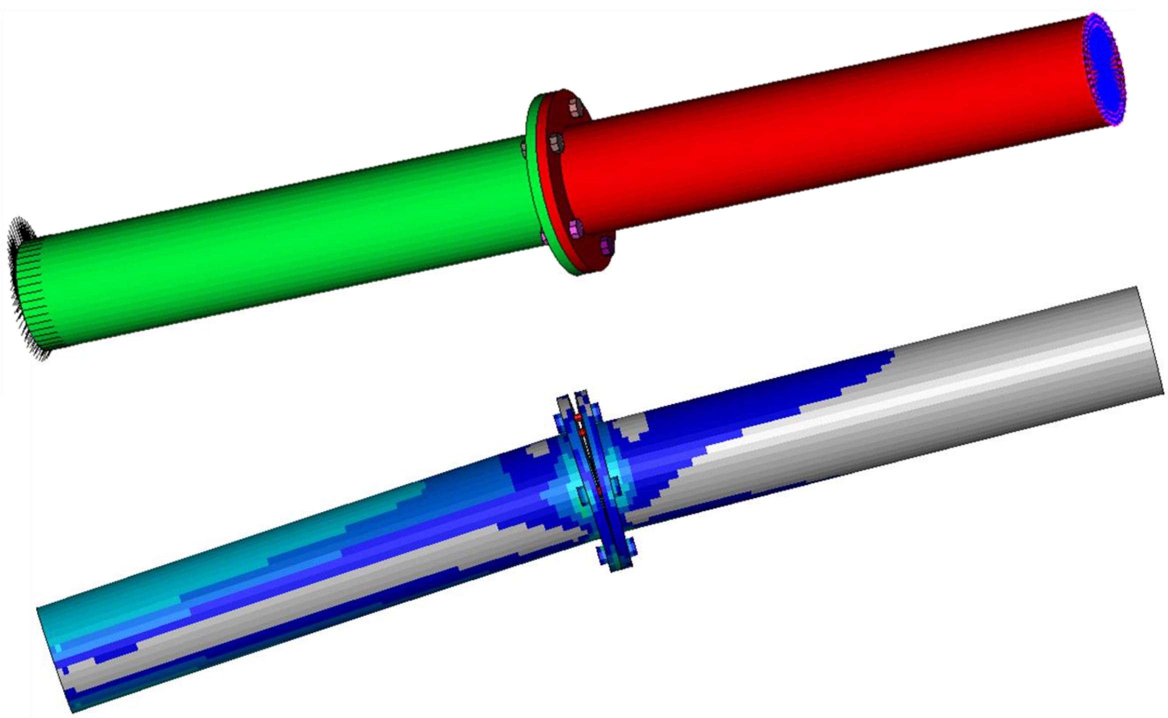
Best Practice

Guideline for Implicit Analyses Using Ansys LS-DYNA Software



2025R1

Table of content

Abstract

The purpose of this document is to reduce the effort of getting started with implicit analysis in LS-DYNA. Basic recommendations on element types, contact formulations, loads and boundary conditions are provided, as well as some small examples.

This document is under continuous development, and future improved revisions will be released.

Disclaimer

By using this Guideline, you hereby consent to this disclaimer and agree to its terms.

All the information in this Guideline, comprised of this document and the accompanying simulation models, is published in good faith and for general information purposes only. Neither Ansys, DYNAmore Nordic AB nor the authors make any warranties about the completeness, reliability, and accuracy of the information in this Guideline. Any action you take upon the information you find in this Guideline is strictly at your own risk. Neither Ansys, DYNAmore Nordic AB nor the authors will be liable for any losses and/or damages in connection with the use of the Guideline. It is always up to the user of this Guideline to verify the results.

1 Introduction

The purpose of the present document is to provide a starting point[1] for setting up implicit analyses in Ansys LS-DYNA® software, with respect to control card settings, element formulations, material models etc. Some elementary examples are also presented. It is assumed that the reader has some previous knowledge of LS-DYNA.

It is recommended to use an include-file structure for the FE-model files, as outlined in Figure 1. By this, the provided include files for control and database cards can be utilized directly. The files are intended for use with LS-DYNA R12.2.1, R14.1, R15.0.2 [1], or later. The control card settings have been developed and tested for everyday use and found to work well for a wide range of applications. Still there may be specific situations where modifications of the provided basic settings will be required. Then at least, the provided include files may serve as a starting point for further modifications.

For users with experience from other implicit FE-solvers, the extensive use of control-cards in LS-DYNA, such as `*CONTROL_IMPLICIT_`..., with many different parameter settings, may seem overwhelming. All the values and flags that must be specified might be seen as a threshold to new LS-DYNA users. By using the supplied include-files with suggested control card settings, the effort of getting started with implicit analyses in LS-DYNA is hopefully reduced.

The present document is aimed at general implicit analyses, without any specific application focus.

For thorough details regarding LS-DYNA keywords and material models, the reader is referred to Ref [1][2]. Appendix P of the Keyword manual [1] contains a detailed description of implicit analyses in LS-DYNA.

An extensive benchmark of linear and non-linear static implicit analyses in LS-DYNA according to DNV Recommended Practice C208 is presented in Ref. [7]. An overview of the implicit capabilities of LS-DYNA is presented in Ref. [47].

For general support, see lsdyna.ansys.com/knowledge-base/. Useful publications from LS-DYNA users and developers may be found on lsdyna.ansys.com/conference-papers/. Example keyword files can be found at lsdyna.ansys.com/knowledge-base/implicit and lsdyna.ansys.com/knowledge-base/nvh-fatigue. For further questions, or if errors are found in this document, please contact your local Ansys LS-DYNA supplier.

Training courses and webinars in implicit analyses using LS-DYNA are available at the Ansys Learning Hub, or on demand from your local Ansys LS-DYNA supplier. Courses in contacts and material modelling are also available. The present document is based on the course notes [4] developed by Dr. Thomas Borrvall and others.

[1] It might very well be the case that modifications or alternative settings are required to better fit a particular application.

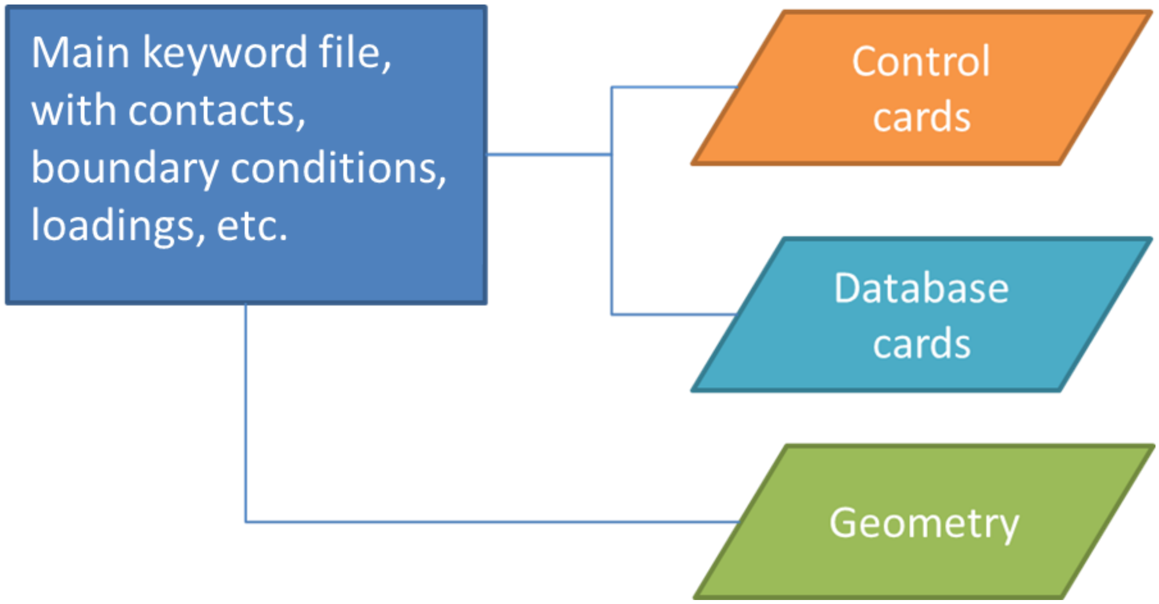


Figure 1. Recommended include-file structure for convenient use of the provided control cards files.

1.1 Workflows for setting up implicit analyses

Historically, keyword-based (or text input) approaches have been widely applied for setting up LSDYNA analyses. This means that the mesh is created in some preprocessor, and then handled as a separate include file. The main keyword file collects include files and contains the load case description, contacts, boundary conditions etc. The main keyword file is then modified using a text editor. The keyword-based approach is assumed in this Guideline, but recently GUI-based approaches are available, which guides the user through the setup, for example the Solution Explorer of LS-PrePost, see Section 1.1.1, or Ansys Workbench, see Section 1.1.2.

1.1.1 LS-PrePost Solution Explorer

From version 4.6 of LS-PrePost, the Solution Explorer, a dedicated GUI for setting up implicit analyses [19] is available, see Figure 2. It is based mainly on general engineering concepts and is not so focused on keywords. With this tool, the user can easily set up different types of implicit analyses in the Ansys LS-DYNA software. The user will be guided with regards to control card setting, element types, contacts etc. Using the Solution Explorer, the recommended settings according to Appendix P of Ref. [1] will be obtained, along with recommended contact settings, elements, materials etc. for implicit analyses.

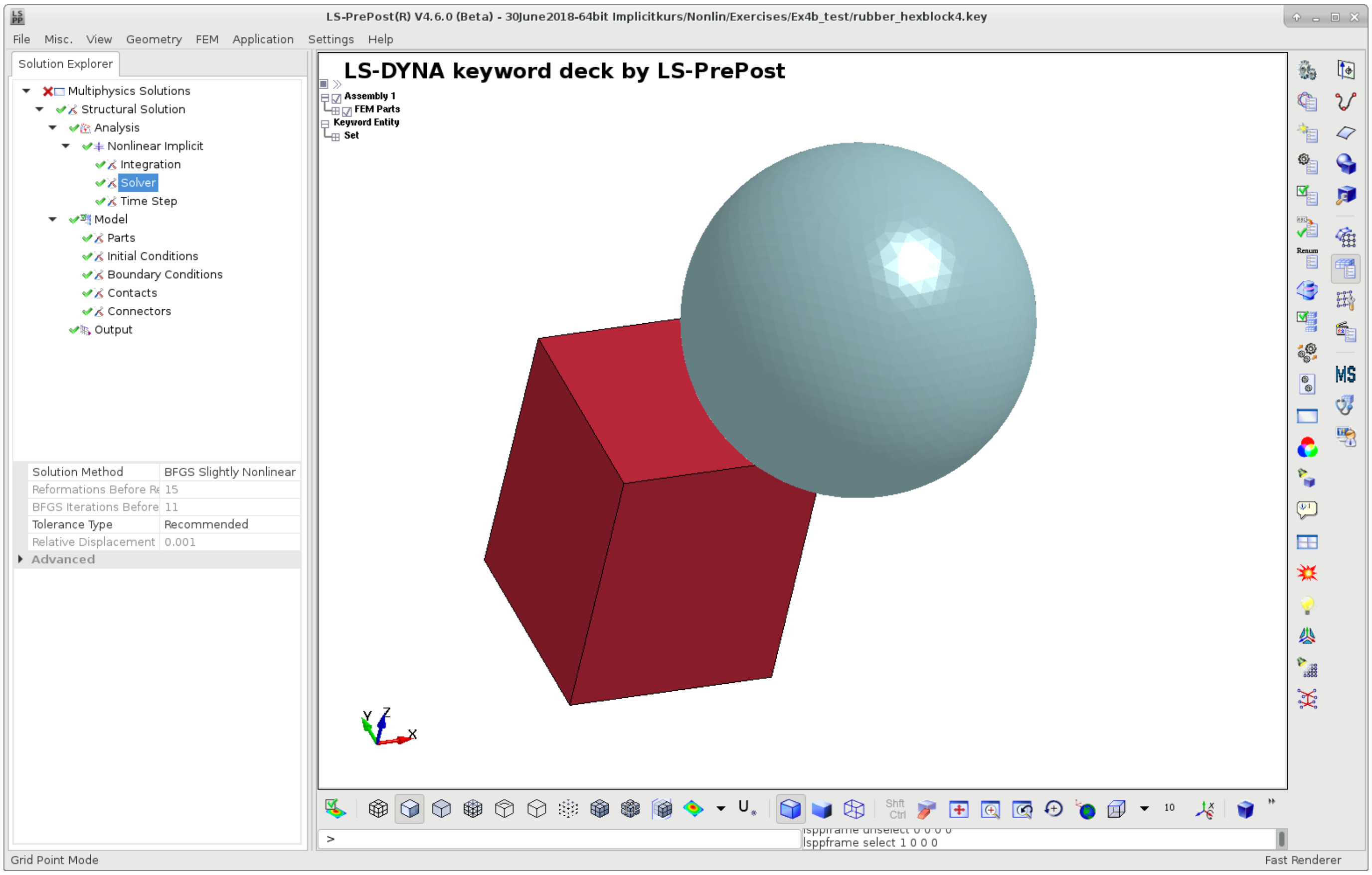


Figure 2. Screenshot from the Implicit GUI in LS-PrePost 4.6.

1.1.2 Ansys Workbench

It is possible to use Ansys Workbench / Ansys Mechanical for setting up implicit LS-DYNA analyses, see Figure 3. To activate implicit analysis, select Analysis Settings > Explicit Solution Only > No. Currently, in 2025R1, the control card settings provided by default by Ansys Mechanical are not in line with the recommendations of this Guideline and Appendix P of the LS-DYNA Keyword Manual. The default control card settings from Ansys Mechanical can be disabled, by Analysis Settings > Advanced > Default Solver Control Cards > Omit. Include files can also be used in the tree-based approach of Ansys Mechanical, by adding Keyword Snippets (Analysis Settings > Insert Command), see Figure 4. By this approach, it is possible to use the control card include files provided by this Guideline also when using Ansys Workbench for model setup.

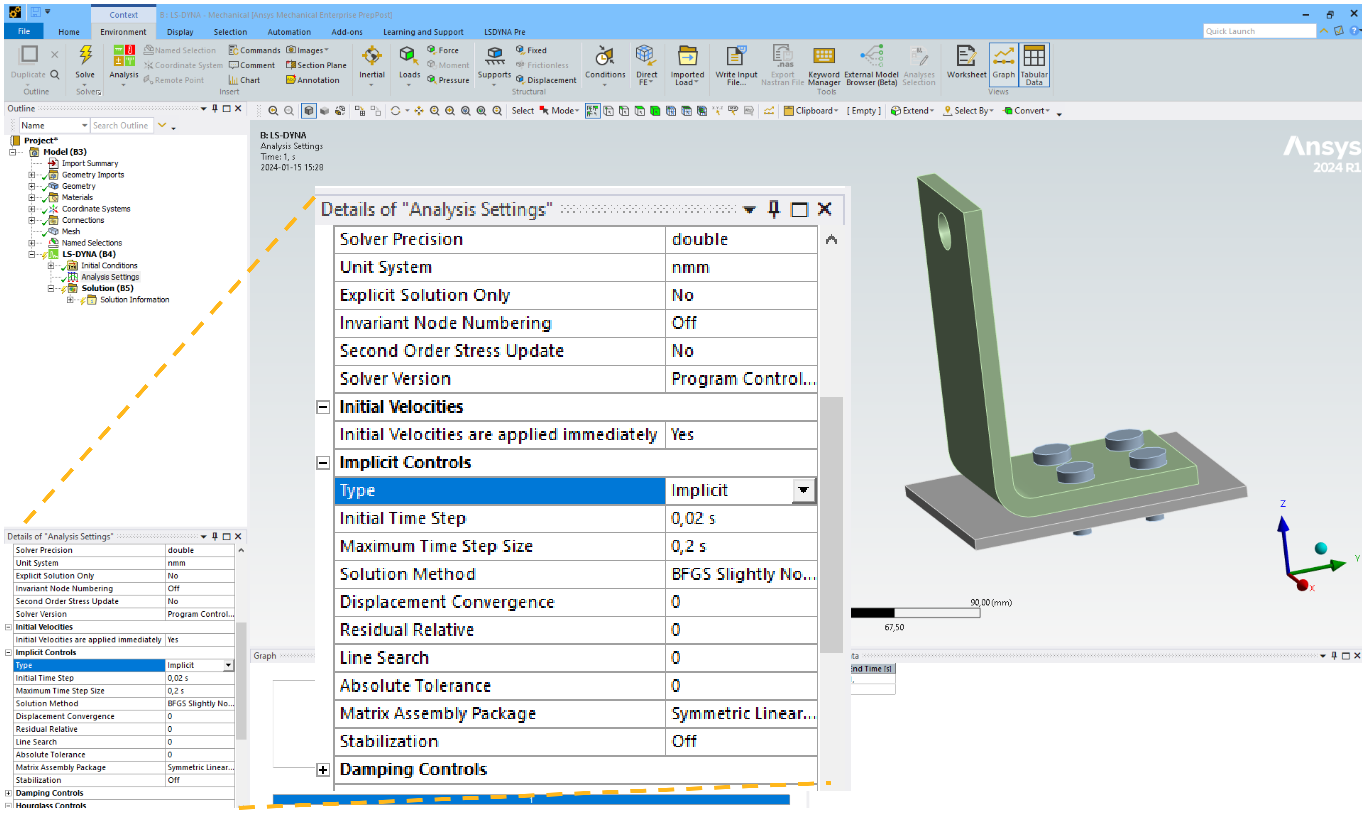


Figure 3. Implicit set up in Ansys Workbench 2024R1. A zoomed-in view of the Analysis Settings is shown.

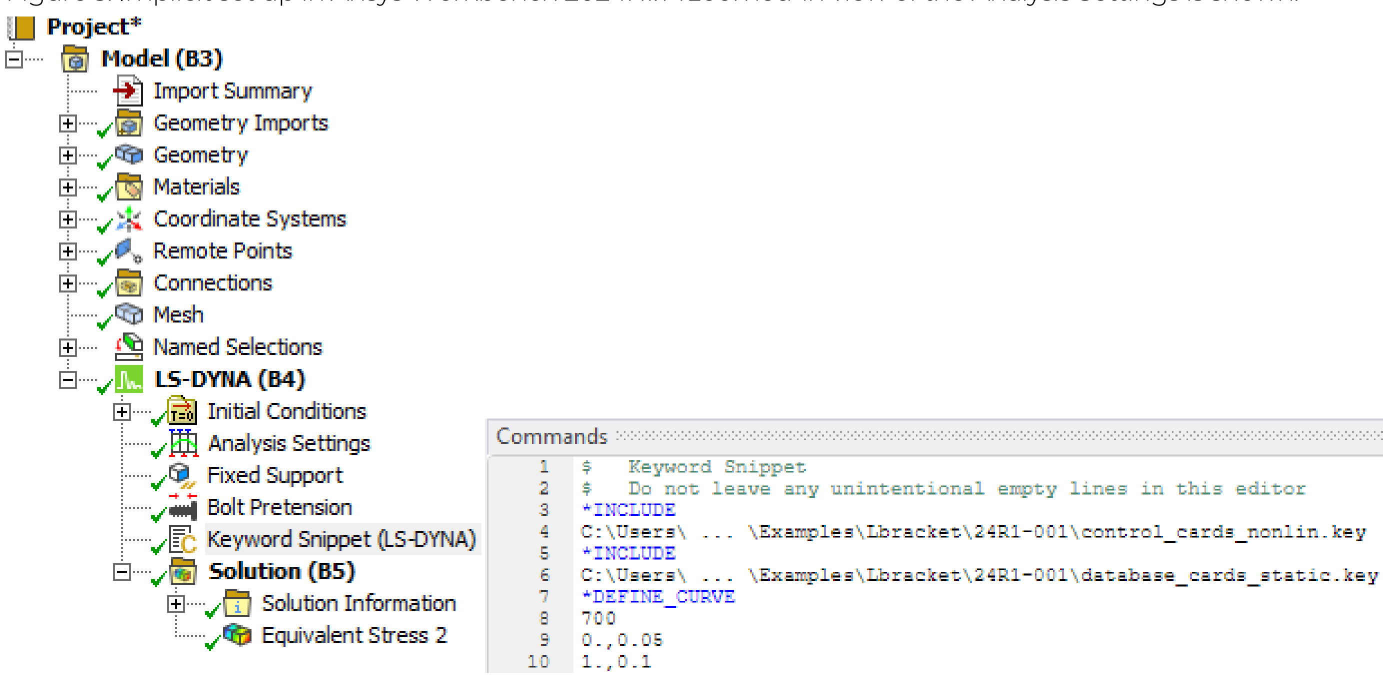


Figure 4. Inserting a Keyword Snippet makes it possible to work with include files also in Ansys Workbench.

2 Overview

In Section 3, output options are briefly discussed. How to set up some common implicit analysis types is presented in Section 4. In Section 5, some recommendations regarding element types for implicit analyses are given. Different contact types are discussed in Section 6. In Section 7, material models for implicit analyses are discussed. Loads and boundary conditions are briefly treated in Section 8.

A listing of appended keyword files can be found in Section Attachments (page 130).

3 LS-DYNA database cards for different analysis types

Results from an LS-DYNA analysis are written in a set of separate output files, see Figure 5 for an overview. The solution diagnostics files (`d3hsp`, `mes*`, `glstat`) are in ASCII format and are useful for tracking the progress of the solution. The files for 3D visualization are in binary format and can be opened using for example LS-PrePost or META. The main results file for visualization of deformation and stresses etc. is the `d3plot` file (or `d3eigv` from eigenvalue analyses). History data (for curve plotting) are output as binary files (`binout*`) from mpp/LS-DYNA, or a family of ASCII – files (`nodout`, `elout`, `rcforc` …) from smp/LS-DYNA.

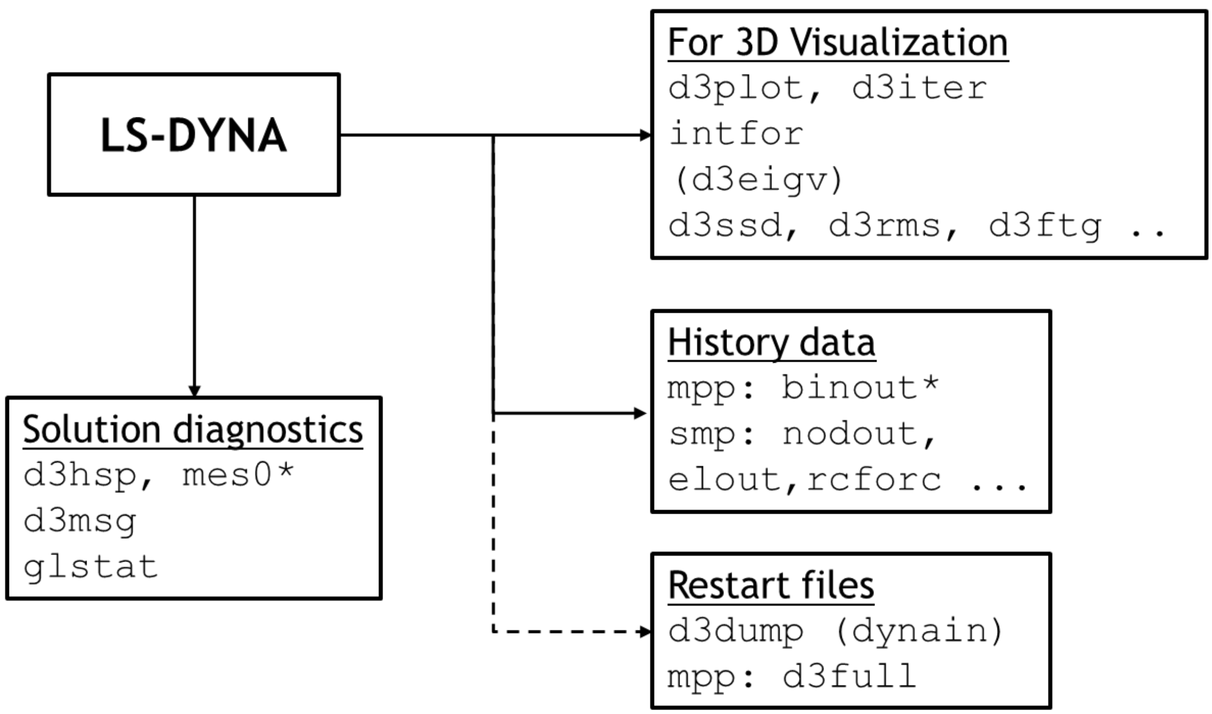


Figure 5. Overview of LS-DYNA output files.

Include files for controlling the output from implicit analyses are appended to the present package. The amount of output provided should be sufficient for most analysis types, but the user is advised to study the contents of the include files and check if some special output type required for a particular analysis type is missing.

In the file `database_cards_static.key`, output frequency is set very high, in order to ensure output after each time step of the analysis. In the file `database_cards_dynamic.key`, output frequency is reduced in order to get output at times that makes physical sense. Please note that the time scale used in this case is ms (10^{-3} s). If other unit systems are used, the user must modify the output frequencies to sensible values.

By default, LS-DYNA will output element results (stresses, strains) averaged at element centroids. Output of element integration point quantities is discussed in Section 5.4. Contact output is discussed in Section 6.3.

For analyses in R10.1 or prior, the include file `database_cards_before_R102.key` should be used. The reason for this is related to contact output, see Section 6.3.

4 Implicit analysis procedures

In this section, different types of implicit analyses are discussed, and control card settings are presented in the form of include files. A quick selection guide, showing which include files to use for different analysis procedures, is presented in Table 1. Example keyword files are also presented in the following Sections.

Note that a double precision version of LS-DYNA is required for implicit analyses. In general, the mpp version of LS-DYNA is recommended, but all features discussed, except for the linear transient modal dynamics (Section 4.7), are available in both mpp and smp versions of LS-DYNA. Linear transient modal dynamics is only available in smp. Analyses in 2D are normally performed using smp-LS-DYNA (double precision).

It should be pointed out that these control card settings are merely recommendations. The user is advised to study the contents of the include files and verify that the settings seem reasonable for a particular application. It might very well be the case that other settings give better results. Some comments to the implicit control card settings of the provided include files are given in Appendix G. Alternative control card settings for implicit analyses in LS-DYNA are presented in for example Refs. [17][18].

In some cases, it might be useful to switch to the explicit solver (see Appendix D and F), and for this purpose also a keyword file with control cards for explicit analyses is provided: `control_database.k`, taken from the Explicit guideline [22], which also contains template keyword files for different explicit analysis types.

For thermal analyses, additional control cards are required, see further Section 4.11.

Table 1. Guide for quick selection of include files for different analysis procedures

Analysis type	Include files control_cards_	database_cards_	Add keywords:(*) *CONTROL_IMPLICIT_
Linear static	linear.key	static.key	
Non-linear static	nonlin.key	static.key	
Linear buckling	(1)	static.key	BUCKLE
Non-linear postbuckling	arc.key	static.key	
Eigenfrequency analysis	(1)	static.key	EIGENVALUE
Linear transient modal dynamics	linear.key	dynamic.key	MODAL_DYNAMIC
Frequency domain analyses(2)	linear.key		
Non-linear implicit dynamics	nonlin.key	dynamic.key	DYNAMICS

Notes: (*) *CONTROL_TERMINATION must always be added. (1) Can be part of both linear and non-linear analysis. (2) Frequency response functions, steady state dynamics etc., see Section 4.8.

4.1 Linear static analysis

From a solution viewpoint, linear static analysis is perhaps the most basic FE-analysis type. To be brief, it consists of the following steps in the solver:

1. Form the stiffness matrix **K** by linearization at the initial configuration.
2. Solve for the displacements **u**: **Ku** = **F**, where **F** is the applied forces.
3. Compute the (engineering) strains from the displacements and stresses via the material routines.

In LS-DYNA, this is a purely linear procedure. The linear equation system is simply solved, as is. Small deformations and small strains are the basic assumption. No iterations or checks of convergence with respect to residuals or equilibrium are performed. If non-linear material models are used, they will be linearized at the initial configuration, and the corresponding tangent stiffness will be used in step 1. But the stresses and strains will still be computed based on the obtained displacements using the corresponding (non-linear) material and element routines, as a postprocessing step. If consistent estimates of stresses and strains from a linear analysis are of interest, it is highly recommended to use a linear elastic material model (*MAT_ELASTIC, see also Section 7) and an element type suited for linear analyses (see for example Table 5). Should the applied force cause a deformation that is not consistent with the assumption of small strains, erroneous results may be obtained.

In the Ansys LS-DYNA software, a set of beam, shell and solid elements have been specifically developed for linear implicit analyses, see further Section 5.

In a linear static analysis, sufficient boundary conditions are required in order to prevent rigid body modes or mechanisms, or the solution step 2 above may result in spurious deformations.

Tied contacts are valid in linear analyses, see also Section 6.2. Other contacts should be used with great care, see further Section 6.1.7. Non-linear constraints in LS-DYNA, such as joints, will be linearized, and thus the results in a linear analysis will not be correct.

The keyword file `control_cards_linear.key` contains control card settings which are generally well suited for linear static analyses. A template for a linear static analysis follows:

```
*KEYWORD
*INCLUDE
control_cards_linear.key
*INCLUDE
database_cards_static.key
*CONTROL_TERMINATION
Define end time of the simulation
*INCLUDE
Include file defining geometry, materials etc.
*LOAD_...
Define nodal loads etc.
*BOUNDARY_...
Data line to prescribe boundary conditions
*TITLE
Simulation title
*END
```

To solve a sequence of independent load cases, see for example Figure 6, without re-forming the stiffness matrix for each load case, it is recommended to set *NSOLVR* = -1 on `*CONTROL_IMPLICIT_SOLUTION` (available from R10.1). The result from each load case will then be output as a new `d3plot` state in an individual file.

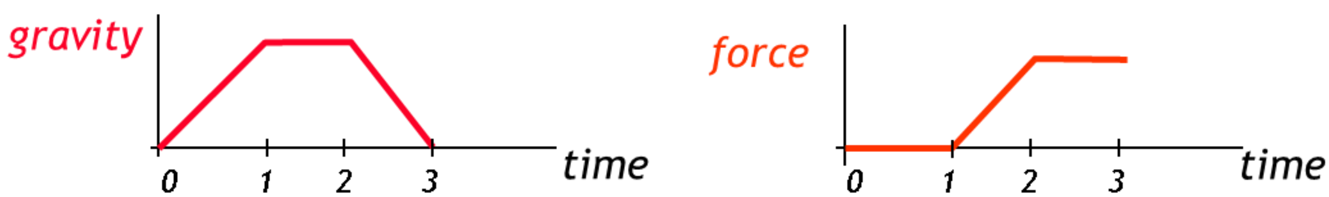


Figure 6. Definition of multiple load cases for linear static analyses. Each time (t=1, 2, ...) corresponds to a new load case. Note that $\Delta t = 1$.

A template keyword file follows:

```
*KEYWORD
*INCLUDE
control_cards_linear.key
*CONTROL_IMPLICIT_SOLUTION
$   nsolvr
        -1
$    dnorm    diverg     istif
                         99999
*INCLUDE
database_cards_static.key
*CONTROL_TERMINATION
Define end time of the simulation
*INCLUDE
Include file defining geometry, materials etc.
```

```
*LOAD_...
Define nodal loads etc.
*BOUNDARY_...
Data line to prescribe boundary conditions
*TITLE
Simulation title
*END
```

4.1.1 Linear static example

The torsional stiffness of a rollover protection structure (ROPS) is to be determined. The geometry of the example is shown in Figure 7. Unit force of ±1 kN are applied to constrained nodal rigid bodies in the corners of the lower longitudinal beams. Additional boundary conditions are applied to prevent large rigid body motions. The example keyword file is `ROPS_linear_stiffness.key`. The resulting displacement magnitude is shown in Figure 8.

Figure 7. The left image shows the FE-model of the ROPS. The right image shows the applied loadings.

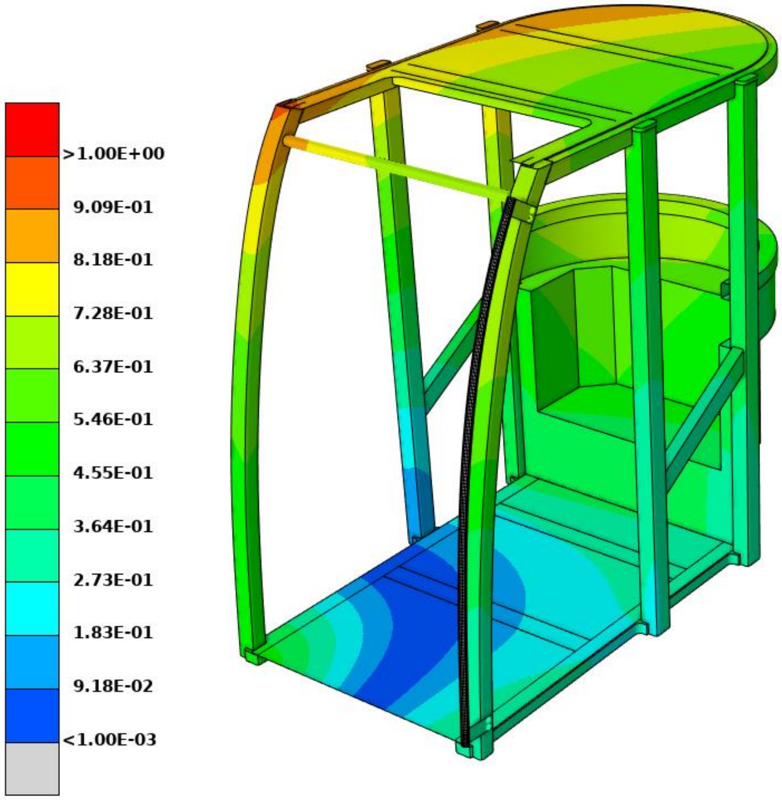


Figure 8. Magnitude of displacement for the ROPS stiffness example.

4.1.2 Inertia relief boundary conditions

Inertia relief can be used for solving a static problem without complete boundary conditions. An acceleration field is applied in order to counter-act the unbalanced forces, which would otherwise lead to infinite displacements. Examples of applications are, for example, static loading of an airplane in flight. In LS-DYNA, rigid body modes are determined based on an eigenfrequency analysis (see further Section 4.6). Inertia relief boundary conditions can be combined with other boundary conditions, such as single point constraints (`*BOUNDARY_SPC_`…). A template for using inertia relief follows:

```
*KEYWORD
.
.
*LOAD_...
Define nodal loads etc.
*BOUNDARY_...
Data line to prescribe boundary conditions
*CONTROL_IMPLICIT_INERTIA_RELIEF
1, THRESH
*TITLE
Simulation title
*END
```

Eigenmodes with corresponding frequencies below *THRESH* are taken as rigid body modes in the following analysis. The threshold value *THRESH* should be a small number, e.g. 0.001 Hz[2]. It is important to input a reasonable value for *THRESH*, it cannot be left blank or zero. Alternatively, the number of lowest eigenmodes to use as inertia relief modes may be specified using the variable *IRCNT*. For example, to use the 6 lowest eigenmodes as rigid body modes, use

```
*CONTROL_IMPLICIT_INERTIA_RELIEF
$  irflag    thresh     ircnt
        1                   6
```

4.2 Linear analysis with nonlinear contact

As an intermediate step between the fully linear soliton (Section 4.1) and the fully nonlinear solution (Section 4.3), a linear solver with iterations in order to account for nonlinearities due to contacts from was introduces from R15 of the Ansys LS-DYNA software. This is activated by setting the variable *NSOLVR* = -12 on the keyword `*CONTROL_IMPLICIT_SOLUTION` [1]. With this option, the model's behavior will be linear (except for that nonlinearities due to contacts will be considered) - material models will be linearized, and small strain deformation theory will be used. A template for using linear analysis with nonlinear contact follows:

```
*KEYWORD
.
.
*INCLUDE
control_cards_nonlin.key
*INCLUDE
Database_cards_static.key
*CONTROL_IMPLICIT_SOLUTION
$#  nsolvr    ilimit    maxref     dctol     ectol     rctol     lstol    abstol
       -12                                                              1.E-20
$#   dnorm    diverg     istif   nlprint    nlnorm   d3itctl
         1         1         1         3         4         1
$#  arcctl    arcdir    arclen    arcmth    arcdmp

$#   lsmtd
         4
*CONTROL_TERMINATION
...
*LOAD_...
Define nodal loads etc.
*BOUNDARY_...
Data line to prescribe boundary conditions
*TITLE
Simulation title
*END
```

[2] Note that the unit for frequency depends on the unit system of the model. For example, if time is in ms, the corresponding frequency unit is kHz.

4.2.1 Press fit example

In this example, the linear solver with nonlinear contacts is applied for resolving a press fit between an axle and a pulley, see Figure 9, by the option *IGNORE* = 3 of the Mortar contact. A simplified version of this example is attached to this Guideline package as `run_pressfit.key`.

Press fit can of course also be resolved using fully nonlinear analysis, see also Section 11.6.

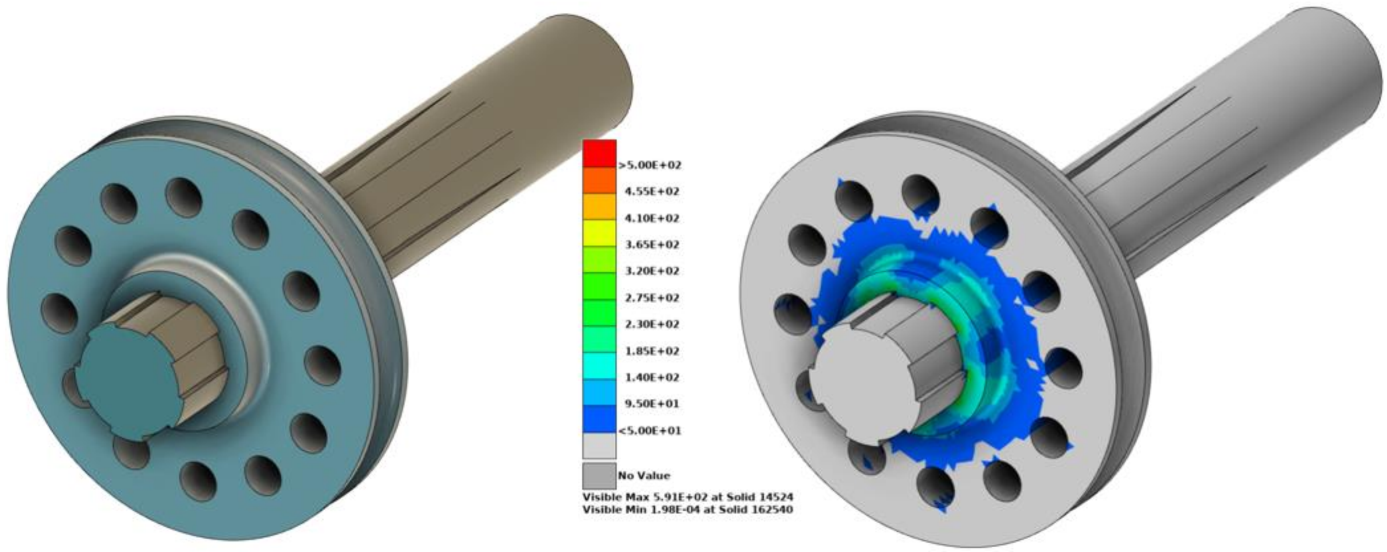


Figure 9. Example of a press fit between a shaft and a pulley. The right image shows the geometry. The left image shows the stress distribution after the press fit has been resolved.

4.3 Nonlinear static analysis

In LS-DYNA implicit, fully non-linear static analysis is the default, including large strains and deformations. Sources of non-linearity in a static analysis may be

- non-linear material models (plasticity)
- contact,
- large deformations,
- non-linear constraints (such as joints),
- non-linear loading (such as follower forces, where the force direction is defined relative to the deformed geometry), or
- stress stiffening (guitar string effect).

In the Ansys LS-DYNA software, an implicit analysis is either fully nonlinear or fully linear. From R15, the option to perform a linear analysis considering nonlinear contacts (see Section 4.2) was added.

The keyword file `control_cards_nonlin.key` contains control card settings which are generally well suited for non-linear static analyses. The automatic time incrementation is controlled by the load curve with ID 700, which the user must define. The purpose of the load curve is

- to define the maximum allowed time increment during the simulation. Reducing the time increment can aid convergence, if substantial non-linearities are present in the model.

- to synchronize the simulation time with the loading: at each time specified in the load curve ID 700, a converged step is obtained[3], see Figure 10.

A template for a non-linear static analysis follows:

```
*KEYWORD
*INCLUDE
control_cards_nonlin.key
*DEFINE_CURVE_TITLE
Implicit time incrementation
700,
0., dt0 (first timestep)
Additional lines to define time incrementation and synchronization with loadings
*INCLUDE
database_cards_static.key
*CONTROL_TERMINATION
Define end time of the simulation
*INCLUDE
Include file defining geometry, materials etc.
*LOAD_...
define nodal loads etc.
*BOUNDARY_...
Data line to prescribe boundary conditions
*TITLE
Simulation title
*END
```

For a static, implicit analysis to converge, it is required that no rigid-body modes or mechanisms exist in the model. This means that the user must provide sufficient boundary conditions in order to prevent rigid body motion of any part. Inertia relief boundary conditions, see Section 4.1.2, may also be applied to non-linear analyses. Loads and boundary conditions are further discussed in Section 8.
Many analyses involve an initial stage where rigid body modes exist in the assembly, which are eliminated when the parts come into contact, for example parts that are kept together by bolt pre-tensioning. This type of situations can be handled by performing the initial stage (until for example contacts are established) using implicit dynamics (LS-DYNA keyword `*CONTROL_IMPLICIT_DYNAMICS`, set *`IMASS`* = 1) which is discussed further in Section 4.3.2.

[3] If possible. It may also be the case that static equilibrium cannot be found, due to for example structural instability or overload (see further Section 4.5).

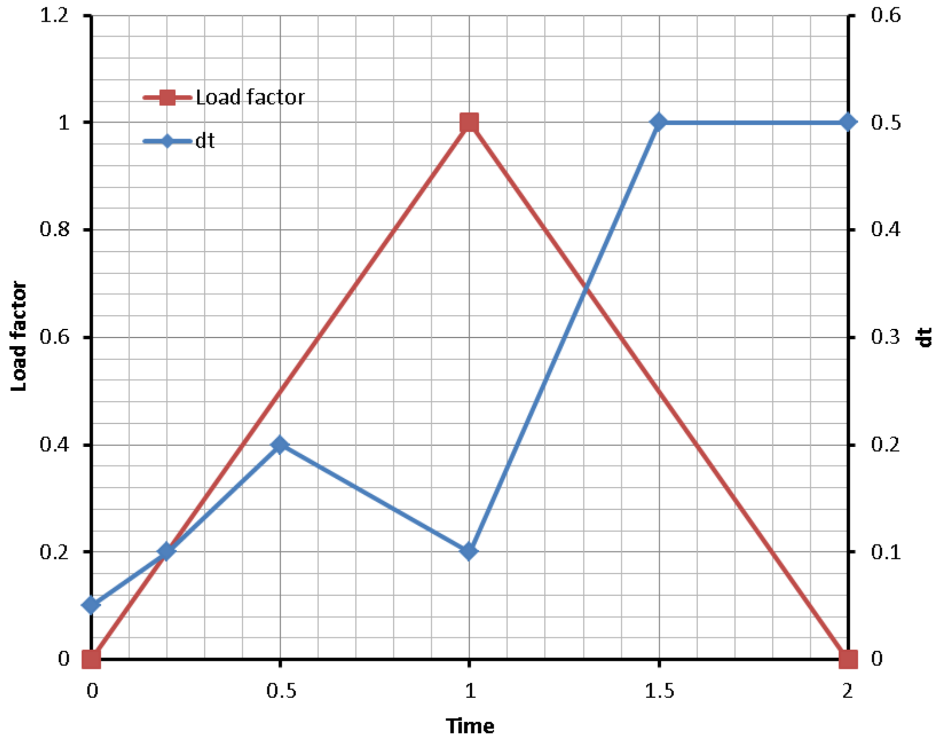


Figure 10. Illustration of the use of the time incrementation curve (blue) to synchronize the simulation with the applied loading (red curve). By specifying a value of (1.0, 0.2) for the time incrementation curve, a step at t = 1.0 (coincident with the maximum loading) is obtained.

4.3.1 Non-linear static example

An aluminium cantilever beam is bent over a rigid support. The geometry for the example is shown in Figure 11. At one end, the beam (yellow) is fully constrained (blue lines) while at the other end, a prescribed displacement is applied. Non-linear material properties are used in the beam. The contact between the beam and the support is modelled using `*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR`, see further Section 6. The example keyword file is `bend001.key`. Results are shown in Figure 12 and Figure 13.

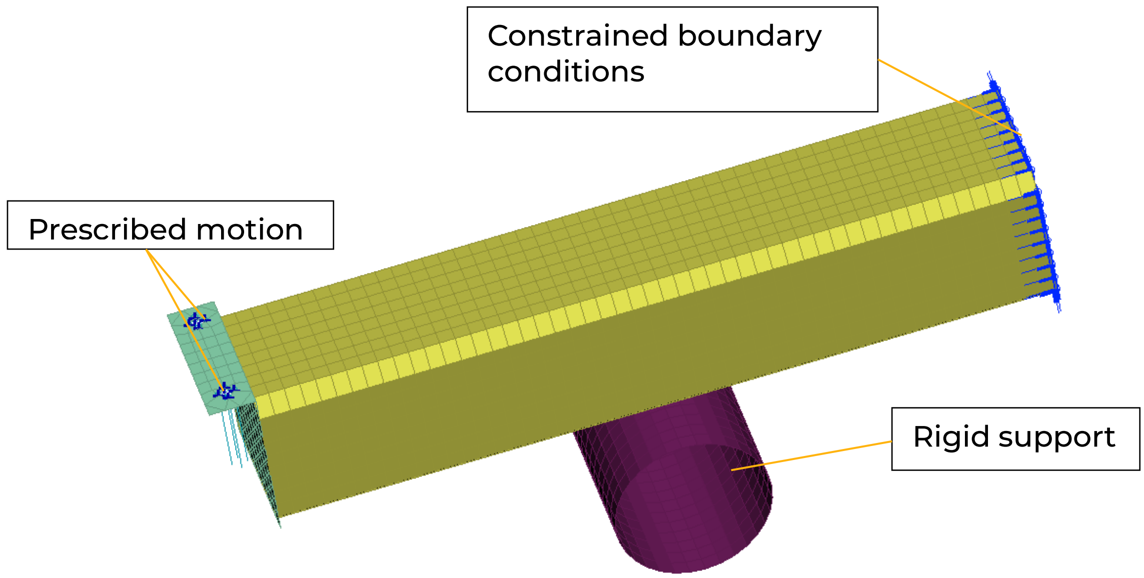


Figure 11. A cantilever beam is bent over a rigid support (keywordfile `bend001.key`).

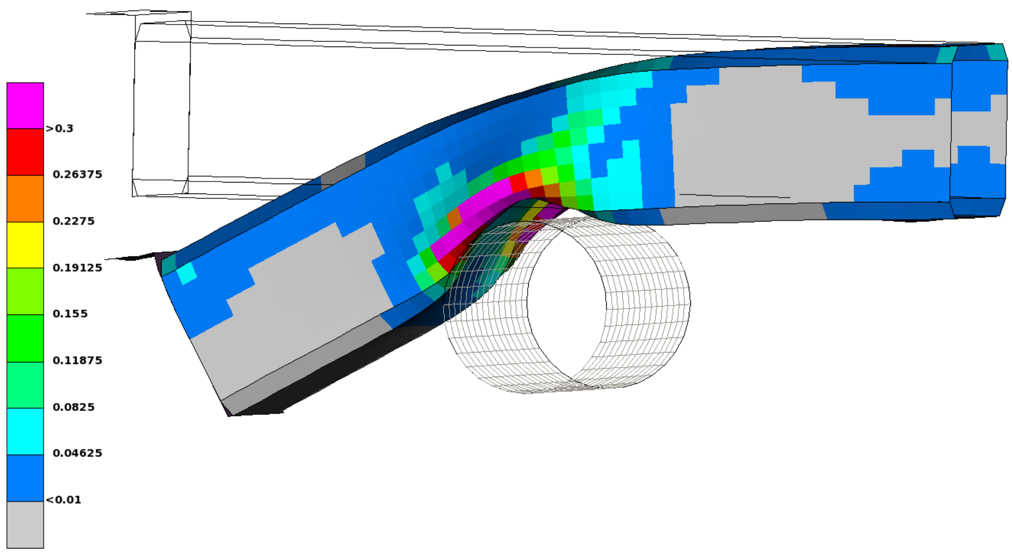


Figure 12. Plastic strains and deformation at the final stage.

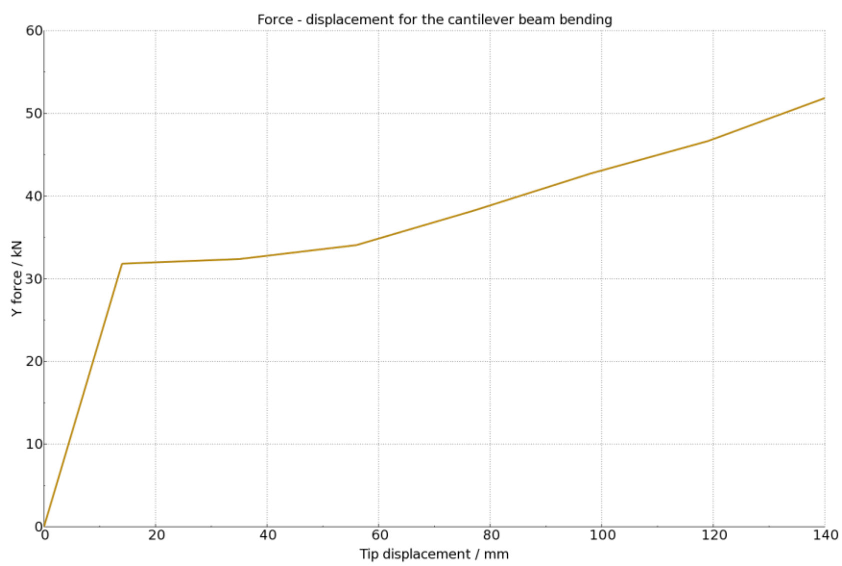


Figure 13. Required Y-force vs. tip displacement.

This example is also available in a version using a less strict convergence criterium by *DNORM* = 2 on `*CONTROL_IMPLICIT_SOLUTION`, see also Section 10 and Appendix G, as `bend002.key`.

4.3.2 Bolt pre-tensioning

A bracket is connected to a base plate by four bolts, see Figure 14. Bolt pre-tensioning is performed between t = 0 and t = 1. The LS-DYNA keyword `*INITIAL_STRESS_SECTION` is used to apply the bolt pre-tensioning. From t = 1 to t = 2, a load is applied at the free hole of the flange via a distributing coupling (`*CONSTRAINED_INTERPOLATION`). The example keyword file is `bolts001.key`.

Implicit dynamics is used during the bolt pre-tensioning in order to overcome the fact that the model initially contains rigid-body modes. A template for using implicit dynamics for this purpose follows:

```
*KEYWORD
*INCLUDE
control_cards_nonlin.key
*DEFINE_CURVE_TITLE
Implicit time incrementation
700,
0., dt0 (first timestep)
Additional lines to define time incrementation
*CONTROL_IMPLICIT_DYNAMICS
1, GAMMA, BETA, 0., TDYDTH, TDYBUR, IRATE
*INCLUDE
database_cards_static.key
*CONTROL_TERMINATION
Define end time of the simulation
*INCLUDE
Include file defining geometry, materials etc.
*LOAD_...
Define nodal loads etc.
*BOUNDARY_...
Data line to prescribe boundary conditions
*TITLE
Simulation title
*END
```

The parameters *GAMMA* and *BETA* of the `*CONTROL_IMPLICIT_DYNAMICS` keyword (see also Section 4.9) control the time integration. Normally *GAMMA* = 0.6 and *BETA* = 0.38 are used in order to introduce some numerical damping, when the purpose is to use implicit dynamics in a (quasi) static analysis to resolve initial rigid-body modes. The parameter *TDYDTH* is the time when the dynamic effects start to ramp off, and at time *TDYBUR* the dynamic effects are completely removed, see Figure 15. Setting the parameter *IRATE* = 1 will switch off the rate effects in material models, see further Section 7.

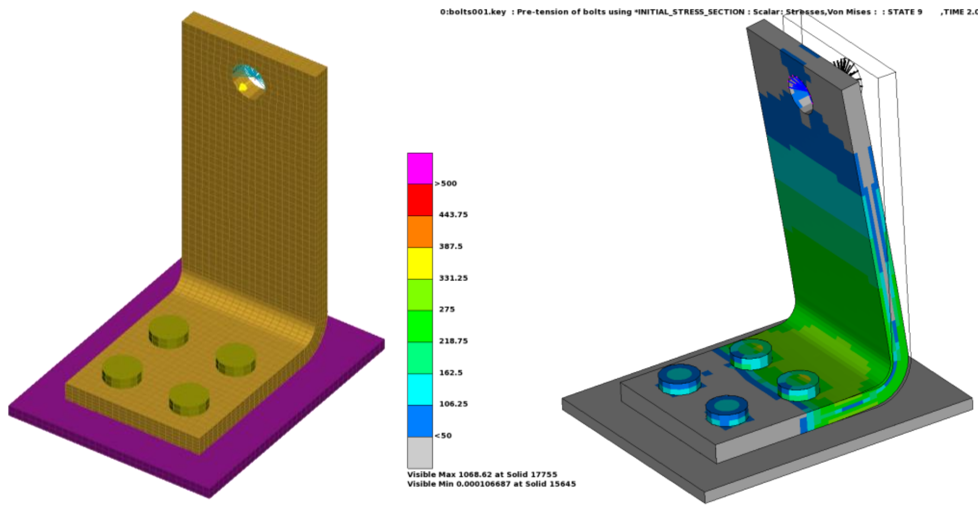


Figure 14. A bracket is attached to a base plate by four bolts. A force (shown as a yellow arrow) is applied by a distributing coupling at the center of the hole in the flange. The left image shows the initial geometry. The right image shows the final state with a fringe plot of von Mises effective stress.

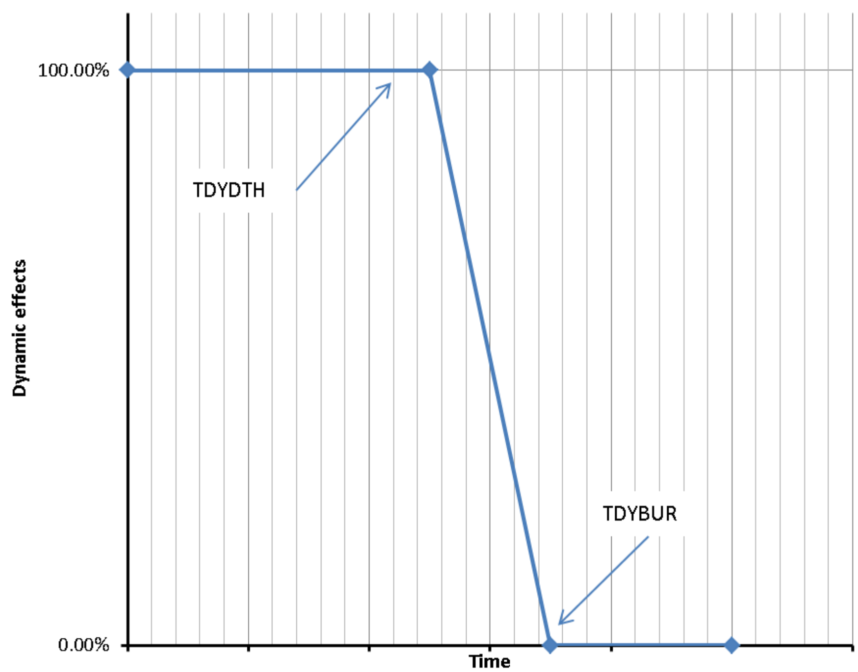


Figure 15. Death and burial time for implicit dynamics.

See also Section 16.1.1 for a slightly modified version of this example. Bolt pre-tensioning in LS-DYNA in general is also discussed in for example Refs. [18], [29], [33] and on [LSDYNA Bolts (ansys.com)]

4.4 Linear buckling analysis

Linear buckling analysis can be used for estimating the critical load, or bifurcation load, of a stiff structure. This is done by solution of the generalized eigenproblem

$$\left(\boldsymbol{K}_m + \lambda \boldsymbol{K}_g(\boldsymbol{\sigma}_0)\right)\boldsymbol{\Phi} = 0,$$

where $\mathbf{K}_g$ is the geometric stiffness matrix (due to stress stiffening effects) and $\mathbf{K}_m$ is the material tangent stiffness matrix. To compute $\mathbf{K}_g$, some loading must be applied to the structure (so that stresses develop). For the numerical solution to work well, the applied loading should be high enough to result in a well-defined geometrical stiffness matrix, compared to the tangent stiffness. The pre-loading can be applied by a static implicit analysis.

In LS-DYNA, linear buckling analysis is activated by using the keyword `*CONTROL_IMPLICIT_BUCKLE`. A linear buckling analysis may be part of a linear or non-linear static analysis. Note that linear (eigenvalue) buckling is a linear procedure, with all the limitations mentioned in Section 4.1. This means that contacts will not be updated, and that material responses are linearized (plasticity is not considered during the buckling phase).

A template for linear buckling analyses follows:

```
*KEYWORD
.
.
.
*INCLUDE
database_cards_static.key
*CONTROL_TERMINATION
Define end time of the simulation
*CONTROL_IMPLICIT_BUCKLE
Define parameters for buckling load evaluation
*INCLUDE
Include file defining geometry, materials etc.
*LOAD_...
Define nodal loads etc.
*BOUNDARY_...
Data line to prescribe boundary conditions
*TITLE
Simulation title
*END
```

By default, the buckling analysis will be performed at the termination time of the simulation, but it is possible to perform intermittent buckling analyses at different stages of the simulation. A curve ID can be specified by entering the LCID as a negative number. The x-values of the curve specifies at which time(s) the buckling analyses are to be performed, and the corresponding y-value specifies the number of buckling modes to compute. An example, based on a non-linear implicit pre-loading of a structure is given, see Section 4.4.1.

Buckling load factors are printed in the `eigout` - file. The buckling load is simply the eigenvalue times the applied load. The buckling mode shapes are output in the `d3eigv` file(s). To get output of stresses in the d3eigv file(s), include `*CONTROL_IMPLICIT_EIGENVALUE` with *NEIG* = 0 and *MSTRES* = 1.

Note that a linear buckling analysis may severely overestimate the physical buckling load of a real-world structure, since non-linear effects (plasticity, failure, large deformations) are neglected (as well as imperfections).

4.4.1 Linear buckling analysis of a panel with a bead

A panel is fully constrained at one edge and loaded by a force distributed around the bolt holes in the flange, see Figure 16. The edge at the flange is constrained in the transverse direction. The example keyword file is `buckle001.key`. First, a pre-loading of 32 kN is applied in a nonlinear implicit analysis, then a linear buckling analysis is performed. The lowest eigenvalue is 12.078, corresponding to a buckling load of 386.5 kN. The buckling mode shape is shown in Figure 17.

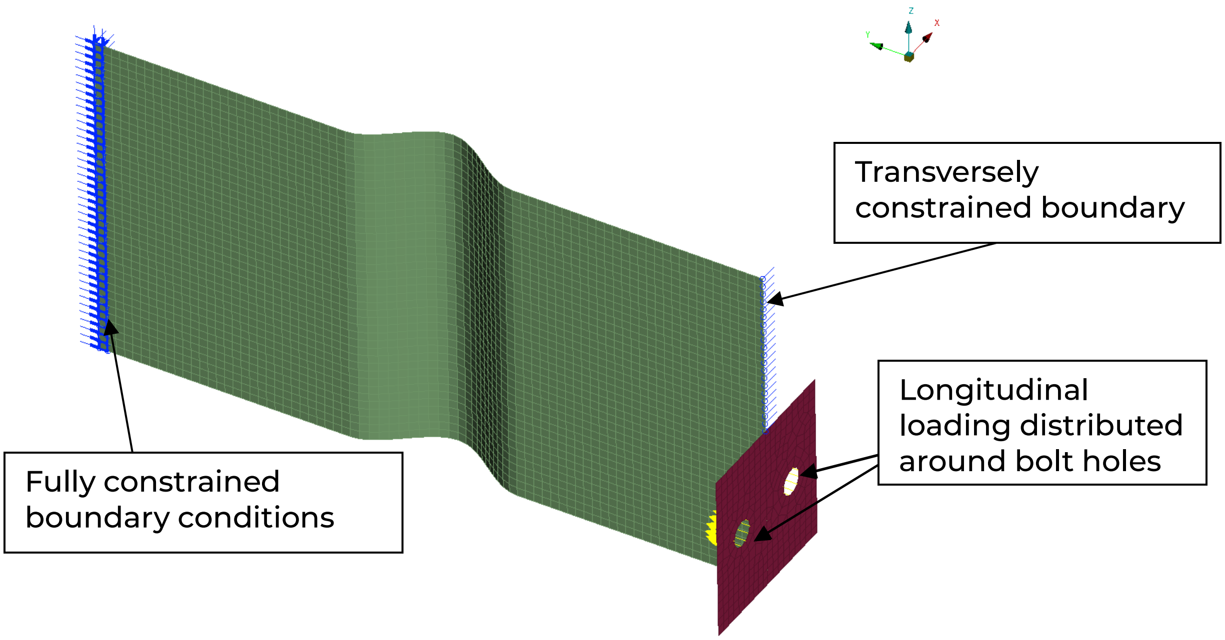


Figure 16. A panel with a bead under compression loading.

Figure 17. The lowest buckling mode shape for the panel.

4.5 Non-linear analysis using the arc-length method

The arc-length method is applied in order to follow a non-monotonically increasing load path, see Figure 18. The arc-length method can be seen as a mixed load and displacement-controlled loading with a highly advanced control system. By use of the arc-length method, situations of load reversal, snap-through and snap-back can be handled. This makes it ideal for non-linear limit load and postbuckling analyses.

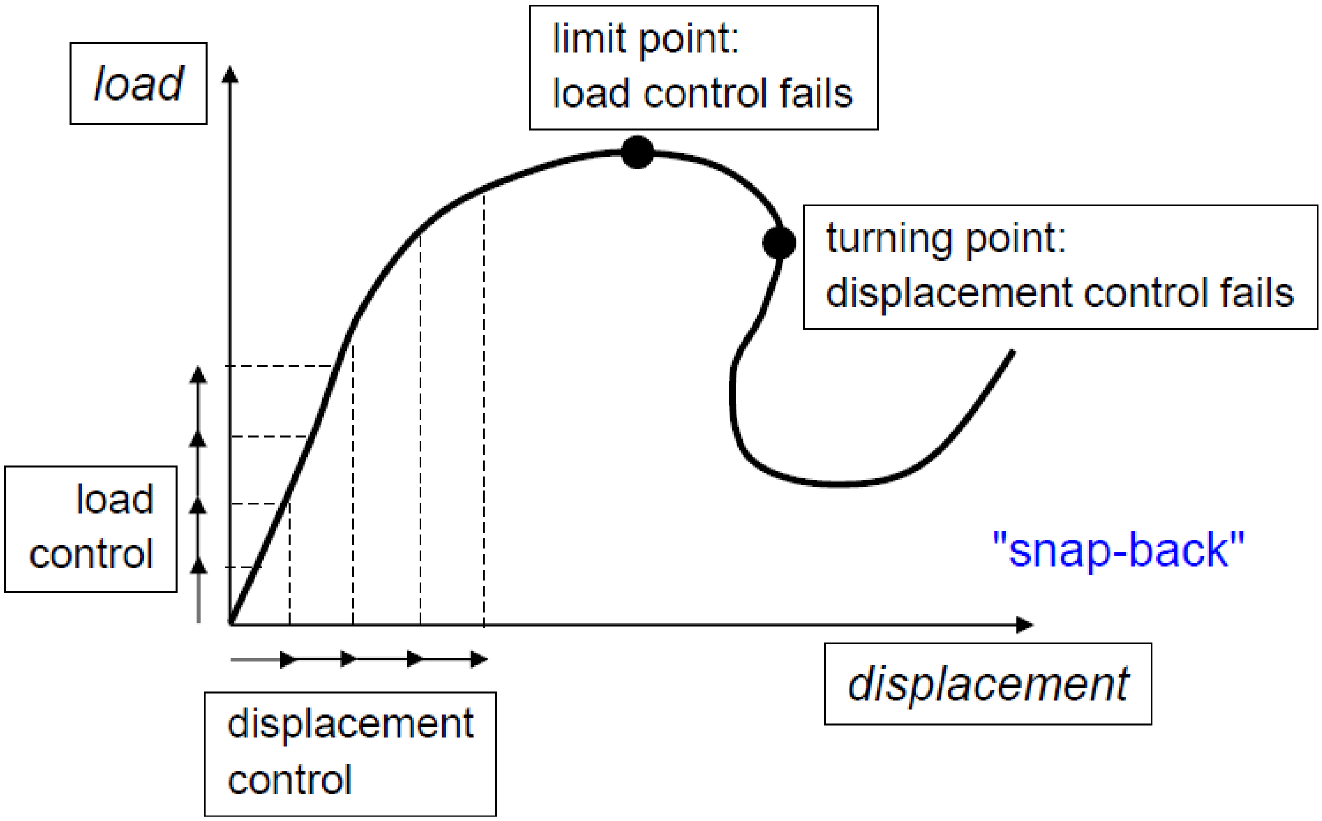


Figure 18. Arc-length control is required in order to follow a non-monotonically increasing (generalized) load path. From Ref. [4].

In LS-DYNA, the arc-length method is activated by a parameter on the `*CONTROL_IMPLICIT_SOLUTION` – card, but this is already predefined in the include file `control_cards_arc.key`. For postbuckling analyses, imperfections are commonly introduced as perturbations of the geometry or material parameters. This can be achieved using the keyword `*PERTURBATION`.

By default, all generalized loadings in the model must use linear loading curves, containing only two points, starting at (0, 0) and increasing to their final value at the termination time, see Figure 19. If multiple loads exist in the model (including thermal loading), all loads will vary simultaneously. See for example Ref. [26] for further discussions on use of the arc-length method for ultimate capacity analyses in LS-DYNA.

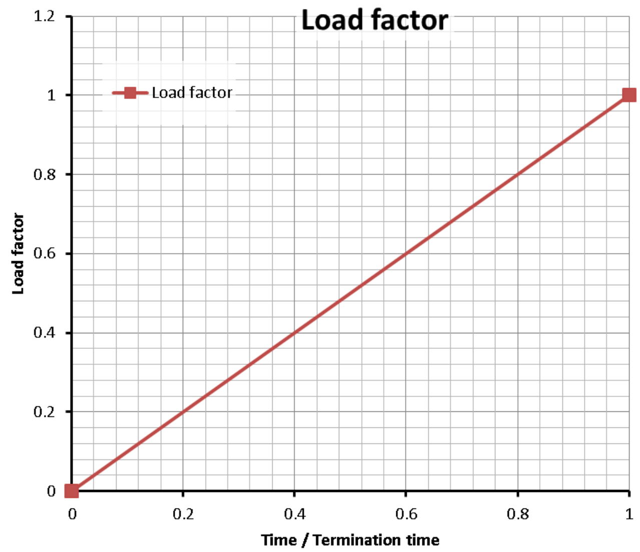


Figure 19. Typical load curve for arc-length analysis.

The keyword file `control_cards_nonlin_arc.key` contains control card settings which may be useful for non-linear post-buckling analyses. In this file, *ARCTIM* is set to zero, which means that the arc-length method will be active from the start of the analysis.

The simulation time in this case takes a clear roll of a load parameter. In general, “time” in an arc-length analysis may increase and decrease to follow the force-displacement response. The “time” may even become negative, indicating complete load reversal. This means that using `*CONTROL_TERMINATION` (based on time) may not make sense, since the specified termination time might never be reached. Alternative termination criteria are then required, for example based on the deformation of the structure. In this case, the keyword `*TERMINATION_NODE` (or `*TERMINATION_BODY)` may be very useful.

A template for a non-linear analysis using the arc-length method follows:

```
*KEYWORD
*INCLUDE
control_cards_nonlin_arc.key
*DEFINE_CURVE_TITLE
Implicit time incrementation
700,
0., dt0 (first timestep)
Additional lines to define time incrementation
*INCLUDE
database_cards_static.key
*CONTROL_TERMINATION
Define end time of the simulation
*TERMINATION_NODE
Define termination criteria based on displacement
*INCLUDE
Include file defining geometry, materials etc.
*PERTURBATION_NODE
Define parameters for creating geometrical imperfections
*LOAD_...
Define nodal loads etc.
*BOUNDARY_...
Data line to prescribe boundary conditions
```

```
*TITLE
Simulation title
*END
```

An example involving a limit load analysis of a panel with a bead is presented in Section 4.5.1.

4.5.1 Limit load analysis of a panel with a bead

A panel is fully constrained at one edge and loaded by a force distributed around the bolt holes in the flange, see Figure 16. The edge at the flange is constrained in the transverse direction. The geometry, boundary conditions and loadings are the same as in Section 4.4.1. The material in the panel is Domex 355. The example keyword file is `limitload001.key`. The force-deflection curve is shown in Figure 20. The peak load is 100 kN (which is significantly lower than the linear estimate of Section 4.4.1).

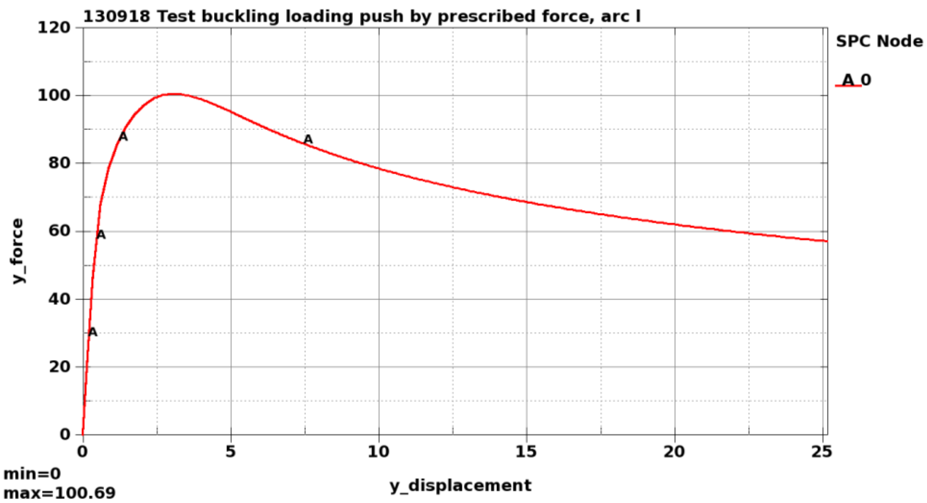


Figure 20. Force-displacement curve for the postbuckling analysis of a panel with a bead.

4.6 Eigenfrequency analysis

Eigenfrequency analysis is a way of determining the fundamental frequencies and shapes of harmonic vibration of a system. The equations of unforced motion can be put as

$$\mathbf{M}\ddot{x}+\mathbf{K}\mathbf{x}=\mathbf{0}.$$

By the ansatz $\mathbf{x} = \mathbf{\Phi}\sin \omega t$, the generalized eigenvalue problem

$$(\boldsymbol{K} - \omega^2\boldsymbol{M})\boldsymbol{\Phi} = 0$$

is obtained, with the eigenfrequency $f = \omega/2\pi$. Eigenfrequency analysis can be used for example to verify that the eigenfrequencies of the system do not coincide with critical excitation frequencies. It can also be used as a tool for model integrity checking, as mechanisms and rigid body modes are effectively revealed. Another common use for eigenfrequency analysis is to calculate a modal basis for further dynamic analyses, see Sections 4.7 and 4.8.

In LS-DYNA, eigenfrequency analysis is activated by using the keyword `*CONTROL_IMPLICIT_EIGENVALUE`. Intermittent eigenfrequency analyses are also possible, as a part of a linear or non-linear analysis[4]. This means that the effect of loading, such as for example bolt pre-tensioning, on the eigenfrequencies, can be accounted for (see also the example of Section 4.6.3). It is also possible to compute modal stresses by setting the parameter *MSTRES* = 1. This is required if strain energy density is to be evaluated, see Section 4.6.1, or if the computed eigenvectors are to be used as a modal basis in a frequency domain analysis where stress evaluations are of interest, see Section 4.8.

Note that non-Mortar sliding contacts shall be used with care in an eigenfrequency analysis, see Section 6.1.7.

A template for eigenfrequency analyses follows:

```
*KEYWORD
.
.
.
*INCLUDE
database_cards_static.key
*CONTROL_TERMINATION
Define end time of the simulation
*CONTROL_IMPLICIT_EIGENVALUE
Card 1: Define number of eigenmodes, frequencies etc.
Card 2: Set MSTRES=1 to request modal stresses
*INCLUDE
Include file defining geometry, materials etc.
*LOAD_...
Define nodal loads etc.
*BOUNDARY_...
Data line to prescribe boundary conditions
*TITLE
Simulation title
*END
```

A basic eigenfrequency example is presented in Section 4.6.2.

By default, the eigenfrequency analysis will be performed at the beginning of the simulation, which then terminates. It is possible to perform intermittent eigenfrequency analyses at different stages of the simulation. A curve ID can be specified by entering the LCID as a negative number. The x-values of the curve specifies at which time(s) the eigenfrequency analyses are to be performed, and the corresponding y-value specifies the number of eigenmodes to compute. When performing intermittent eigenfrequency analyses, keypoints corresponding to the times for the eigenfrequency analysis must be added to the synchronization curve (LCID 700, compare Figure 10). See also the example of Section 4.6.3. Another option for performing intermittent eigenvalue analyses is to use the multistage approach, see Section 12.2.1 and Appendix X of Ref. [1].

The eigenfrequencies are printed in the `eigout` - file. The database for visualizing modal deformations (eigenvectors) is called `d3eigv`. The eigenvectors in LS-DYNA are mass normalized, which implies that

[4] For intermittent eigenvalue analyses as a part of an explicit analysis, see Section 16.5.

$$\Phi^{\mathrm{T}} M\Phi = \mathbf{I}$$

but when output in the `d3eigv` – file(s) they are also rescaled for improved visibility. The scale factor can be obtained from LS-PrePost, as described in Figure 21. In versions starting with R12.2, the scale factors are also printed in the `eigout` – file, search for "`EIGENVECTOR SCALING FACTOR`".

1. Open the `d3eigv` – file(s),
2. Select to visualize the mode of interest,
3. Go to FEM > Post > Output
4. Click "Curr" (middle button in lowest row)
5. Toggle the "Normalization" checkbox to update the scale factor value.
6. The scale factor is shown in the text box next to the "Normalization" checkbox.

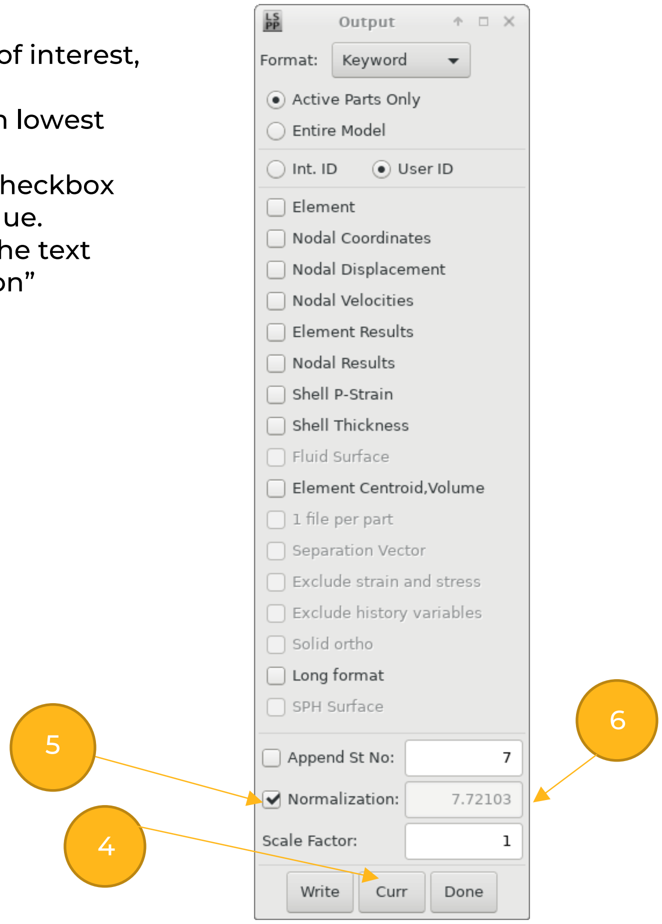


Figure 21. Output of scale factor for mode shapes from LS-PrePost.

Another option for obtaining the unscaled eigenvectors is to use the functionality for dynamic condensation and superelement creation of the keyword `*CONTROL_IMLPICIT_MODES`. This will, in addition to the `d3eigv` files, also output the unscaled eigenvectors in the `d3mode` binary database. The `d3mode` file(s) can be opened in LS-PrePost for 3D visualization and postprocessing.

If the `d3eigv` – files are to be used in mode-based analyses (see Sections 4.7 and 4.8) they must be in double precision format for all mode-based analyses (do not use `*DATABASE_FORMAT` to request single precision output in this case).

The default solution method for eigenvalue problems in LS-DYNA is block-shift invert Lanczos. This is normally a fast and efficient method, especially in mpp/LS-DYNA. Since R11.1, the MCMS method is available in smp/LS-DYNA [20] for computing many (>1000) eigenvalues of large models (> 1E6 elements). This is activated by setting *EIGMTH* = 101 on card 1 of `*CONTROL_IMPLICIT_EIGENVALUE`. If only a few (< 50) eigenvalues of a large model are to be determined, the LOBPCG method may be of interest. It can be activated by setting EIGMTH = 102. The LOBPCG is only available in smp in versions prior to R14, but from R14 also an mpp implementation exists.

4.6.1 Visualization of strain energy and strain energy density

Strain energy and strain energy density are commonly used measures, indicating the areas of the structure which are under the greatest stress, and contribute most to the stiffness, for a particular mode. In LS-DYNA, it is required to request output of modal stresses by setting `MSTRES = 1`. Then LS-PrePost (and some 3rd party post-processors as well) can be used for visualizing the strain energy and strain energy density from the `d3eigv` file, see Figure 22 (For additional required output settings, it is assumed that the control and database card settings provided with this Guideline are used, for example `control_cards_linear.key` and `database_cards_static.key`) . Note that the strain energy is not computed by the LS-DYNA solver but is calculated as a postprocessing result by the postprocessing software.

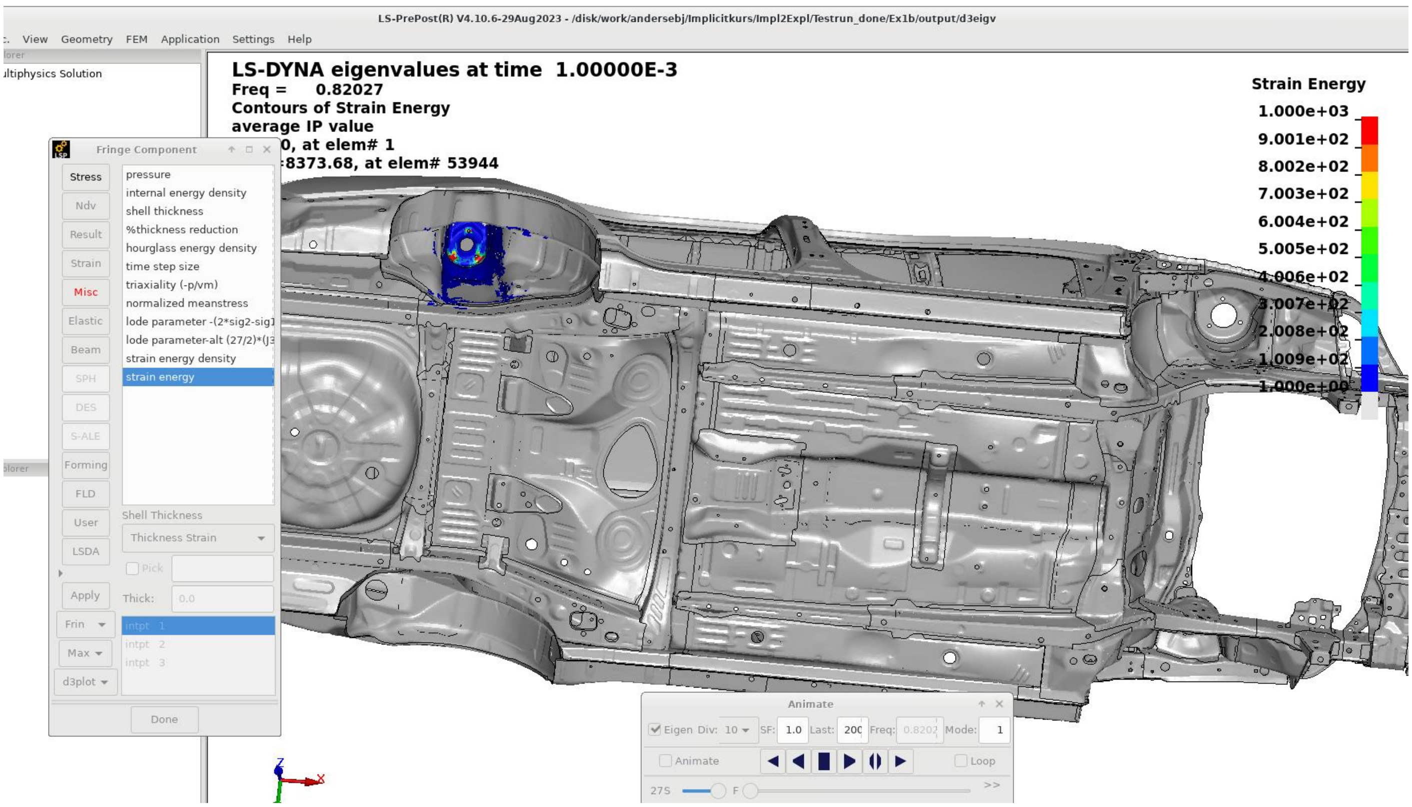


Figure 22. Example of visualization of strain energy using LS-PrePost. Model [45] courtesy of NCAC. The work of the Center for Collision Safety and Analysis at the George Mason University is gratefully acknowledged.

4.6.2 Eigenfrequency analysis of a panel with a bead

A panel is fully constrained at one edge, see Figure 23. The example keyword file is `eigen001.key`. The lowest eigenfrequency is 44.3 Hz, and the mode shape is a side-to-side swinging motion.

4.6.3 Intermittent eigenfrequency analysis of a bolted L-bracket

Eigenfrequency analyses of the bolted assembly of Section 4.3.2 at t = 0 (initial configuration without bolt pre-tension), t = 1 (bolt pre-tension applied) and t = 2 (bolt pre-tension and loading is applied). The example keyword file is `eigen002.key`.

At t = 0, six rigid body modes are found, indicating that the assembly, at this stage, is not connected. After bolt pre-tensioning, at t = 1, the lowest eigenfrequency is 479 Hz. At t = 2, when also the loading is applied, the lowest eigenfrequency decreases to 104 Hz.

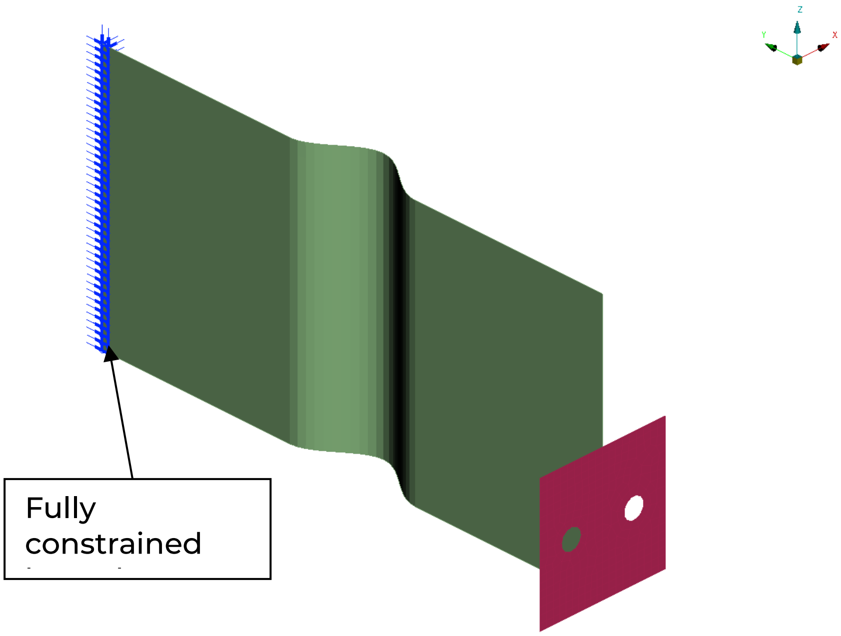


Figure 23. Eigenfrequency analysis of a panel with a bead.

Intermittent eigenvalue extraction can also be part of an explicit analysis, see Section 16.5.

4.7 Linear transient modal dynamic analysis

Transient linear dynamic analyses can often be performed very efficiently using a modal basis, denoted as Φ, since the number of active degrees of freedom can be drastically reduced. Instead of working with all the physical degrees of freedom of the system **x**, a limited number of generalized (modal) degrees of freedom, z, is used: **x** = Φ**z.** Also, by the orthogonality properties of Φ, the transformed set of equations of motion is diagonalized.

In LS-DYNA, linear transient modal dynamic analysis is activated by the keyword `*CONTROL_IMPLICIT_MODAL_DYNAMIC`. At present, this feature is only available in the double precision smp-version[5] of LS-DYNA. Either eigenmodes computed previously and stored in `d3eigv`-files or computed in the same run using `*CONTROL_IMPLICIT_EIGENVALUE`, can be used as a modal basis. Note that the `d3eigv` – file must be in double precision format (set `IBINARY` = 0 on `*DATABASE_FORMAT`) for all mode-based analyses.

The default is that all eigenmodes are used in the modal basis, but it is possible to select which eigenmodes to use by the keyword `*CONTROL_IMPLICIT_MODAL_DYNAMIC_MODES`.

Modal damping can be specified, either as a

[5] Versions R10.1, R10.2, R14, R15.0.2 are recommended for transient modal dynamics.

- constant value, applied to all modes, using the parameter *ZETA* on `*CONTROL_IMPLICIT_MODAL_DYNAMIC`, or as a
- function of frequency, or per mode, using the keyword `*CONTROL_IMPLICIT_MODAL_DYNAMIC_DAMPING`.

As an alternative, (approximatively) frequency-independent damping can be applied by use of `*DAMPING_FREQUENCY_RANGE`.

A template for a linear transient modal dynamic analysis follows:

```
*KEYWORD
*INCLUDE
control_cards_linear.key
*INCLUDE
database_cards_static.key
*CONTROL_TERMINATION
Define end time of the simulation
*CONTROL_IMPLICIT_EIGENVALUE
Parameters for computation of eigenmodes
*CONTROL_IMPLICIT_MODAL_DYNAMIC
1, Define global modal damping
*CONTROL_IMPLICIT_MODAL_DYNAMIC_MODES
Data lines to specify which modes to use
*CONTROL_IMPLICIT_MODAL_DYNAMIC_DAMPING
Data lines to specify modal damping
*INCLUDE
Include file defining geometry, materials etc.
*LOAD_...
Define nodal loads etc.
*BOUNDARY_...
Data line to prescribe constrained boundary conditions
*TITLE
Simulation title
*END
```

This is a linear analysis procedure, which means that material models will be linearized. Contacts shall be used with care in linear analyses, see Section 6.1.7.

Stress recovery from a linear transient modal dynamic analysis is activated by setting the flag *MSTRES* = 1 on the 2nd card of `*CONTROL_IMPLICIT_EIGENVALUE` and also setting the flag *STRCMP* = 1 (3rd position) on `*CONTROL_IMPLICIT_MODAL_DYNAMIC`. Stresses will always be computed for shell element formulation 18.

From R9.0.1, it is possible to use prescribed displacement boundary conditions for this analysis type, by use of `*BOUNDARY_PRESCRIBED_MOTION`. To activate this feature, set *MDFLAG* = 2 on `*CONTROL_IMPLICIT_MODAL_DYNAMIC`. In addition, the node (or node set) subjected to prescribed displacement must be specified in a node set for constraint modes using `*CONTROL_IMPLICIT_MODES`. A small example follows, where node set ID 4 is subjected to a prescribed displacement in the global Y-direction, according to load curve ID 100:

.

.

.

`*CONTROL_IMPLICIT_MODAL_DYNAMIC`

2, Define global modal damping, {1 – optional to compute stresses}

`*CONTROL_IMPLICIT_MODES`

4, 0, 0, Define offset value for internally created DOFs

`*BOUNDARY_PRESCRIBED_MOTION_SET`

4, 2, 2, 100

.

.

.

Time integration is by default done in an explicit manner, but from R11, implicit time integration may be activated by setting *INTEG* = 2 on `*CONTROL_IMPLICIT_MODAL_DYNAMIC`. This option will apply the time integration parameters *BETA* and *GAMMA* from `*CONTROL_IMPLICIT_DYNAMICS` and use *DT0* from `*CONTROL_IMPLICIT_GENERAL` as time step.

4.7.1 Transient loading of an L-beam

An L-shaped beam is loaded by a transverse, transient (triangular) loading, see Figure 24. A constant modal damping of 3 % is applied. The transient displacement response of the loaded node is shown in Figure 25.

The example keyword file is `transient001.key`.

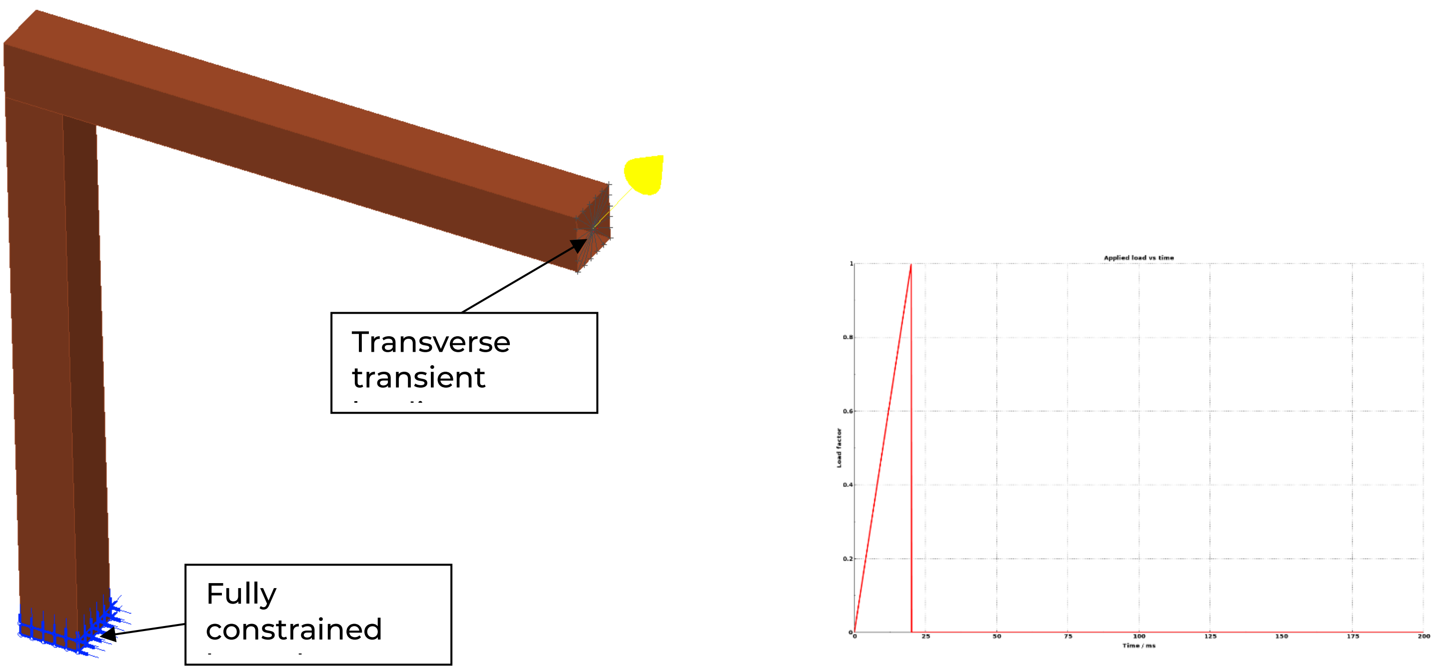


Figure 24. Transient loading of an L-shaped beam. The left image shows the geometry, with boundary conditions and loading. The right image shows the loading history.

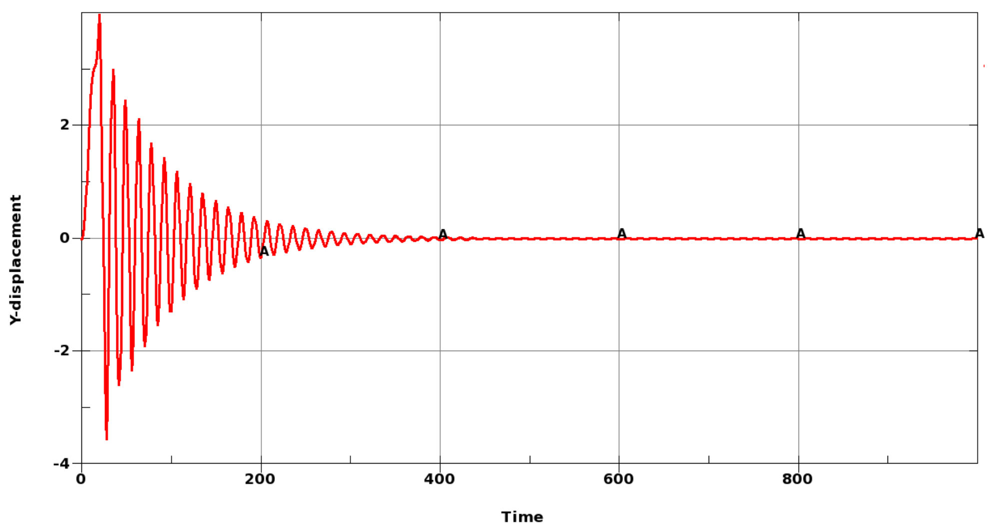


Figure 25. Displacement history in the transverse direction of the loaded node.

4.8 Frequency domain analyses

For many dynamic analysis types, such as NVH or earthquake analyses, it is convenient to study the response of a structure in the frequency domain instead of the time domain. Several frequency domain analysis types are available in LS-DYNA:

- calculation of frequency response functions,
- steady state dynamics,
- response spectrum analysis,
- random vibration analysis, and
- FEM and BEM acoustics.

Up to version R11 of LS-DYNA, these analyses are performed using a modal basis, which implies that an eigenfrequency analysis, see Section 4.6, is required. Note that if stresses are of interest in the frequency domain analyses, for example for random vibration fatigue or steady state dynamics analyses, they must already have been calculated during the eigenfrequency analysis. This can be achieved by setting the parameter *MSTRES* = 1 on card 2 of the `*CONTROL_IMPLICIT_EIGENVALUE` – keyword. The eigenfrequency analysis can be a part of the frequency domain analysis, or performed in a separate, preceding analysis. The `d3eigv` – files must be in double precision format. From R11 of (smp) LS-DYNA, a direct solution for steady state dynamics is available by the keyword(s) `*FREQUENCY_DOMAIN_SSD_DIRECT` and `*FREQUENCY_DOMAIN_SSD_DIRECT_FREQUENCY_DEPENDENT`.

Note that the frequency domain analyses are linear, which means that the same restrictions as mentioned in Section 4.1 apply (see also Section 6.1.7). The appropriate control card file is `control_cards_linear.key`.

An overview of the frequency domain analysis capabilities is presented in Refs. [8][43], and a thorough description is given in Ref. [9]. For more examples of NVH analyses in LS-DYNA, please visit https://lsdyna.ansys.com/knowledge-base/nvh-fatigue/.

In LS-DYNA, the related keywords all begin with `*FREQUENCY_DOMAIN` [1], and involve almost the complete problem definition, including loadings and damping definitions, on several cards. Output for the different frequency domain analysis types can be requested by using the keywords `*DATABASE_FREQUENCY_BINARY_`*OPTION*, and from R10 also the keyword `*DATABASE_FREQUENCY_ASCII_`*OPTION*.

From LS-PrePost 4.3, there is a wizard for setting up different types of frequency domain analyses. It is accessed from the top menu bar, Application > NVH, see Figure 26. LS-PrePost also has dedicated functionality, the NVH Fringe component, for convenient post-processing of frequency domain results.

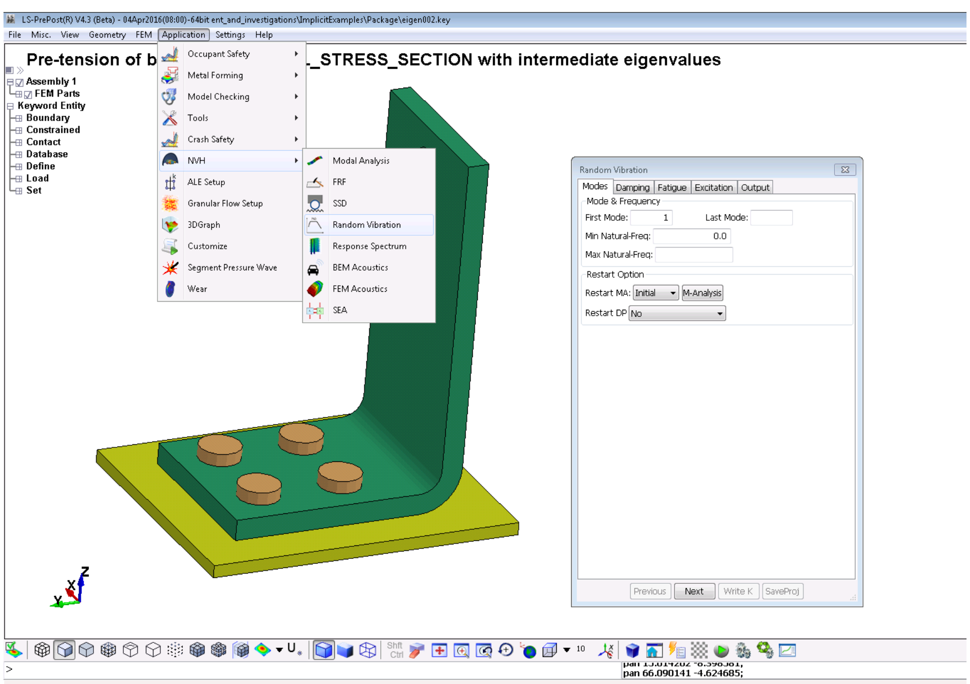


Figure 26. Wizard for frequency analysis set-up in LS-PrePost 4.3.

4.8.1 Frequency response functions

Frequency response functions (FRFs) of a dynamic system can be seen as the transfer functions between the excitation in one point and the response in another point, see Figure 27.

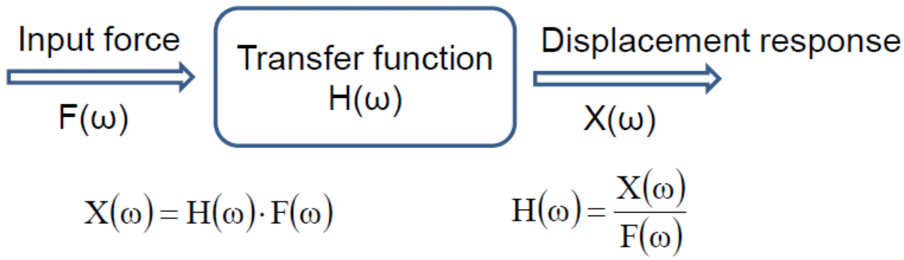


Figure 27. The frequency response functions are the transfer functions of a dynamic system. From Ref. [8].

Calculation of FRFs such as dynamic stiffness, impedance etc. is activated by the keyword `*FREQUENCY_DOMAIN_FRF`. The excitation can be base velocity, base acceleration, base displacement, nodal force, or pressure on a segment set. The response can be velocity, acceleration, displacement, or nodal force. The output is given in the ASCII-files `frf_amplitude`, `frf_angle` (alternatively by setting the parameter *OUTPUT* = 1, as `frf_real` and `frf_imag` for the real and imaginary part, respectively).

A template for the frequency response function – keyword follows:

```
*FREQUENCY_DOMAIN_FRF
Card 1: Define excitation and select modes
Card 2: Define damping
Card 3: Define the response
Card 4: Define output parameters
```

4.8.2 Steady state dynamics

Steady state dynamics is used in order to obtain the linearized steady state response of a structure subjected to harmonic excitation, for example a fuel pump attached to a vibrating engine. In LS-DYNA, steady state dynamics is activated by the keyword `*FREQUENCY_DOMAIN_SSD`. Note that if stresses are of interest, it is required to set the parameter *MSTRES* = 1 on card 2 of the `*CONTROL_IMPLICIT_EIGENVALUE` – keyword in the preceding eigenfrequency analysis.

Output from the steady state dynamics analyses can be obtained as 3D – binary plot files `d3ssd` (containing displacements, stresses etc., similar to `d3plot`) and as ASCII history database files. The output frequencies for the `d3ssd` – databases are specified using the `*DATABASE_FREQUENCY_BINARY_D3SSD` – keyword. This keyword will control which frequencies are analyzed in the SSD analysis. Linear, logarithmic or biased spacing of output frequencies are possible. Nodal and element history output are specified in the same way as for a time-domain analysis, using `*DATABASE_HISTORY_`*OPTION*, but the output files for steady state dynamics are `nodout_ssd` and `elout_ssd`. From R10, the keyword `*DATABASE_FREQUENCY_ASCII_`*OPTION* can be used to specify separate, or additional, output frequencies for these files.

In LS-DYNA R9 and earlier, enforced nodal motion (acceleration, velocity, displacement) is not explicitly implemented for steady state dynamics. If this type of excitation is desired, the large mass method can be used. A very large mass m_L (recommended is 10^6 times the total mass of the structure) is then applied to the nodes where the enforced motion is applied, using the keyword

*ELEMENT_MASS_{OPTION}. The desired motion is achieved by application of a corresponding force p as follows,

- For nodal acceleration $\ddot{u}$, $p = m_L\ddot{u}$
- For nodal velocity $\dot{u}$, $p = i\omega m_L\dot{u}$
- For nodal displacement u, $p = -\omega^2 m_L u$

Note that no other boundary conditions (*BOUNDARY_SPC or *BOUNDARY_PRESCRIBED_MOTION) in the direction of the desired motion shall be applied to the nodes where an enforced motion is to be applied by use of the large mass method. The corresponding rigid body modes must be included in the consecutive frequency domain analyses in order to obtain the desired result. The application of the large masses may alter the eigenfrequencies or mode order, which means that care shall be taken when selecting the eigenmodes.

From LS-DYNA R10, the large mass method has been semi-automated. Via the keyword *CONTROL_FREQUENCY_DOMAIN, the (large mass) per node is specified, and LS-DYNA will convert the prescribed velocity, acceleration, or displacement (*VAD* = 5 – 7) to the appropriate nodal force. The actual mass elements (*ELEMENT_MASS_{OPTION}) must still be added by the user. Note also that from R9 to R10, the excitation input type codes have been changed, see Table 2.

Table 2. Overview of excitation input types (*VAD*) for *FREQUENCY_DOMAIN_SSD

VAD	LS-DYNA R9	LS-DYNA R10 and later
0	Nodal force	
1	Pressure	
2	Base acceleration	Base velocity
3	Enforced velocity(1)	Base acceleration
4	Enforced acceleration(1)	Base displacement
5	Enforced displacement(1)	Enforced velocity(2)
6	N/A	Enforced acceleration(2)
7	N/A	Enforced displacement(2)

Notes: (1) Not explicitly implemented. (2) Semi-automatic implementation by the large-mass method

A template for a steady state dynamics analysis follows:

```
*FREQUENCY_DOMAIN_SSD
Card 1: Select modes to be used in the analysis
Card 2: Define damping
Card 3: Define output for subsequent acoustic analysis
Card 4: Define the excitations (repeat if multiple excitations are present)
*DATABASE_FREQUENCY_BINARY_D3SSD
Define frequency range, spacing etc. for output
*ELEMENT_MASS_{OPTION}
Define masses for enforced motion by the large mass method
```

For each excitation, two load curves are used (*LC1* and *LC2* of Card 4) for defining amplitude and phase (or real and imaginary component) as a function of frequency are required. These load curves must have the same number of abscissa values.

From R9.0.1[6], there is an option `_FATIGUE` for the steady state dynamics analyses, for computing also the fatigue damage due to the dynamic load case(s). The load duration for each frequency must be specified via a load using the parameter *LCFTG* on position 1 of Card 8 (or for versions prior to R12: *LC3* on position 7 of Card 4). Also, the fatigue properties (S-N-curves) for each material, respectively, must be specified using the keyword `*MAT_ADD_FATIGUE`. Note that stresses are required for the fatigue analysis, which in turn means that it is required to set the parameter *MSTRES* = 1 on card 2 of the `*CONTROL_IMPLICIT_EIGENVALUE` – keyword in the preceding eigenfrequency analysis. Also note that `*DATABASE_FREQUENCY_BINARY_D3FTG` must be specified in order to get 3D – binary plot files for visualization of fatigue analysis results. Consider switching to Lobatto quadrature for shell elements when performing fatigue analyses, see Section 5.2, to get integration points exactly on the inner/outer surface.

A basic example of a steady state dynamics analysis of a bracket, including fatigue evaluation, has been attached, see `ssd_fatigue001.key`, and Figure 28.

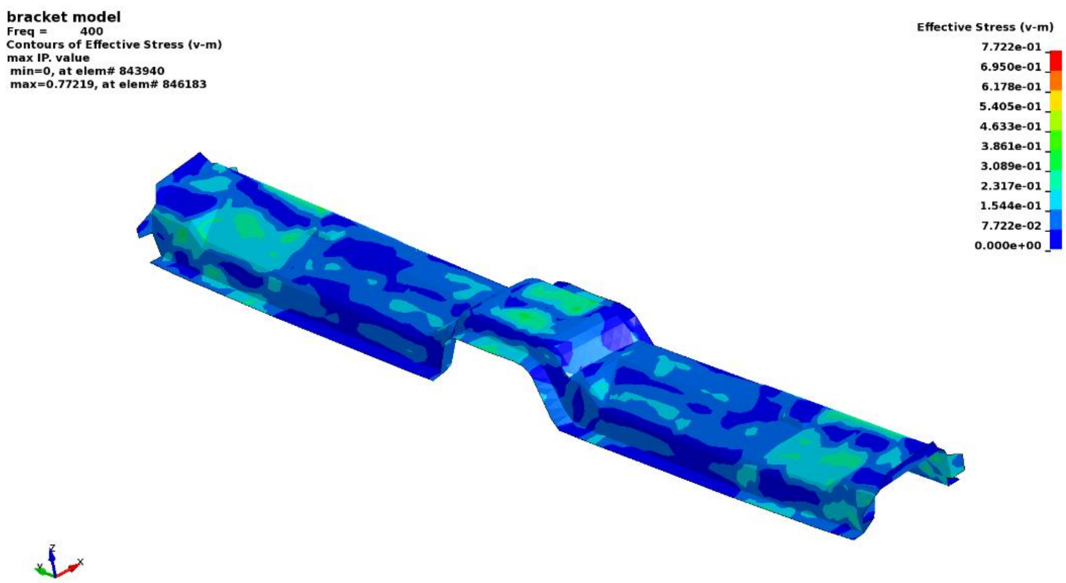


Figure 28. Steady state dynamics analysis of a bracket. The image shows the effective stress (von Mises) response at 400 Hz.

From R11.1 of LS-DYNA, a direct solution option is available for steady state dynamics, activated by the keyword `*FREQUENCY_DOMAIN_SSD_DIRECT` (or `*FREQUENCY_DOMAIN_SSD_DIRECT_FREQUENCY_DEPENDENT` for cases with frequency dependent material properties). Using a direct solution approach may be advantageous if only a few excitation frequencies of a large model are to be analyzed, since in such a case the eigenvalue analysis may be quite time-consuming. To get output of stresses it is required to

[6] Available in R9.0.1 smp only, but in R10 and later, both smp and mpp implementations are available.

set the variable *ISTRESS* = 1 on Card 3 of the *FREQUENCY_DOMAIN_SSD_DIRECT keyword. The direct solution option is only available in smp/LS-DYNA (up to and including R15.0.2). See Ref. [31] for further details.

4.8.3 Random vibration and fatigue analyses

Random vibration is non-deterministic motion, for example a vehicle riding on a rough road. Structures subjected to random vibrations are usually studied using statistical or probabilistic approaches. The power spectral density (PSD) is commonly used to specify a random vibration process [8]. In LS-DYNA, a stationary random process, meaning that the statistics describing the process does not change over time, is assumed. The keyword *FREQUENCY_DOMAIN_RANDOM_VIBRATION activates the random vibration analysis, and by adding the option _FATIGUE also the fatigue damage of a component subjected to the random process can be evaluated. Many different fatigue evaluation methods are implemented in LS-DYNA, such as

- Dirlik's method based on the 4 moments of PSD,
- Wirschinig's method,
- The narrow band method,
- Steinberg's three-band technique, considering the number of stress cycles at the levels 1σ, 2σ, and 3σ levels,

and some more. Dirlik's method is considered [8] to be the most useful for general purpose applications.

Note that stresses are required for the fatigue analysis, which in turn means that it is required to set the parameter *MSTRES* = 1 on card 2 of the *CONTROL_IMPLICIT_EIGENVALUE – keyword in the preceding eigenfrequency analysis. It is also required to request the binary database d3rms using the *DATABASE_FREQUENCY_BINARY_D3RMS keyword, as well as the *DATABASE_FREQUENCY_BINARY_D3FTG keyword in order to get 3D binary plot files for visualization of fatigue analysis results. Consider switching to Lobatto quadrature for shell elements when performing fatigue analyses, see Section 5.2, to get integration points exactly on the inner/outer surface, and also activating output of individual shell in-plane integration points, see Section 5.4.

For a random vibration analysis, the PSD of displacements, velocities, accelerations and stresses are output in the 3D-binary database d3psd. Also, the root mean square results for the same quantities are output in the 3D-binary database d3rms. Nodal and element PSD results are available in the binout results elout_psd and nodout_psd. Fatigue results are output as stress components in the 3D binary database, see Table 3. LS-PrePost will present the correct result labels when a d3ftg file is opened.

The irregularity factor is a real number from 0 to 1, where a sine wave has an irregularity factor of 1 and white noise has an irregularity factor of 0. The less the value is, the closer the process is to the broad band case; the larger the value is, the closer the process is to the narrow band case.

Table 3. Fatigue analysis results are output as stress components in the d3ftg 3D binary database.

Result / component	Fatigue analysis results	Comment
σ_{XX}	Cumulative damage ratio	
σ_{YY}	Expected fatigue life	
σ_{ZZ}	Zero-crossing frequency	
σ_{XY}	Peak-crossing frequency	
σ_{YZ}	Irregularity factor	
σ_{XZ}	Expected fatigue life	From R12
ε_p (1)	Counted cycles	From R12

Notes: (1) Accumulated effective plastic strain.

A template for a random vibration fatigue analysis follows:

```
*FREQUENCY_DOMAIN_RANDOM_VIBRATION_FATIGUE
Card 1: Select modes
Card 2: Define damping. Note! This is mandatory. Some damping must be specified
Card 3: Define loading type and the number of PSD loadings and fatigue data
(SN-curves) definitions
Card 4: Define exposure time and type of S-N curves
Card 5: Define the excitation PSDs (repeat if multiple excitations are present)
Card 6: Define the fatigue data for each PID
*DATABASE_FREQUENCY_BINARY_D3PSD
Define frequency range, spacing etc. for output
*DATABASE_FREQUENCY_BINARY_D3RMS
1,
*DATABASE_FREQUENCY_BINARY_D3FTG
1,
```

The excitation PSDs must be defined for positive frequencies only, zero or negative frequencies in the load curve definitions will lead to Errors. Note that it is required to specify some kind of damping; Card 2 cannot be all zero, or blank.

From R9.0.1, the fatigue properties (S-N-curves) for each material, respectively, may instead be specified using the keyword `*MAT_ADD_FATIGUE` if the variable *NFTG* on card 5 is set to -999.

In random vibration analysis, special care must be taken when computing the RMS of the von Mises effective stress; it cannot be obtained directly from the RMS of the stress components. In LS-DYNA, the method of Ref.[12] is implemented, and the von Mises stress is stored in `d3psd` and `d3rms` instead of plastic strain. LS-PrePost (version 4.3 and later) will display this value instead of computing from stress components[7].

A basic example of random vibration analysis of a link arm, including fatigue evaluation, has been attached, see keyword file `set_4.key` and Figure 29.

[7] Other post-processors might not be aware of this. Then check if the plastic strain and the von Mises stress gives the same values from a `d3rms` – file.

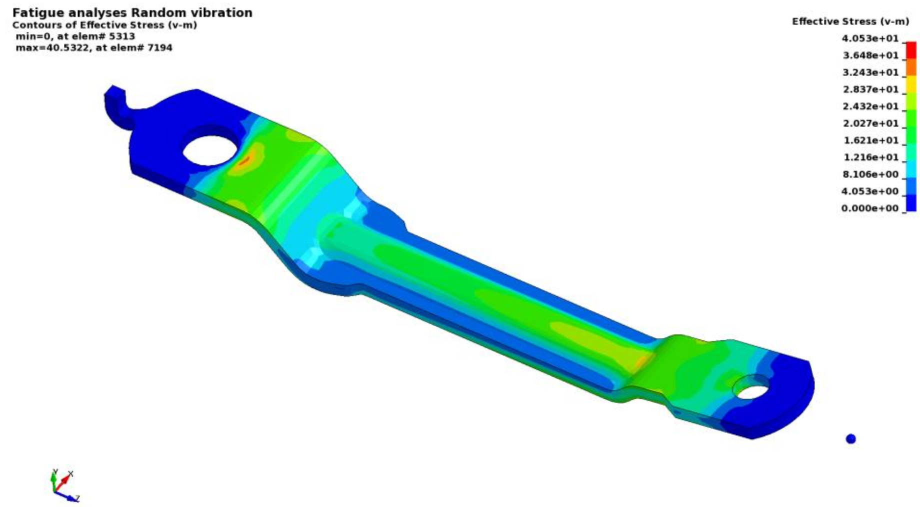


Figure 29. RMS effective stress from random vibration analysis of a link arm.

4.9 Non-linear implicit dynamic analysis

Fully non-linear (including contacts, material non-linearities and large deformations etc.) implicit dynamics, is activated in LS-DYNA by the keyword `*CONTROL_IMPLICIT_DYNAMICS`. The main purpose is obviously for performing fully non-linear, transient dynamic analyses, but the procedure can also be useful for the initial phase of analyses of assemblies with "loose" parts, like a bolted joint, see Section 4.3.2. A template for using implicit dynamics follows:

```
*KEYWORD
*INCLUDE
control_cards_nonlin.key
*DEFINE_CURVE_TITLE
Implicit time incrementation
700,
0., dt0 (first timestep)
Additional lines to define time incrementation
*CONTROL_IMPLICIT_DYNAMICS
1, GAMMA, BETA, , , , , ALPHA
*INCLUDE
database_cards_dynamic.key
*CONTROL_TERMINATION
Define end time of the simulation
*INCLUDE
Include file defining geometry, materials etc.
*LOAD_...
Define nodal loads etc.
*BOUNDARY_...
Data line to prescribe boundary conditions
*TITLE
Simulation title
*END
```

The parameters *GAMMA* and *BETA* of the `*CONTROL_IMPLICIT_DYNAMICS` keyword control the time integration and the amount of numerical damping that is introduced by the implicit time-integration scheme. The default values are `GAMMA` = 0.5 and `BETA` = 0.25, which corresponds to energy being conserved. Even for purely transient dynamics simulations, numerical damping may be beneficial for convergence and stability reasons. For quasi-static loading, assembly of parts by bolt pre-tensioning or similar situations, higher values are recommended, for example `GAMMA` = 0.6 and `BETA` = 0.38.

For long duration events (typically several seconds) or for structures undergoing curved or rotational motion, a composite time-integration scheme is available, and can be activated by setting the parameter `ALPHA` > 0. By setting `ALPHA` = 0.5, and using the default values of γ and β, the Bathe time integration scheme, which is reported to preserve energy and momentum to a reasonable degree, is obtained.

4.9.1 Damping options in non-linear implicit dynamics

To obtain realistic simulation results (that correlate with physical measurements) it is often required to consider the energy dissipation (internal friction, viscous effects etc.) or *damping* in vibrating structures. In LS-DYNA, several options exist for introducing damping, on global and part level, and the relevant keywords all start with `*DAMPING_` ..., see Ref [23]for an overview. To introduce Rayleigh damping, which forms the damping matrix **C** as a linear combination of the mass matrix **M** and stiffness matrix **K**,

$$C = \alpha M + \beta K$$

use the keywords `*DAMPING_GLOBAL` or `*DAMPING_PART_MASS` for the mass-proportional term, and the keyword `*DAMPING_PART_STIFFNESS` for the stiffness proportional term. Note that Rayleigh damping is frequency dependent, with a damping ratio ξ of

$$2\xi = \frac{\alpha}{\omega} + \beta\omega$$

see also Figure 30 for an illustration of the varying damping ratio. This also implies that the damping coefficients α and β will depend on the unit system applied. Some optimization process may be required to find values of α and β that give a damping that is close enough to the desired damping ratio.

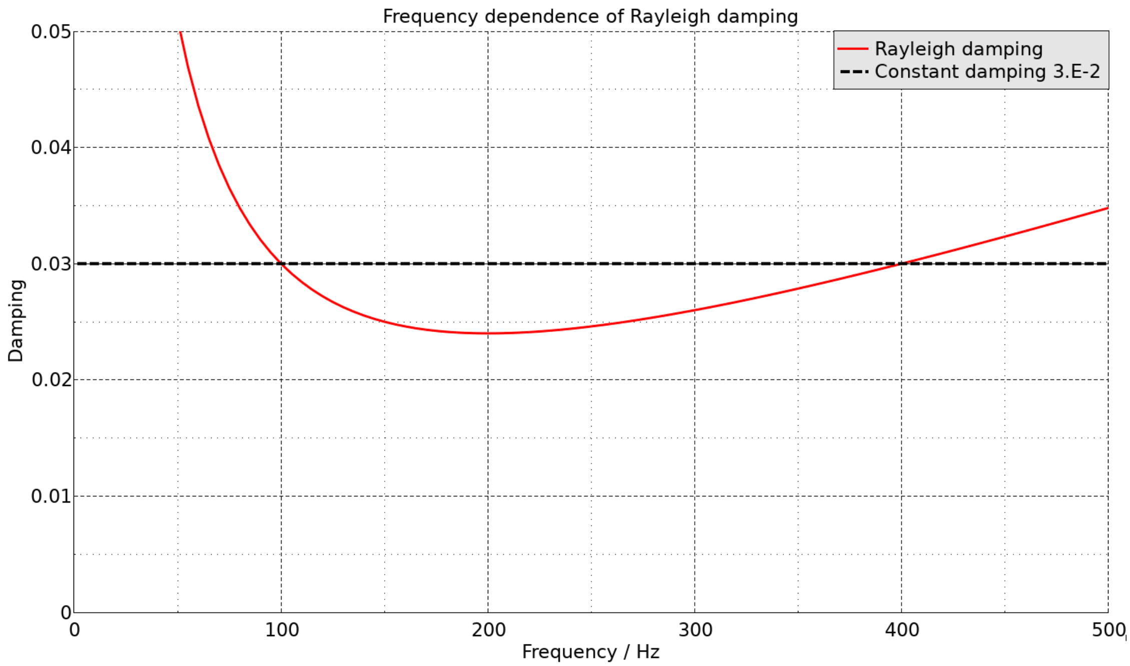


Figure 30. Frequency dependent damping ratio of Rayleigh damping (red line) versus constant damping of 3 %.

The default implementation of the stiffness damping (`*DAMPING_PART_STIFFNESS`) is best suited for damping out high frequency oscillations in explicit analyses. To obtain classical stiffness damping, input a negative damping value, *COEF* = $-\beta$. The stiffness damping is introduced via a viscous term in the material routines. This requires that *IRATE* < 1 on `*CONTROL_IMPLICIT_DYNAMICS` for the stiffness damping to have effect.

For example, to introduce Rayleigh damping of approximately 3 % in the frequency range 100 to 300 Hz on a global level, the following keyword template can be used:

```
*DAMPING_GLOBAL
0, 30.159
*SET_PART_LIST_GENERATE_TITLE
All parts for stiffness damping
       338
1, 99999999
*DAMPING_PART_STIFFNESS_SET
338, -1.9099e-5
*CONTROL_IMPLICIT_DYNAMICS
1,
```

Approximately frequency independent damping can be applied using the keyword `*DAMPING_FREQUENCY_RANGE_DEFORM`. In this case, the fraction of critical damping is given, and the frequency range for application, $F_{\text{low}} < F < F_{\text{high}}$, is specified. The method is best suited for small damping ratios (< 0.05) and frequency ranges such that $10 \le F_{\text{high}}/F_{\text{low}} \le 300$. The drawback with this method is that the dynamic stiffness of the model will increase, leading to slight over-estimation of eigenfrequencies. This can be overcome by reducing the Young's modulus of the materials. For example, for 1 % damping across a frequency range of 30 to 600 Hz, the average error across the frequency range is about 2 %. It would therefore be appropriate to reduce the stiffness by $(1.02)^2$, that is, by 4 % in this case.

For example, to introduce approximately frequency independent damping of 3 % in the frequency range 100 to 300 Hz on a global level, the following keyword template can be used:

```
*DAMPING_FREQUENCY_RANGE_DEFORM
3.E-2, 100., 300.,
```

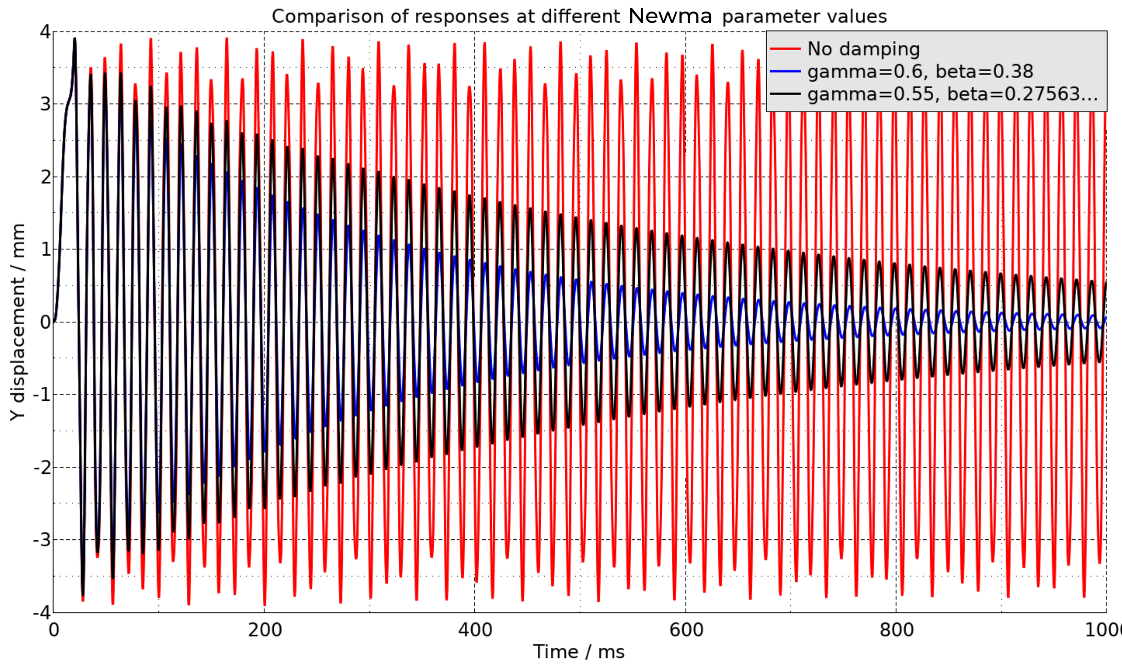


Figure 31. Comparison of tip deflection for different non-linear solutions of the L-beam example of Section 4.7.1.

The Newmark time integrator will also introduce *numerical* damping in the solution, see Figure 31 where this is illustrated for the non-linear solutions of the L-beam example of Section 4.7.1. This artificial numerical damping is unphysical but may in some situations be beneficial since it can act to stabilize the solution and promote convergence. In addition to the settings of *GAMMA* and *BETA*, the numerical damping will depend on the (lowest) natural frequency of the structure, and the time step used in the numerical solution. See for example Ref. [32] for a discussion of numerical damping associated with implicit time integrators.

4.9.2 Transient loading of a L-beam with contact

This example is very similar to the one of Section 4.7.1, but a fully constrained, rigid transverse support (green in Figure 32) is added. Loads and boundary conditions are the same as in Section 4.7.1. A contact condition is defined between the support and the beam using `*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR`. A frequency-independent (for the range 50 - 300 Hz) damping of 3 % of critical is applied using `*DAMPING_FREQUENCY_RANGE`. Moderate numerical damping is also applied by using *GAMMA* = 0.55 and *BETA* = 0.27563. The transient displacement response of the loaded node is shown in Figure 33. The response is quite chaotic at the beginning, but decays quickly due to the combined effect of the damping definitions.

The example keyword file is `transient_nonlin001.key`.

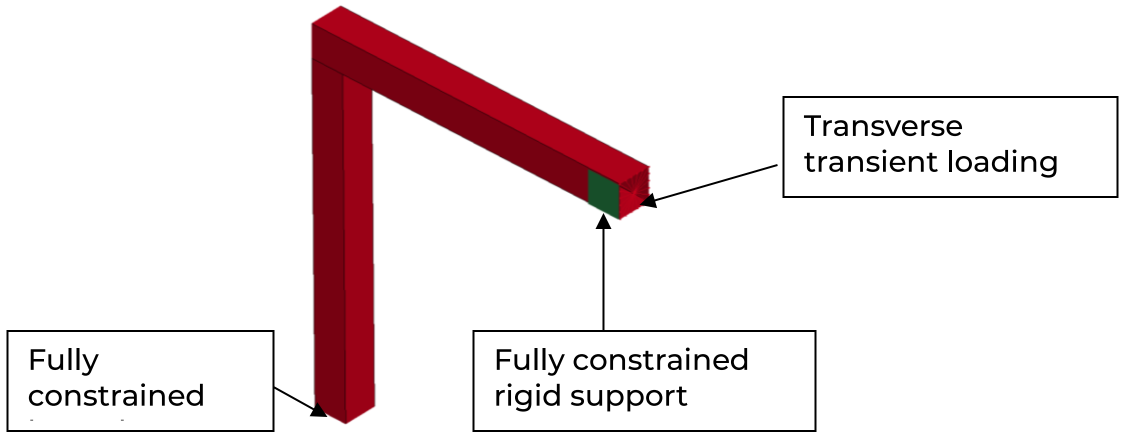


Figure 32. Geometry for the L-beam (red) with rigid support (green).

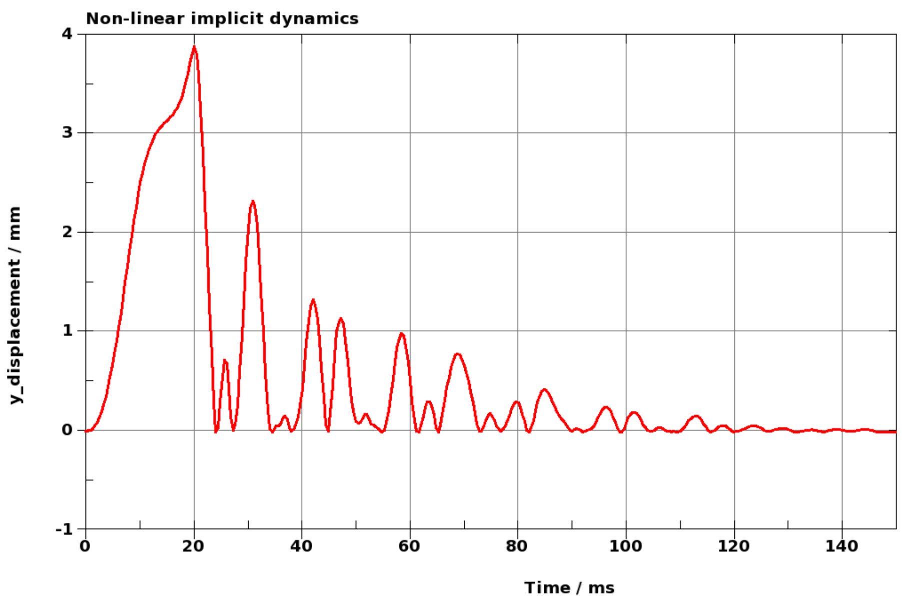


Figure 33. Displacement history in the transverse direction of the loaded node.

4.10 Rotational dynamics

The rotational dynamics analysis involves studying the vibrational or transient response of a rotating (sub) structure, such as balance shafts, turbines and rotating disks in hard disk drives. Rotational dynamics in LS-DYNA [15] is activated by the keyword `*CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS`.

The transient analysis option is supported in both smp and mpp, while the eigenvalue computations (for generation of Campbell diagrams) is currently only supported in smp. The eigenvalue analysis option (activated by setting *NOMEG* > 0) will automatically activate the non-symmetrical eigensolver.

The direction of the rotational axis is defined using `*DEFINE_VECTOR`, or `*DEFINE_VECTOR_NODES`, where in the latter case the direction will be updated during the simulation according to the motion of the referenced nodes. For a transient analysis, either a constant or a varying angular velocity can be prescribed. The analysis for the rotating parts can be performed using a fixed or rotating coordinate system. If a fixed coordinate system is used, the rotational motion of the rotating subassembly should be applied using for example *INITIAL_VELOCITY_GENERATION. If a rotating coordinate system is used, then optionally the rotational motion can be superimposed to the deformation of the rotating subassembly for visualization purposes.

For brake squeal analyses, it is mandatory to activate non-symmetrical matrix storage and factorization (set *LCPACK* = 3 on *CONTROL_IMPLICIT_SOLVER).

4.10.1 A spinning and rotating shaft with an off-center disk

A disk is attached slightly off-center to a rotating shaft (ω = 200). The shaft is attached to a rigid fixture, see Figure 34. The fixture in turn is spinning around the X-axle. The rotational axis of the shaft is defined using *DEFINE_VECTOR_NODES and two nodes in the rotating fixture. A rotating coordinate system is applied for the rotational dynamics analysis.

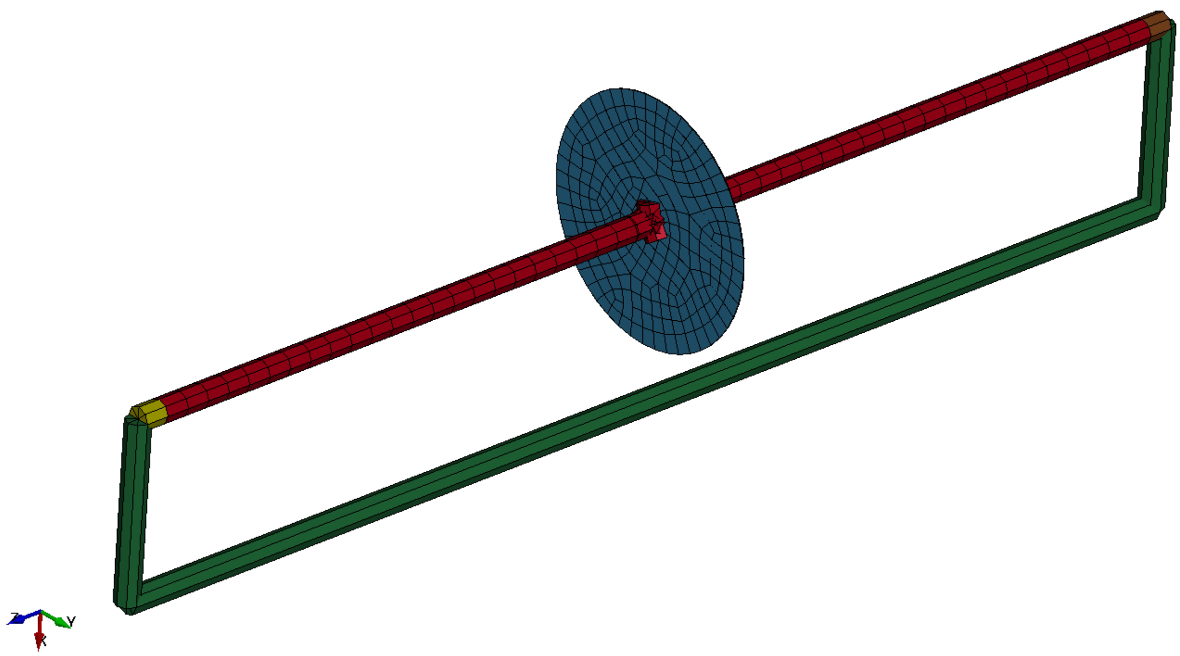


Figure 34. Example of rotational dynamics. A deformable shaft (red) with an off-center disc (blue) is attached to a rigid fixture (green). The fixture spins about the global X-axle.

The keyword file for this example is run_rotodyn.key. The analysis is performed using non-linear transient dynamics (compare Section 4.9) and the control cards of control_cards_nolin.key. To account for the rotational motion, the composite Bathe time integration scheme is used (*ALPHA* = 0.5 on *CONTROL_IMPLICIT_DYNAMICS), compare Section 4.9.

4.11 Thermal analyses

In order to perform a thermal analysis in LS-DYNA (thermal only or coupled structural thermal) some additional control cards to the ones presented in Table 1 are required. It is also required to assign thermal material properties to all parts of the model (*TMID* on *PART), see Section 7.1.

Suggested *additional* control card settings for thermal analyses are given in

- control_themal_steady.key, for linear steady state analyses, and

- `control_themal_transient.key`, for transient non-linear thermal, or coupled thermal and structural, analysis.

The idea is to use these control card files in conjunction with the structural settings, as presented in Table 1.

A template for steady state linear thermal analyses follows:

```
*KEYWORD
*INCLUDE
control_cards_linear.key
*INCLUDE
database_cards_static.key
*INCLUDE
control_thermal_steady.key
*CONTROL_TERMINATION
Define end time of the simulation
*INCLUDE
Include file defining geometry, materials etc.
*INITIAL_TEMPERATURE_...
Define initial temperatures
*BOUNDARY_TEMPERATURE…
Define nodal temperature boundary conditions
*BOUNDARY_CONVECTION...
Define convection boundary conditions
*DATABASE_HISTORY_NODE_...
Define nodal points of interest for tabular output
*DATABASE_TPRINT
1.,
*TITLE
Simulation title
*END
```

For transient thermal and coupled thermal structural analyses, instead use the thermal control card file `control_thermal_transient.key`. The keywords `*CONTROL_SOLUTION` (for selecting analysis type) and `*CONTROL_THERMAL_TIMESTEP` must also be defined.

- On `*CONTROL_SOLUTION`, set *SOLN* = 1 for a thermal only analysis, or 2 for a coupled thermal structural analysis.
- On `*CONTROL_THERMAL_TIMESTEP`, specify thermal timestep parameters and for thermal only an optional[8] limit on maximum temperature change per time step.

A template for a transient thermal only analysis follows:

```
*KEYWORD
*INCLUDE
Include file defining geometry, materials etc.
*INCLUDE
control_cards_nonlin.key
*INCLUDE
database_cards_static.key
```

[8] Set a high value of *DTEMP* to deactivate this feature.

```
*INCLUDE
control_thermal_transient.key
*CONTROL_SOLUTION
$#     soln       nlq     isnan     lcint
          1         0         0      1000
*CONTROL_THERMAL_TIMESTEP
Specify time step parameters (optional limit on temperature increment)
*CONTROL_TERMINATION
Define end time of the simulation
*INITIAL_TEMPERATURE_...
Define initial temperatures
*BOUNDARY_TEMPERATURE…
Define nodal temperature boundary conditions
*BOUNDARY_CONVECTION...
Define convection boundary conditions
*DATABASE_HISTORY_NODE_...
Define nodal points of interest for tabular output
*DATABASE_TPRINT
1.E-5,
*TITLE
Simulation title
*END
```

For a thermal only analysis, detailed thermal time stepping control can be used (`*CONTROL_THERMAL_TIMESTEP`), by specifying upper and lower limits on the thermal time step (*TMIN, TMAX*) or a load curve with break points (*LCTS*). Also, the time step will be limited such that the max temperature increment in a time step is ≤ *DTEMP*. Note that *DTEMP* must be specified since the default value is 1 – set a high value to deactivate this control feature.

For a coupled thermal structural analysis, using a constant time step is recommended. Simply set the initial thermal time step as the max structural time step. Naturally, care must be taken when synchronizing thermal and structural events in a coupled analysis.

A template for a coupled thermal structural analysis follows:

```
*KEYWORD
*INCLUDE
Include file defining geometry, materials etc.
*INCLUDE
control_cards_nonlin.key
*INCLUDE
database_cards_static.key
*INCLUDE
control_thermal_transient.key
*CONTROL_SOLUTION
$#     soln       nlq     isnan     lcint
          2         0         0      1000
*DEFINE_CURVE_TITLE
Implicit time incrementation
700,
0., dt0 (first timestep)
Additional lines to define time incrementation and synchronization with loadings
*CONTROL_THERMAL_TIMESTEP
Set constant thermal time step = max structural timestep
```

```
*CONTROL_TERMINATION
Define end time of the simulation
*LOAD_...
Define nodal (structural) loads etc.
*BOUNDARY_...
Data line to prescribe structural boundary conditions
*INITIAL_TEMPERATURE_...
Define initial temperatures
*BOUNDARY_TEMPERATURE…
Define nodal temperature boundary conditions
*BOUNDARY_CONVECTION...
Define convection boundary conditions
*DATABASE_HISTORY_NODE_...
Define nodal points of interest for tabular output
*DATABASE_TPRINT
1.E-5,
*TITLE
Simulation title
*END
```

4.11.1 Output from thermal analyses

History output of nodal temperatures and heat flux, as well as thermal contact results, can be output to the file `tprint` by the keyword `*DATABASE_TPRINT`. Averaged results per part are also output. By default, this will output results of all nodes in the model and is therefore not included in the suggested database card files (see Table 1). In order to limit the thermal output, define nodes of interest using `*DATABASE_HISTORY_NODE(_SET)`.

In order to visualize temperature gradients through shells, set the variable *THERM* = 3 on `*DATABASE_EXTENT_BINARY`. This can be visualized in LS-PrePost by turning on the visualization of shell thickness but may make the database incompatible with other 3rd party post-processors [1], and is therefore not included in the suggested database card files.

4.11.2 Coupled structural thermal analysis example

A flat panel is fully constrained at one edge, see Figure 35. The initial temperature of the moving block (yellow in Figure 35) is 500 K, while all other parts have an initial temperature of 300 K. First, the moving block is placed on the panel (by gravity) for a duration of 100 s, then a motion is applied to close the contact with the fixed block (blue in Figure 35). For more details on thermal contacts, see Section 6.1.8. The temperature distribution after 1 s is shown in Figure 36 (cut view) and after 500 s in Figure 37. Note that the shell thickness is also visualized in these images, and that the fringe limits differ.

The example keyword file is `thermal.key`. For further examples of thermal analyses, see also lsdyna.ansys.com (https://lsdyna.ansys.com/knowledge-base/thermal/).

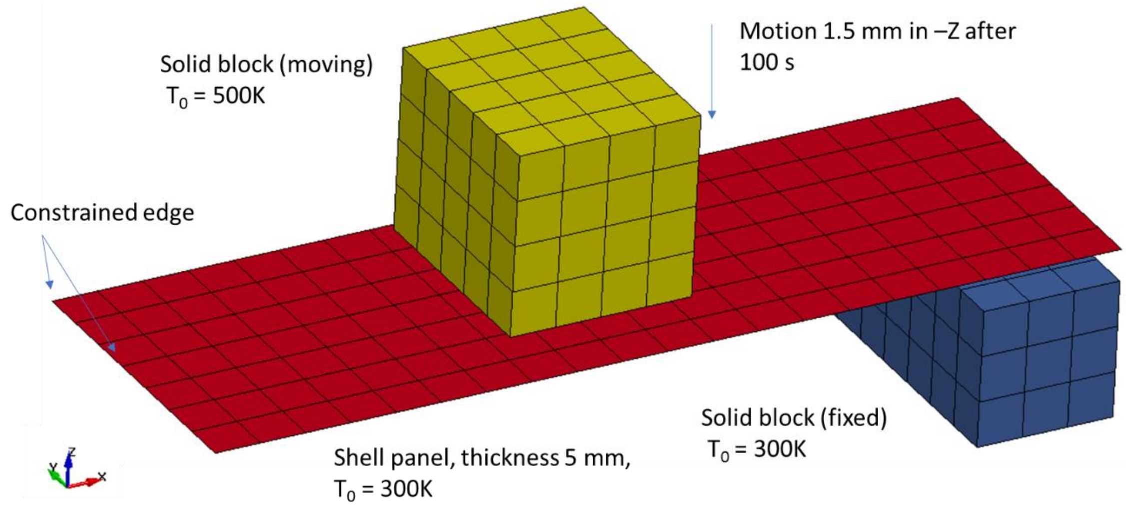


Figure 35. Thermal and mechanical loading of a panel (red, shown without shell thickness). The initial gap between the panel and the fixed support (blue) is 1.0 mm.

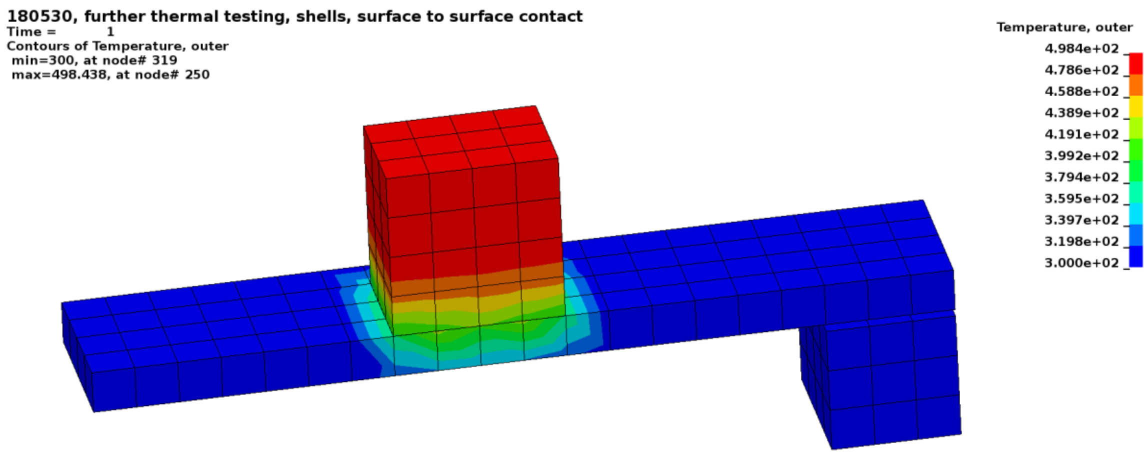


Figure 36. Cut view through the assembly after 1 s. Fringe plot of temperature (300 to 498 K).

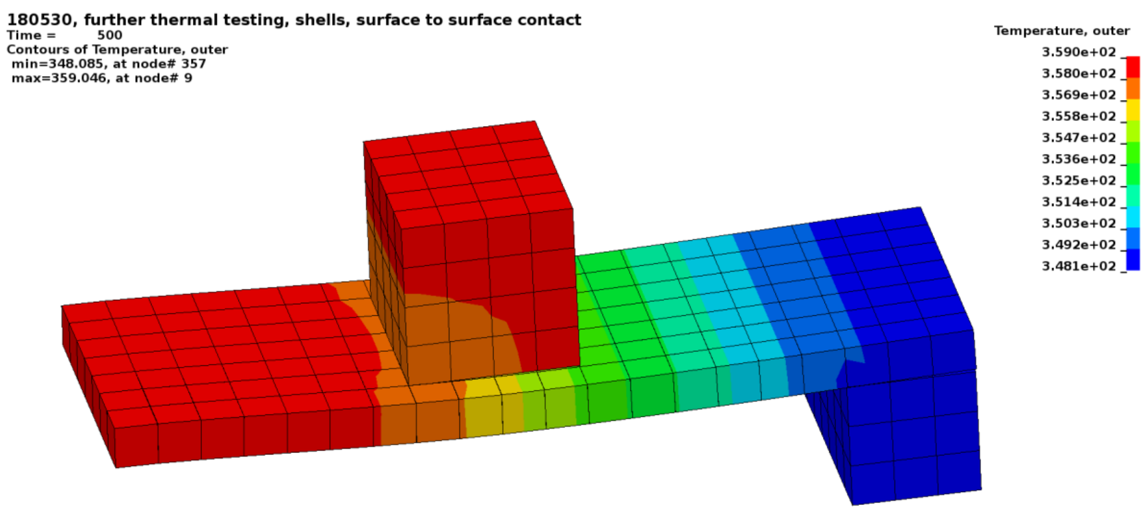


Figure 37. Fringe plot of temperature after 500 s (from 348 K to 359 K).

5 Element types

In the Ansys LS-DYNA software, many different element types and element techniques are available. This section presents a brief overview of elements that may be used in implicit analyses, see Table 4. Depending on the application, other element types or modeling techniques may be a better fit. There are several special beam, shell, and solid element formulations for linear analyses, see Table 5. See Section 4.1 for some more details regarding linear static analyses.

Table 4. Some element types for non-linear implicit analyses

Element type	Comment	LS-DYNA keyword	Element formulation
Beam	Structural beams	`*SECTION_BEAM`	1
	For bolts w. pre-tension		9
	For springs, dampers etc.		6
Shell	1st order	`*SECTION_SHELL`	16
	2nd order		23
Solid	1st order hex	`*SECTION_SOLID`	-2 (-18 from R13)
	1st order tet		(13)
	2nd order tet		16 (17)
	2nd order hex		23

Table 5. Specialized element types for linear implicit analyses

Element type	Comment	LS-DYNA keyword	Element formulation
Beam	Structural beams for linear static analyses	`*SECTION_BEAM`	13(1)
	For springs, dampers etc.		6
Shell	1st order	`*SECTION_SHELL`	18 (20, 21, 99)
Solid	1st order hex	`*SECTION_SOLID`	18 (99)
	1st order tet		(10)

Notes: (1) Beam elforms 1, 2 or 11 are not allowed for linear static analyses in versions prior to R11.

An additional functionality of `*CONTROL_IMPLICIT_EIGENVALUE` (see also Section 4.6) is to specify element types for implicit analyses. This makes it possible to switch element formulations very quickly in, for example a crash model to element formulations better suited for implicit analysis. Note that section properties for beams are not converted when beam element formulation is switched. A template for switching to shell element formulation 20 and solid formulation 18 follows:

```
*CONTROL_IMPLICIT_EIGENVALUE
Data line for eigenfrequency calculation[9]
18,,20
```

5.1 Beam elements

For non-linear analyses, beam element formulation 1 is generally recommended. In some cases, with thick and short beams, element formulation 9 (spotweld beam) is preferred. This element formulation requires the use of the spotweld material model `*MAT_SPOTWELD`. Element formulation 9 can also be used for applying pre-tensioning by the keyword `*INITIAL_AXIAL_FORCE_BEAM`.

For linear static analyses, beam element formulations 13 is recommended. For eigenvalue analyses it is recommended to activate the improved computation of the torsional mass by setting *ITORM* = 1 on `*SECTION_BEAM`. Beam element formulation 13 cannot be used for linear buckling analyses (Section 4.4). Formulations 1, 2 and 11 are not available for linear static analyses in versions prior to R11. For discrete elements (springs, dashpots) beam elform 6 or `*ELEMENT_DISCRETE` may also be used for linear statics.

5.1.1 Discrete elements, springs, and dashpots

For modelling discrete elements, such as springs or dashpots, two different element families are available: the `*ELEMENT_BEAM` with element formulation 6, and the `*ELEMENT_DISCRETE` formulation.

If a discrete element acting along the line between two nodes N1 and N2 is desired, the `*ELEMENT_DISCRETE` – formulation may be used. It is also possible to create a spring between N1 and ground by setting N2 = 0. The spring/damper properties are defined by the material model, for example `*MAT_SPRING_ELASTIC` (the available material models are denoted as `*MAT_S01` – `*MAT_S15` in Ref. [2]). The discrete elements can be assigned an initial pre-tension, or offset, specified by a load curve, using `*ELEMENT_DISCRETE_LCO`.

The `*ELEMENT_BEAM` element formulation 6 option may be applied when a more general behavior is required (for example if an axial displacement should cause a transverse force). The spring/damper properties are defined by the material model, for example `*MAT_LINEAR_ELASTIC_DISCRETE_BEAM` (many more discrete element materials [2] are available, with more complex behavior, typically denoted as `*MAT_..._DISCRETE_BEAM`). These materials can also be assigned a density, which in combination with the *VOL* and *INER* – parameters of the `*SECTION_BEAM` card means that the discrete elements defined using `*ELEMENT_BEAM` element formulation 6 can be assigned mass and inertia properties. In order to obtain a spring which is always aligned between two nodal points, set `SCOOR` = ± 12 on `*SECTION_BEAM`. Orientation of spring elements can also be obtained by a coordinate system (CID on `*SECTION_BEAM`).

[9] If applicable. Leave blank if the intention is to only change element formulations.

5.2 Shell elements

In general, the 1st order shell element formulation 16 works well both for linear and non-linear analyses. It is used in the examples of Sections 4.3.1 and 4.5.1. For some cases with extreme deformation, use of hourglass type 8 (full projection warping stiffness) may be beneficial, see Figure 38 for an example.

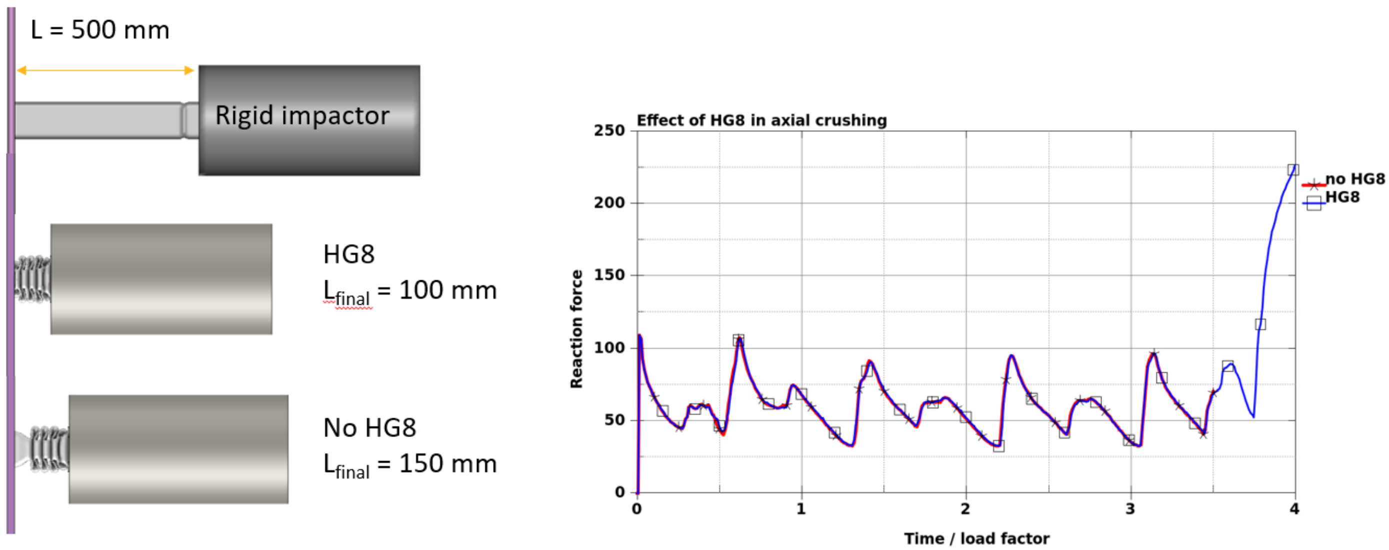


Figure 38. In this example of crash box compression, converted from the explicit simulation of Ref. [22], adding hourglass type 8 makes an additional 50 mm of compression possible before implicit convergence stops.

Also 2nd order shell elements are available for non-linear analyses. Element formulation 23 is an 8-node quadratic quadrilateral shell, and element formulation 24 is a 6-node quadratic triangular shell, see Ref. [39] for details. From R14, the elforms 23 and 24 can be used also in linear static analyses.

There are several shell element formulations that are developed for linear analyses. Shell element formulation 18 uses Kirchhoff plate theory, which makes it suited for thin-walled structures. This element formulation can also be used in linear transient modal dynamics analyses if stresses are of interest. Shell element formulation 20 uses Mindlin plate theory, which makes it suited for thick-walled structures. The element formulation 21 is similar to the CQAD4 element of Nastran.

Quadrilateral and triangular elements can be mixed in the same LS-DYNA part `(*PART)`, they will be automatically sorted to the correct element formulation internally by LS-DYNA[10].

In order to remove the bending stiffness, the user can set the number of through-thickness integration point layers (the parameter *NIP* on the `*SECTION_SHELL` - card) to one for the shell elements. The default number of integration point layers is two, which may be sufficient for linear analyses (if stresses on the inner/outer surfaces of the shells are not of interest). By setting *NIP* ≥ 3 and activating Lobatto

[10] Depending on the setting of the *ESORT* - flag on `*CONTOL_SHELL`. In all the provided control cards files, this feature is active.

quadrature (set *INTGRD* = 1 on *CONTROL_SHELL) stresses at integration points placed exactly on the inner/outer surface of the shell are obtained. This option can be appealing for e.g., fatigue analyses.

For non-linear analyses with plasticity, it is recommended to use (at least) five through-thickness integration point layers ($NIP \geq 5$).

For composite materials, the shell element formulation is specified on the *PART_COMPOSITE card, where also materials are assigned to the different layers of the composite. The default in LS-DYNA is to use one integration point per layer, but it is recommended to activate 3 points per layer Simpson's rule integration by setting OTPTC and *IRPL* = 103 on Irregular Optional Card 2 (unless this split of layers into multiple integration points has not been done manually).

A general review of shell element formulations in LS-DYNA is presented in Ref. [14].

5.3 Solid elements

For linear analyses, the linear solid hexahedron element formulation 18 is recommended. For non-linear analyses, the linear solid hexahedron element formulation -2 is recommended. Note that the solid element formulation 2 will give an overly stiff response in bending for aspect ratios > 1.5 - 2. Element formulations -2 and -1 are enhanced in order to work also for aspect ratios > 2 [5]. From R13.0.0, an enhanced assumed strain hexahedral element with incompatible modes is available as element formulation -18 [44]. This formulation eliminates parasitic shear that generally may cause an overly stiff result in bending for linear elements.

It is possible to mix 1st order hexa and penta elements in the same part, they will be automatically sorted to the corresponding element formulation internally by LS-DYNA.

First order tetrahedra are available as element formulation 10 and 13. It should be noted that the 1st order tetrahedra can give results for bending deformations that are very much too stiff due to volumetric locking. However, 1st order tetrahedra may be useful for example for modelling foam materials. In general, the nodal averaged pressure formulation of elform 13 suffers less from the overly stiff behavior [10], at least when plastic deformation is dominating, but it is only supported by a subset of materials (including MAT_103). Note also that the advantages of the nodal averaging might be lost in some cases, for example when a deformable part is connected to a rigid part using shared nodes and elform 13 is used for both parts (the remedy may then be to switch to elform 10 in the rigid part). Should 1st order tetrahedra be required for linear analyses (see Section 4.1) currently only elform 10 is fully supported.

Two different types of 10-noded tetrahedra are available: element formulation 16 and 17. Elform 16 is in general recommended. It is a standard 2nd order tetrahedral element, which works well with Mortar contact. Also segment based loadings (*LOAD_SEGMENT) are handled consistently.

Elform 17 is modified to have equal nodal weighting factors, so that an applied constant pressure corresponds to equal (and positive) nodal forces. Compared to elfrom 16, the computational cost for elform 17 is higher. In LS-DYNA versions prior to R14.0.0, the implementation of elform 17 is not strongly objective, meaning that finite rotations must be divided into small enough time steps to obtain correct results.

Solid element formulation 23 provides a 20-noded, quadratic hex element (but there is no matching 15-noded quadratic penta element).

A special kind of solid elements are the Cosserat point [21] elements (CPE). They are based on a re-formulation of the standard elements to make them better suited for large deformation analyses of, for example, rubber (see Appendix A for more details). A linear hexahedral CPE is obtained by using element formulation 1 in combination with hourglass type 10 (`*SECTION_SOLID` with *ELFORM* = 1 and `*HOURGLASS` with *IHQ* = 10). A quadratic tetrahedral CPE is obtained by using element formulation 16 in combination with hourglass type 10 (`*SECTION_SOLID` with *ELFORM* = 16 and `*HOURGLASS` with *IHQ* = 10). Note that the CPE implementation is similar to an under-integrated element, which means that hourglass energy can arise. The amount of hourglass energy should be checked to ensure that the simulation results are valid.

A general review of solid element formulations in LS-DYNA is presented in Ref. [10].

5.4 Element integration point output for 3D post-processing

By the provided files `database_cards_...key`, LS-DYNA will output the average element stresses and strains for both shells and solids in the `d3plot` files for 3D postprocessing. For shells, the average element stresses and strains at the top, mid and bottom surfaces of the shells will be output in the `d3plot` files. In order to obtain shell element quantities at the integration points instead (4 points at the top, 4 points at the mid and 4 points at the bottom surface), set *MAXINT* = -3 on `*DATABASE_EXTENT_BINARY`. From R12.0.0 of LS-DYNA, the shell element integration point stresses can be extrapolated to the nodes by setting *SHLSIG* = 1 on `*CONTROL_OUTPUT`.

If stresses and strains at the outmost layer of a shell element are of interest, for example in a fatigue analysis, it is possible to switch to Lobatto integration by setting *INTGRD* = 1 on `*CONTROL_SHELL`. By Lobatto integration, the inner and outer layer of integration points will coincide exactly with the inner and outer surface of the shell, respectively.

For solids, the average element stresses and strains at the center point will be output in the `d3plot` files. In order to obtain solid element quantities at the integration points instead, set *NINTSLD* = 8 on `*DATABASE_EXTENT_BINARY`. In combination with this, stresses from solid elements can be extrapolated to the nodes by setting *SOLSIG* > 0 on `*CONTROL_OUTPUT`.

Another option for obtaining stress results at nodal points of solids or shells is to apply Zienkiewicz-Zhu's superconvergent Patch Recovery method [28]. This is activated by the keyword `*DATABSE_RECOVER_NODE`. By this, the nodal acceleration components in the `d3plot` files will be replaced by the recovered nodal stress values, for a specified part set. An example of this is attached to this Guideline package, as `element_output.key`, see also Figure 39. Note also that different postprocessors may display these results slightly differently.

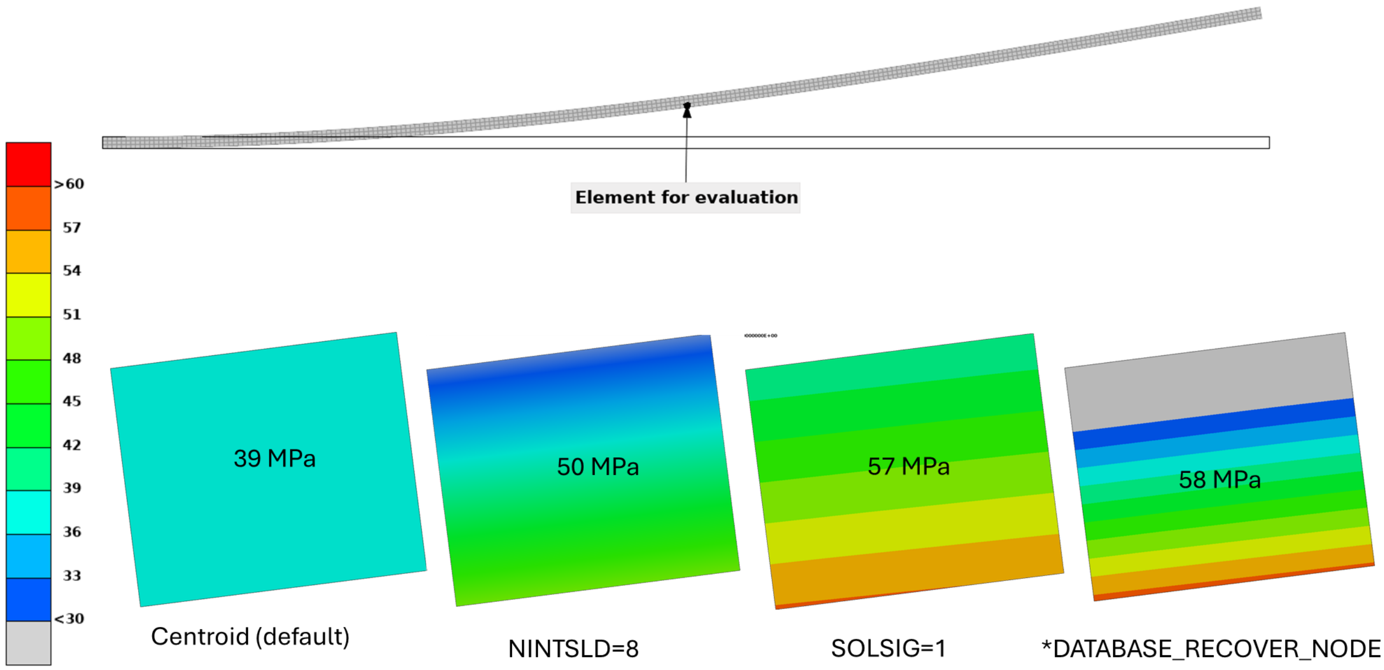


Figure 39. Example of different output options for element results. The Images show fringe plots of the von Mises stress for one element in the center of a cantilever beam subjected to bending by a tip force, using different extrapolation/interpolation options. The numbers show the peak stress from the visualization.

5.5 Elements for 2D analysis

Both axisymmetric, plane strain and plane stress (linear) elements are available in LS-DYNA. The 2D elements are defined in the XY–plane (Z=0). For axisymmetric analyses, the Y-axis (X=0) is the symmetry axis, and all nodes should have X-coordinate ≥ 0.

The respective analysis type is automatically activated by selecting a 2D element type. By choosing axisymmetric elements, the analysis becomes axisymmetric as well. Note that the 2D and 3D element types must not be mixed in the same analysis, and different types of 2D elements must not be used together.

The 2D solid elements are seen as shell elements in LS-DYNA (`*SECTION_SHELL`) and 2D shell elements as beams (`*SECTION_BEAM`). The normals of shells used for defining 2D solids should point in the positive Z-direction. Setting *NIP* = 4 on `*SECTION_SHELL` gives fully integrated 2D solids.

An overview of the suggested element formulations for 2D analyses is presented in Table 6.

Table 6. Element types for implicit 2D analyses

Element type	Comment	LS-DYNA keyword	Element formulation
Shell	Plane strain	*SECTION_BEAM	7
	Axisymmetric		8
Solid	Plane stress	*SECTION_SHELL	12
	Plane strain		13
	Axisymmetric		15

5.5.1 Tube expansion by 2D analysis

An elastic steel cone is inserted into a tube (σ_Y = 580 MPa) with a variable diameter using prescribed motion [16], see Figure 40. The analysis is performed using non-linear statics, and the control cards of `control_cards_nonlin.key`. Since the displacement in this case is quite large (400 mm), the solution control was modified, setting *DNORM* = 1 on `*CONTROL_IMPLICIT_SOLUTION`. A total of 809 axisymmetric solid elements (shell element formulation 15) are used. The example keyword file is `run_2d.key` (NOTE that this example should be run in smp/LS-DYNA double precision. From R14, mpp/LS-DYNA can also be used). The axial force-displacement curve is shown in Figure 41. Note that in axisymmetric analyses, reported forces should be multiplied by 2π to obtain the total force.

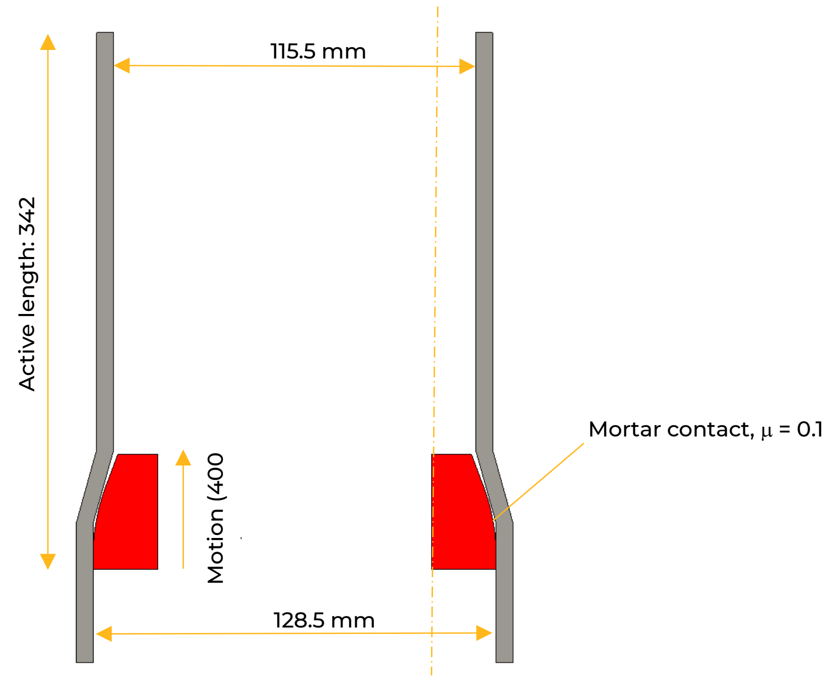


Figure 40. A conical insert (red) is pushed into a tube (gray).

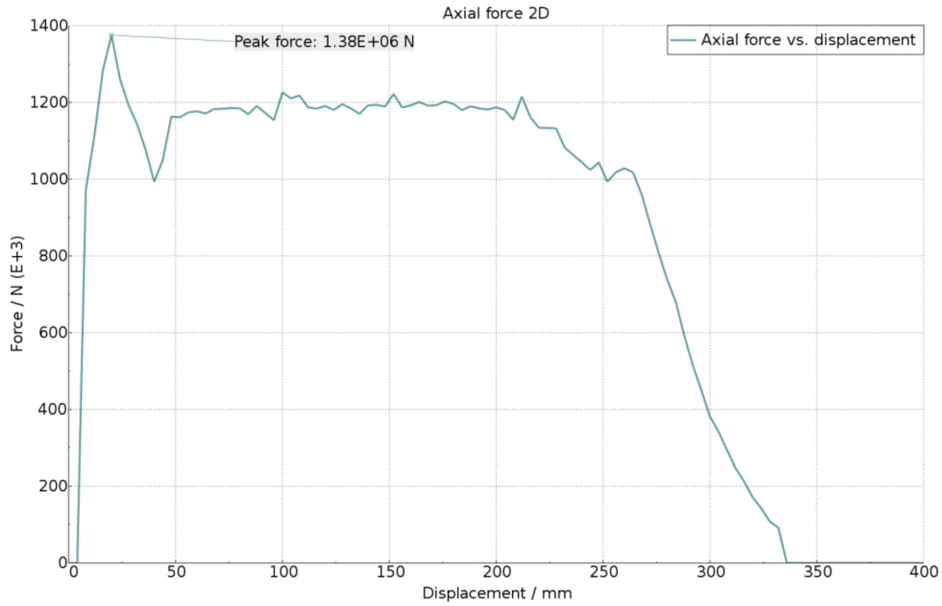


Figure 41. Total axial force vs. displacement for the axisymmetric example.

5.6 Elements for thermal analyses

The structural elements are automatically augmented for thermal analyses. This means that the same element definitions can be used for mechanical as well as thermal and coupled thermo-mechanical analyses. By setting *THSHEL* = 1 on `*CONTROL_SHELL`, a temperature gradient is calculated through the shell thickness (this is active in the provided control card files). To get output of temperatures on shell top/bottom sides, set *THERM* = 3 on `*DATABASE_EXTENT_BINARY`.

5.7 Superelements

Superelements can be applied for replacing parts of structures with their equivalent static and dynamic stiffness properties, see for example Refs. [48][50], in recent versions of LS-DYNA (R15 smp, from R16 also mpp). The procedure consists of two steps:

1. Create the superelement This is a linear procedure, with required keyword `*CONTROL_IMPLICIT_MODES`. This will create a superelement in DMIG (ASCII) format, and binary d3mode – database file(s) for 3D visualization. For large models, where only the superelement creation is of interest, it is possible to disable output of the d3mode – files by setting *ID3MODE* = 1 on Card 2, which can save substantial amounts of disk space.
2. Use the superelement in the "main" simulation. This can be an explicit or implicit linear or non-linear analysis, but the superepelement is not valid for large deformations or finite rotations. The keyword `*ELEMENT_DIRECT_MATRIX_INPUT` imports the previously generated superelement from the DMIG file. To connect the superelement to the main simulation model, either matching node numbers between Creation (Step 1) ans Use (Step 2) can be used, or alternatively, the superelement may be attached by use of tied contact.

A basic example of the procedure for creating and using superelements has been attached in the folder "SuperElement" of the Guideline package, see Figure 42.

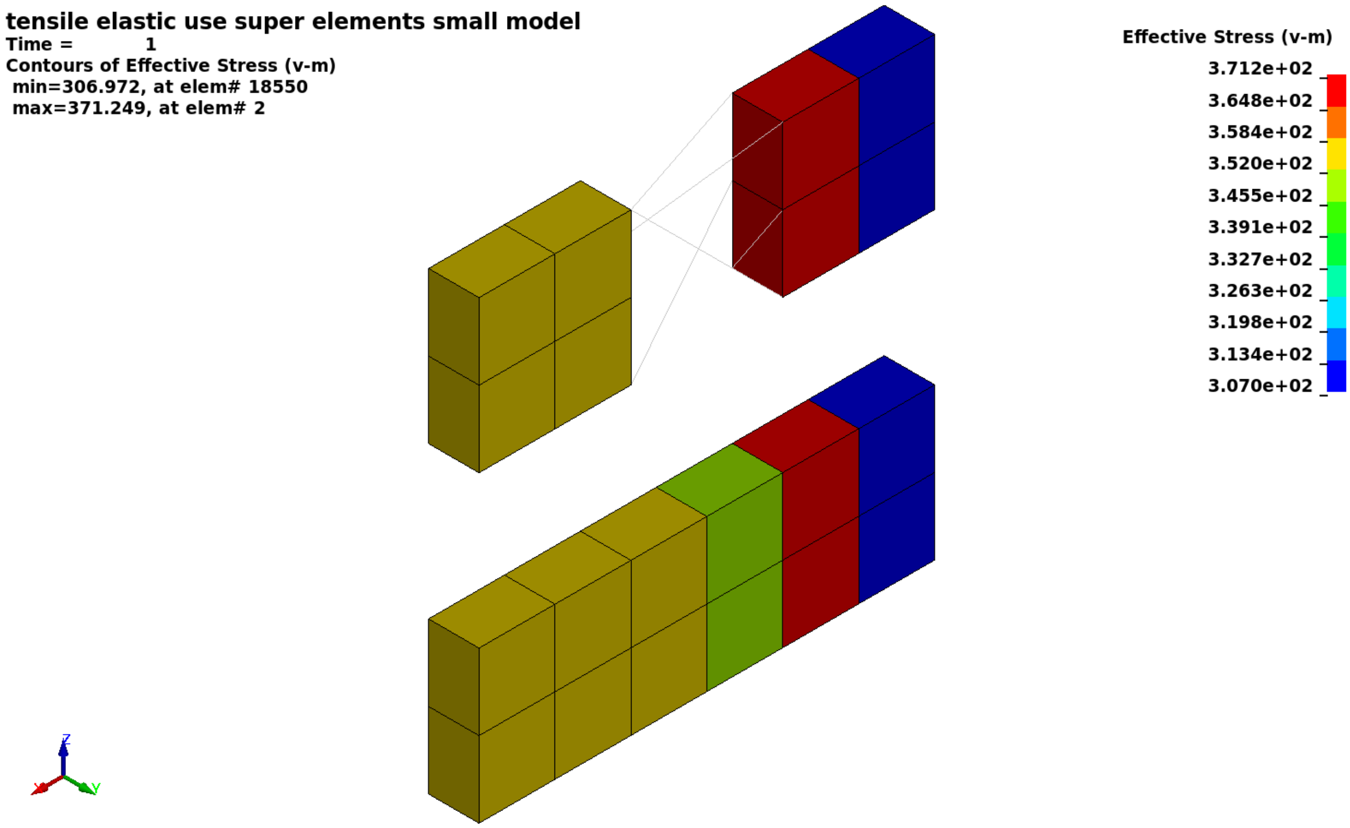


Figure 42. A small example model of a tensile test, where a part of the upper test specimen has been replaced by a superelement.

6 Contacts for implicit analyses

Contacts can be categorized as either

- sliding contacts, with the purpose of allowing movements between parts in a FE-model, without the parts intersecting each other (see Section 6.1), or
- tied contacts, with the purpose of coupling parts together with a constant distance (see Section 6.2).

For a general overview of contact definitions in LS-DYNA, the course "Contacts in LS-DYNA" (or similar) is recommended. See the Webinar, Ref. [40], for an overview of contacts for implicit analyses.

6.1 Sliding interface contact

The Mortar [6] contacts are recommended for modelling sliding contact in implicit analyses. The Mortar contacts are segment based, using a penalty formulation, and allow for finite sliding in each implicit time step. It is a very general contact, which can handle edge-to-edge contacts for shells and solids, as well as contact situations involving beam elements, and element erosion.

The input syntax for the Mortar contacts has been slightly changed from R10 of LS-DYNA, see Section 6.1.4. Not all features of the Mortar contacts are presented here; see the General remark section of the `*CONTACT` – keyword in the LS-DYNA keyword manual [1] for a complete description of the present features of the Mortar contacts.

It is recommended to define the sliding contacts based on parts or part sets. For surfaces involving higher order elements, part or part sets must be used in the contact definitions.

6.1.1 Surface-to-surface contact

For situations where contacts can be defined pairwise in a convenient way between two obvious surfaces, and self-contact need not be considered, the contact type `*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_ID` [6] is recommended. It is generally recommended to use the softer (less stiff material, or coarser mesh) part as tracked[11] surface. A template for using the surface-to-surface Mortar contact (versions R10 and later) follows:

```
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_ID
Define contact ID, Heading
Card 1: Define what shall be in contact using parts or part sets
Card 2: Define friction
Card 3: Define alternative penalty stiffness (SFSA)and thicknesses (SAST) for shells,
leave blank to get defaults
Card 4: blank / optional
Card 5: Define contact thickness for solids (PENMAX)or leave blank
Card 6: optional / Define IGAP and IGNORE
```

Cards 1 - 3 must be defined, while Cards 4 - 6 are optional.

If solid elements are involved in the contacts, a “contact thickness” for these may be specified using the *PENMAX* - parameter[12] on Optional card B (Card 5). The “contact thickness” for solids can be seen as the thickness of the zone below the surface of the solid part where the penalty forces will be applied in order to push out penetrating nodes, see Figure 43. The relative penetration distance, used for calculating the penalty force, is related to this distance (see

Figure 44). This means that by reducing *PENMAX*, a stiffer contact (for solid parts) can be obtained, but at the same time the risk that a tracked segment will be released from the contact increases. It is important to note that the *PENMAX* – parameter for solids specifies a physical distance, which means that the value must be reasonable with respect to the mesh size, as well as to the physical dimension of the involved parts.

If *PENMAX* is left blank, LS-DYNA automatically calculates a value for the contact thickness for solids. This works well in most cases, provided that the mesh in the contact surfaces is reasonably regular.

---

[11] Previously denoted ”slave”.
[12] In versions prior to R10 the “contact thickness” for solids was defined by the *SAST* parameter, see Section 6.1.4.

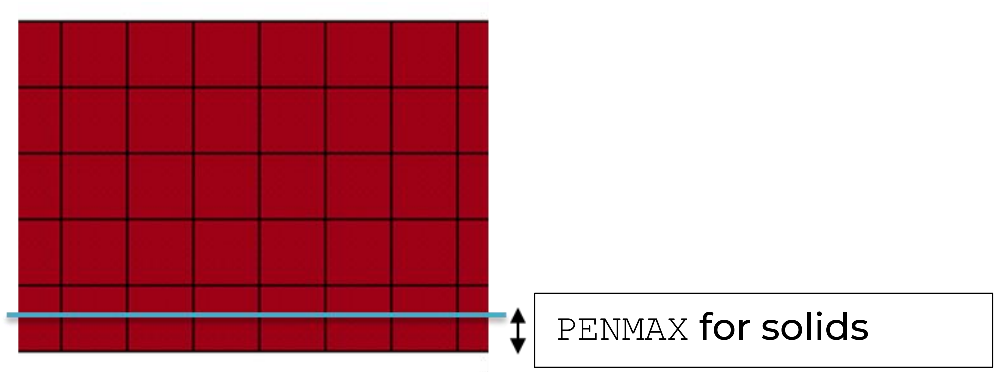


Figure 43. Use of the *PENMAX* – parameter to define the contact thickness for solid parts.

The Mortar contact uses a penalty formulation. This means that there will always be small penetrations between parts in contact. The penetration is required in order to transfer a contact force. The Mortar contact will report both relative as well as absolute values of the maximum penetrations during the simulation in the `mes0*` - files, before each implicit step begins, for example:

```
Contact sliding interface       1010
Number of contact pairs         2784

Maximum penetration is  0.1122220E-02 between
elements      14606 and      10263 on this processor

Maximum relative penetration is  0.9978402E-01 % between
elements      14606 and      10263 on this processor
```

Based on this the user can judge what is acceptable in terms of penetration distance. To reduce the penetrations (if required), the contact stiffness can be increased by the two parameters: *SFSA* and *IGAP*, see Figure 44. The stiffness scale factor *SFSA* controls the initial slope of the contact stiffness and can effectively be increased in order to reduce penetrations. The parameter *IGAP* controls the ramp-up of the penalty stiffness and can be increased to reduce the risk of contacts being released. From R14 of the Ansys LS-DYNA software, the option *IGAP* < 0 will switch to a linear relationship between the penetration and contact pressure, which might be helpful if relatively large penetrations are observed for small contact pressures. In most cases, modifying the default contact stiffness settings is not necessary.

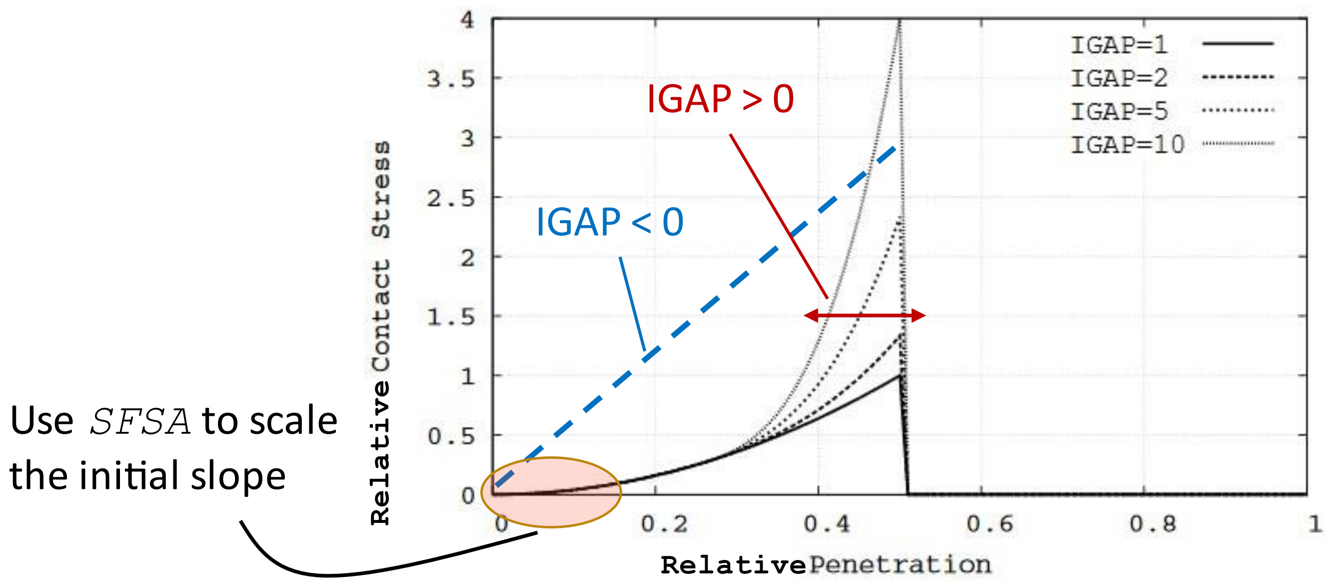


Figure 44. Mortar penalty force as a function of penetration (schematic). Image from Ref. [1].

The *IGNORE* parameter can be used to allow initial[13] penetrations. For two parts with initial penetrations, the option *IGNORE* = 1 will track the penetrations, and if the parts are separated and put into contact again, the first contact will occur at the physical surface of the parts, see Figure 45. The option *IGNORE* = 2 (which is the default behavior) means that the contact surface is moved, so that when the parts come into contact again, it will be at the original position, see Figure 46.

The options *IGNORE* = 3, 4 can be used for resolving initial interferences, such as in a press-fit situation. The penetrations will then be resolved linearly between t = 0 and t = *MPAR1*, as specified on Card 6 of the contact card. The option *IGNORE* = 3 is useful for initial penetrations that are small enough to be detected by the contact algorithm. For larger penetrations, the option *IGNORE* = 4 can be used, in combination with a user-specified search distance (*MPAR2* on Card 6), which shall be at least as large as, and on the order of, the maximum initial penetration. The use of *IGNORE* = 4 for resolving a press-fit between a rubber sleeve and a (rigid) steel housing is illustrated in the example `Insert_and_interference.key`.

[13]Within reasonable limits. It is recommended to use the capabilities of your preferred preprocessing tool to check for and fix unintended initial contact penetrations.

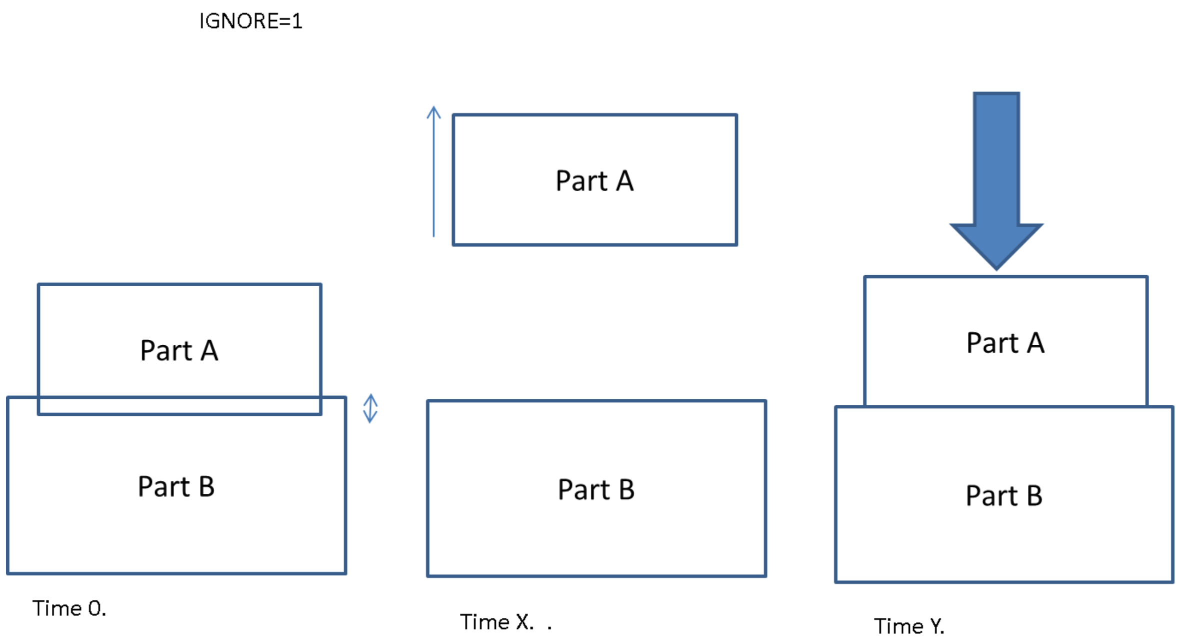


Figure 45. Schematic illustration of IGNORE = 1. Parts A and B are initially penetrating at Time 0. At Time X, the parts are separated, for example by prescribed displacements. At Time Y, the parts are brought back into contact. Contact then occurs at the physical surface of the parts.

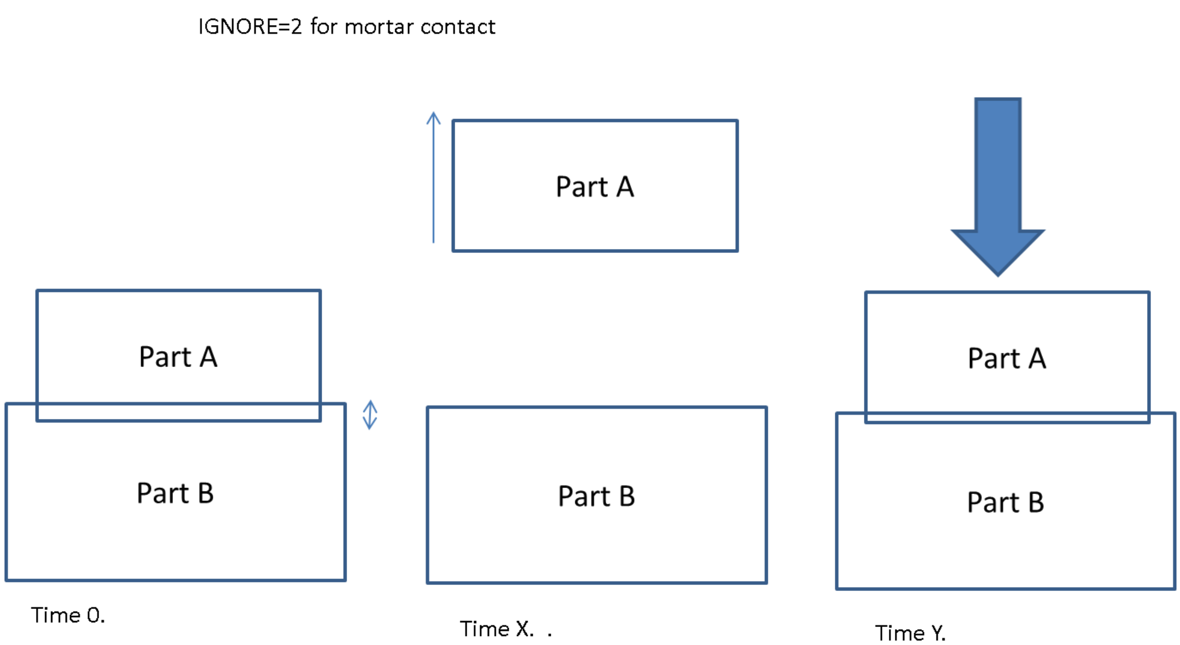


Figure 46. Schematic illustration of IGNORE = 2 for mortar contacts. Parts A and B are initially penetrating at Time 0. At Time X, the parts are separated, for example by prescribed displacements. At Time Y, the parts are brought back into contact. Contact then occurs at the original intersecting surface of the parts.

6.1.2 Single-surface contact

For situations where self-contact must be considered, or a single contact definition is desired, it is recommended to use `*CONTACT_AUTOMATIC_SINGLE_SURFACE_MORTAR_ID`.

A template for using the automatic single-surface Mortar contact follows:

```
*CONTACT_AUTOMATIC_SINGLE_SURFACE_MORTAR_ID
Define contact ID, Heading
Card 1: Define what shall be in contact (only SURFA14) by part or part set
Card 2: Define friction
Card 3: Define alternative penalty stiffness (SFSA)and thicknesses (SAST) for shells, 
leave blank to get defaults
Card 4: blank / optional
Card 5: Define contact thickness for solids (PENMAX)or leave blank
Card 6: optional / Define IGAP and IGNORE
```

For the single-surface mortar contact option, it is possible to specify a version that ignores possible self-contact within the same part, by setting `IGNORE` < 0 (for example setting `IGNORE` = -2 will have the same meaning as `IGNORE` = 2, but contacts between segments in the same part are ignored).

6.1.3 Contact damping

In general, it is not recommended to use contact damping in implicit analyses. For the Mortar contacts, the contact damping settings will be ignored in an implicit static analysis.

6.1.4 Mortar contacts in R9 and earlier

From R10 (rev. 118243) of LS-DYNA, the input for the Mortar contact was modified.

The "contact thickness" for solids (cf. Figure 43) was, in versions prior to R10, defined using the `SAST` –parameter on Card 3. (From R10, the "contact thickness" is defined by the `PENMAX` parameter, see Section 6.1.1.)

A template for using the Mortar surface-to-surface contact in LS-DYNA versions before R10 follows:

```
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_ID
Define contact ID, Heading
Card 1: Define what shall be in contact using parts or part sets
Card 2: Define friction
Card 3: Define alternative penalty stiffness (SFSA, SFM) and thicknesses (SAST)
or leave blank to get defaults
Card 4: blank / optional
Card 5: blank / optional
Card 6: optional / Define IGAP and IGNORE (MPAR1, MPAR2)
```

Note that in version prior to R10, when shells and solids are treated by the same single-surface Mortar contact, it is not possible to use the `SAST` parameter for specifying a contact thickness of the solids, since `SAST` will then also be used as the contact thickness for the shells.

6.1.5 Non-Mortar contacts, *SOFT*, *IGAP* and sticky contact

In some specific cases, it might be desirable to use non-mortar contacts (for example `*CONTACT_SURFACE_TO_SURFCAE_INTERFERENCE_ID` may in some situations be useful for resolving a press-fit). For all non-mortar contacts, a "sticky" behavior is active by default. This means that contact forces are transferred before the surface gap is closed, and that a certain amount of negative contact pressure can be transmitted between surfaces that are pulled apart again, see Figure 47. This behavior

[14]Previously denoted "slave".

is controlled by the *IGAP* parameter on Card 6 (optional card C) of the contact definition. In some situations, using sticky contact (*IGAP* = 1) can aid convergence, but in general, it is recommended to turn it off (*IGAP* = 2). By setting *IGAP* > 2, the sticky contact will be active for the first *IGAP*-2 iterations, and then turned off.

For non-mortar contacts, a segment-based algorithm can be activated by setting *SOFT* = 2.

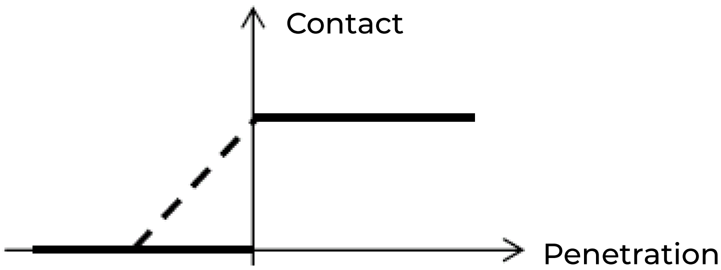


Figure 47. Illustration of contact stiffness as a function of penetration for IGAP = 1 (dashed line) and IGAP = 2 (solid line) for non-Mortar contacts.

6.1.6 Rigid walls

The `*RIGIDWALL` option in LS-DYNA can be seen as a special case for sliding contact, where the user can specify contact (including friction) with an analytical rigid surface. This contact type is mainly developed for explicit (crash event) analysis. Even though there may be situations where rigid walls work[15] also for implicit analysis, it is in the general case recommended to mesh the rigid wall as a "normal" part, using `MAT_RIGID`, and use a Mortar contact definition for imposing the desired contact condition.

6.1.7 Sliding contacts in eigenvalue analyses and linear implicit analyses

Non-Mortar sliding contacts will (in some cases) be linearized in eigenvalue analyses: if the contact surfaces initially are slightly over-closed or at least touch exactly, the surfaces may be "tied" in the normal direction, but free to move relative to each other in the tangential direction. If the gap between the contact surfaces is initially open, no contact forces will be transferred, see Figure 48.

The linearization procedure for non-Mortar contacts may lead to unintended overestimation of eigenvalues. For example, if an FE-mesh is taken from an automotive crash simulation model, where initial penetrations may exist, and non-Mortar contacts are used, the linearization may "tie" parts together, which then in turn leads to unrealistically high eigenvalues. In cases like this, care must be taken so that tied contacts are only introduced where intended, in a way controlled by the user. Some of the options for eigenvalue analyses are then:

- If no additional tied contacts are intended, simply remove the non-Mortar contact definitions, or alternatively replace them all with Mortar contacts.

[15] For example, shell structures involving metallic materials may work reasonably well with `*RIGIDWALL` in implicit, while for example plastic components meshed with solid elements may end up with unrealistically large penetrations.

- If a behavior where parts appear as "glued" or "welded" together is desired, introduce tied contacts (see Section 6.2) in the appropriate areas.
- If linearized contacts (where parts in contact are kept together but can slide against each other) are desired: start with a penetration-free model, apply preloading (bolt pre-tension, gravity etc.) in a non-linear step and then do an intermittent eigenvalue analysis (see Section 4.6.3).

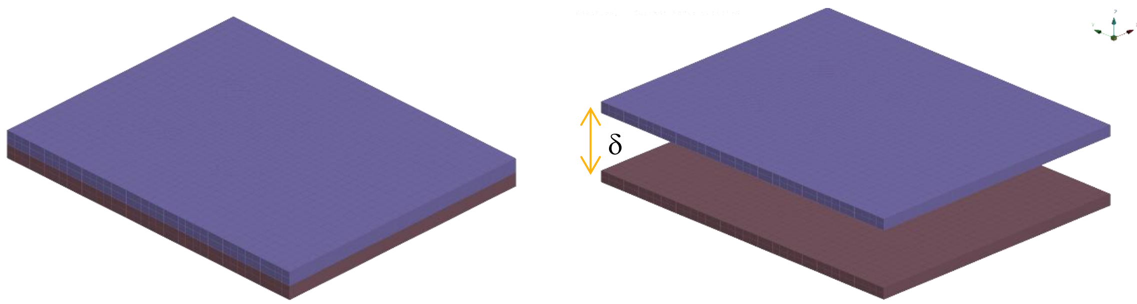


Figure 48. The left image shows two contact surfaces that are initially touching. For this case, using non-Mortar contacts in a linear analysis, the surfaces may be tied in the normal direction but free to move in the tangential direction. The right image shows two contact surfaces with an initial gap. For this case, no contact forces will be transferred in a linear analysis.

The Mortar sliding contacts do not transfer forces in eigenvalue analyses (unless pre-loading is introduced by a preceding non-linear case).

Sliding contacts will be appropriately linearized during intermittent eigenvalue analyses and intermittent linear buckling analyses, if activated by for example pre-loading, see also Section 4.6.3. This means that surfaces that come into contact during the pre-loading and have a contact pressure acting on them will, in the eigenvalue analysis appear as connected by stiff springs (from the contact stiffness) in the normal direction while being able to slide in the tangential direction (depending on the coefficient of friction). No new contacts can occur during the eigenvalue analysis, and the ones detected during the non-linear preload cannot be removed or unloaded.

Contact forces will be computed and output (see further Section 6.3) also in linear transient modal dynamic analyses but not applied to the structure. By this, the contact force can be used as an indicator of initially open contact gaps closing during the simulation.

In linear static analyses (`nsolvr` = ± 1 on `*CONTROL_IMPLICIT_SOLUTION`, see also Section 4.1) the use of sliding contacts is in general not recommended. Sliding contacts are non-linear interactions, incompatible with purely linear analysis. The contact forces are computed and applied to the structure, but since no equilibrium iterations are performed, the results will make little sense. Due to the "sticky" option, `IGAP` = 1 of non-Mortar contacts, see Section 6.1.5, they may appear as tied in linear analyses. If such behavior is desired, it is preferable to replace the non-Mortar contacts with proper tied contacts.

6.1.8 Contacts for thermal analyses

To include heat transfer by contacts and gap radiation in thermal or coupled thermal structural analysis, use `*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_THERMAL_ID`. The thermal contacts will also account for heat generated by friction. See Section 4.11.1 for an example of a coupled thermal mechanical analysis involving thermal contact. A template for using the automatic surface-to-surface Mortar contact (versions R10 and later) with thermal options follows:

```
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_THERMAL_ID
Define contact ID, Heading
Card 1: Define what shall be in contact using parts or part sets
Card 2: Define friction
Card 3: Define alternative penalty stiffness (SFSA)and thicknesses (SAST) for shells, o
leave blank to get defaults
Card 4: Define thermal contact properties (gap conductance, radiation factor etc.)
Card 5: Define contact thickness for solids (PENMAX)or leave blank
Card 6: optional / Define IGAP and IGNORE
```

In the following, the separation or gap in the contact is denoted as l_{gap}. The parameters that can be defined are

- *LMIN* specifies the contact gap (l_{gap}) for which the thermal contact is treated as closed. For l_{gap} > *LMAX* the thermal contact is not active.
- *H0* is the heat transfer conductance for a closed contact gap, used for contact gaps in the range $0 \leq l_{gap} \leq$ *LMIN*.
- *K*, which can be used to model a fluid between the contact surfaces. Then *K* is the thermal conductivity of the fluid, and the conductance calculated by LS-DYNA will be $h_{cond} = \mathrm{K} / l_{gap}$.
- *FRAD* is the radiation factor between the contact surfaces. This can be used to model heat transfer due to radiation of surfaces close to each other. Based on the reference[16]-side temperature T_m and the tracked-side temperature T_s, LS-DYNA will calculate a radiant heat transfer conductance as $\mathrm{h_{rad}} = \mathrm{f_{rad}}(\mathrm{T_m} + \mathrm{T_s})(\mathrm{T_m^2} + \mathrm{T_s^2})$.
- *FTOSLV* – describes how to distribute the sliding friction energy between the tracked and reference[16] surfaces.

In summary, the contact conductance *h* will be given by

$$h = \begin{cases} \mathrm{H0} & 0 \leq l_{gap} \leq \mathrm{LMIN} \\ h_{cond} + h_{rad} & \mathrm{LMIN} < l_{gap} \leq \mathrm{LMAX} \\ 0 & l_{gap} > \mathrm{LMAX} \end{cases}$$

See under the section `THERMAL` of the `*CONTACT` – keyword of Ref. [1] for more details. By setting the variable *BC_FLG* = 1, convection boundary conditions (`*BOUNDARY_CONVECTION`) applied to a surface which is also involved in a thermal contact will be de-activated for the segments where the thermal contact is closed.

By use of the option `_THERMAL_FRICTION`, temperature dependent friction coefficients can be defined, and the thermal contact parameters can be specified as temperature and contact pressure dependent.

6.1.9 Contacts for 2D analysis

The Mortar contact is also available for 2D analyses (NOTE! smp-LS-DYNA only, up to and including R12.0.0) as `*CONTACT_2D_AUTOMATIC_..._MORTAR_ID`. A template for using the Mortar surface-to-surface 2D contact follows:

[16] Previously denoted ”master”.

```
*CONTACT_2D_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_ID
Define contact ID, Heading
Card 1: Define what shall be in contact, and friction
Card 2: Specify activation/deactivation times, or leave blank
```

See Section 5.5.1 for an example using the 2D Mortar contacts.

For single surface contact, the `*CONTACT_2D_AUTOMATIC_SINGLE_SURFACE_MORTAR_ID` may in some situations become quite time-consuming. Should this occur, consider switching to a non-Mortar single surface 2D contact.

When use of a non-Mortar contact is required, it is recommended to set the closing-and-opening flag *COF* = 1 on Card 2 (similar to disabling the "sticky" option for *IGAP* of non-Mortar 3D contacts, compare Figure 47).

6.2 Tied contacts

The purpose of tied contact is to couple parts at a constant distance. In this section, a brief overview of the tied contact (`*CONTACT_TIED_...`[17]) options in LS-DYNA is presented. There are two main families of tied contacts:

- `*CONTACT_TIED_NODES_TO_SURFACE`, where only the translational degrees of freedom are tied, or
- `*CONTACT_TIED_SHELL_EDGE_TO_SURFACE`, where all available degrees of freedom are tied.

The family `*CONTACT_TIED_SHELL_EDGE_TO_SURFACE` should be used when structural elements (beams, shells ..., or nodes of these element types) are involved in a tied contact, if it is desired to involve all degrees of freedom, including rotations, in the tied contact. Using `*CONTACT_TIED_NODES_TO_SURFACE` when beam elements are involved may create rigid body modes (spinning beams or hinges in the model).

By adding the option `OFFSET` to these contacts (for example `*CONTACT_TIED_NODES_TO_SURFACE_OFFSET`), a penalty-based tie formulation is obtained. This is required when rigid nodes or segments (either a rigid part from `*MAT_RIGID` or a nodal constraint, for example `*CONSTRAINED_NODAL_RIGID_BODY`, or `*BOUNDARY_SPC_...`) are involved in a tie contact. Even without using the `OFFSET` option, it is possible to let LS-DYNA automatically create a backup penalty-based tied contact for the rigid nodes/segments by setting *IPBACK* > 0 on the optional Card E. By this, LS-DYNA will first try to use kinematic constraints to tie the SURFA nodes, and for those nodes who fail due to "conflicting constraints", a penalty-based constraint is created.

A tracked node is only tied to the reference segment if it is within a distance δ to the reference segment. The distance δ is computed as

---

[17] By setting *IACC* ≥ 1 on `*CONTROL_ACCURACY` (active in the attached include files), a strongly objective formulation for some of these contacts is activated, see §29.10 of Ref. [11] for further details.

$$\delta_1 = 0.60 * (thickness\_slave\_node + thickness\_master\_segment)$$

$$\delta_2 = 0.05 * min(master\_segment\_diagonals)$$

$$\delta = max(\delta_1, \delta_2)$$

It is possible for the user to explicitly specify the tie distance by entering a negative value for the *SAST* and *SBST* parameters on card 3. The tie distance will then be computed as $\delta = 0.6 * |SST + MST|$. It is strongly recommended to check the tied contacts using a graphical preprocessor, such as LS-PrePost, to assure that the nodes get tied as intended. It is also recommended to check in `messag` (`mes00*`) file(s) for warning messages for nodes that are not being tied.

If a tracked node is tied (within the distance δ as above), and the `OFFSET` option is not active, it will also be moved (but without causing any stresses or strains) to the reference segment (without considering shell thickness) automatically. This means that the mesh may be modified, which can lead to mesh distortion. One way to visualize these nodal movements is to create a vector plot of the displacements at t = 0. The nodal movements will also be printed in the `d3hsp` file. In case of severe mesh distortion due to nodes being moved by tied contact, a warning message will be issued:

```
*** Warning 41240 (SOL+1240)
     Volume of solid element # XXX drastically reduced
     in first cycle, this may be due to inappropriately
     applied tied contact constraints.
```

If the `OFFSET` option is active, the tracked nodes will not be moved. In general, tied contacts with only the `OFFSET` option will not account for the moments due to separation between tracked nodes and reference segments. This means that there may be a risk of creating an artificial resistance against (large) rotations if the separation is too big.

If a constraint-based contact that does not move the tracked nodes is desired, the option `CONSTRAINED_OFFSET` can be used. In addition, this option will also account for the moment created due to separation between tracked nodes and reference segments. An overview of the most common tied contact options is presented in Table 7.

An alternative to `*CONTACT_TIED_NODES_TO_SURFACE` is the `SURFACE_TO_SURFACE` type. For this latter type, segment sets may be specified for the tracked side, which is not allowed for the `NODES_TO_SURFACE` type. For the `NODES_TO_SURFACE` type, node sets may be specified for the tracked side, which is not allowed for the `SURFACE_TO_SURFACE` type.

For maximum control over which nodes that get tied, a tracked node set (*sstyp* = 4) should be specified. If 2nd order elements are involved in a tied contact, it is recommended to use a tracked node set (including mid-side nodes) and define the reference side by part ID or part set ID (*SURFBTYP* = 2 or 3).

Table 7. Overview of some tied contact options

*CONTACT_TIED_	Constraint formulation	DOFs	Move nodes[1]	Moment transferred[2]
NODES_TO_SURFACE	Kinematic	1 – 3	Yes	
NODES_TO_SURFACE_OFFSET	Penalty	1 – 3	No	No
NODES_TO_SURFACE_CONSTRAINED_OFFSET	Kinematic	1 - 3	No	Yes
SHELL_EDGE_TO_SURFACE	Kinematic	All[3]	Yes	
SHELL_EDGE_TO_SURFACE_OFFSET	Penalty	All[3]	No	No
SHELL_EDGE_TO_SURFACE_BEAM_OFFSET	Penalty	All[3]	No	Yes
SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET	Kinematic	All[3]	No	Yes

Notes: (1) Only tracked nodes within the search distance will be moved. (2) Due to separation between tracked nodes and reference segments. Only applies to the _OFFSET_ option. (3) All available degrees of freedom, depending on element type of the tracked node.

For the case of T-joint-like junctions, the *CONTACT_TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET may be too weak in bending, see Ref. [42], while the constrained offset option (*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET) transfers moments more correctly. This tied contact is in general recommended for implicit analyses, and can also handle situations involving conflicting constraints if the *IPBACK* option is activated.

An example of a tied contact definition follows:

```
*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET_ID
Define contact ID, Heading
Card 1: Define what shall be tied
Card 2: Blank
Card 3: Define alternative tie distance via SAST, SBST < 0 (optional, can be blank line
Card 4- D: Optional / blank
Card E: Optional / recommended / set IPBACK > 0 (position 3)
```

There is also an option to change mortar contacts to penalty-based tied contacts, *CONTACT_..._MORTAR_TIED. This can for example be useful when debugging convergence problems by switching from sliding contact to tied contact.

Thermal properties can be specified for tied contacts, in the same way as for a sliding contact, see Section 6.1.8, using the keywords *CONTACT_TIED_..._THERMAL_ID.

A brief overview of tied contacts in LS-DYNA, focusing on explicit simulations, is presented in the Webinar "Tied and tiebreak contacts in LS-DYNA" [35].

6.2.1 Making contact surfaces stick together

In some situations, it is desired to have two contact surfaces stick to each other as soon as they come into contact: initially the contact surfaces are separated and can move individually, but as soon as they touch, they will be "glued" together, as in a tied contact. This may in some sense be seen as having "infinite" friction (compare *FRICTION, ROUGH in Abaqus). In LS-DYNA, this can be achieved by using the contact option *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK_ID with *OPTION* = 1 on

Card 4. It is also recommended to set *IGAP* = 2 in order to avoid force transfer before the contact is established. An example of a tiebreak contact definition follows:

```
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK_ID
Define contact ID, Heading
Card 1: Define what shall be tied
Card 2: Define friction (optional)
Card 3: Blank
Card 4: Set OPTION = 1
Card 5 - 6: Blank / optional
Card 7: Set IGAP = 2
```

The tie constraint is enforced via a penalty formulation. This means that the above option can be used also for rigid bodies.

An example of `*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK_ID` is supplied as `sticking_contact.key`. A rubber tube is compressed between two rigid plates. When the plates are brought back to their original position, the rubber tube remains attached to the plates, see Figure 49.

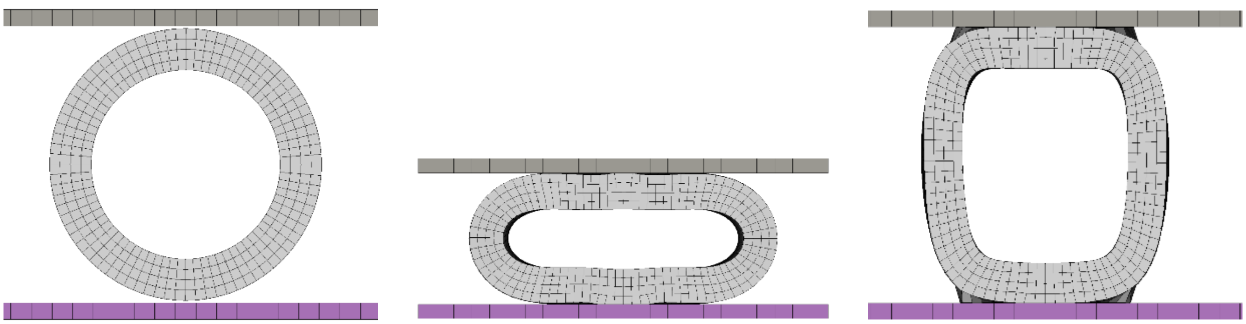


Figure 49. Example of a rubber tube being compressed by two rigid plates using tiebreak contact with option = 1.

6.3 Contact output

Contact forces are output in the `rcforc` file. This is a tabular result, which can be used for 2D postprocessing. Contact forces between certain parts of a surface-to-surface or single-surface contact may be extracted by use of force transducers, `*CONTACT_FORCE_TRANSDUCER_PENALTY_ID`. For Mortar contacts, the force transducers must be defined based on parts or part sets.

It is also possible to obtain output of contact gaps and contact pressure for 3D visualization in the `intfor` file. Then, the following steps must be followed:

- On card 1 of the contact definition, set the parameters `SPR` and `MPR` to one.
- Also, the keywords `*DATABASE_BINARY_INTFOR_FILE` (available from R10.2) and `*DATABASE_EXTENT_INTFOR` are required to specify the file name, output frequency and content of the 3D interface force file. These are already defined in the include files `database_cards_static.key` and `database_cards_dynamic.key`.
- For analyses in R10.1 or prior, the include file `database_cards_before_R102.key` should be used. It is then required to add `s=intfor` on the command line for submitting the LS-DYNA analysis.

Constraint-based tied contacts will not output contact pressures into the interface force file (intfor), while penalty-based tied contacts will.

The Mortar contacts will output contact diagnostics in the message file (`messag` or `mes00*`):

- A report of initial penetrations during initialization, and
- at the beginning of each implicit time step, a report of the number of segments in contact, the maximum penetration (absolute and relative) and warnings if there are nodes that are close to being released by the contact.

This can be very useful information for debugging convergence problems related to contacts. From R12 of LS-DYNA, penetration information is also output to the `d3plot` and `intfor` files for fringe plotting, and to the `sleout` file (binout-format only) for curve (history) plotting.

7 Material models

This Section contains only a few general recommendations regarding material models for implicit analyses in LS-DYNA. For a detailed description of the material models available in LS-DYNA, see Ref. [2]. Presently, not all material models are supported for implicit analysis. If unsupported materials are used, LS-DYNA will print an error message in the `d3hsp` file and terminate.

The material model `*MAT_PIECEWISE_LINEAR_PLASTICITY` (or `MAT_24`) is commonly used in explicit LS-DYNA analyses to characterize metallic materials with plasticity, and it performs well also for implicit analyses[18].

An alternative to `*MAT_24`, if anisotropy or kinematic hardening is of interest, may be `*MAT_103`. A template for basic use of `MAT_103`, with isotropic hardening and no strain rate effects, follows:

```
*MAT_ANISOTROPIC_VISCOPLASTIC_TITLE
Define material title
MID, RO, E, PR, 0., 2, LCSS, 1.
Blank line × 5
```

The parameter *`LCSS`* is the load curve ID of the hardening curve. After the line defining material ID, density etc., five blank rows must follow. For mixed or kinematic hardening with a non-linear hardening curve, `*MAT_`225 could be of interest. The `*MAT_DAMAGE_3` (MAT_153) is a Chaboche-type material model for nonlinear mixed or kinematic hardening, including a possibility to specify temperature dependent material parameters.

If a material model with linear isotropic hardening using a tangent modulus is required, it is recommended to use `*MAT_24`. Then, only specify the yield stress (*`SIGY`*) and the tangent modulus (*`ETAN`*) on card 1 (and of course also the elastic parameters) and leave the other cards blank. For this case, it is not recommended to use `*MAT_3` (`*MAT_PLASTIC_KINEMATIC`) with the parameter *`BETA`* = 1.

For linear implicit analyses, the material response is linearized, but stresses will be computed from the strains obtained via the displacement solution to the linear problem. Since no equilibrium iterations are performed in a linear analysis, the obtained stresses can be nonsensical. If consistent estimates of

[18] From R9, if *`IACC`* = 1 on `*CONTROL_ACCURACY` (this is active in the attached include files).

stresses are of interest in a linear analysis, it is recommended to use *MAT_ELASTIC, which is an elastic material model[19].

It is however not recommended to use *MAT_ELASTIC to model rubber-like behavior in a non-linear analysis, since it is a hypoelastic material model, which may cause instabilities at large strains. For modelling of rubber materials, it is instead recommended to use *MAT_HYPERELASTIC_RUBBER (MAT_77). An alternative material model for rubber materials is *MAT_SIMPLIFIED_RUBBER_FOAM (MAT_181), which allows for direct input of results from materials testing. The material model *MAT_MOONEY-RIVLIN_RUBBER (MAT_27) is in general not recommended for modelling of rubber or rubber-like materials, see Appendix A for further details. If an elastic response is desired, *MAT_ORTHOTROPIC_ELASTIC (MAT_2) can be used, since it is a hyperelastic material model which works also for large elastic deformations.

For modelling of creep effects, LS-DYNA offers many possibilities [37]; for example, *MAT_THERMO_ELASTO_VISCOPLASTIC_CREEP (MAT_188) may be useful.

For modelling fabrics, *MAT_FABRIC can be used. It is recommended to use the option *FORM* = -14. By this, an elastic coating (thickness *TCOAT* and Young's modulus *ECOAT*) can be applied to the fabric, which can improve convergence of implicit analyses. For the mesh of the fabric part, shell elform 9 is recommended.

For the foam materials *MAT_LOW_DENSITY_FOAM and *MAT_FU_CHANG_FOAM, it can be beneficial for convergence to set the variable *HU* = 0 (possibly also the variable *SHAPE* = 0). For these materials, it is also recommended to check that the Youngs modulus (variable *E*) is consistent with the initial slope of the stress-strain curve.

7.1 Rate effects in implicit analyses

Many elastic-plastic material models in LS-DYNA allow for modelling the influence of strain rate on the hardening, either via a tabular definition of hardening curves, or via for example Cowper-Symonds relation. This is normally a desired feature in explicit analyses, where the strain rate effect on the hardening may be of importance, and the small time-step size allows for good resolution of the strain rates.

In an implicit static analysis, however, time can be a parameter that does not correspond to physical time, or the time-step size may be too large to adequately resolve the strain rate. For such cases, it is recommended that strain rate effects (on the yield limit) be disabled. This can be achieved on a global level, for all constitutive models in a simulation, by setting *IRATE* = 1 on *CONTROL_IMPLICIT_DYNAMICS. This flag will have effect, even if the dynamic analysis option is inactive (*IMASS* = 0). By *IRATE* = 2, strain rate effects are disabled in both implicit and explicit analyses.

[19] From R15 of LS-DYNA, material models are automatically linearized in linear analyses.

From R11 of LS-DYNA, strain rate effects are disabled by default for implicit static analyses but can be activated by *IRATE* = -1. If viscoelastic or creep effects are of interest, using for example `*MAT_BERGSTROM_BOYCE_RUBBER`, it is recommended to use *IRATE* = -1.

Note that setting *IRATE* ≥ 1 in implicit dynamics will cancel the effect of stiffness damping specified by `*DAMPING_PART_STIFFNESS` (see also Section 4.9.1).

7.2 Thermal materials and thermomechanical materials

The thermal material properties (specific heat, thermal conductivity etc.) are defined separately from the mechanical material properties, by setting *TMID* on the `*PART` card. The thermal material models in LS-DYNA all begin with `*MAT_THERMAL_...`, for example, use `*MAT_THERMAL_ISOTROPIC` to define isotropic, temperature-independent thermal properties.

For coupled analyses, it is often of interest to use temperature-dependent material properties as well. A quite general thermomechanical material model is then `*MAT_ELASTIC_VISCOPLASTIC_THERMAL` (MAT_106), which allows for temperature- and rate-dependent plasticity using tables or analytical expressions, while `*MAT_ELASTIC_PLASTIC_THERMAL` (MAT_4) is simpler, allowing for linear hardening specified at 8 temperatures.

8 Loads and boundary conditions

In this Section, some aspects regarding loadings and boundary conditions for implicit analyses are briefly discussed.

Fixed nodal constraints in LS-DYNA are in general applied by the keyword `*BOUNDARY_SPC_{OPTION}`. Prescribed non-zero boundary conditions (displacements, accelerations of velocities) are applied by the keyword `*BOUNDARY_PRESCRIBED_MOTION_{OPTION}`.

In case a (fixed) boundary condition is to be active only during part of an analysis, it is recommended to use `*BOUNDARY_PRESCRIBED_MOTION_{OPTION}`, since it offers more flexibility than `*BOUNDARY_SPC_{OPTION}_BIRTH_DEATH` which only can be applied once to a given node. Note that it is required to define a zero-valued curve in order to apply a fixed boundary condition by `*BOUNDARY_PRESCRIBED_MOTION_{OPTION}`. As an example, the keywords required to keep the NSID 112 constrained between t = 0 and t = 0.5 follow:

```
*DEFINE_CURVE_TITLE
Zero curve
      1001
0.,0.
1.,0.
*BOUNDARY_PRESCRIBED_MOTION_SET
$#    nsid       dof       vad      lcid        sf       vid     death     birth
       112         1         2      1001       1.0         0       0.5
       112         2         2      1001       1.0         0       0.5
       112         3         2      1001       1.0         0       0.5
```

In order to constrain rigid parts (a `*PART` defined as rigid by use of `*MAT_RIGID`, or a `*CONSTRAINED_NODAL_RIGID_BODY`) it is recommended to use `*BOUNDARY_PRESCRIBED_MOTION_RIGID` and a zero-valued curve. This will then constrain the center of gravity of the rigid part. In case the motion is desired around a specific point which does not coincide with the inherent center of gravity of the rigid part, the `_INERTIA` option can be used to specify a coordinate or a node ID for the desired center of gravity. It is in general (for LS-DYNA versions up to and including R11.1) not recommended to use `*BOUNDARY_SPC` to constrain nodes of rigid parts, since these constraints then may be ignored by LS-DYNA. In that case, a warning message is printed in the `d3hsp` file:

```
 *** Warning 60257 (IMP+257) (processor # 0)
     skipping spc on rigid body node X
       tcode = Y rcode = Z
```

It may be tempting to make use of the possibility to constrain rigid parts defined by `*MAT_RIGID` by using the *CMO* ≠ 0 and *CON1*, *CON2* options on the `*MAT_RIGID` card, and normally this will enforce the desired constraints to the rigid part. But up to version R11 of LS-DYNA, the reaction forces due to constraints specified on the `*MAT_RIGID` card cannot be obtained for post-processing. In case the corresponding reaction forces are of interest, it is recommended to use `*BOUNDARY_PRESCRIBED_MOTION_RIGID` and a zero-valued curve.

Loads in LS-DYNA are in general specified by the `*LOAD_{OPTION}` keywords. Bolt pre-load is applied using either `*INITIAL_STRESS_SECTION` (see the example of Section 4.3.2) for bolts modelled by solid elements, or `*INITIAL_AXIAL_FORCE_BEAM` for bolts modelled by beam elements. It is recommended to apply loadings gradually, for example by linear ramps. An example follows, where a distributed pressure of 10 MPa is applied to segment set ID 100:

```
*LOAD_SEGMENT_SET
$#    ssid      lcid        sf        at
       100       100       10.       0.0
*DEFINE_CURVE_TITLE
Loading ramp
      1001
0.,0.
1.,1.
2.,1.
```

Loading a structure via prescribed displacement is often beneficial for convergence since it reduces the possibilities for instabilities or rigid body motions. Forces can be evaluated from contacts (`rcforc`) or boundary conditions (`spcforc`) or prescribed motions (`bndout`). In implicit statics, convergence for the force-based loading will fail if the structure has a non-monotonic (global) force-displacement response, see Figure 50. By using a prescribed motion of the loading device, it is possible to follow the deformation even after the peak force has been reached. In case the ultimate capacity and post-peak behavior is sought using force control, the arc length method can be applied, see Section 4.5.

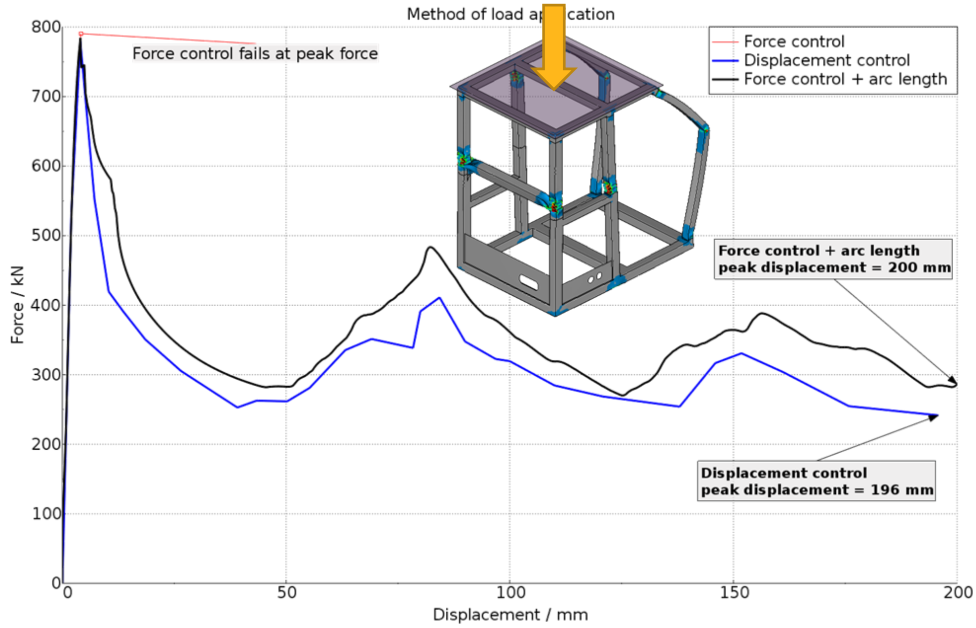


Figure 50. Example of different load application approaches. A generic frame structure is compressed by a rigid loading device.

9 Other implicit analysis types

There are many additional implicit analysis types available in LS-DYNA which are currently not described in the present document.

Many of the multi-physics solvers in LS-DYNA [3], for example the ICFD-solver and the EM-solver, use implicit solution schemes. These solvers are not treated in this document.

10 Modifications of control card settings

It is hard to define LS-DYNA control card settings that are universally valid for all possible analysis types. The provided include files supply settings which work in most situations, but sometimes modifications are necessary. A few recommendations follow:

- For highly non-linear problems, the default solution scheme, using BFGS (low rank) updates to the stiffness matrix during the iterations, may be inefficient. For these cases, a full Newton solution scheme can be activated by setting *ILIMIT* = 1 and *MAXREF* to for example 55 on `*CONTROL_IMPLICIT_SOLUTION`. In this case, the settings of *ITEOPT* and *ITEWIN* on `*CONTROL_IMPLICIT_AUTO` may also require modifications.
- The inclusion of geometric stiffness (by setting *IGS* = 1 on `*CONTROL_IMPLICIT_GENERAL`) can in some cases be beneficial for convergence, for example in simulations of bolt pre-tensioning or tensile testing.
- The choice of line search method 5 on `*CONTROL_IMPLICIT_SOLUTION` may sometime be overly conservative. It may then help to use the default line search method by leaving the *LSMTD* field blank.
- In some rare cases, the provided convergence settings may be too strict. Indications of this may for example be repeated occurrence of RETRIEs and decreasing time increment, and eventually the solution may fail to converge. One remedy in cases with convergence problems may possibly be to increase *DCTOL* to, for example, 0.01 or change to *DNORM* = 2 on `*CONTROL_IMPLICIT_SOLUTION` in order to troubleshoot the convergence problems. Before

making any changes to convergence tolerances of *CONTROL_IMPLICIT_SOLUTION, it is strongly recommended to go through the trouble shooting guide of Appendix C.

- From R8, a non-symmetric equation solver is available. In some cases, activating this (set *LCPACK* = 3 on *CONTROL_IMPLICIT_SOLVER) may aid convergence, for example in cases involving contacts with high coefficients of friction. For brake squeal analyses, this option is mandatory.
- If many separate load cases are to be solved in a linear static analysis, it is possible to reuse the factorization of the stiffness matrix by setting *NSOLVR* = -1 on *CONTROL_IMPLICIT_SOLUTION. See Section 4.1.

For a more extensive troubleshooting guide, see Appendix C and Ref. [24]. Some further comments on the most important control cards regarding implicit analyses are given in Appendix G.

/ Appendix A

11 Rubber modeling for implicit analysis

11.1 Background

This appendix contains a description of recommended settings when analyzing rubber structures in implicit LS-DYNA.

In some cases, the default or general recommended settings need some modifications for simulation of rubber materials. The purpose of this document is to serve as a guideline for the user in these special cases. The features that are described in this document are included in LS-DYNA version R7.1.1 and later.

11.2 Material models

There are many material models available for analyzing rubber structures in implicit LS-DYNA, for example

- `*MAT_HYPERELASTIC_RUBBER (*MAT_077)`
- `*MAT_SIMPLIFIED_RUBBER/FOAM (*MAT_181)`
- `*MAT_MOONEY_RIVLIN (*MAT_027)`

Our recommendation is to use `*MAT_077` as the model of choice for rubber materials. If the user experiences problems with the curve fitting, try `*MAT_181`. The Mooney-Rivlin model is not recommended, but for relatively simple cases it may work, and is therefore included in this document.

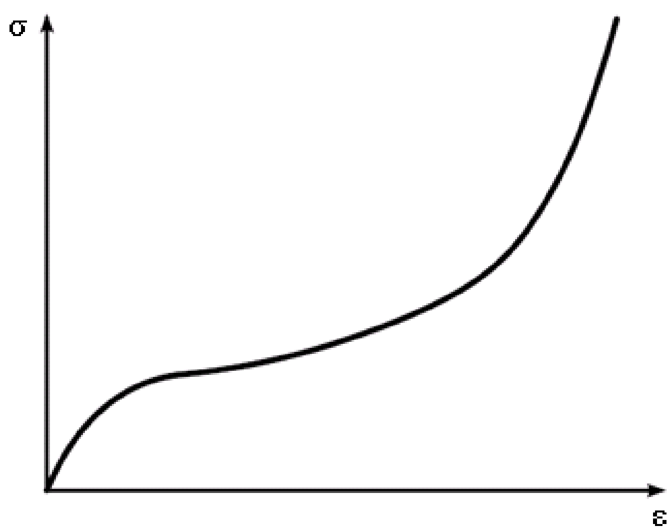


Figure 51. General rubber response curve.

11.2.1 *MAT_HYPERELASTIC_RUBBER

This is the recommended material model to use when modelling rubber structures for implicit simulations in LS-DYNA. In this model, the user may specify up to six terms to directly describe the material behavior, see Figure 52. If only C10 and C01 is defined the model is equal to a Mooney-Rivlin rubber model. The difference between *MAT_027 and *MAT_077 is in that case only the stability of the material model.

The user may also input data from tests to fit the parameters (C_{nn}) to the test curve. This is available when n>0, see Figure 53. The user then specifies sgl, sw, st and lcid according to Ref. [2]. For this option, LS-DYNA will print the fitted parameter values in the d3hsp file, search for “Fit to General Hyperelastic material“. For comparison using the XY Plot tool in LS-PrePost, the file curveplot_1 is created.

```
*MAT_HYPERELASTIC_RUBBER_TITLE
MAT_077_N=0
$#     mid        ro        pr         n        nv         g      sigf
         8 1.2000E-9  0.499500         0         0     0.000     0.000
$#     c10       c01       c11       c20       c02       c30
  0.550800  0.137700     0.000     0.000     0.000     0.000
```

Figure 52. Hyperelastic rubber, n=0.

```
*MAT_HYPERELASTIC_RUBBER_TITLE
MAT_077_N>0
$#     mid        ro        pr         n        nv         g      sigf
       100 1.2000E-9  0.499500         1         0     0.000     0.000
$#     sgl        sw        st     lcid1      data     lcid2    bstart     tramp
  1.000000  1.000000  1.000000      1234     0.000         0     0.000     0.000
```

Figure 53. Hyperelastic rubber, n>0.

The strain energy functional is defined as:

$$\mathrm{W}(\mathrm{J}_1, J_2, J) = \sum_{p,q=0}^{n} C_{pq}(J_1 - 3)^p (J_2 - 3)^q + W_H(J)$$

Where :

$$J_1 = I_1 I_3^{-\frac{1}{3}}$$

$$J_2 = I_2 I_3^{-\frac{2}{3}}$$

$$\mathrm{I}_1, \mathrm{I}_2, \mathrm{I}_3 = \text{Invariants of right Cauchy - Green Tensor C}$$

11.2.2 *MAT_SIMPLIFIED_RUBBER/FOAM

This material model may also be used for modeling rubber behavior. The user does not specify any rubber parameters such as a, b or C_{nn}. Instead, the bulk-modulus and test data are provided, see Figure 54. The model is developed for incompressible polymers and is a good alternative if, for example, the Poisson’s number is less than 0.495. Note that the tensile and compressive part of the curve are dependent on one another, thus correct and complete test data are crucial to obtain useful results, see Figure 55 for an example.

```
*MAT_SIMPLIFIED_RUBBER/FOAM_TITLE
TPU
$#      mid        ro         k        mu         g      sigf       ref     prten
         11 1.1700E-9  1000.000     0.000     0.000     0.000     0.000     0.000
$#      sgl        sw        st   lc/tbid   tension     rtype    avgopt   pr/beta
   1.000000  1.000000  1.000000      1154 -1.000000     0.000     0.000  0.000000
```

Figure 54. Example of simplified rubber/foam material card.

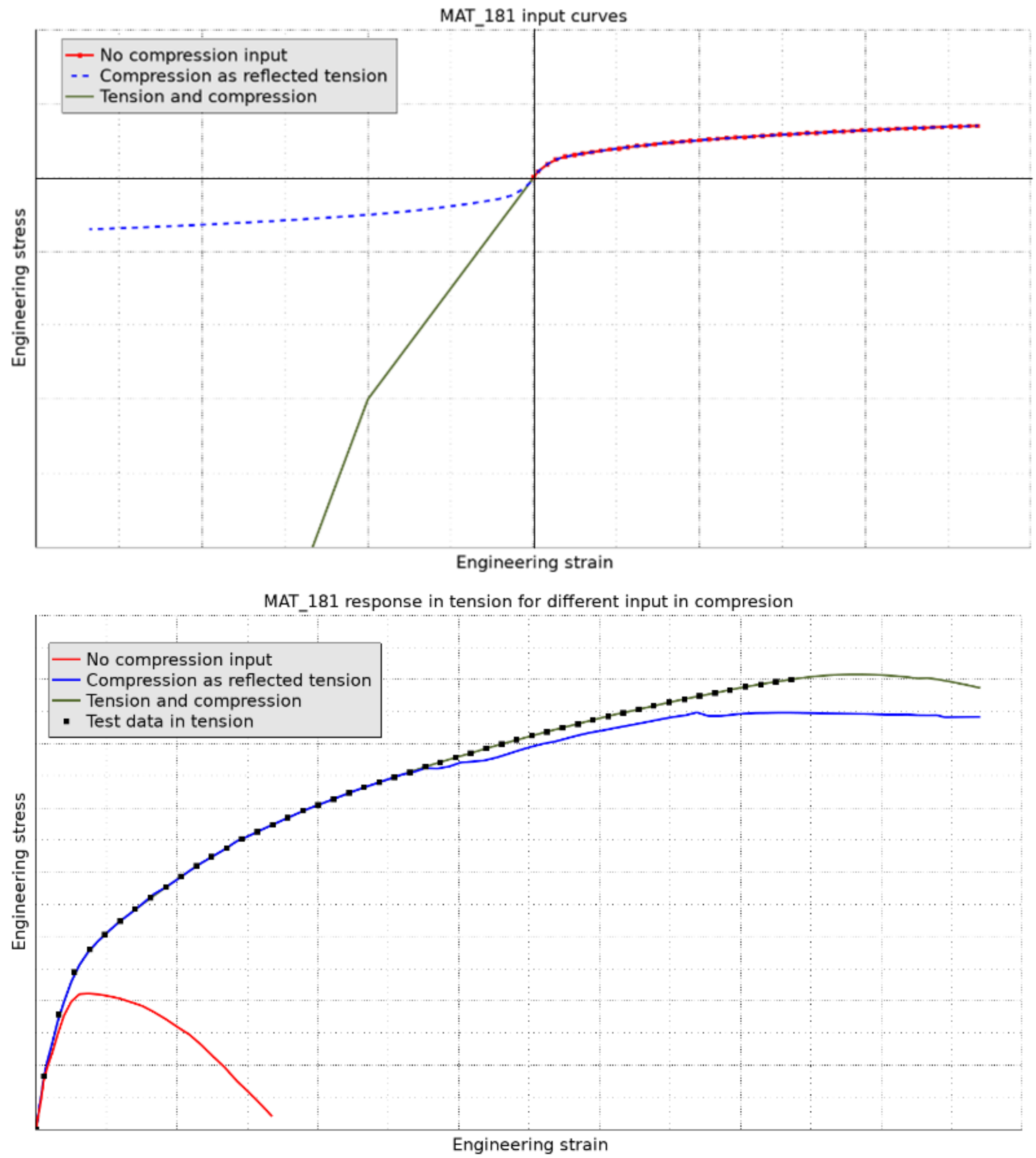


Figure 55. The top image shows three different input curves, engineering stress vs. engineering strain. The bottom image shows how the input of compression data affects the result in tension for *MAT_SIMPLIFIED_RUBBER_FOAM.

11.2.3 *MAT_MOONEY-RIVLIN_RUBBER

Even though this material model may work for simple cases, it is in general not recommended for modelling of rubber materials. The model is popular due to its simplicity but may in some cases experience instability and/or other convergence issues, especially for large deformations. The Mooney-Rivlin model is a two-parameter rubber model, see Figure 56. Since only two parameters (a and b) are used to directly describe the material response, the response curve must be very “rubber like”, see

Figure 51. If fillers are used or if the material response is too unlike a general rubber response curve for some other reason, this model may not be the best choice. There is also an option for fitting the parameters a and b to a test curve by using sgl, sw, st and lcid. Mooney-Rivlin rubber can be modelled by using `*MAT_077`. In such case, only C10 and C01 are defined.

```
*MAT_MOONEY-RIVLIN_RUBBER_TITLE
Mooney-rivlin_rubber
$#     mid        ro        pr         a         b       ref
       101 1.2000E-9  0.499500     0.500    0.1377     0.000
$#     sgl        sw        st      lcid
     0.000     0.000     0.000         0
```

Figure 56. Mooney-Rivlin rubber.

The strain energy function is defined as:

$$W = A(I-3) + B(II-3) + C(III^{-2}-1) + D(III-1)^2$$

Where :

$$C = 0.5A + B$$

$$D = \frac{A(5\nu - 2) + B(11\nu - 5)}{2(1 - 2\nu)}$$

ν = Poisson's ratio

$2(A+B)$ = Shear modulus of linear elasticity

I, II, III = Invariants of right Cauchy-Green Tensor C

From this, it follows that if ν = 0.5, the expression for D becomes singular. Instead, use for example ν = 0.495.

11.3 Elements

11.3.1 Continuum elements

The recommended element types for rubber modeling are:

Hexahedral: -1, -2

Tetrahedral: 13

The -1 and -2 hexahedral formulations are especially useful for elements with poor aspect ratios as may become the case for large deformations of rubber. Elform -1 seems to be more efficient than elform -2 in most rubber cases.

Elform 13 was developed to avoid volumetric locking by applying nodal pressure averaging. Our experience is that elform 13 is a robust element formulation that in general works well for analyses involving rubber.

If warnings/errors due to negative volumes occur, try a mesh with larger element size.

In recent development of element technologies, the Cosserat point element [21] formulation (CPE) has been implemented for underintegrated 1st order hexa and 2nd order tets, see also Section 5.3. It is a formulation originally developed for handling large deformation of hyperelastic materials and should therefore be an interesting alternative for analyses involving rubber. The Cosserat point formulation is activated by combining hourglass type 10 (*IHQ* = 10 on *HOURGLASS) with solid elform 1 (8-noded hexa) or 16 (10 node tet). It is available from R10 of LS-DYNA. Note that the CPE implementation is similar to an underintegrated element, which means that hourglass energy can arise. The amount of hourglass energy should be checked to ensure that the simulation results are valid.

From R15 of LS-DYNA, displacement-pressure elements are available for fully incompressible hyperelastic materials [46]. These are activated by setting Poisson's ratio to exactly 0.5 of *MAT_HYPERELASTIC_RUBBER or *MAT_MOONEY-RIVLIN_RUBBER and using solid element formulation 2 (hexahedral), 10 (4-noded tetrahedral) or 16 (10-noded tetrahedral).

11.3.2 Element free methods

If the mesh becomes very distorted it may be hard to get a converged solution using structural solid elements. In that case, EFG may be used. EFG, or Element Free Galerkin, is a meshless method where only the nodes of the mesh are used. Consequently, badly shaped or even inverted elements will not be an issue when using EFG. The drawbacks of using this method are that it may be computationally expensive and that more work may be needed in order to get contact regions to converge.

In current versions of LS-DYNA (up to and including R12.0.0) only the element formulation 42 on *SECTION_SOLID_EFG can be used with *MAT_HYPERELASTIC_RUBBER. For this EFG formulation, it is also recommended to activate pressure smoothing, by setting *IPS* = 1 on Card 3. A template follows:

```
*SECTION_SOLID_EFG_TITLE
RUBBER EFG TET
$#    secid     elform        aet
          1         42          0
$#       dx         dy         dz    ispline      idila       iebt       idim     toldef
        1.1        1.1        1.1          0          0          3          2       0.01
$#      ips      stime       iken         sf       cmid        ibr         ds       ecut
          1
```

11.4 Contacts

We always recommend the MORTAR contact for implicit analysis. Also, set *MINFO* = 1 on *CONTROL_OUTPUT to check the contact status. This option is active in the database cards of the include files provided with this package. See Section 6.1 for settings and recommendations regarding MORTAR contact.

In some cases, it might serve as an additional stabilization to add an extra Mortar single surface contact for the rubber part(s), see for example Ref. [24].

11.5 Solver settings

The solver setup is (almost) identical to what is recommended in control_cards_nonlin.key, but with a few tweaks.

The recommended line search method (*LSMTD*) to be used is 5 or 6. In control_cards_nonlin.key, line search method 5 is active.

- *LSMTD* = 5 will, as usual, minimize the energy, but, in addition to this it will also apply constraints on the change of the residual force in each iteration. This means that the residual force cannot change too much between two iterations. Of course, this governs a more convergence friendly behavior, but it is a more time-consuming method compared to the default method (*LSMTD*=4).

- *LSMTD* = 6. Same as 5 but minimizes the residual norm whenever convenient. In some cases, the constraint on the residual in *LSMTD*=5 can result in a too small change between two iterations. If so, the message “Line search step size zero” will appear and the solution will not converge. *LSMTD*=6 will prevent this from happening.

11.6 Examples

The following example problems are included in this document:

- Insert_and_interference.key

Press fit between a rubber and a (rigid) steel part is resolved by using *IGNORE*=4, *MPAR1* and *MPAR2* on the MORTAR contact card followed by a large deformation and sliding contact load case.

- Large_deformation.key

General rubber deformation analysis.

A rubber plate is heavily deformed between two rigid, corrugated panels.

- TPU.key, compare.png

Test of TPU using `*MAT_SIMPLIFIED_RUBBER/FOAM`

and compare to physical test.

Appendix B

12 Restart of analyses

In many situations, it is required to perform an initial pre-loading stage (for example bolts pre-tensioning) followed by a set of independent (static) load cases of the pre-loaded structure. It may then be desirable to perform the pre-loading as a separate analysis, and the consecutive load cases as restart analyses, starting from the pre-loading analysis but changing or adding loads or boundary conditions, and prolonging the termination time.

In LS-DYNA, there are two main possibilities for restarting analyses, or continuation based on a previously defined state. The approach using `d3dump` or `d3full` files (traditionally called “restart files” , see the Appendix “Restart input data” of Ref. [1]) is described in Section 12.1. If this approach is to be used, the keyword `*CONTROL_MPP_IO_NODUMP` must not be present in the keywordfiles. The approach using binary-format `dynain.lsda` files (traditionally called “dynain files”) is described in Section 12.2.

See also Ref. [30] for a review of different restart options in LS-DYNA.

12.1 Restart using d3dump or d3full files

In the following, only the “small restart” and “full restart” options will be discussed.

By default, LS-DYNA will write restart files (`d3dump` and `d3full`) after a completed analysis (and after completed dynamic relaxation). It is possible to get restart files also during a simulation using `*DATABASE_BINARY_D3DUMP`, but for this database keyword the number of cycles between printouts is specified. For example, to get a `d3dump`/`d3full` file after each converged step, specify *CYCL* = 1. Restart files can also be requested during a simulation using sense switches: simply put the text “sw3” in a file called `d3kil` in the folder where the simulation is running, and LS-DYNA will write `d3dump` and `d3full` files for the current configuration. It is recommended to output the `d3dump` and `d3full` restart files in double precision[20]. Note that when `d3dump` or `d3full`files are used for restart of analyses, the same LS-DYNA version should be used for both the initial and the following analysis.

12.1.1 Small restart

A “small restart” means that only a limited number of changes can be made to the model in the restart run. It is possible to modify curve definitions, remove contacts and delete elements and parts, but it is not possible to add contacts or change boundary conditions (but constraints can be added). The advantage is that normally, a “small restart” has a more robust functionality than a “full restart”. Note that the same number of cores must be specified both for the first run and the restart run.

In practice, a “small restart” can be used if the load cases following the pre-loading are defined by prescribed forces (`*LOAD_` ...). Then, all load cases can be prepared already in the pre-loading stage, and the loadings in the consecutive runs can be controlled by modifying the corresponding load curves using `*CHANGE_CURVE_DEFINITION`. For example, if two load cases are to be studied, the corresponding

[20] Avoid setting *IBINARY* = 1 on the `*DATABASE_FORMAT` keyword.

load curves are defined in the keyword file for the pre-loading, but with zero magnitudes. Then, in the following "small restart" analyses, the curves are redefined by use of `*CHANGE_CURVE_DEFINITION` and `*DEFINE_CURVE_TITLE`. Note that the number of points in the curve cannot change in a small restart. Note also that if offset of the abscissa is used in the first keyword file (`OFFA` of `*DEFINE_CURVE_TITLE`) this will in practice create extra points in the load curve. If the number of points in a curve that is changed in a "small restart" is incorrect, LS-DYNA will terminate (Error termination). It is also important to remember that the simulation time continues from the first simulation. For example, if bolt pre-tensioning is applied in the first simulation from T=0 to T=1, the following restart will start at T=1. In a "small restart", LS-DYNA will continue writing to existing `ASCII/binout` files, which means that it is important that these files are present in the running directory, otherwise output to these files may be lost. The generated plot states will be numbered in the same sequence as the first analysis. To exemplify: if the last plot state of the pre-loading analysis was `d3plot05`, the first plot state of a "small restart" will be `d3plot06`.

A "small restart" is activated by setting `r=d3dump01` as a command line argument to LS-DYNA.

A template for a "small restart" follows:

```
*KEYWORD
*CONTROL_TERMINATION
Define end time of the simulation
*CHANGE_CURVE_DEFINITION
Specify curve ID to modify
*DEFINE_CURVE_ID
Specify re-definition of load curves
*TITLE
Simulation title
*END
```

12.1.2 Small restart example: Bolt pre-tensioning followed by prescribed loading

In the first stage, bolt pre-tensioning is applied to a bolted flange joint, see Figure 57. Non-linear material properties are used both in the pipes and the bolts. The example keyword file is `pretens001.key`. This keyword file also includes the definitions of the applied loadings.

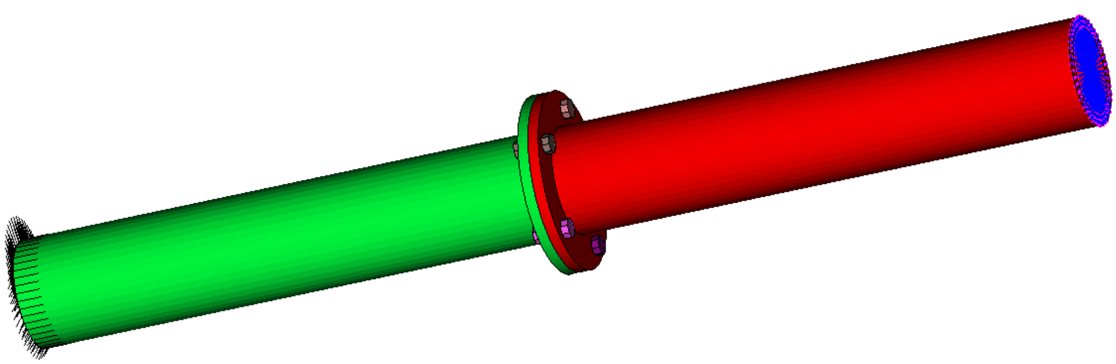


Figure 57. Pre-tensioning of a bolted flange joint. At one end, the pipe (green) is fully constrained (black lines), while at the other end, a prescribed loading is to be applied (at the constrained nodal rigid body, shown with blue lines).

In the following analysis (`small_restart.key`), loading is applied in the Y-direction, by redefinition of the load curves using the keyword `*CHANGE_CURVE_DEFINITION`. The result is shown in Figure 58. For this example, R11.2, R13.1 or R14.1 of LS-DYNA is recommended.

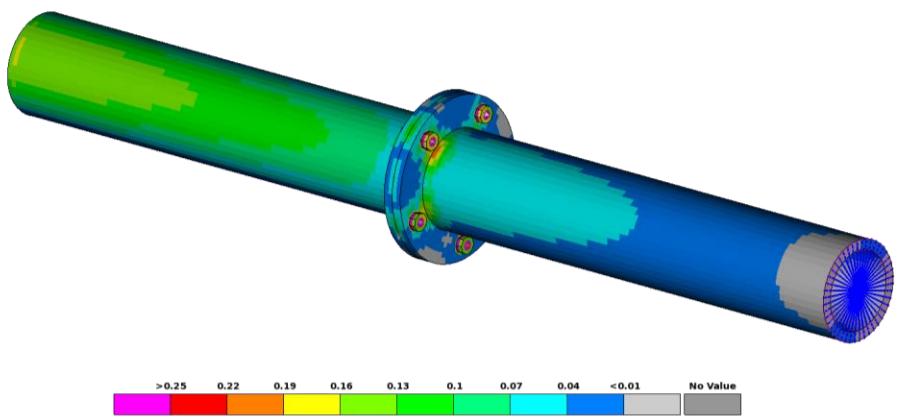


Figure 58. Effective stress due to applied transverse loading of 10 kN.

12.1.3 Full restart

In a "full restart", a complete redefinition of the simulation model is possible, including loads, boundary conditions, contacts etc. (From R14, it may even be possible to switch from implicit to explicit analysis, see Section 16.3) It is of course the responsibility of the user to ensure that the modifications make sense with respect to the previous analysis. In the keyword file of the restart analysis, the keyword `*STRESS_INITIALIZATION` must be present. The restart keyword file must also contain a complete model definition, in a similar way as in the initial analysis. In the same way as in a "small restart", the analysis time will continue from the previous analysis. For example, if bolt pre-tensioning is applied in the first simulation from T=0 to T=1, the following "full restart" will start at T=1.

A "full restart" is based on the `d3full` file from the previous run in mpp LS-DYNA and is activated by setting `n=d3full01` as a command line argument. The generated plot states will be numbered in a new sequence, by default starting at `d3plotaa`.

A template for a "full restart" follows:

```
*KEYWORD
*STRESS_INITIALIZATION
*INCLUDE
control_cards_nonlin.key
*DEFINE_CURVE_TITLE
Implicit time incrementation
700,
0., dt0 (first timestep)
Additional lines to define time incrementation
*INCLUDE
database_cards_static.key
*CONTROL_TERMINATION
Define end time of the simulation
*INCLUDE
Include file defining geometry, materials etc.
*LOAD_...
Define nodal loads etc.
```

```
*BOUNDARY_...
Data line to prescribe boundary conditions
*TITLE
Simulation title
*END
```

12.1.4 Full restart example: Bolt pre-tensioning followed by prescribed displacement

The same bolted joint as in Section 12.1.2 is studied, but in this case the pre-tensioning (`pretens001.key`) is followed by a prescribed transverse displacement of 50 mm (`full_restart.key`). The stress result is shown in Figure 60, and the moment vs. tip displacement curve is shown in Figure 59.

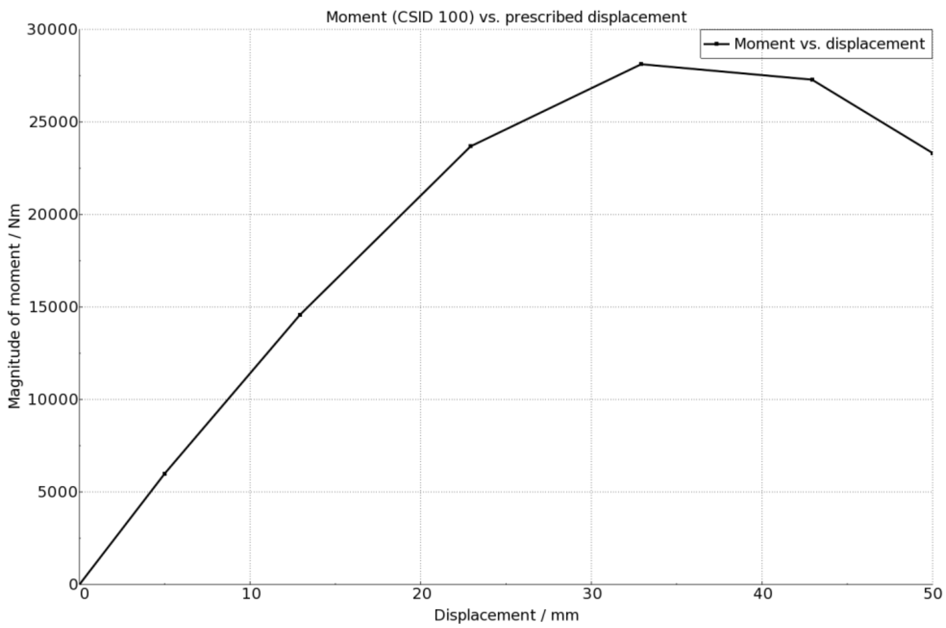


Figure 59. Cross-sectional moment in the lower pipe as a function of applied tip displacement.

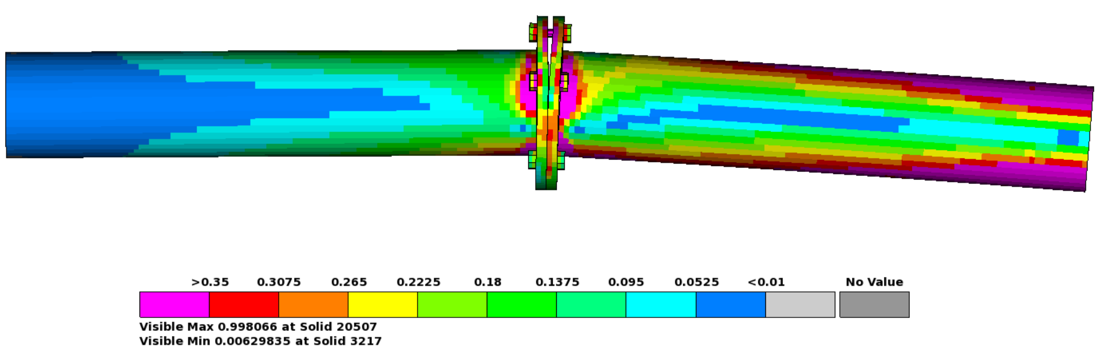


Figure 60. Effective stress due to 50 mm prescribed tip displacement.

An additional example of full restart is provided in Section 16.3.

12.2 Restart using dynain.lsda

The `dynain` file is basically a keyword file containing the final state (deformed configuration, stresses and strains, history variables etc. defined via keywords `*NODE`, `*ELEMENT_...`, `*INITIAL_STRESS_...`) of an analysis. Traditionally the `dynain` file has been used for implicit springback analyses after sheet metal forming.

The binary format `dynain.lsda` [19] has been augmented in order to be useful for continuation of (quasi static) analyses, and should contain the necessary information to start a subsequent analysis from the same point where the preceding analysis finished. The contact state of Mortar contacts and tied contacts is also stored in the `dynain.lsda` file, meaning that for example stresses due bolt pre-tension or press fits should not be lost. Note that the contact state of non-Mortar sliding contacts is not stored.

The binary format `dynain.lsda` is available from R11.2.0 and R12.0 of LS-DYNA. It is a relatively new[21] feature and coming developments for future versions include support for e.g., more element types (the 2nd order shells elforms 23, 24 are supported from R13.1, and 1st order tet elform 13 may be problematic in mpp/LS-DYNA R12.0. The 2nd order hexa elements, solid elform 23 is currently not supported).

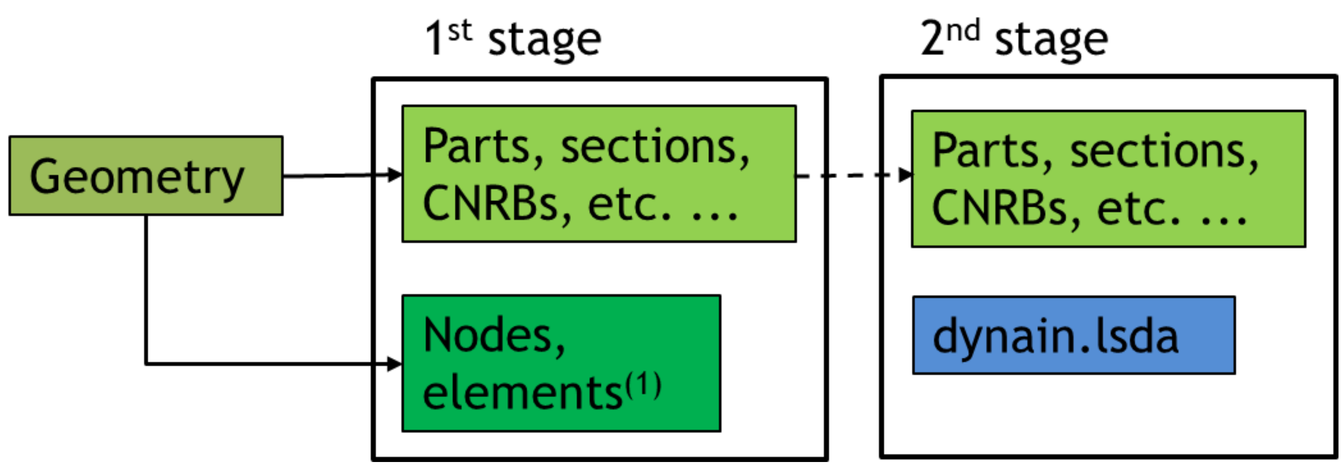


Figure 61. Partitioning of the geometry for convenient use of the dynain.lsda feature. Note: (1) For example `*ELEMENT_MASS` will not be output to `dynain.lsda`.

Since node and element definitions are included in the `dynain.lsda` file, the keyword file for the subsequent analysis must be modified in order to avoid double definition of these entities. It can be convenient to partition the geometry keyword file into two parts: one part with node and element definitions, and one part with the remaining keywords (parts, sections, constraints, sets etc.) see Figure 61. Note that not all element types will be included in the `dynain.lsda` file. See Table 8 for a list of entities included in the `dynain.lsda` file.

[21] As of 2021-02-01

Table 8. Contents of the `dynain.lsda` file

Output to dynain.lsda	Not output to dynain.lsda
*NODE, *ELEMENT_...	*ELEMENT_MASS
*CONSTRAINED_ADAPTIVITY	*CONSTRAINED_*{OPTION}*
*BOUNARY_SPC (1)	*LOAD_...
*INITIAL_STRESS_...	*MAT_...
*INITIAL_FOAM_REF…	*DEFINE_...
Contact state	*CONTACT_
	*PART, *SECTION
	*SET_...

Notes: (1) Output of SPC constraints to the `dynain.lsda` file is in many cases not desired since it limits the flexibility and can be suppressed by the `*INTERFACE_SPRINGBACK_EXCLUDE` keyword.

In a keyword (pseudo-code) template format, the main keyword file of the first analysis would look something like:

```
*KEYWORD
*INCLUDE
geo001_parts_sections.key
*INCLUDE
geo001_node_elements.key
*INCLUDE
control_cards…
*CONTROL_TERMINATION
Define end time of the simulation
*CONTACT_AUTOMATIC_SINGLE_SURFACE_MORTAR_ID
       101Global contact
...
*LOAD_...
Define nodal loads etc.
*BOUNDARY_...
Data line to prescribe boundary conditions
*INTERFACE_SPRINGBACK_LSDYNA
Parameters for requesting a dynain.lsda - file
*TITLE
Simulation title
*END
```

and the main keyword file of the following analyses would look something like:

```
*KEYWORD
*INCLUDE
geo001_parts_sections.key
*INCLUDE
dynain.lsda
*INCLUDE
control_cards…
*CONTROL_TERMINATION
Define end time of the simulation *CONTACT_AUTOMATIC_SINGLE_SURFACE_MORTAR_ID
       101Global contact
...
*LOAD_...
Define nodal loads etc.
*BOUNDARY_...
```

```
Data line to prescribe boundary conditions.
*INTERFACE_SPRINGBACK_LSDYNA
Parameters for requesting a dynain.lsda file
*TITLE
Simulation title
*END
```

The `dynain.lsda` file is automatically recognized by LS-PrePost, from version 4.8, making it possible to open and visually inspect also the subsequent analyses once the `dynain.lsda` file is obtained.

Note that the IDs of contacts in the following analysis must be kept identical to the ones used in the first analysis for the contact state to be properly initiated from the `dynain.lsda` file.
A template for requesting a binary `dynain` file follows:

```
*SET_PART_LIST_GENERATE_TITLE
All parts for dynain.lsda
       338
         1  99999999
*INTERFACE_SPRINGBACK_LSDYNA
$#    psid      nshv     ftype         _    ftensr    nthhsv     rflag   intstrn
       338       999         3         0         0         0         1         0
$#    optc      sldo      ncyc    fsplit    ndflag     cflag     hflag
OPTCARD            0         0         0         1         1         1
*INTERFACE_SPRINGBACK_EXCLUDE
$#                                                                     kwdname
BOUNDARY_SPC_NODE
```

where, of course, the part set ID is arbitrary, and it can contain all or only some of the parts in the model, depending on the nature of the subsequent analyses. The last exclude option is to suppress the printing of boundary conditions to the `dynain.lsda` file. By this, it will be possible to use either the same boundary conditions or completely redefine the boundary conditions for the following analysis. In each analysis, the simulation time goes from T = 0 to T = *endtime* of `*CONTROL_TERMINATION` as defined in each keyword file, respectively. The `dynain.lsda` approach offers similar flexibility as the Full restart option (see Section 12.1.3) with respect to changes of the model definition, control card settings etc.

Normally, the `dynain` file is output at the end of an analysis when the termination time is reached. From R12.0.0 of LS-DYNA, it is also possible to request output of the `dynain` file using sense switches: put the text “SWB”, “SWC” or “SWD” in a file called `d3kil` in the folder where the simulation is running, and LS-DYNA will write a `dynain` file, according to what is specified on the `*INTERFACE_SPRINGBACK_LSDYNA` keyword. The different options are

- SWB: a `dynain` file is written, and LS-DYNA continues,
- SWC: a `dynain` file and restart files are written, and LS-DYNA continues,
- SWD: a `dynain` file and restart files are written, and LS-DYNA terminates.

Note that nodal velocities and accelerations are not stored in `dynain.lsda`, making it less useful if a transient dynamic analysis is to be carried out using the multistage approach.

12.2.1 Combining dynain.lsda and *CASE for multistage analyses

The restart capabilities of the `dynain.lsda` format can conveniently be combined with the built-in case driver of LS-DYNA for automating the analysis of a sequence of loadings. By this, a sequence of load cases can be run after each other automatically, and the state information is propagated between each analysis using the `dynain.lsda` file. This requires that the `*CASE_` - keywords [1] are added to the keyword file, and that "case" is added to the execution line of LS-DYNA when the main job is submitted. See Appendix X of Ref. [1], and Ref. [34] , for more details on this concept for multistage analysis.

In the most basic form, a sequence of analyses, starting with bolt pre-tensioning followed by X- and Y-loadings, can be set up as

```
*KEYWORD
*CASE_BEGIN_1
*INCLUDE
run_pretens.key
*CASE_END_1
*CASE_BEGIN_2
*INCLUDE
case1.dynain.lsda
*INCLUDE
run_xload.key
*CASE_END_2
*CASE_BEGIN_3
*INCLUDE
case2.dynain.lsda
*INCLUDE
run_yload.key
*CASE_END_3
*END
```

using keyword pseudo-code. Using the `*CASE_` keywords, the output files (`d3plot`, `d3hsp`, `binout` etc.) of each analysis will be separated using the case ID as a prefix, for example `case1.d3hsp`, `case2.d3plot`, etc. Keywords outside the `*CASE_BEGIN_`*N* and `*CASE_END_`*N* will be common to all the cases. This makes it possible to place the main model definition, like control cards, boundary conditions and contacts, in the common block and limit the content of each `*CASE_` block to the load-case specific keywords, as outlined in Figure 62. Note that the `*CASE_` keywords must be put in the main keyword file (submitted to LS-DYNA as `i=`*main_keyword.k*). The `*CASE_` keywords cannot be part of include files (referenced by the main keyword file using `*INCLUDE`).

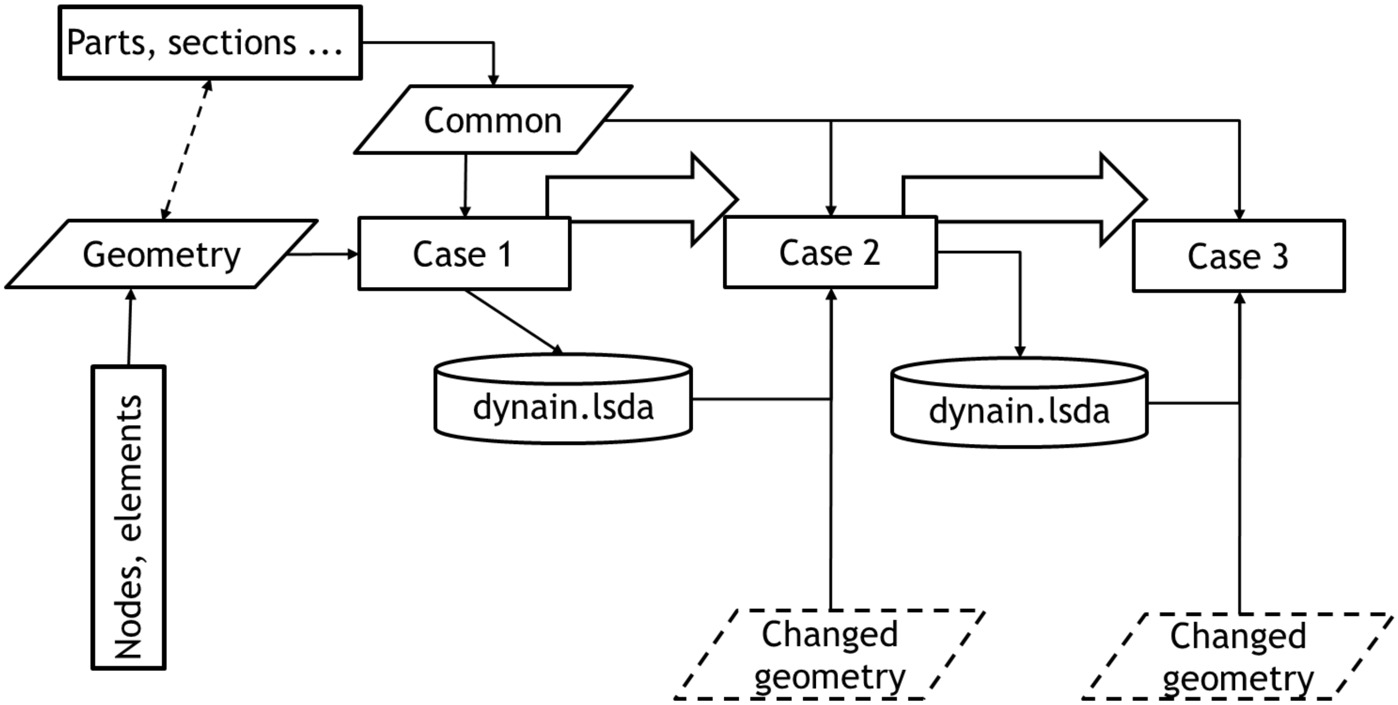


Figure 62. Flowchart indicating the possibilities for analyzing a sequence of load cases using dynain.lsda and the *CASE functionality of LS-DYNA.

12.2.2 Example of a sequence of loadings using *CASE and dynain.lsda

An L-shaped bracket assembly is subjected initially to bolt pre-tensioning followed by loadings in the X and Y directions, see Figure 63. The analysis of these load stages is set up as a sequence of three cases, as described in Table 9.

Table 9. Loading of the assembly using the *CASE concept

Case #	Load description	Simulation time
1	Bolt pre-tensioning	0 → 1
2	X loading ...	0 → 1
	and unloading	1 → 2
3	Y loading ...	0 → 1
	and unloading	1 → 2

The information is propagated between the analyses using the `dynain.lsda` binary format files. The main run file is `L_bracket_dynain.key`. For reference, also an all-in-one analysis is set-up (`L_bracket_allin1.key`), where the loadings vary with simulation time according to Table 10.

The history of peak effective stress in each part for the two analysis approaches are compared in Figure 64, where the results from the `multistage` approach have been time-shifted to match the all-in-one analysis. The general agreement is very good, indicating the capability of the `dynain.lsda` file to carry the state information forward between analyses. In Figure 65, fringe plots of the effective stress at the peak Y-loading are compared between the two analysis approaches. Only very small differences (0.1 %) can be noted.

Table 10. Loading of the assembly using time history

Load description	Simulation time
Bolt pre-tensioning	0 → 1
X loading ...	1 → 2
and unloading	2 → 3
Y loading ...	3 → 4
and unloading	4 → 5

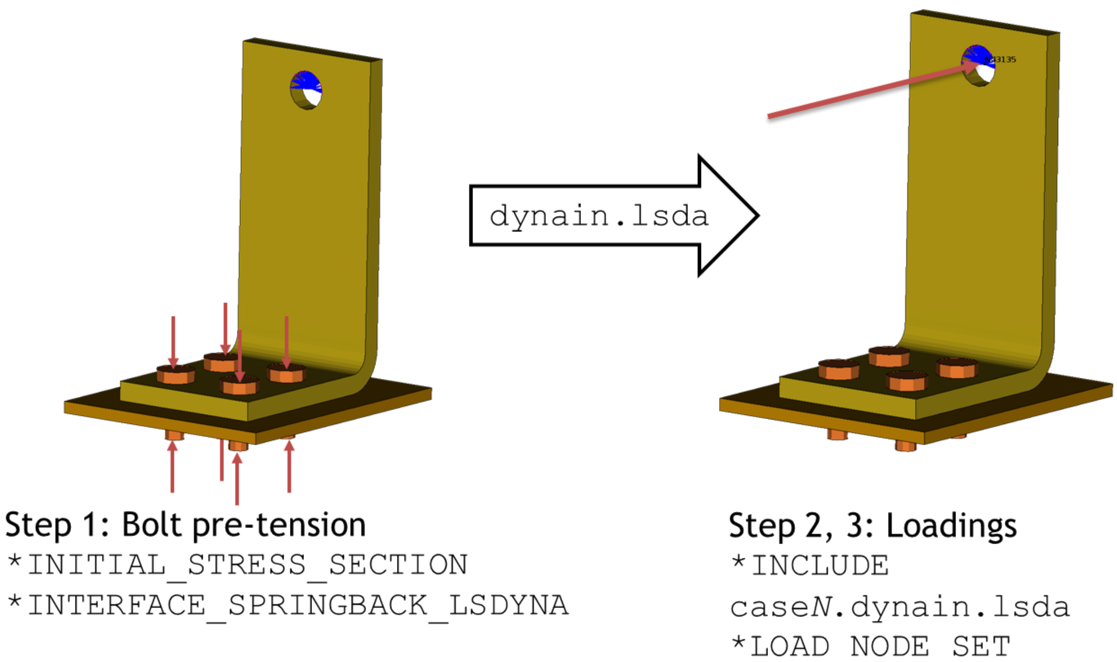


Figure 63. The L-bracket assembly is subjected to a sequence of loadings.

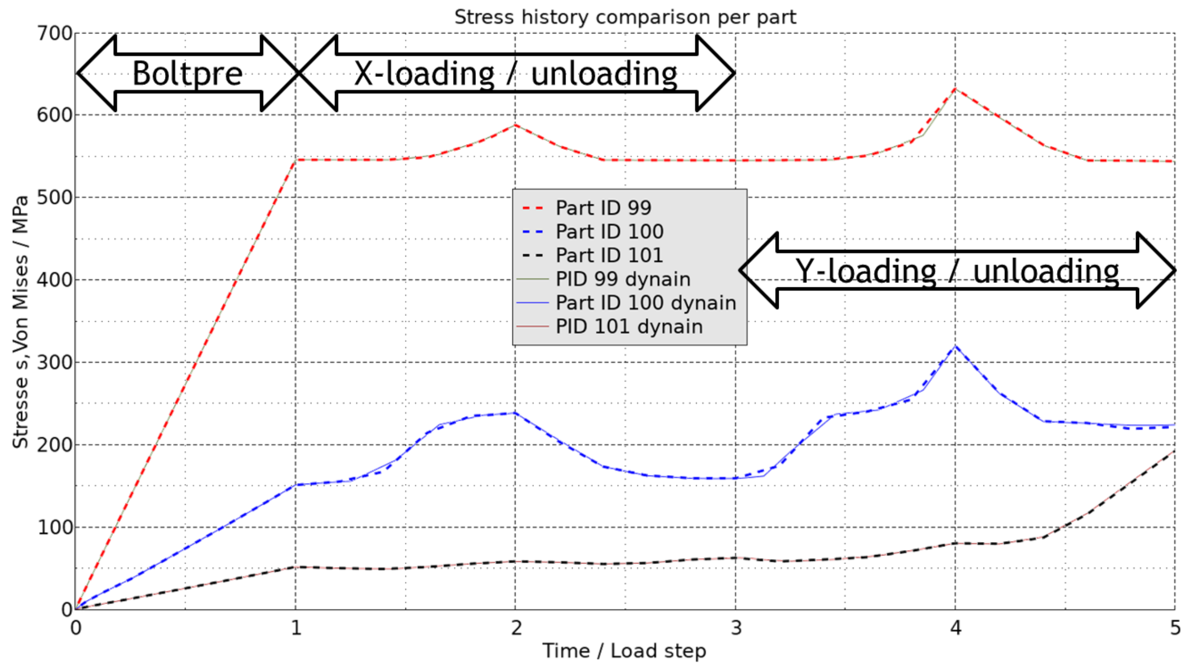


Figure 64. Comparison of peak stress per part, between all-in-one analysis (dashed lines) and the multistage analysis, using dynain.lsda and *CASE (solid lines).

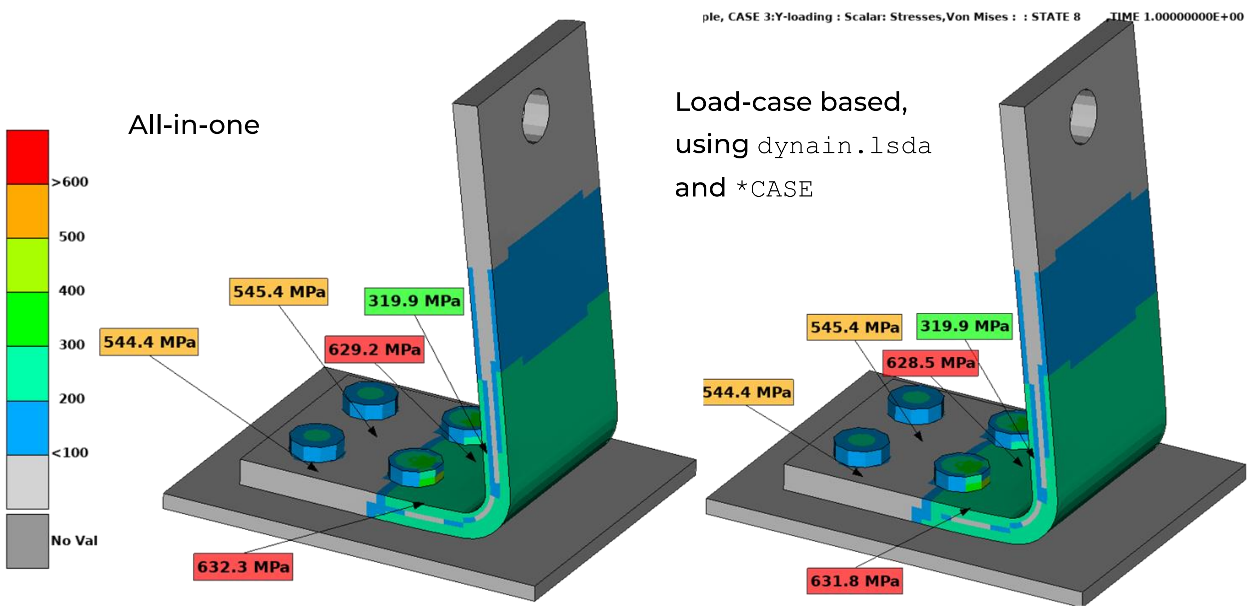


Figure 65. Comparison of effective stress (von Mises) at the peak Y-loading. The left image shows the stress distribution of the all-in-one analysis, and the right image shows the results of the load-case analysis.

Appendix C

13 Troubleshooting convergence problems

It shall be emphasized that convergence problems in (static) non-linear implicit finite element analyses can arise in any FE-solver, it is in general nothing specific for LS-DYNA. The reason for convergence problems is in many cases simply that finding static equilibrium in a non-linear FE analysis (involving contacts, material non-linearity possibly including material failure, and large deformations) is very difficult.

In this Appendix, some tips on how to identify and possibly circumvent convergence problems in implicit analyses in LS-DYNA are presented. The content of this Appendix is also presented in the Webinar [24]. Tips on general model checking can be found in Ref. [25]. Troubleshooting convergence problems is also treated in Appendix P of the Keyword Manual [1].

LS-DYNA outputs information, warnings, and error messages in ASCII text format to the `d3hsp`-file, and the `mes*` files (one per mpp thread in mpp/LS-DYNA, while smp/LS-DYNA writes one `messag` file) during the progress of the analysis. These files also contain information on the progress of the implicit solution, see Figure 66 for an example. Iteration information with respect to the relative norms are printed in the `d3hsp` file. This is also printed in the `mes0*` files, where in addition line search progress is printed, and before each time step also Mortar contact information is output. Additional information from the linear solver which may be of interest in a troubleshooting phase can be requested by setting *LPRINT* = 2 or 3 on `*CONTROL_IMPLICIT_SOLVER`.

Studying the evolution of displacement, energy and force residuals during the iterations can give a good indication to if the solution is converging well or not. In addition to reading this information in a text editor, the D3hsp View tool in LS-PrePost provides a graphical visualization of the residual norms, see Figure 69 and Figure 74.

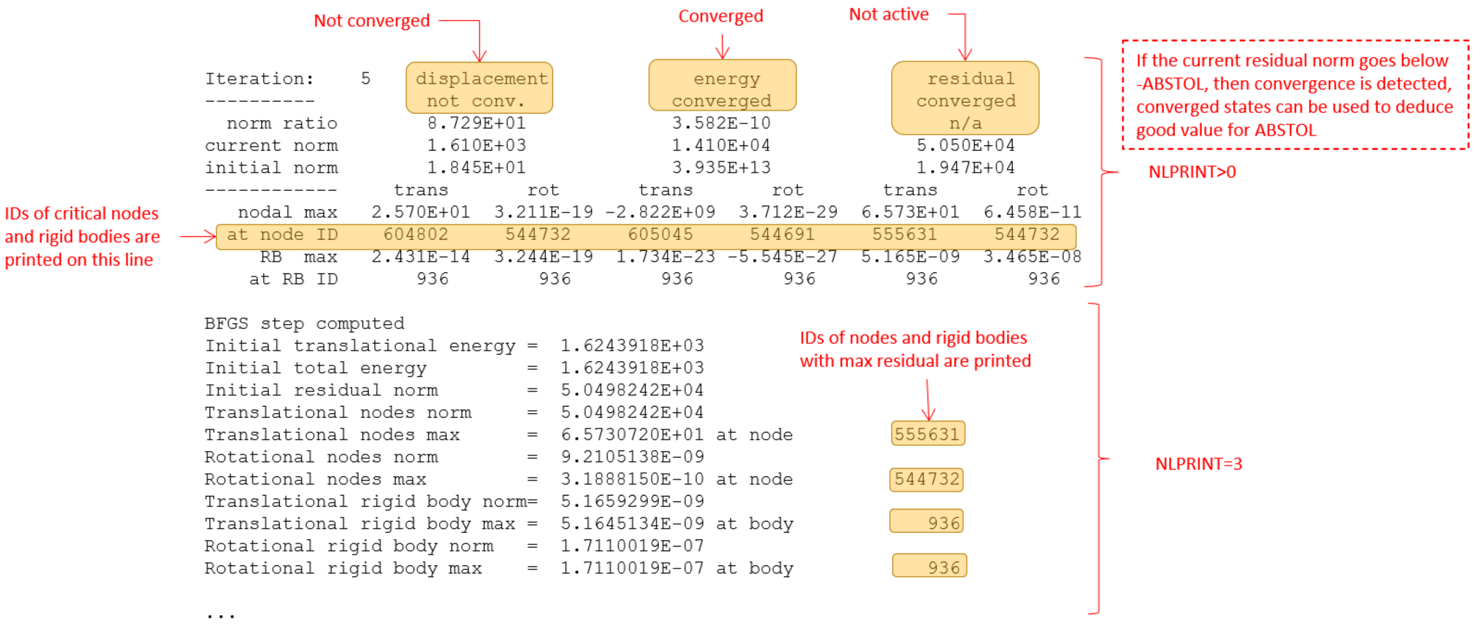


Figure 66. Tracking convergence information in the mes0* files.

If the solution converges well, typically the norms decrease with almost each iteration (see the left image of Figure 67), line search converges in less than 10 iterations, and convergence of a time step is achieved within less than 50 iterations, and the penetrations reported from the Mortar contacts stays within reasonable limits.

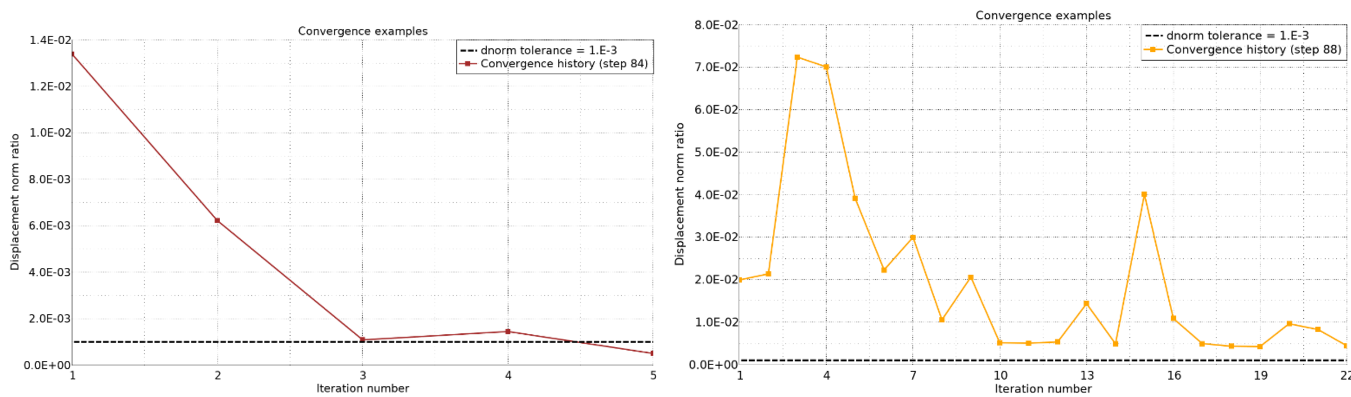


Figure 67. Examples of convergence histories. The left image shows rapid decrease of the displacement norm, and the required tolerance is met in 5 iterations. The right image shows problematic convergence, and the step is abandoned after 21 iterations.

In case the solution progress deviates from this ideal behavior in some way, for example if the residual norms do not decrease in a consistent way but seem to “jump up and down” (see the right image of Figure 67) or if very many (>100) iterations are required to find equilibrium, convergence problems may be at hand. If equilibrium cannot be established within the given number of iterations, the message:

```
automatic time step size DECREASE, RETRY step:
```

is printed, and the time step is attempted again with a reduced time increment. If the time-step size decreases below the minimum allowed the error message:

```
*** Error 60004 (IMP+4)
      *****************************************************
      *                                                   *
      *                 -  FATAL ERROR  -                 *
      *                                                   *
      *  Nonlinear solver failed to find equilibrium.     *
      *                                                   *
      *****************************************************
```

is printed and the solution terminates. Troubleshooting in this context denotes the process of finding out the reason for the problematic behavior and resolving the problems.

The first step in troubleshooting is to make use of the available information output from LS-DYNA, see Figure 68.

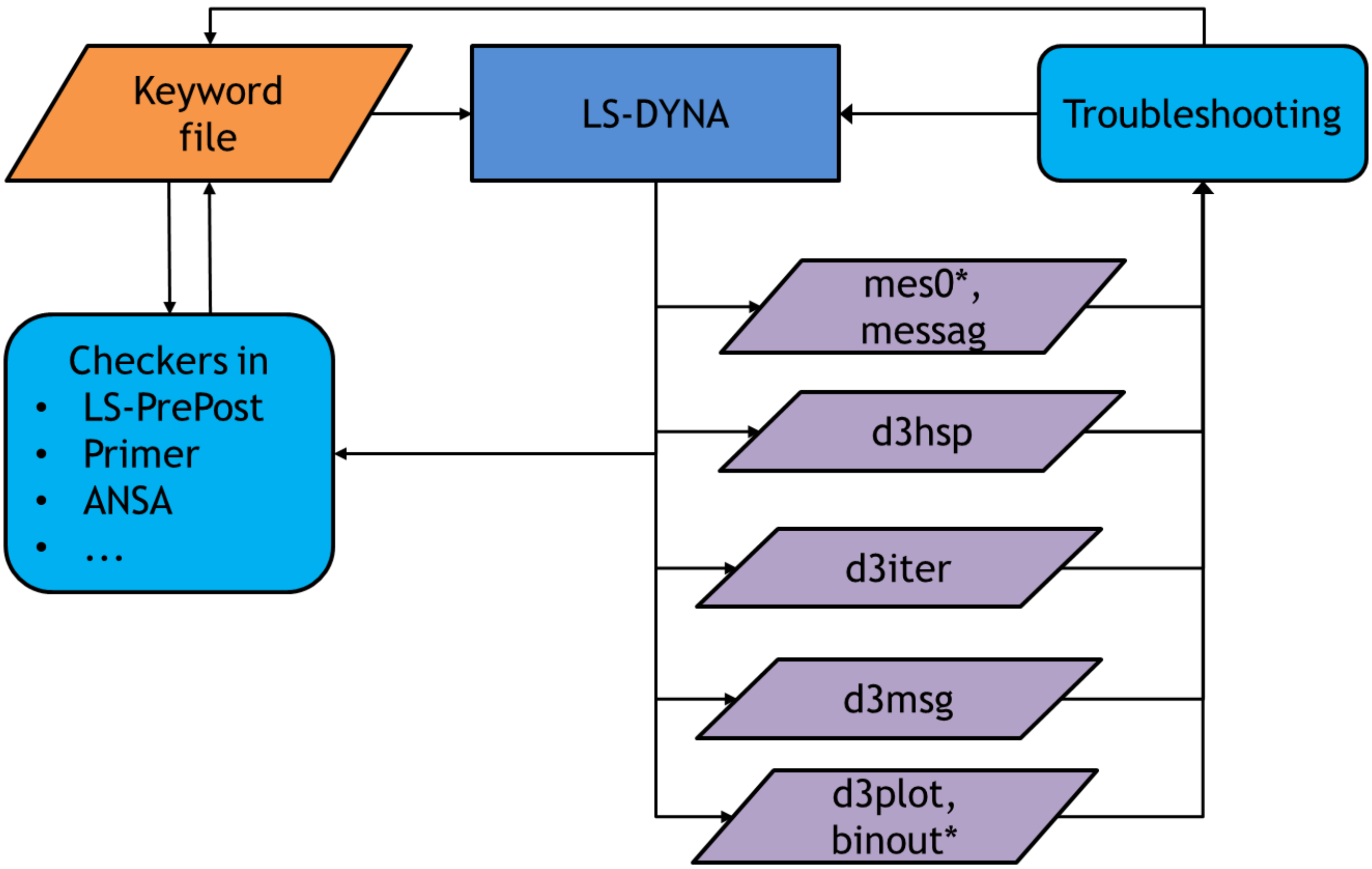


Figure 68. The output files from LS-DYNA which may be relevant in a troubleshooting process.

Inspect the `d3hsp` and the `mes0*` files using a text editor (for example Notepad++ or NEdit or similar) and look for Warning or Error messages, for example the Warning message:

```
*** Warning 60124 (IMP+124)
    XX negative eigenvalues detected
```

indicating the presence of rigid body modes in the model, or the information message

```
AUTOSPC CONSTRAINTS summary information
        AUTOSPC tolerance                                1.000E-08
        no. imposed on translation modes                         0
        no. imposed on rotational modes                          0
```

```
no. imposed on rigid body translation modes          1
no. imposed on rigid body rotational modes           0
```

indicating that the AUTOSPC functionality is active, probably also due to rigid body modes in a static model. Information on errors and warnings can also be browsed using LS-PrePost and the D3hsp View tool (available from the main menu, Misc > D3hsp View) including possibilities of highlighting the model entities causing Warnings or Errors on the 3D model, see Figure 69.

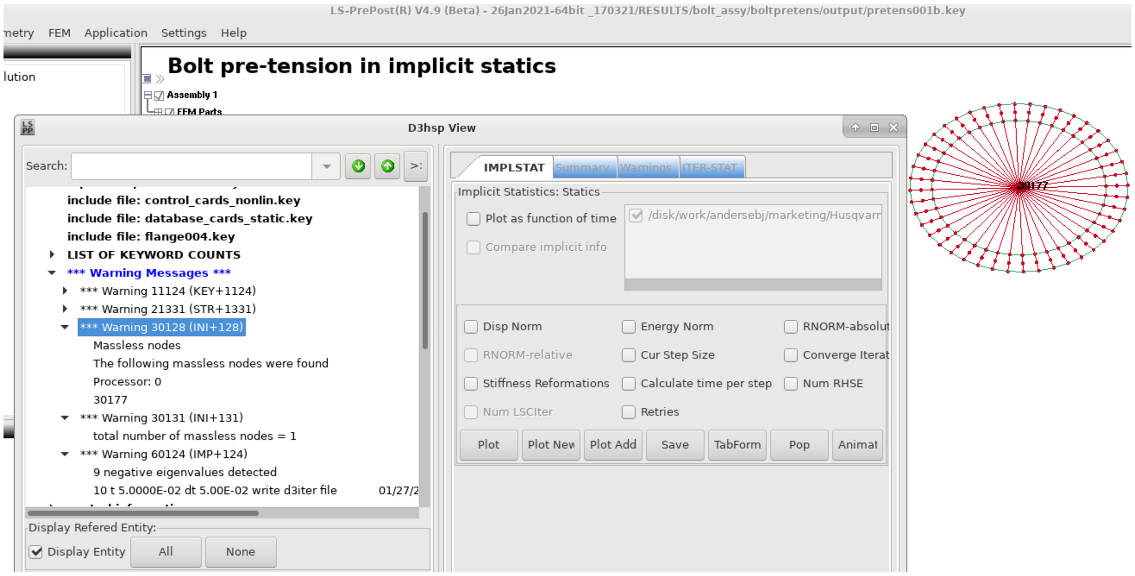


Figure 69. The D3hsp View-tool in LS-PrePost. It is also possible to highlight directly on the 3D-model some of the entities causing Warnings, in this case a massless node.

The `mes0`* files also contain information from the Mortar contacts, regarding penetrations and active segment counts etc., printed before each implicit time step starts. Tracking this information can give a hint to whether the convergence problems are related to the contacts. Information regarding evolution of residual norm and progress of iterations are output in the `d3hsp`-file and with additional information regarding line search also in the `mes0`* files (see also Figure 66).

The `d3iter` binary file (for 3D visualization in LS-PrePost) shows how the model deforms during the non-linear iterations. It can provide very useful information on (unexpected) deformations as likely causes of non-convergence, for example parts "flying away" due to rigid body modes, or severe deformations due to for example inconsistent use of unit systems. Of course, looking at `d3plot` files from previous converged steps can also provide hints to what may cause convergence problems in the following steps.

The `d3msg` file is output after termination and contains more detailed explanations for some warning and error messages. The information in the `d3msg` file could contain fictious ID:s and names of entities.

Once the available information has been carefully inspected, the actual troubleshooting work can begin. The objective is to try to find a reasonable explanation for the non-convergence and hopefully also a remedy. At this point it is strongly recommended to **keep the control card settings** (as attached to this document, see Table 1, or from other sources, or your private preferences). The many options and possible combinations of control card settings in LS-DYNA perhaps makes it tempting to modify these in order to obtain convergence or to assume that the convergence problem is caused by some

LS-DYNA control card setting. However, before making any modifications to the control card settings, it is strongly recommended to first investigate other possible reasons for the convergence problems. A suggested action[22] list follows:

- Perform a basic model check, as outlined in in Ref. [25]. Use for example LS-PrePost (Application > Model Checking > General Checking > Keyword Check).
  - Inspect the model carefully with respect to element quality; small (or even negative) initial volume of solid elements, and small Jacobian values.
  - To some extent, the explicit time step (which is always computed) listing in the d3hsp file can be used as an element quality indicator. For example, if the first listed element has an explicit time step which is several orders of magnitude lower than the others, it can be an indication that this is maybe an element which violates the quality criteria. See Figure 70 for an example.

```
100 smallest timesteps
------------------------------------------------
element        number                 part    timestep
solid          864683                   28  1.6249E-12
solid          841031                   26  2.7367E-09
solid          630230                   27  3.6608E-09
solid          874582                   28  4.3065E-09
solid          233799                    6  6.4381E-09
solid         1166300                    5  6.9968E-09
solid          874583                   28  7.0312E-09
solid         1167573                    5  7.1905E-09
```

Figure 70. Example of listings of explicit time steps in the d3hsp file. Element number 864683 (highlighted in yellow) has a time step which is about 1000 times smaller than the other elements. This indicates poor element quality.

  - The warning for bad Jacobian, for example,

```
*** Warning solid element #   XX has a bad jacobian value -3.991E-01
```

    is in general an indication that re-meshing is required.
  - From R14, lists of shells and solid elements with the worst aspect ratio are printed. Search for "Listing the top" in the d3hsp file. Also, a check for elements with nearly coincident nodes is added, for example,

```
*** Warning 60421 (IMP+421)
     Solid element XX has a slightly bad edge/diag ratio,
     check if nodes are coincidental and adjust mesh.
```

    or in more severe cases, the warning

```
*** Level3-Warning 60421 (IMP+421)
     Solid element XX has a fatally bad edge/diag ratio,
     check if nodes are coincidental and adjust mesh.
```

[22] It is assumed that the appropriate control and database include files, see Table 1, are used. Possible modifications of, for example, control card settings, are described with respect to these as a baseline.

  - Check that no unintended cracks in the mesh are present.
  - Check that consistent units are used (for materials, loadings, accelerations, time etc.)
  - Check for duplicate elements, needle elements, etc.
  - Split pyramid elements (if any) to tetras (there is no proper pyramid element in LS-DYNA)
  - Delete free, unreferenced nodes from the model.
- Check the model connectivity. Unconnected sub-assemblies or “loose parts” will cause rigid body modes in implicit statics, potentially causing convergence problems. One indication of this is negative eigenvalue warnings: look for

```
*** Warning 60124 (IMP+124)
      XX negative eigenvalues detected
```

  in the `d3hsp` file. Unintentionally loose sub-assemblies that start to spin can cause slow convergence in implicit dynamics. Ways of checking model connectivity are:

  - Perform an eigenvalue analysis, (`*CONTROL_IMPLICT_EIGENVALUE`), see Section 4.6. Inspecting the obtained `d3eigv` file normally reveals rigid body modes or mechanisms in a very efficient and visual way.
  - By inspecting the `d3iter` files using LS-PrePost, rigid body modes or parts “flying away” can also be identified.
  - Some preprocessors have dedicated functionality for detecting unconnected assemblies, based on contacts and connectors defined in the model.

  In many analyses, the presence of rigid body modes is intentional. A typical example of this is an assembly that is to be connected via bolts and contacts. Before the bolt pre-tension is applied, such an assembly has many rigid body modes[23]. Set *IMASS* = 1 on `*CONTROL_IMPLICIT_DYNAMICS` to activate implicit dynamics (at least during the initial phase, until contacts are fully established). This normally solves convergence problems due to intentional rigid body modes (see for example Section 4.3.2). In some cases, inertia relief boundary conditions (applied by `*CONTROL_IMPLICIT_INERTIA_RELIEF`) are relevant for handling intentional rigid-body modes. In cases where the rigid body modes are unintentional due for example to forgotten boundary conditions, the solution is to simply apply the appropriate boundary conditions.

  - Check that the element connectivity is valid. Take care when continuum elements (typically solids) and structural elements (beams, shells) share nodes. Invalid connectivity can cause unintended hinges, joints, or other mechanisms. For example, connecting a beam element between two solids using shared nodes only will leave the possibility for the beam to spin freely around its axis. Connecting shells to solids along a single row of nodes could create a hinge. In some cases, LS-DYNA will issue warning messages for potentially problematic connectivity, for example:

```
*** Warning 60301 (IMP+301)
     Using *CONSTRAINED_SPOTWELD with nodes without rotational dofs.
```

  - Avoid the use of release conditions on constrained nodal rigid bodies. It is not recommended to use *DRFLAG, RRFLAG* $\neq$ 0 on CNRBs (other than for two-noded CNRBs in linear implicit analyses, *NSOLVR* = ±1 on `*CONTROL_IMPLICIT_SOLUTION`). If release of degrees of freedom is required, use joints (`*CONSTRAINED_JOINT_...`) instead.

[23] In many cases LS-DYNA may still find static equilibrium, even when rigid body modes are initially present in the model.

From R11 of LS-DYNA, joint stiffness and damping can be applied globally on *CONTROL_RIGID, by the variables *GJADSTF*, *GJADVSC*, *TJADST* and *TJADVSC*. By this, a small artificial stiffness can be added to aid convergence of models containing many joint definitions. As a troubleshooting step, it is recommended to disable joint failure (switch *CONSTRAINED_JOINT_{*TYPE*}_FAILURE to *CONSTRAINED_JOINT_{*TYPE*}) at least in the initial model development phase.

- Check tied contacts. This is often closely related to the model connectivity, since tied contacts are often used for assembling different parts or sub-assemblies of a model.

    - Most preprocessors have built-in tools for checking tied contacts, for example LS-PrePost (accessed via Application > Model Checking > General Checking > Contact Check > Tied).
    - Make sure that the intended tracked nodes get tied. Not too many (which may result in surprisingly rigid behaviour), not too few (which may for example cause unintended rigid body modes).
    - *CONTACT_TIED_... will report to the d3hsp file, nodes that are not tied. The result can be visualized in LS-PrePost using Misc > D3hsp View (or similar functionality in other preprocessors).
    - Non-offset tied contacts (see Table 7) will move the tracked nodes to the reference segments. In some cases, this can cause mesh distortions, especially if a large search distance is given. Look for:

      ```
      *** Warning 41240 (SOL+1240)
      ```

      The remedy for this may be to switch to an offset tied contact or reducing the search distance.
    - If 2nd order elements are involved in a tied contact, it is recommended to use a node set (including mid-side nodes) on the tracked side and define the reference side by part ID or part set ID (*SURFBTYP* = 2 or 3).
    - If structural elements (shells, beams, ... or nodes of these element types) are involved in a tied contact, use *CONTACT_TIED_SHELL_EDGE_TO_SURFACE_{*OPTION*} in order to avoid spinning beams or hinges in the model.

  Using a tracked node set gives better control over the possible tied nodes than if a tracked part set is used. The tie distance can be manually adjusted as described in Section 6.2. When using constraint-based tied contact, nodes subjected to other constraints (*CONSTRAINED_NODAL_RIGID_BODY, *BOUNDARY_SPC, ...) cannot be tied. One remedy may be to set *IPBACK* > 0 on Optional Card E of the *CONTACT_TIED_ definition. By this, LS-DYNA will automatically create a penalty-based tied contact for nodes that are subjected to other constraints.

- Avoid unintended initial penetrations in contacts. The *IGNORE* = 2 option of the Mortar contacts can handle “reasonably large” initial penetrations, but a penetration free initial configuration is always preferred.
    - Most preprocessors have built-in tools for checking and correcting initial penetrations, for example LS-PrePost (accessed via Application > Model Checking > General Checking > Contact Check > Penet) or other preprocessors.
    - The Mortar contacts will report initial penetrations in the mes0* files. Look for:

      ```
      Mortar contact information for contact ID …
      ```

      This information can be used to verify that the contact penetration checks of the preprocessors agree with how LS-DYNA interprets the initial contact state.

- It may occur that initial penetrations are reported in the `d3hsp` or `mes0*` files, but cannot be found in the preprocessor, see Figure 71 for an example. This may be remedied by specifying a reasonably small “contact thickness” for solids using the *PENMAX* parameter (compare Figure 43).

```
Face of solid  element 169691 is penetrating
 face of solid element 179567 by 0.2540949E+01

 Face of solid element 169691 is penetrating
 edge of solid element 179567 by 0.2512578E+01
```

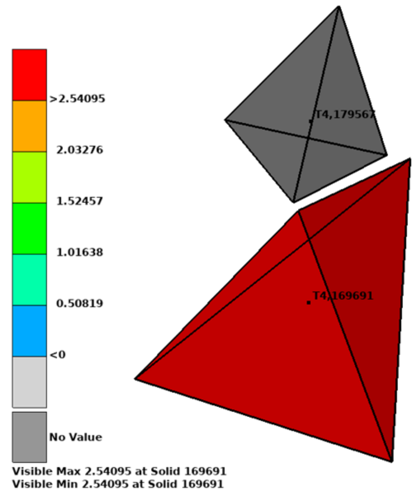


Figure 71. Example of spurious penetration detected by the Mortar contact. The left image shows an extract from the `d3hsp` file where the elements are reported as penetrating, even though they are clearly separated, as shown in the right image.

- From R12 of LS-DYNA, penetration information can also be visualized (fringe plotted) from the `d3plot` file by setting *PENOUT* = 1 or 2 on `*CONTROL_OUTPUT`.
- In cases when it is known beforehand that no self-contact within the same part will take place, and a single surface Mortar contact is used perhaps more out of convenience, the option *IGNORE* = -2 on can be beneficial for convergence since it will neglect self-contact within the same part, thus avoiding spurious self-contact.
- Use *IGNORE* = 3 or 4 of the Mortar contact[24] to resolve press-fits or other intended initial penetrations.

- When shell parts with varying thickness are involved in Mortar contacts, it is often beneficial for convergence to define the thickness variation by `*ELEMENT_SHELL_THICKNESS`, rather than splitting up a (physical) part in many different parts with the thickness variation defined by different `*SECTION_SHELL`.
- Relaxing the penalty stiffness of the Mortar contacts (by decreasing the scale factor for tracked-side stiffness *SFSA* on the `*CONTACT` card) is normally beneficial for convergence since it will give a smoother response from the model.
  - This may be an effective remedy when the convergence is “erratic”, with residual norms (see Figure 66, Figure 67) “jumping up and down” between a low value and a significantly higher value for every other two iterations.
  - Changing the penalty stiffness can be a step in the troubleshooting process, perhaps helping to reveal other problems with the analysis. In most cases, modifying the default contact penalty settings is unnecessary.

[24] Or alternatively `*CONTACT_SURFACE_TO_SURFACE_INTERFERENCE_ID`.

- The Mortar contact will report both relative as well as absolute values of the maximum penetrations during the simulation in the `mes0*` files, before each implicit step begins, for example:

```
Contact sliding interface       1010
Number of contact pairs         2784

Maximum penetration is  0.1122220E-02 between
elements      14606 and      10263 on this processor

Maximum relative penetration is  0.9978402E-01 % between
elements      14606 and      10263 on this processor
```

Based on this the user can judge what is acceptable in terms of penetration distance.

- The Mortar contacts will issue warnings in the `mes0*`-files if the penetration becomes too big to handle for the penalty-based approach. Search for:

  ```
  *** Warning Penetration is close to maximum before release
  ```

  In case the contact is released, a very unphysical configuration may result, making convergence impossible.

  - If initial penetrations are present, try minimizing them.

Try increasing the ramp-up of the penalty stiffness by setting (for example) *IGAP* = 5, 10, 50, 100 (etc.) see also

  - Figure 44.
  - Also, increasing the contact penalty stiffens may help to reduce risk of contact release.

- From R11 of LS-DYNA, the Mortar contacts can also interact with the nonlinear solver. Should very large penetrations occur during the iterations, the Mortar contact can cause the solver to retry the implicit time step with a smaller time increment. The following message is then printed in the d3hsp file (for example):

  ```
  Contact algorithm rejected current iterate,
  list of element pairs affected follows (at most 10):

     Element #          488758 vs          686147

  ------------------------------------------------------------
       automatic time step size DECREASE, RETRY step:
    dt(old) =  1.00000E-02      dt(new) =  4.64159E-03
  ------------------------------------------------------------
  ```

  In this example, the elements 488758 and 686147 are listed as problematic. Inspect the model using the d3iter and d3plot files, in the vicinity of the listed elements, perhaps it could give an indication of what is causing the contact problems. On the other hand, it may very likely be that reject due to contact problems occur only a few times, and the simulation still goes to completion with satisfactory results.

  - From R16 of LS-DYNA, it is possible to disable this possibility for the Mortar contacts to cause the nonlinear solver to reject an iterate. It is done by setting the 2nd filed of the 7th card on `*CONTROL_CONTACT` to 1, see below. This can be used as a step in the troubleshooting process, to see if the solution can progress further, and if possible, more information from the solution can be found.

```
*CONTROL_CONTACT
...
$Card#7      noreject                    TIEOPT
                    1                         1
```

- When suspecting that the contacts are causing the convergence problems, one step in the troubleshooting process can be to switch to the Mortar tied contacts (`*CONTACT_AUTOMATIC_..._MORTAR_TIED_ID`). If convergence is improved by this, it can be an indication that the problem indeed lies in the contacts, for example contact release (see above), but it may also be an indication of for example that rigid body modes were present in the model (switching from sliding to tied contacts will probably cause couplings between parts, which in turn can reduce or remove rigid body modes in the model). Note that switching to Mortar tied contact is not to be seen as a final solution to convergence issues, but is presented in the context of troubleshooting, as a possible step in such a process. When switching to tied contacts, perhaps the analysis can progress further which in turn may help to reveal other problems with the model.

- Do not use shell elform 2 for parts involved in Mortar contacts if shell thickness update is active (*ISTUPD* ≠ 0 on `*CONTROL_SHELL`). Use the fully integrated formulation 16 instead.
  - See Section 5 for recommendations regarding element formulations.

- For 2D analyses, use the appropriate 2D contacts, see Section 6.1.9.

- Check the material models.
  - Avoid using `*MAT_ELASTIC` with $\nu \approx 0.5$ for modelling rubber, at least for finite deformations. Instead, use `*MAT_HYPERELASTIC_RUBBER` or `*MAT_SIMPLIFIED_RUBBER_FOAM` (see Appendix A).
  - If a simple elastic-plastic material model with isotropic hardening is desired, use `*MAT_PIECEWISE_LINEAR_PLASTICITY` rather than setting *BETA* = 1 for `*MAT_PLASTIC_KINEMATIC`.
  - Inspect hardening curves and avoid a negative slope of the last segment, see for example Figure 72. Since LS-DYNA will extrapolate the hardening curve based on the slope of the last segment, negative yield stress values may result (at some un-converged point in the iterations) which in turn may lead to a configuration from which convergence is impossible.

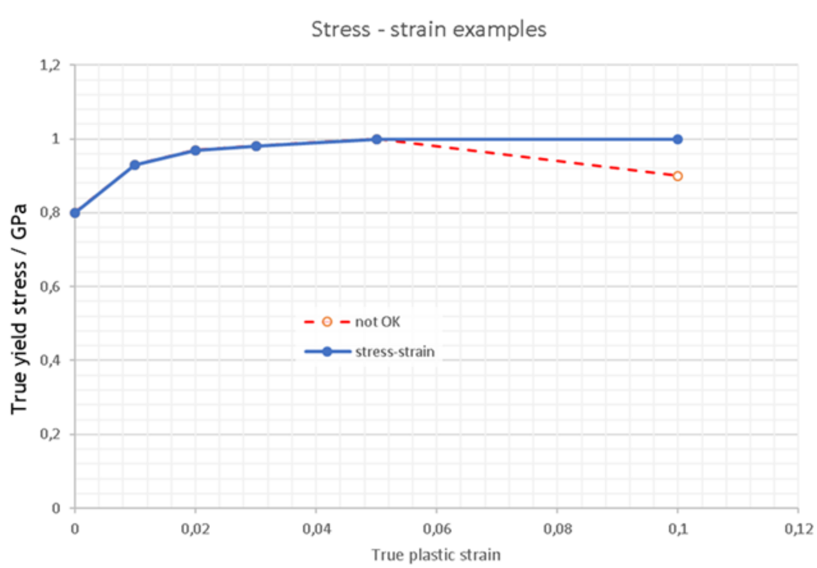


Figure 72. Example of hardening curves. A negative slope for the last segment (dashed red line) should be avoided in implicit analyses.

- Note! Some material models are not supported for implicit analyses.
- A step in the troubleshooting process can be to switch all material models to `*MAT_ELASTIC` (with sensible parameters) and rerun the simulation. If convergence is improved, this may be an indication that the problem is related to the original material models. In some cases, switching material models may result in a simulation that can proceed a bit further, perhaps revealing other problems with the model.
- Avoid involving parts with `*MAT_NULL` in tied contacts.
- If `*MAT_NONLINEAR_ELASTIC_DISCRETE_BEAM` (`*MAT_067`) is used in the model, and you suspect that it may cause convergence problems, try switching to `*ELEMENT_DISCRETE` with `*MAT_SPRING_NONLINEAR_ELASTIC` (`*MAT_S04`).

- From R11 of LS-DYNA, there is a tighter integration between some material models (MAT_24 and MAT_123) and the nonlinear implicit solver. By this, the material routines can cause a retry of an implicit time step if the increment of plastic strain is too big. A message is then printed in the d3hsp file,

```
Material model rejected current iterate,
 list of elements affected follows (at most 20):

     Element #            1234
```

- This could indicate a problematic area of the model. Inspect the model using the d3iter and d3plot files, in the vicinity of the listed elements (1234 in the example above).
- Inspect the material model and doublecheck units, hardening curves etc.
- The message may simply indicate that the time step used was too big, and should in those cases not be seen as too alarming.
- Should the reject occur very frequently, one option to test could be to switch from MAT_24 to MAT_103 (`*MAT_ANISOTROPIC_VISCOPLASTIC`, see Section 7).
- From version R16 of the Ansys LS-DYNA software, material rejects can be disabled using the settings of `*CONTROL_MAT`,

```
*CONTROL_MAT
$#    maef          -      umchk     oldint  noreject
                                                    1
```

Disabling rejects, either by switching material model or by that above control option (from R16), may be seen as a part of the troubleshooting process, but should the obtained results be acceptable with respect to solution quality (force and energy balance, etc.) it may also be part of a final solution to the convergence problems.

- Use a moderate initial time-step size. If the initial time step is too small, it might be hard to find a useful search direction towards the first converged equilibrium. Note that the automatic step size control will cause LS-DYNA to retry the step with a smaller step size if convergence fails (due to too big initial step size, or another reason). It cannot increase step size (even though this resolves the convergence problem in some cases).
  - In some cases, for example transient dynamic analyses, a very small time-step may be required in order to resolve a rapid event. Then also decrease *DTMIN* on `*CONTROL_IMPLICIT_AUTO`, to perhaps 1 % of the smallest target time step, so that LS-DYNA is given the possibility to do at least one halving of the time step if a retry is necessary.

- If the solution of an implicit time step fails, causing a time step reduction (RETRY), then converges nicely with the reduced time-step size , causing a time step increase, but then again fails, causing a retry, then convergence, etc., see for example Figure 73, this is probably an indication that the given maximum time step is too large. In such cases, reducing the maximum time-step size (given by LCID 700 with the attached control cards) may be an effective solution to this type of slow convergence. This can also be combined with less aggressive settings of `*CONTROL_IMPLICIT_AUTO` (reducing *ITEOPT* and *ITEWIN*).

Implicit Solution Statistics

STEP	ATT	EQUIL ITERS	TOTAL TIME	INC OF TIME
1	1	4	0.0100	0.0100
2	1	6	0.0258	0.0158
3	1	5	0.0510	0.0251
4	1	3	0.0908	0.0398
5	1	28	0.1539	0.0631
6	1	11	0.2539	0.1000
7	1	12	0.4124	0.1585
8	1	16	0.6636	0.2512
9	1R	4	1.0617	0.3981
9	2	15	0.8483	0.1848
10	1	26	1.1412	0.2929
11	1R	2	1.6054	0.4642
11	2	19	1.3566	0.2154
12	1	25	1.6981	0.3415
13	1R	3	2.1981	0.5000
13	2	7	1.9302	0.2321
14	1R	3	2.2980	0.3678
14	2	6	2.1009	0.1707
15	1R	3	2.3715	0.2706
15	2	13	2.2265	0.1256
16	1	7	2.4256	0.1991
17	1	19	2.7410	0.3155
18	1R	3	3.2410	0.5000
18	2R	46	2.9731	0.2321
18	3	16	2.8488	0.1077

Figure 73. Implicit solution statistics, by the Tab Form function of the D3hsp View tool in LS-PrePost. In this case, the maximum allowed time step was too big (0.50) for an efficient solution (but after many RETRIES, still Normal termination was obtained).

- The `d3iter` file in many cases provides very useful (and visual) information indicating reasons for the convergence problems. Looking at the deformation of a non-converged state shows if loose parts “fly away” (due to rigid body modes).
    - Scale up the displacement, or look at a fringe plot, in order to identify areas of the model where large changes in displacement take place between iterations. This will highlight problematic areas of the model.
    - Extreme deformations may be an indication that inconsistent units are used.
    - By setting *RESPLT* = 1 on Card 4 of `*DATABASE_EXTENT_BINARY`, it is possible to fringe plot the force residual from the `d3iter` binary database. This can also be very useful for pinpointing the areas of the model where convergence is the most challenging.

- Is the physical problem in fact unstable? Will collapse, buckling, or bifurcation occur? If the applied loading exceeds the ultimate capacity of a structure, convergence in implicit statics will in practice be impossible to reach.
  - Try switching from load to displacement control (see Figure 50).
  - Activate the arc-length solver (see Section 4.5)
  - Activate implicit dynamics (see Section 4.9). Note that in implicit dynamics, time means physical time. If for example the unit system kg–mm–ms is used, an appropriate load application time would be on the order of 1000 ms.

- For analyses involving rubber (or other incompressible materials) try disregarding the initial geometric stiffness effect by setting *IGS* = 2 on `*CONTROL_IMPLICIT_GENERAL`.
  - Disregarding the geometrical stiffness can be beneficial for convergence in other situations as well. It is disabled in the control card files provided with this Guideline.

  - If negative eigenvalue warnings are issued, which cannot be related to rigid body modes or mechanisms, they are probably due to severe deformations of the elements. In such situations it might help to use under integrated elements (solid element formulation 1) or the CPE elements, activated by choosing Hourglass formulation 10 for solid element formulation 1 or 16.

  - In some cases of extreme deformation of shell structures, it may be beneficial to activate the hourglass type 8 for shells of element formulation 16, compare Figure 38.

- Switching to a full Newton solution scheme may be efficient for solving highly non-linear problems. This means that the stiffness matrix is reformulated and factorized for each iteration.
  - If the default BFGS method consistently requires very many (>100) iterations to converge, or if the residual norms decrease very slowly with each iteration, it may be in place to switch to full Newton.
  - Set *ILIMIT* = 1 and increase *MAXREF* to 30–60 on `*CONTROL_IMPLICIT_SOLUTION` to activate a full Newton solution. The settings of `*CONTROL_IMPLICIT_AUTO` may have to be adjusted to account for the reduced number of total iterations allowed.

- Activating the non-symmetrical equation solver may aid convergence in some cases, for example follower loads or “snap-through” deformation, or contacts with high ($\mu \approx 0.3$ or above) friction. It can also be beneficial for cases involving anisotropic material models.
  - Set *LCPACK* = 3 on `*CONTROL_IMPLICIT_SOLVER`.

- Check for “too easy” convergence. This means that LS-DYNA has previously accepted a state, even though it was not converged “enough” (some residual forces or deformations remain in parts of the model). When proceeding from this state, the residuals might lead to non-convergence, since the problems from previous steps remain to be resolved.
  - Check for “noisy” contact force histories or unsatisfied global equilibrium (for example if `spcforc` does not add up to the external forces) or large force residuals are printed in the `d3hsp` file.
  - To remedy this, try tightening the tolerances slightly on `*CONTROL_IMPLICIT_SOLUTION`, for example by setting *DCTOL* = 5E-4 and/or adding residual force criterion *RCTOL* ≈ 0.02 – 0.1.

- Some clues may be found by tracking the convergence information, see Figure 66, regarding the line search history. Typically, the line search should converge within 10 iterations. In cases where the line search requires significantly more than 10 iterations to converge, and the residuals remain high even with very small step sizes, this may be due to that the loading being unreasonably high (severe overloading of the structure). In such cases, check that consistent units are used (so that the loading has the correct order of magnitude). If the loading is correct, it might help to reduce the time-step size.
- Note that the node ID/rigid body ID with the maximum residuals are listed in the iteration information (see Figure 66). Inspecting the model at these ID:s (and perhaps their surroundings)

might give clues to what is causing the convergence problems . This can for example be poor element quality or unconnected parts. This is especially likely if the same ID:s show up as critical for several consecutive iterations. These entities can be highlighted on the model using the D3hsp View tool, see Figure 69 and Figure 74.

- Converting the model for explicit analysis (see Appendix D for further details) might give valuable insight for troubleshooting (like unconnected parts “flying away”).
    - In cases when convergence cannot be obtained using the implicit solver, switching to explicit analysis is an alternative option for obtaining a solution.

Residual histories, iterations to converge, etc. may also be plotted in LS-PrePost (from version 4.5) using Misc>D3hsp View, see Figure 74.

If the problems (convergence related or other) occur during a restart, it might help to test other options for continuation of analyses.

- If a small restart (using d3dump) fails, try a full deck restart instead (using d3full)
- Other options/workarounds would for example be using a `dynain` file (from `*INTERFACE_SPRINGBACK_LSDYNA`, see Section 12.2) or a `drdisp.sif` file from a dynamic relaxation (set *IDRFLG*=2 on `*CONTROL_DYNAMIC_RELAXATION` to initiate to a predefined geometry).

If you suspect that the convergence problems are related to bolt-pretension, it might be worth investigating the effect of setting *IZSHEAR* = 0 on `*INITIAL_STRESS_SECITON` (for solid bolts) or *KBEND* = 0 on `*INITIAL_AXIAL_FORCE_BEAM` (for beam element bolts). By this, the effect of the bending stiffness of the section to be pre-tensioned is neglected, which may improve convergence, but could also cause unphysical results if the bolt as a consequence undergoes bending due to the pre-tension (for example due to uneven stiffness distribution of the flanges).

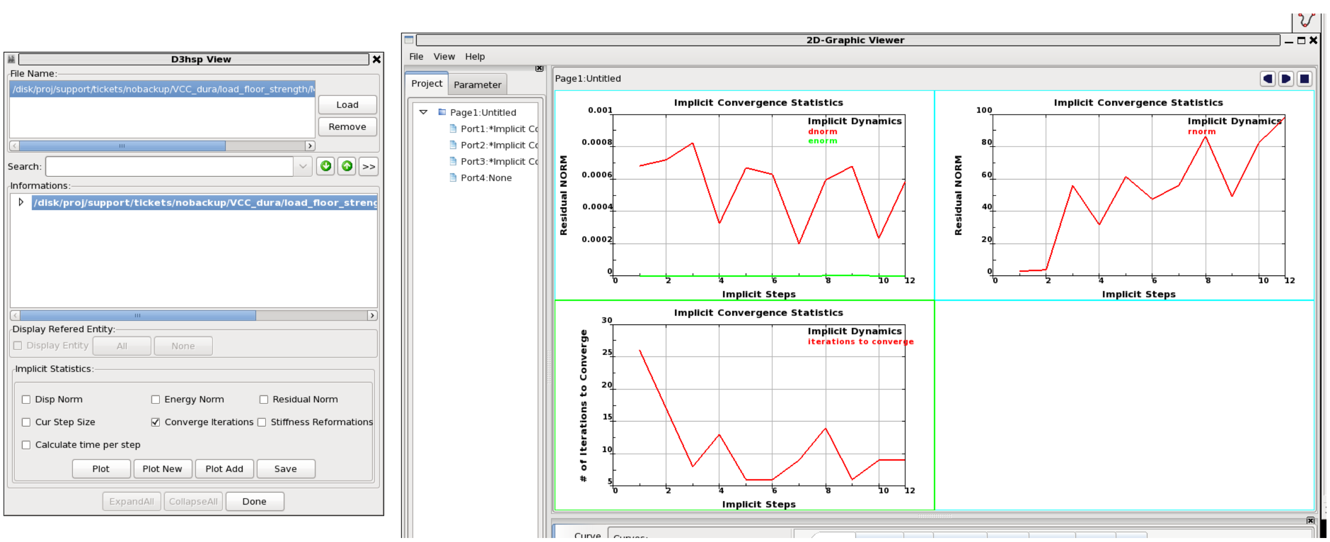


Figure 74. Tracking the convergence histories using the D3hsp View tool in LS-PrePost.

For extended interactive troubleshooting, it is possible to send so-called sense switch controls to the Ansys LS-DYNA software. This can be done by creating a text file named `d3kil` in the directory where the current simulation is running. The text file should contain the appropriate text string, for implicit analysis, the relevant ones in this context are

- `conv`, to force implicit nonlinear convergence for current time step.
- `ttrm`, to terminate iterations for the implicit time step, and retry time step with a reduced time increment.
- `rtrm`, to terminate implicit at end of current time step.
- `lpri`, totoggle implicit lin. alg. solver output on/off.
- `nlpr`, to toggle implicit nonlinear solver output on/off.
- `Iter`, to toggle implicit output to `d3iter` database on/off.

The sense switch `conv` may be interesting for troubleshooting purposes, since it can be used to force the current time step to converge and push the solution on step further, which may be useful to gather further information from the evolution of the solution and find more clues to reasons for nonconvergence. Note that the sense switch `conv` is only for troubleshooting, it should not be a part of a permanent solution.

If convergence problems prevail, do not hesitate to contact your local supplier of the Ansys LS-DYNA software.

Appendix D

14 Converting an implicit model for explicit analyses

This Appendix gives a brief description of how to convert an implicit model for explicit analysis. Training courses and webinars in explicit analyses using LS-DYNA can be found on the Ansys Learning HUB (ALH). The Guideline for explicit analyses [22] provides more details on how to set up different types of explicit analyses.

The control cards and database cards for explicit analyses, taken from Ref. [22], are provided in the keyword file `control_database.k`. Note that these control card settings are merely recommendations, and the user must verify the applicability to the current application. To these, the user will have to add keywords `*CONTROL_TIMESTEP` for reasonable mass scaling (see further Section 14.1) and `*CONTOL_TERMINATION`. A parametrized version of the database cards is provided in `database_cards_expl.key`, where the `*PARAMETER` for controlling the output frequency is `dbparu`. A template for explicit analyses follows:

```
*KEYWORD
*INCLUDE
control_database.k
*CONTROL_TIMESTEP
Define time incrementation and mass scaling (DT2MS) and selective mass scaling (IMSCL)
*PARAMETER
Rdbparu, <output time step for d3plot>
*INCLUDE
database_cards_expl.key
*CONTROL_TERMINATION
Define end time of the simulation
*INCLUDE
Include file defining geometry, materials etc.
*LOAD_...
Define nodal loads etc.
*BOUNDARY_...
Data line to prescribe boundary conditions
*TITLE
Simulation title
*END
```

In many cases, explicit analysis is a very efficient alternative for solving highly non-linear problems and rapid events (such as impact or crash tests). It may also be an effective and visual tool for troubleshooting model problems, since an explicit analysis will reveal model “flaws”, like unconnected regions, missing contact definitions etc. However, quasi-static analyses of weak structures may be challenging. Explicit analyses always include dynamic effects and are performed in physical time, which means that a very long load application time may be required for structures with low eigenvalues. To some extent, mass scaling (see Section 14.1) and damping (see Section 14.5) can be used to overcome these difficulties.

14.1 Time step control and mass scaling in explicit analyses

In explicit analyses, the time-step size will be limited by the Courant-Friedrichs-Lewy-criterion, stating that no information can propagate across more than one element per time step. Roughly, the critical timestep ($\Delta t_{critical}$) can be related to the element size[25] l_e, material stiffness E_e, density ρ_e, and damping d as

$$\Delta t_{\text{critical}} = \frac{2}{\omega_{\max}}\left(\sqrt{1+\xi^2}-\xi\right), \qquad \xi = \frac{d}{2m\omega_{\max}}$$

$$\frac{2}{\omega_{\max}} \approx \min_e (l_e / c_e), \qquad c_e \sim \sqrt{\frac{E_e}{\rho_e}}.$$

The most convenient way of increasing the critical time step (for a given mesh size) is to use the mass scaling features of `*CONTROL_TIMESTEP`. Two versions are available:

- Conventional mass scaling, by setting *DT2MS* < 0, where LS-DYNA will add mass where it is required in order to meet a given time step |*DT2MS*|, and
- selective mass scaling, by setting *IMSCL* = -`PSID`, where `PSID` is the part set listing which parts should undergo selective mass scaling.

Conventional mass scaling adds mass to the model, while maintaining the pure diagonal structure of the mass matrix. This means that the added mass will couple directly to the translational degrees of freedom. The selective mass scaling adds mass also to the off-diagonal entries in the mass matrix. This means that less mass will be added to the translational DOFs but comes with the associated cost of the now non-diagonal mass matrix having to be factorized. LS-DYNA will print out the amount of added mass in the `d3hsp`, `glstat` and `matsum` (per `*PART`) files. The mass scaling will affect the dynamic properties of the model, and the allowable amount is highly problem dependent. For transient dynamic analyses, a moderate amount (5 – 10 %) is recommended, while for example less dynamic forming simulations can be run with over 100 % mass scaling.

14.2 Contacts for explicit analyses

The Mortar contacts are optimized for accuracy and general applicability in implicit analyses, while other contacts in LS-DYNA are more optimized for performance in explicit analyses. The Mortar contacts will work also in an explicit analysis, but the computational cost may be high, and it might therefore be an idea to switch to `*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_ID` or `*CONTACT_AUTOMATIC_SINGLE_SURFACE_ID`. It is recommended to use contact damping in explicit analyses (set *VDC* = 20 – 40 on Card 2 of the contact).

For further details on contacts in explicit, the course "Contacts in LS-DYNA" is warmly recommended. See also the section "Contact cards" of Ref. [22]. A template for using the automatic single surface contact follows.

[25] For 2nd order elements, the shortest distance between nodes along element edges. An overview of how changing from 1st to 2nd order elements affects the time step in explicit analyses is presented in Ref. [10].

```
*CONTACT_AUTOMATIC_SINGLE_SURFACE_ID
Define contact ID, Heading
Card 1: Define what shall be in contact (only SURFA)
Card 2: Define static and dynamic friction, and contact damping (VDC)
Card 3: Define alternative penalty stiffness (SFSA)and thicknesses (SAST) or
leave blank to get defaults
Card 4: Set SOFT = 1 or 2
```

It shall be noted that this is merely a recommendation. It does not cover all possible contact situations that can arise. Thera are many situations where special care must be taken with the contact definitions in an explicit analysis.

14.3 Element formulations for explicit analyses

An explicit analysis consists of very many small time steps, and in each cycle, the element routines for all elements of the model have to be processed. This means that element calculations will make up a (relatively speaking) big part of the computational cost. This also motivates the use of under-integrated elements for explicit analyses. Recommended under-integrated elements are shell element formulation 2, and solid element formulation 1. See Table 11 for an overview of recommended elements for general structural analyses in explicit.

Table 11. Recommended elements for explicit analyses

Element type	Comment	LS-DYNA keyword	El. form.	Under-integrated el. form.
Beam		`*SECTION_BEAM`	1	
	For bolts w. pre-tension		9	
	For springs, dampers etc.		6	
Shell	1st order	`*SECTION_SHELL`	-16	2
	2nd order		23	
Solid	1st order hex	`*SECTION_SOLID`	-1 (-2)	1
	(1st order tet)			(10(1), 13)
	2nd order tet		16 (17)	
	2nd order hex		23	

Notes: (1) Not all material models support tet elform 13 in explicit analysis, see Ref. [1].

Fully integrated 1st order elements (for example shell elform ±16 and solid elform -2) can also be used in explicit analyses but may result in a higher computational cost. Under-integrated elements require hourglass control. It is recommended to use the LS-DYNA keyword `*HOURGLASS` and assign hourglass control per part. Note that under-integrated elements cannot guarantee mesh convergence in a strict mathematical sense.

The use of 2nd order elements will increase computational cost, both due to the increased computational cost associated with the element routine, and the reduces critical time step [10].

14.4 Load curves

In order to introduce a minimal amount of high frequency excitations, all curves used in loadings or displacements should be as smooth as possible. It is recommended to use prescribed velocities rather than displacements in order to obtain a prescribed motion. The purpose is to reduce the spikes in the acceleration signal. For example, if a displacement is prescribed using a linear ramp, this corresponds to a constant velocity. But at t=0, this corresponds to a jump in acceleration, which can introduce dynamics into the model. It is recommended to use linear ramps (or smoother) up to constant velocities. This reduces the peak velocity required, and thus also the inertial forces in case of a reversed direction of travel. The keyword `*DEFINE_CURVE_SMOOTH` can be used to generate smooth curves for velocities, see Figure 75. In addition, for forces and other loadings, smooth curves (at least linear ramps) should be used.

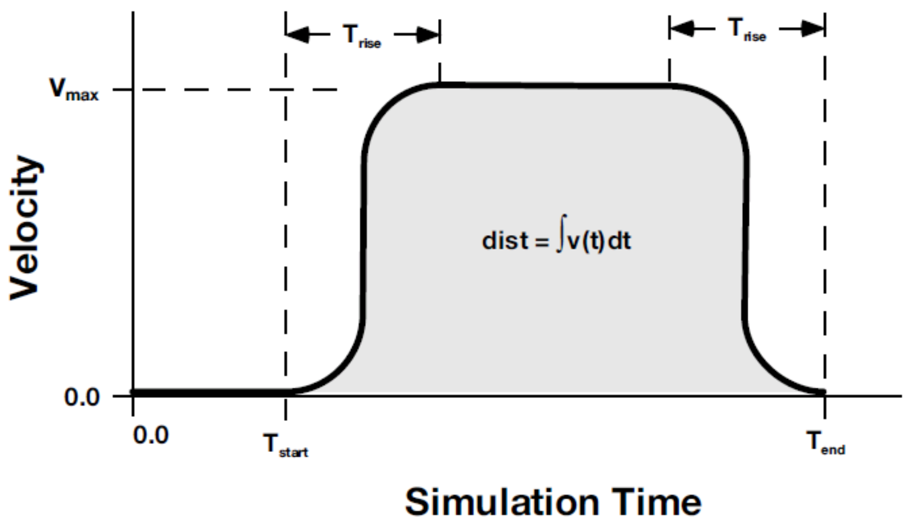


Figure 75. The keyword `*DEFINE_CURVE_SMOOTH` can be used to create a smooth velocity curve based on a desired displacement distance.

In explicit analyses, it is also recommended to extend the load curves beyond the termination time. A reason for this is that in explicit analyses, since a fixed time step is used, it might occur that the simulation terminates at a time slightly after the termination time (as specified on `*CONTROL_TERMINATION`).

14.5 Damping

See Ref. [23] for a review of different damping options for explicit analyses.

Stiffness damping, `*DAMPING_PART_STIFFNESS_SET`, with a value of 0.02 – 0.05 (corresponding to 2 – 5 %) may be applied [13] in order to reduce high frequency noise in the solution. For simulations involving pre-tensioning of beam-style bolts by means of `*INITIAL_AXIAL_FORCE_BEAM`, the use of stiffness damping is recommended [1]. Mass damping (`*DAMPING_GLOBAL` or `*DAMPING_PART_MASS_{SET}`) can in some situations be applied to reduce unwanted oscillations in the solutions that arise due to the dynamics that are introduced when the loading is ramped up. This is illustrated[26] by an example in Figure 76, where first a pre-tensioning (blue curve) is ramped up until 20 ms. When the blue curve ends, pre-tensioning is completed, and then no further loadings are applied for 10 ms. During this time, the damping (red curve) is ramped up, after which full damping is applied for 10 ms, and then removed. Care must be taken in order not to introduce unphysical effects due to the mass damping (typically, high values of mass damping can constrain rigid body motions).

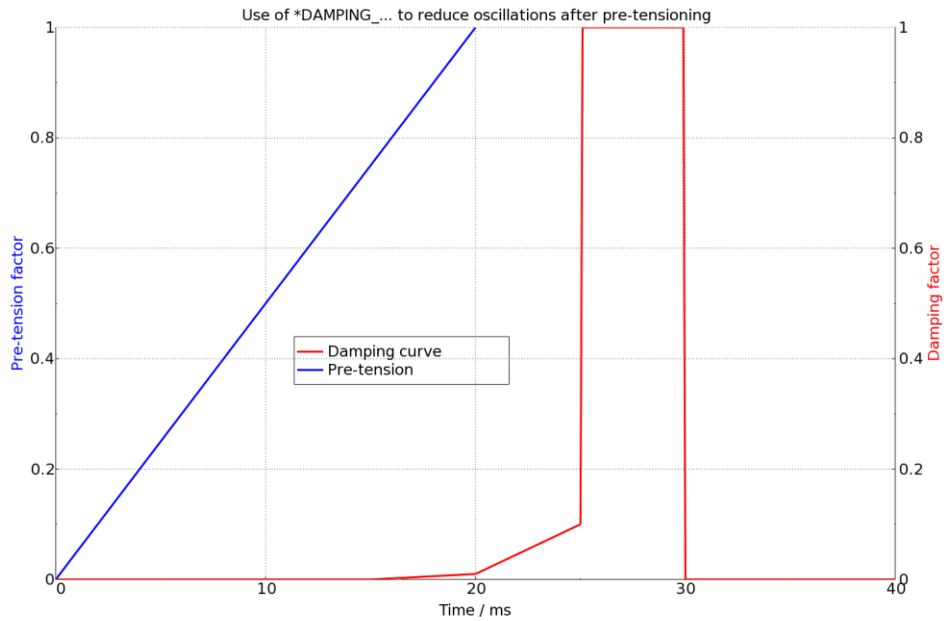


Figure 76. The use of mass damping to reduce oscillations after load application.

14.6 Global energy balance

The global energy balance from the `glstat` file, see Figure 77 for an example, can be used as a basic check of the model validity. In general, negative sliding energy indicates problems with the contacts (in this case, an initial press-fit has been performed, which explains the negative values). Contact energies can be studied per contact ID in the `sleout` file. Often, if large negative sliding energy is found in a contact, some part has a corresponding increase in internal energy. Internal energies can be studied per part in the `matsum` file. By searching the `matsum` file for parts which have a noticeable increase in internal energy at the time when the negative sliding energy increases, problematic parts or areas may be identified.

From under-integrated elements, a small amount of hourglass energy may be acceptable. For quasi-static analyses, check that the kinetic energy is reasonably low compared to the total energy.

[26] Note that this is only an example! In the general case, ramp times and damping values suited for each particular model must be used.

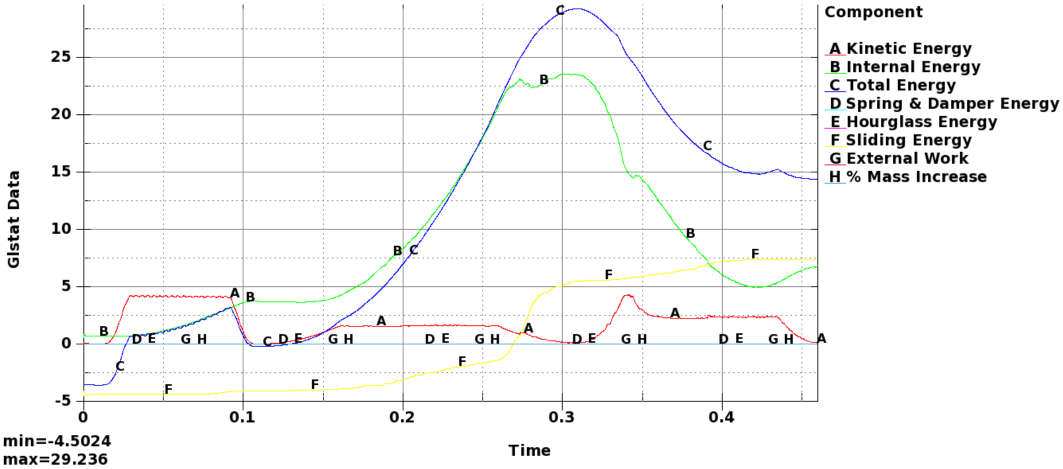


Figure 77. Example of global energy balance plot from `glstat`.

Indications of a problematic analysis may be

- Energies going to +/- infinity,
- Negative sliding energy (> 10 % of total energy)
- Large amounts of hourglass energy.
- Large amounts of kinetic energy, if the objective is a quasi-static analysis.

14.7 Massless nodes

For explicit analyses, it is in general recommended to have a model free from massless nodes. Warning messages will be printed in the d3hsp-file, and many preprocessors have built-in checks for identifying massless nodes in the model. In explicit analyses, great care must be taken when applying loads or boundary conditions to massless nodes (for example center nodes of CNRB "spiders").

14.8 LS-DYNA versions

In general, it is recommended to use a single precision mpp version of LS-DYNA, since it will save computational cost. In some cases, with very many (> 10^6) explicit time steps, round-off errors may, in single precision, cause numerical instabilities, and in those cases, it is recommended to switch to double precision.

Appendix E

15 Converting an explicit model for implicit analyses

A correct and working explicit LS-DYNA model is in general a good starting point for creating an implicit LS-DYNA model. A very short general description of the conversion process follows:

- Check the model for features that are not supported in implicit analysis. Some material models, like `*MAT_NULL` for solids, are not supported in implicit analysis. If user-defined material models are used, these must also meet the requirements for an implicit analysis or be switched to an appropriate alternative standard LS-DYNA material model. Note that material failure by element erosion can also be studied in implicit analysis but may make convergence very challenging. It is recommended to deactivate material failure in the initial stages of model development. When a working implicit model is obtained, additional analyses with material failure active may be performed. Some element techniques (for example DES/DEM, and ALE) are not supported in implicit analysis.
- Switch control cards to the appropriate implicit settings. See Table 1 for an overview.
- Proper boundary conditions, that suppress rigid body modes, are required in implicit static analyses.
- Check that the model is connected as intended. Check tied contacts and boundary conditions. Use `*CONTROL_IMPLICIT_EIGENVALUE` to perform an eigenvalue analysis and look for eigenvalues close to zero. Inspect the eigenmodes in the `d3eigv` files. This is a very visual way of identifying parts that are not connected. An option for unconnected models may be to use implicit dynamics (see Section 4.9).
- Parts or assemblies connected through joints (`*CONSTRAINED_JOINT_{TYPE}`) may cause rigid body modes or mechanisms. Additional constraints, or adding a joint stiffness (`*CONSTRAINED_JOINT_STIFFNESS_{OPTION}`) may be a remedy. From R11, there are also options on `*CONTROL_RIGID` to add joint stiffness globally to all joints in a model. In the same way as with material failure, it is recommended to disable joint failure (switch `*CONSTRAINED_JOINT_{TYPE}_FAILURE` to `*CONSTRAINED_JOINT_{TYPE}`) at least in the initial model development phase.
- Switch contacts to Mortar, see Section 6.1. Preferably also switch rigid walls to proper meshed parts with `*MAT_RIGID` and Mortar contact, see Section 6.1.6.
- Under-integrated elements (shell elform 2, solid elform 1 etc.) can be used also for implicit analysis. Just like in the explicit analysis, hourglass control is then required for these element formulations. In most cases, the hourglass control can be kept from the explicit setup (hourglass control type will be automatically switched to a type which is valid in implicit analysis). Still, in many cases it might be worth switching to fully integrated elements, see Table 4 for an overview.

Should convergence problems arise, see Appendix C for some troubleshooting tips. Don't hesitate to contact your LS-DYNA supplier for further help with converting explicit models for implicit analyses.

Appendix F

16 Implicit-explicit switching

By implicit/explicit switching, the implicit solver can get help from the explicit solver when convergence is difficult (for example du to material failure), by automatic switching, or the implicit solver can be used during a pre-load phase followed by a rapid event (for example a crash analysis or a blast loading) or during a “post-load” phase for computing the elastic springback in for example forming analyses.

Explicit dynamic relaxation (for example bolt pre-tensioning or press-fit) followed by implicit analysis is another example of implicit to explicit switching.

Switching an analysis entirely to use explicit time-integration (See Section 14) may also be a troubleshooting tool. Visual inspection of results from an explicit analysis may give clues about convergence issues in implicit analysis, for example if the wrong unit system is used when calculating the loading.

An overview of some different possibilities for implicit/explicit switching is shown in Figure 78.

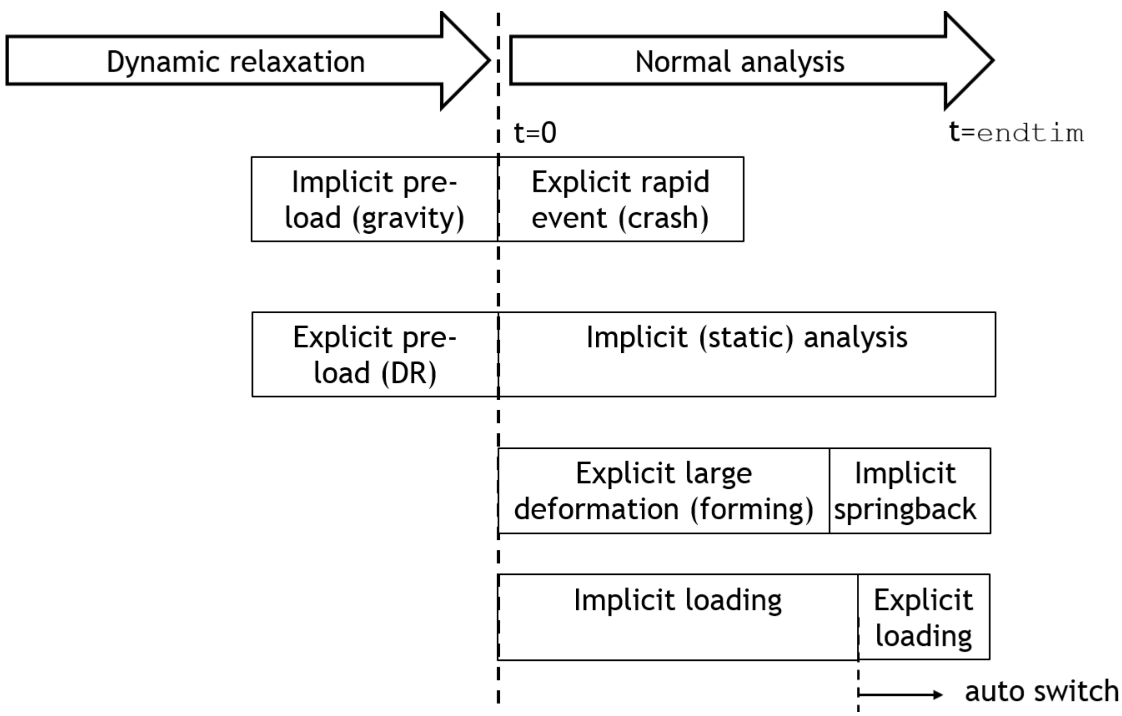


Figure 78. Overview of some possibilities for implicit to explicit switching.

When involving the explicit solver as a part of the solution, it is suggested to use stiffness damping (see Section 14.5) to reduce high-frequency numerical noise, and also time step control (mass scaling) see Section 14.1. It is also important to remember that the explicit solver always involves dynamic effects, as a function of physical time. For more details on how to set up different types of explicit analyses, see Ref. [22]. A review of different options for implicit/explicit switching is presented in the Webinar [38].

16.1 Dynamic relaxation

Dynamic relaxation in LS-DYNA is a possibility for applying pre-loadings (for example gravity or bolt pre-tension) prior to the main loading (for example a rapid event like a crash or a quasi-static ultimate capacity load case). Dynamic relaxation has its own time scale, starting at zero. When the dynamic relaxation phase is completed, and the normal phase begins, the time is reset to zero. Dynamic relaxation can be performed either using the explicit or implicit solver.

Dynamic relaxation in explicit analysis can perhaps be viewed as a fixed-point iterative solution, where a very large amount of damping is applied to obtain an approximate static equilibrium. One application for explicit dynamic relaxation is bolt pre-tensioning followed by implicit statics. This procedure may be very efficient for bolt pre-tensioning, since the rigid body modes of the initially "loose" bolts are handled by the explicit dynamics. It is recommended to minimize gaps between parts that are to be pre-tensioned using explicit dynamic relaxation. To activate explicit dynamic relaxation

- add the keyword `*CONTROL_DYNAMIC_RELAXATION`, and set *`IDRFLG`* = 1,
- set *`SIDR`* = 1 or 2 on any load curve (`*DEFINE_CURVE`),
- and it is also recommended to use time step control (`*CONTROL_TIMESTEP`) and stiffness damping (see Sections 14.1 and 14.5 for further details).

The explicit dynamic relaxation is completed either when the time *`DRTERM`* is reached (`*CONTROL_DYNAMIC_RELAXATION`) or when the ratio of current distortional kinetic energy to peak distortional kinetic energy (the convergence factor) falls within the convergence tolerance (*`DRTOL`*).

Dynamic relaxation can also be performed in implicit analysis:

- Add the keyword `*CONTROL_DYNAMIC_RELAXATION`, and set *`IDRFLG`* = 5, *`NRCYCK`* = 1. Also specify the time when the pre-loading is completed by the *`DRTERM`* parameter (as in any non-linear implicit analysis it is recommended to gradually ramp up the loadings also in implicit dynamic relaxation, to facilitate convergence).
- Set *`SIDR`* = 1 or 2 on any load curve (`*DEFINE_CURVE`),

Note that if required, separate implicit control settings can be specified for the dynamic relaxation phase, using the keywords `*CONTROL_IMPLICIT_AUTO_DYN`, `*CONTROL_IMPLICIT_DYNAMICS_DYN` and `*CONTROL_IMPLICIT_SOLUTION_DYN` (this is not active in the provided control card files).

Output for 3D visualization from the dynamic relaxation phase can be requested by the keyword `*DATABASE_BINARY_D3DRLF` (this is not active in the provided control card files). By this, LS-DYNA will write a sequence of binary databases called `d3drlf` (similar to `d3plot` files).

See also Ref. [18] for an overview of different bolt pre-tensioning techniques, and Ref. [33] for a general overview of dynamic relaxation.

Note that dynamic relaxation cannot be combined with the arc-length method (see Section 4.5) in current versions of LS-DYNA.

16.1.1 Bolt pre-tensioning by explicit dynamic relaxation

This example is similar to the one presented in Section 4.3.2. A bracket is connected to a base plate by four bolts, see the left image of Figure 79. Bolt pre-tensioning is performed using explicit dynamic relaxation. Loading is applied at the free hole of the flange via a distributing coupling (`*CONSTRAINED_INTERPOLATION`). From t = 0 to t = 1, loading in the transverse direction is ramped up to 1 kN and then ramped down to zero at t = 2, and then from t = 2 to t = 3 loading in the longitudinal direction is ramped up to 1 kN and then down to zero at t = 4 (termination time). The example keyword file is `bolts002.key`. The right image of Figure 79 shows the initial stress distribution of the normal phase (which is the final result of the explicit dynamic relaxation phase). The section forces in the bolts are show in Figure 80. The initial drop (by 1.7 % in this case) in section force is due to the approximate character of the solution obtained by the explicit dynamic relaxation.

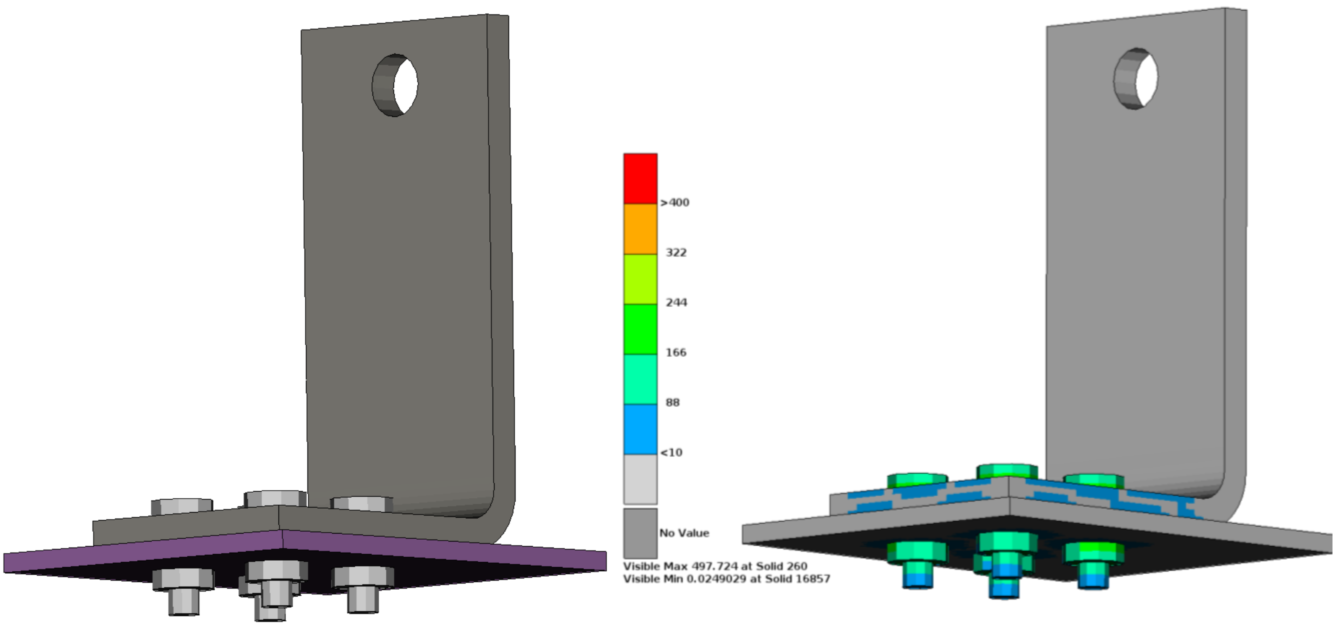


Figure 79. The left image shows the geometry of the bracket (gray), base plate (purple) and bolts (silver). The right image shows the initial effective stress distribution (at t = 0 of the normal phase, after explicit dynamic relaxation).

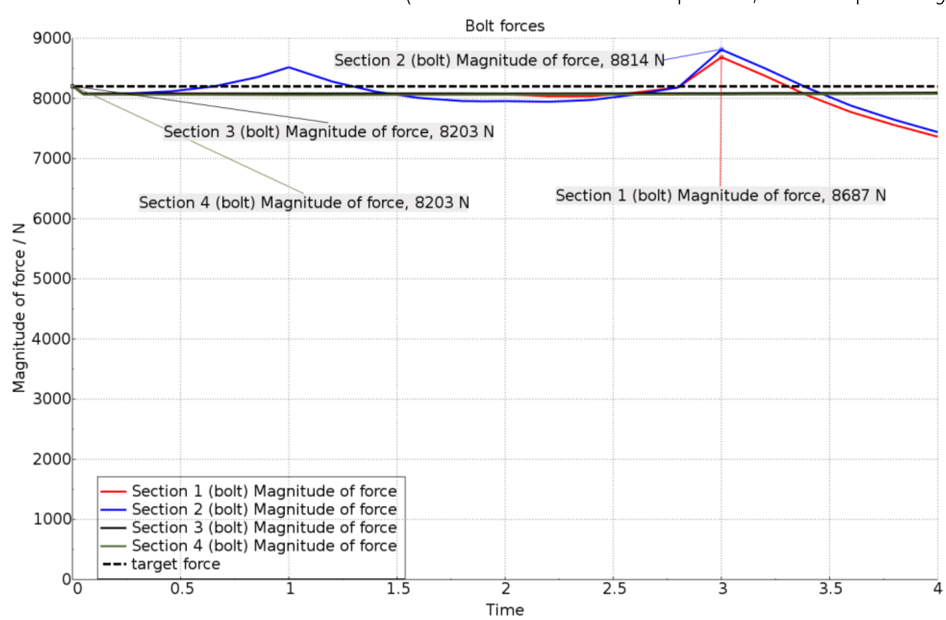


Figure 80. Cross section forces in the bolts as a function of time. The initial drop in section force indicates the approximate character of the explicit dynamic relaxation solution.

16.2 Manual implicit/explicit switching

The user can specify implicit/explicit switching by a load curve id (`*DEFINE_CURVE`) on the keyword `*CONTROL_IMPLICIT_GENERAL`. This is done by setting *IMASS* = -*LCID* (load curve id). An abscissa value > 0 will give an implicit solution, while a zero value gives explicit. A brief template follows:

```
*INCLUDE
control_cards_nonlin.key
*CONTROL_IMPLICIT_GENERAL
$   IMFLAG       DT0     IMFORM       NSBS        IGS      CNSTN       FORM
  -1300000      0.10          2          0          2
*DEFINE_CURVE_TITLE
Implicit/explicit switching
   1300000
0.,1.
1.,0.
4.,0.
4.0001,1.
8.,0.
10.,0.
```

This means that the analysis will start out as implicit, and switch to explicit at t = 1 (when the curve value goes to zero) and use explicit time-integration up to t = 4, and then switch back to using implicit time-integration until t = 8. A converged implicit state will be obtained (unless convergence fails) at the switch times from implicit to explicit (at t = 1.0 and t = 8.0 in this example). If a distinct switch from explicit to implicit, for example at t_1, is desired, use a steep slope of the curve, from (t_1, 0) to from (t_1+ ε, 1).

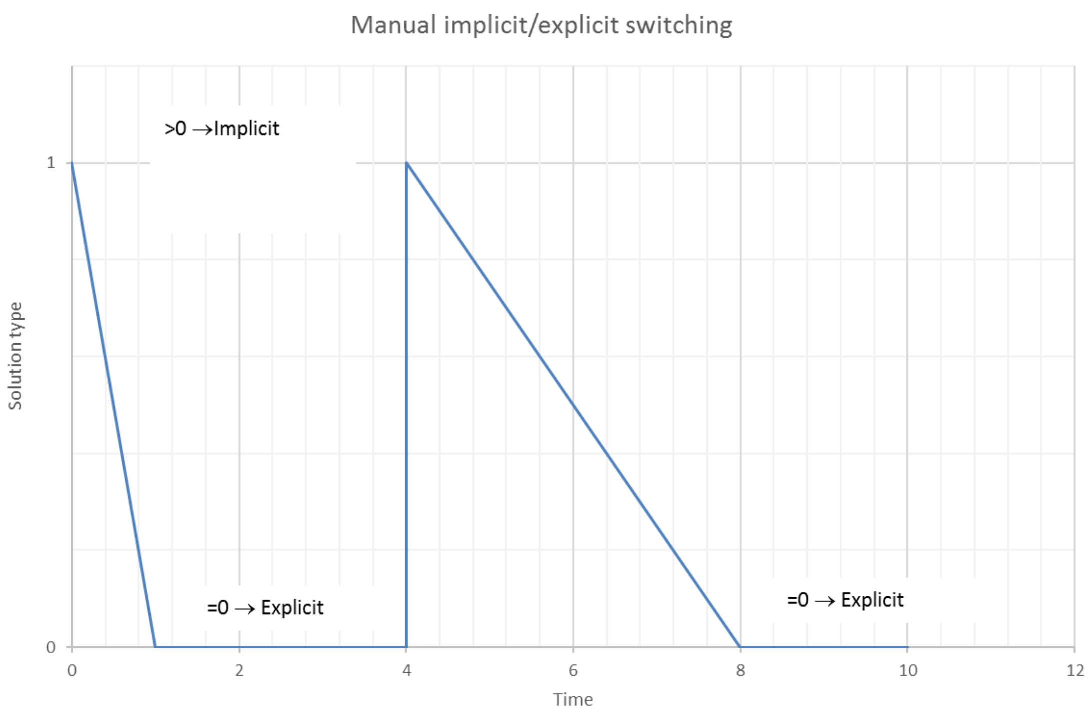


Figure 81. Example of load curve for specifying implicit/explicit switching manually.

16.2.1 Implicit bolt pre-tensioning followed by explicit impact analysis

An example is attached in the `Examples/implexpl/all_in_one` directory of the Guideline package. A pipe assembly (similar to the one used in Sections 12.1.2 and 12.1.4), see Figure 82, is subjected to a load history consisting of two load stages:

1. Bolt pre-tensioning in implicit analysis, for the first 10 ms, with static finish.
2. Impact with a block in explicit analysis, for the following 10 ms.

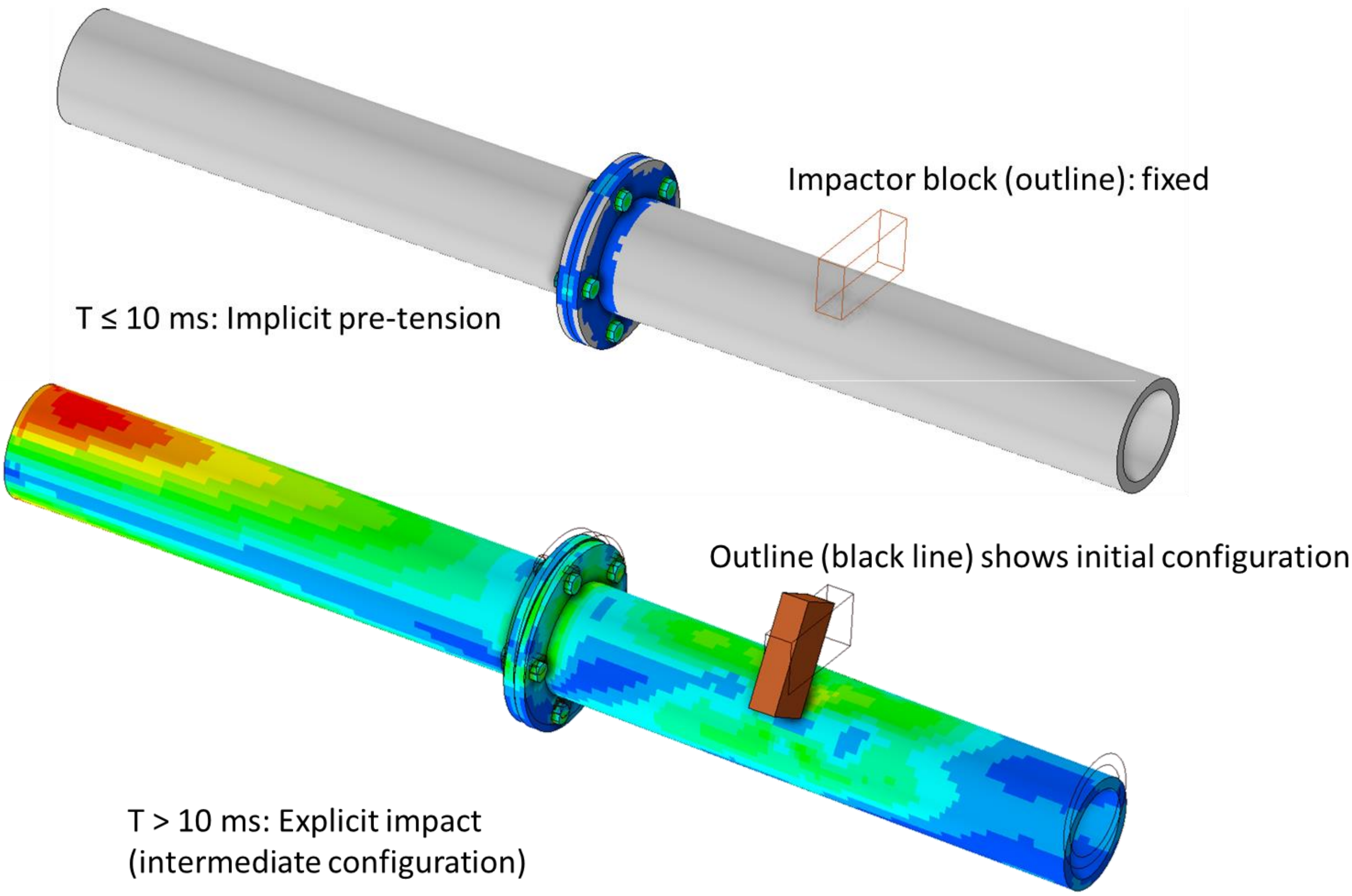


Figure 82. Manual implicit to explicit switching using a load curve. Bolt pre-tensioning is run in implicit analysis, while the impact is run in explicit analysis. The fringe colors show the effective stress from 10 MPa (blue) to 500 MPa (red).

The curve used in `*CONTROL_IMPLICIT_GENERAL` for manually switching from implicit to explicit is shown in Figure 83.

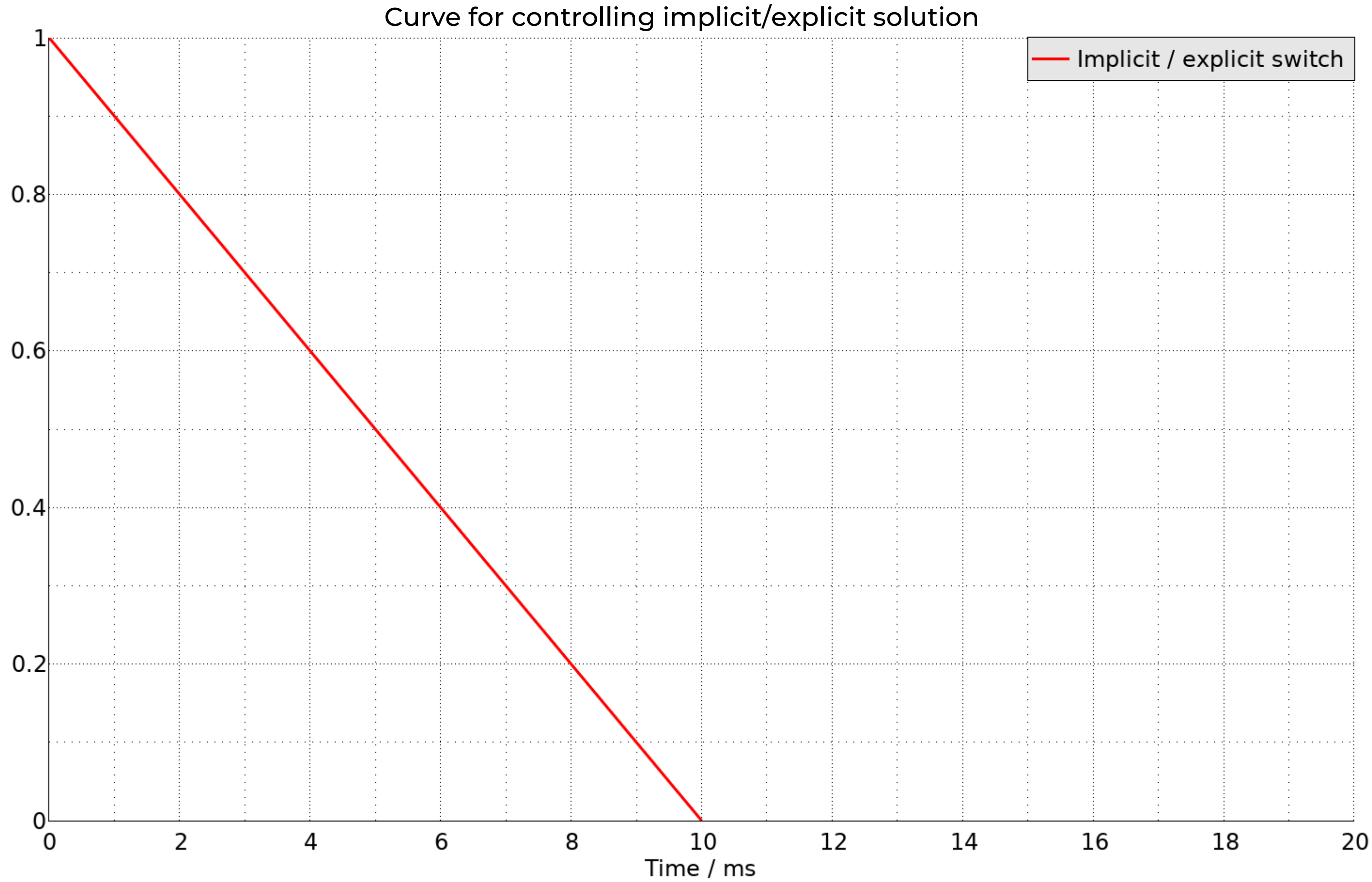


Figure 83. Curve for manually switching between implicit and explicit analysis.

16.2.2 Manual implicit/explicit switching using *CASE and dynain.lsda

The use of `*CASE` and `dynain.lsda`, see Section 12.2.1, makes it possible to run the explicit and implicit stages as separate analyses in a sequence. This, in turn, makes it possible to use different control cards and output requests for implicit and explicit stages. Even element formulations may in some situations, depending on the loading, be switched to under-integrated for the explicit stages, possibly saving some computational cost. In order to keep contact state and pressure consistent between implicit and explicit stages using this approach, Mortar contacts should be used throughout the analysis. A keyword template, using pseudo-code, follows:

```
*KEYWORD
*CASE_BEGIN_1
*INCLUDE
geo001_parts_sections.key
*INCLUDE
geo001_node_elements.key
*INCLUDE
control_cards_nonlin.key
*INCLUDE
database_cards_static.key
*CONTROL_TERMINATION
Define end time of the implicit stage
*CONTACT_AUTOMATIC_SINGLE_SURFACE_MORTAR_ID
       101Global contact
...
```

```
*LOAD_...
Define nodal loads etc.
*BOUNDARY_...
Data line to prescribe boundary conditions
*INTERFACE_SPRINGBACK_LSDYNA
Parameters for requesting a dynain.lsda file
*TITLE
Implicit stage
*CASE_END_1
*CASE_BEGIN_2
*INCLUDE
geo001_parts_underintegrated_sections.key
*INCLUDE
case1.dynain.lsda
*INCLUDE
control_database.k
*PARAMETR
Rdbparu, 1.E-3
*INCLUDE
database_cards_expl.keye
*CONTROL_TERMINATION
Define end time of the explicit stage
*CONTROL_TIMESTEP
Explicit time-stepping settings
*CONTACT_AUTOMATIC_SINGLE_SURFACE_MORTAR_ID
       101Global contact
...
*LOAD_...
Define nodal loads etc.
*BOUNDARY_...
Data line to prescribe boundary conditions
*INTERFACE_SPRINGBACK_LSDYNA
Parameters for requesting a dynain.lsda file
*TITLE
Explicit stage
*CASE_END_2
*END
```

An example is attached in the `Examples/implexpl/multistage` folder of the Guideline package. Note that the attached example requires LS-DYNA R13.1 or later to run. A pipe assembly, (similar to the one used in Sections 12.1.2 and 12.1.4), see Figure 82, is subjected to a load history consisting of two load stages:

- Stage 1: Bolt pre-tensioning in implicit analysis,
- Stage 2: Block impact in explicit analysis.

This is the same geometry and loading as the all-in-one analysis of Section 16.2.1, see Figure 82. Also, the obtained results are very similar, and therefore not presented here.

In this case, fully integrated element formulations are kept throughout all stages, to avoid loss of pre-load force.

16.3 Manual implicit/explicit switching using full restart

From R14.1 of LS-DYNA it is possible to use full deck restart (see Section 12.1.3) for switching from implicit to explicit analysis. In the full restart analysis, control card settings can be switched and adjusted for explicit analyses. An example is attached in the `Examples/implexpl/restart` folder of the Guideline package. A pipe assembly (similar to the one used in Sections 12.1.2 and 12.1.4), see Figure 82, is subjected to a load history consisting of two load stages:

- First analysis (`run_pretens.key`): Bolt pre-tensioning in implicit analysis, for 10 ms
- Full restart analysis (`run_impact.key`): Block impact in explicit analysis, for 10 ms. In the full restart analysis, control cards are replaced with the recommended settings of The Explicit Technical Guide [22].

This is the same geometry and loading as the all-in-one analysis of Section 16.2.1, see Figure 82. Also, the obtained results are very similar, and therefore not presented here.

In this case, fully integrated element formulations and the Mortar contact are kept also in the explicit restart analysis, to avoid loss of pre-load force. The cross-sectional forces in the bolts are shown in Figure 84.

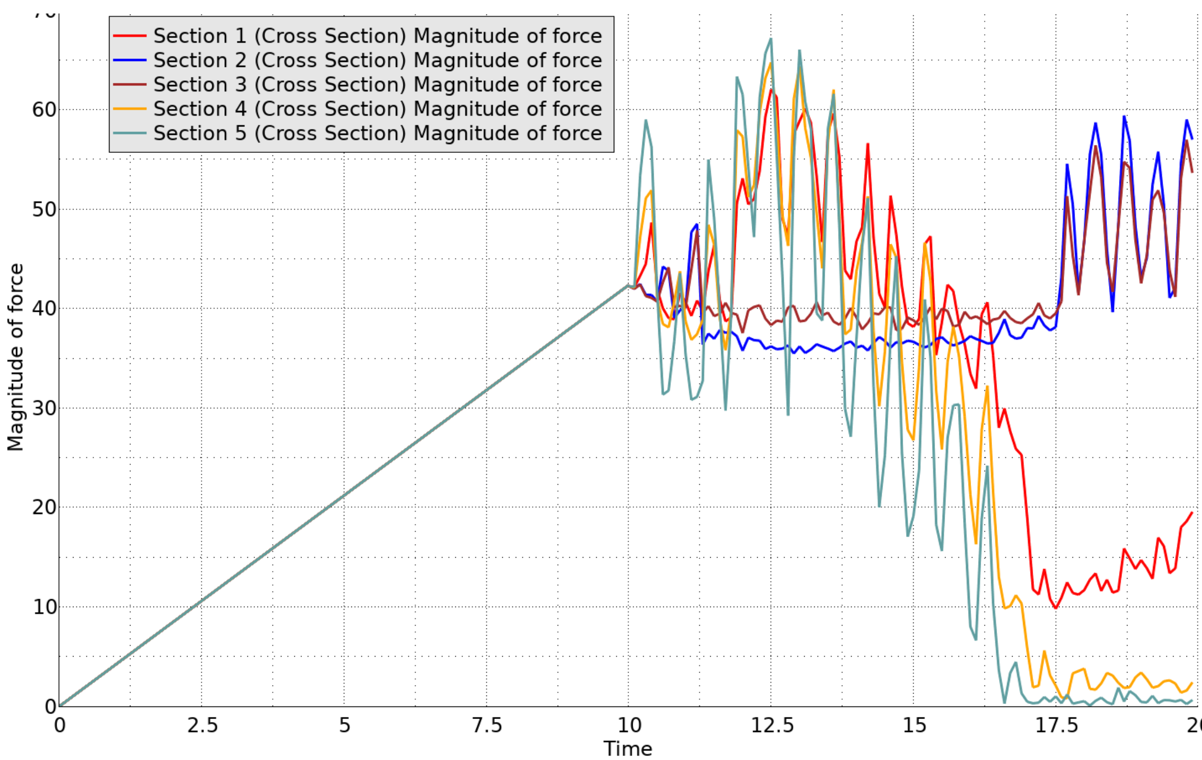


Figure 84. Bolt force time histories. The first 10 ms are from the implicit pre-loading, while results from 10 to 20 ms are from the explicit "full restart" simulation.

16.4 Automatic implicit to explicit switching

By automatic implicit to explicit switching, the implicit solver can get help from the explicit solver to (possibly) overcome convergence difficulties that may occur during the analysis. When the implicit convergence fails, LS-DYNA will switch to explicit and run a specified duration using explicit dynamics, and then switch back to the implicit solver again. In this case, it is important to remember that the explicit solver always involves dynamics, as a function of physical time. This implies that the entire problem should be formulated in physical time, even if it starts out as purely static implicit analysis (where normally the time scale is arbitrary and can be seen as a load factor). For example, if the kg–mm-ms unit system is used, a load application time of say 100 ms could be used, rather than using a ramp from 0 to 1 (as in a purely static analysis). It is also recommended to add keywords for time step control (`*CONTROL_TIMESTEP`) and stiffness damping (see Sections 15.1 and 15.5 for further details) for the explicit phases of the simulation.

To activate automatic implicit/explicit switching:

- Set *IMFLAG* = 4 or 5 on `*CONTROL_IMPLICIT_GENERAL` (5 means mandatory implicit finish)
- Specify the duration to run in explicit dynamics, by the parameter *DTEXP* on `*CONTROL_IMPLICIT_AUTO`.
- Set *KFAIL* on `*CONTROL_IMPLICIT_AUTO` to specify how many failed attempts ("cut-backs", search for RETRY in the `d3hsp` file) to converge are acceptable before the solution type is switched to explicit.

If only one switch from implicit to explicit is desired, set *IMFLAG* = 4 on `*CONTROL_IMPLICIT_GENERAL` and *DTEXP* > termination time on `*CONTROL_IMPLICIT_AUTO`. A template for this case follows:

```
...
*INCLUDE
control_cards_nonlin.key
*CONTROL_TIMESTEP
Specify dt2ms for the explicit part of the solution
*CONTROL_TERMINATION
      100.
*CONTROL_IMPLICIT_GENERAL
$    IMFLAG       DT0    IMFORM      NSBS       IGS     CNSTN      FORM
          4      0.10         2         0         1         0         0
*CONTROL_IMPLICIT_AUTO
$     IAUTO    ITEOPT    ITEWIN     DTMIN     DTMAX     DTEXP     KFAIL
          1       100        20        0.     -700.      500.         3
```

One application of the one-switch approach may be force-controlled limit load analysis, where the explicit solver can take over and explore the structural behavior after the peak load carrying capacity is reached (where static implicit equilibrium fails).

Note that automatic switching from implicit solution to explicit solution and back to implicit solution again may be problematic if the arc-length solver is active. For this switching procedure to work properly, set *ARCMTH* = 0 on `*CONTROL_IMPLICIT_SOLUTION`.

16.5 Explicit analysis with intermittent eigenfrequency analyses

Another possibility for implicit to explicit switching is to combine an explicit analysis with intermittent eigenfrequency analyses (compare Section 4.6.3). The eigenfrequency analyses will be performed on a linearization of the current configurations, which means that effects of loading, pre-tensioning and (linearized) contacts will be included. This procedure is activated by adding

*CONTROL_IMPLICIT_GENERAL with *IMFLAG* = 6 and *CONTROL_IMPLICIT_EIGENVALUE with *NEIG* = -LCID to the original explicit run file. A template follows:

```
.
.
*CONTROL_TERMINATION
101.,
*CONTROL_IMPLICIT_GENERAL
6,
*CONTROL_IMPLICIT_EIGENVALUE
-702,
Optionally set MDSTRS = 1 for calculation of modal stresses
*DEFINE_CURVE_TITLE
Implicit eigenvalues
702,
10.,150
100.,150
.
.
```

Note that if an eigenfrequency analysis of the final configuration (at termination time) is desired, it is recommended to specify a time slightly before the termination time or extend termination time slightly beyond the time for the last eigenfrequency analysis.

16.6 A note on output when using implicit to explicit switching

For an implicit static analysis, it is often desired to get a `d3plot` database for each converged step, since normally a limited number of steps is expected (often << 100). Also, it is possible afterwards to delete the `d3plot` files corresponding to “irrelevant” intermediate steps, in order to save disk space. This motivates the very high output frequency that is specified in the database requests of `database_cards_static.key`. However, when going to explicit, miniscule time steps are used, which may lead to enormous amounts of output databases. When using implicit to explicit switching, it is instead recommended to use the include file `database_cards_expl.key` and to define the output interval using the parameter `dbparu`. A template follows:

```
*PARAMETER
Rdbparu, <output time step>
*INCLUDE
database_cards_expl.key
```

Alternatively, load curves may be used for the `*DATABASE_...` – requests, in order to specify the output interval as a function of problem time throughout the simulation.

Appendix G

17 Some comments on control card settings

In this Appendix, some comments to the control card settings of the appended keyword files `control_cards_...key`, see Table 1, are given. This Appendix will be extended in coming revisions of this Guideline.

The most important control cards specific to implicit analyses are:

- `*CONTROL_IMPLICIT_GENERAL`, where the variable *IMFLAG* (set to a non-zero value) activates the implicit solution scheme (explicit dynamics is the default in LS-DYNA). The variable *DT0* sets the initial time-step size, in combination with what is specified on the automatic time stepping control (`*CONTROL_IMPLICIT_AUTO`) via Curve ID 700 in the present implementation. The initial time-step size will be

  min(*DT0*, value of curve ID 700 at t = 0.)

  The variables *IMFORM* and *NSBS* are related to springback analyses, a sheet metal forming application. An important variable is *IGS*, which controls how the geometric (or stress stiffening) contributions to the stiffness matrix is handled. The stiffness matrix ***K*** is composed of a material and geometric part [4]

  $$\boldsymbol{K} = \boldsymbol{K}_{mat} + \boldsymbol{K}_{geo}$$

  where $\boldsymbol{K}_{mat}$, which is always considered, depends on the material properties, and is often positive definite. The geometric stiffness $\boldsymbol{K}_{geo}$ depends on the stresses, which means that compressive stresses may lead to negative contributions to the stiffness matrix. By considering both material and geometrical contributions, the analysis will sense buckling, bifurcations and instabilities by an indefinite or singular stiffness matrix, but for a general non-linear simulation, these indefinities may deteriorate convergence. By default, the geometrical contribution $\boldsymbol{K}_{geo}$ is neglected, but it can be included by setting `IGS = 1`. A keyword template follows:

```
*CONTROL_IMPLICIT_GENERAL
$    IMFLAG         DT0     IMFORM        NSBS         IGS       CNSTN        FORM
          1        0.10          2           0           1           0           0
```

From R12.0.0 of LS-DYNA, new options for treating the geometric stiffness are available. By setting *IGS* = -1 * *PSID*, the geometric stiffness is only applied to the parts included in the part set *PSID*. Negative principal stresses in the geometric stiffness are neglected by setting *DT0* < 0. The absolute value |*DT0*| is used when calculating the initial time-step size.

- `*CONTROL_IMPLICIT_SOLUTION`, where the variable *NSOLVR* specifies the solution type (12 for non-linear or ±1 for linear, see Section 4.1). The values of *ILIMIT* and *MAXREF* will determine the type of non-linear solution scheme applied by LS-DYNA. In the default BFGS (quasi-Newton) method, the stiffness matrix is reformed every *ILIMIT* iterations. In between reformations, a low-rank stiffness update is applied. By setting `ILIMIT = 1`, a stiffness reformation is performed in every iteration, which is equivalent to the full Newton[27] method. *MAXREF* is the maximum number of stiffness reformations before a step is abandoned (and attempted again with a smaller time

[27] With line search

increment).

The variables *DCTOL*, *ECTOL*, *ABSTOL*, *RCTOL*, *LSTOL* control the convergence criteria for the non-linear solution. The variable *DCTOL* gives a maximum value of the relative displacement norm, and the variable *ECTOL* gives a maximum value on the relative energy norm. A step counts as converged when the relative displacement norm is within *DCTOL*, and the relative energy norm is within *ECTOL*. In practice, it is almost always the relative displacement norm requirement that is the hardest to fulfill. The variable *ABSTOL* can be useful for driving convergence based on the absolute value of the force residual. If for example a maximum force residual of *X* Newton is acceptable, set *ABSTOL* = −*X*, in case the unit system of the model corresponds to Newton for forces. In this case, convergence will be detected if the force residual is below *X* Newton, or both the relative displacement and energy norm requirements are met. Otherwise, to avoid premature convergence the recommended value is *ABSTOL* = 1.E-20.

The variable *DNORM* determines how to calculate the relative displacement norm. With the LS-DYNA default value *DNORM* = 2, the total displacement is used as a reference. This implies that convergence will be easier as large displacements accumulate during a simulation. For most applications, the option *DNORM* = 1 is recommended, which means that the displacement over the current step is used as a reference for the relative displacement norm.

The variables *DIVERG* and *ISTIF* control how and when the stiffness matrix should be reformed. The default value (*DIVERG* = 1 means that the stiffness matrix is reformed when the force residual increases, and *ISTIF* = 1 means that the stiffness matrix is reformed at the start of each step) are recommended.

The variables *NLPRINT* and *D3ITCTL* control additional output from the non-linear solver. By the recommended setting *NLPRINT*=3, additional information regarding convergence and line search progress is printed in the `d3hsp` and `mes0*` files, which is useful for tracking the convergence (and potentially troubleshooting convergence issues). By setting *D3ITCTL* > 0, LS-DYNA will output `d3iter` files, a binary database for 3D visualization (use LS-PrePost) also of the non-converged iterations. The `d3iter` files can be very useful for troubleshooting convergence issues.

By setting the variable *ARCMETH* = 3, the arc length solver is activated. By this, the structural behavior can be determined using prescribed force, even after the peak load is exceeded. The problem time will correspond to a load factor, which will also be treated as an unknown for LS-DYNA to solve. After the peak load is reached, a negative time step means that the force is decreased as the structure continues to deform. The problem time can even become negative, which corresponds to a complete load reversal. See Section 4.5 and for example Ref. [26] for details. The variable *ARCTIM* can specify an activation time of the arc length solver. By this, some computational cost can be saved on the way to the peak force (in case a decent estimate is available beforehand) by running the initial part as a standard implicit analysis.

The variable *LSMTD* determines which line search method is used. Line search in this context means a way of determining the size of the region of validity of the local linear approximation at the current iteration. A conservative, and often useful, approximation is normally given by setting *LSMTD* = 5. In some cases, this estimate may be too conservative, then choosing a more aggressive line search by *LSMTD* = 4 may aid convergence.

For rotating structures, such as turbines or gears, activating a curved line search by setting *LSDIR* = 4 and *IRAD* = 1 may aid convergence.

A keyword template follows:

```
*CONTROL_IMPLICIT_SOLUTION
$#  nsolvr    ilimit    maxref     dctol     ectol     rctol     lstol    abstol
        12         1        65     1.E-3     1.E-2                       1.E-20
$#   dnorm    diverg     istif   nlprint    nlnorm   d3itctl
         1         1         1         3         4         1
$#  arcctl    arcdir    arclen    arcmth    arcdmp    arcpsi    arcalf    arctim

$#   lsmtd     lsdir      irad
         5
```

- `*CONTROL_IMPLICIT_DYNAMICS`, for activating the dynamic terms of the implicit solution. (The default is a static solution). See also Section 4.8. Implicit dynamics can be useful for relatively slow or long duration dynamic events. Fast, or relatively short dynamic events, like drop tests or crash analyses, are better handled by the explicit solver of LS-DYNA (see also Appendix D).
  A very useful application of implicit dynamics is to handle rigid body modes in a model; the typical example is an assembly which is to be connected by bolts. Initially, such an assembly has many potential rigid body modes, as many parts of the model are “loose”, but when the bolt pre-tensioning is applied, the model becomes connected via the contacts and bolt pre-loads. Such an analysis in LS-DYNA is typically run as initially dynamic, and when contacts are established, the dynamic contributions can be ramped down and the solution may continue as purely static.

  Implicit dynamics is activated by setting the variable *IMASS* = 1. The variables *GAMMA*, *BETA* and *ALPHA* control the numerical time integration. For very slow dynamic events, or when relatively strong numerical damping is desired, the values *GAMMA* = 0.60, *BETA* = 0.38 can be used. If the main objective is a detailed dynamic analysis, the values *GAMMA* = 0.55, *BETA* = 0.27563 can be used. For rotating structures, or long duration events, setting *ALPHA* = 0.5 is recommended.

  In implicit analyses, the time step is often too big to get a good estimate of strain rates. The variable *IRATE* can then be used to disable the strain rate effects in material models. By *IRATE* = 1, strain rate effects are disabled in implicit analyses, and by *IRATE* = 2, strain rate effects are disabled in both implicit and explicit analyses. From R11 of LS-DYNA, strain rate effects are disabled by default for implicit static analyses, but can be activated by *IRATE* = -1. Note that setting *IRATE* ≥ 1 in implicit dynamics will cancel the effect of stiffness damping specified by `*DAMPING_PART_STIFFNESS`.
  The keyword contains three variables to control when the dynamic terms should be considered. The variable *TDYBIR* sets the time when dynamic starts. The variable *TDYDTH* sets the time when the removal of dynamic effects begins. Dynamics are then gradually ramped down and completely removed at time *TDYBUR*. Should these options be insufficient, it is also possible to control the dynamics via a curve. Then set *IMASS* = -*LCID*.

  Note that implicit dynamics cannot be combined with the arc length method (see `*CONTROL_IMPLICIT_SOLUTION`) but by use of *ARCTIM*, initially unconnected assemblies can be solved using dynamics until the contacts are established, then the solution is switched to static and then the arc length method is activated.

  A keyword template follows:

```
*CONTROL_IMPLICIT_DYNAMICS
$    IMASS     GAMMA      BETA    TDYBIR    TDYDTH    TDYBUR     IRATE    ALPHA
         1       0.6      0.38                 0.5       0.8         1
```

- `*CONTROL_IMPLICIT_SOLVER`, for setting parameters controlling the behavior of the linear solver. Normally, this keyword can be left out since the default settings seldom need modification.

  The variable *LPRINT* controls the number of print-out messages from the linear solver. Setting *LPRINT* = 3 will provide extensive output regarding, for example automatically applied constraints to rigid body modes (AUTOSPC) and is recommended when troubleshooting convergence issues.

Setting the variable *LCPACK* = 3 activates non-symmetrical matrix treatment. This may aid convergence in the case of high friction (μ > 0.3) and stick-slip conditions, as well as some other highly non-linear analysis types, like cases involving anisotropic material models, but it will also double the required amount of memory for storing matrices. The non-symmetrical solver cannot be used with inertia relief (*CONTROL_IMPLICIT_INERTIA_RELIEF).

From R11 of LS-DYNA, the variable *RDCMEM* may be of importance. It controls how much of the available memory that LS-DYNA implicit may occupy. The default value of 0.85 (corresponding to a maximum limit of 85 % of the available RAM) normally works well in a Linux environment, if (at most) one simulation is run on each cluster node. If more than one (relatively large) simulation is to be run on the same cluster node, it may be required to reduce *RDCMEM* in order to even out the use of memory resources between the simulations. From R13 of LS-DYNA, the variable *ABSMEM* may be used for putting an absolute limit on the memory used by the LS-DYNA implicit solver. See Refs. [27][36] for details regarding memory management in LS-DYNA.

From R15, some new options to use mixed precision [49] by setting ISINGLE > 0 were introduced. This can potentially save 15 – 50 % of RAM usage.

A keyword template follows:

```
*CONTROL_IMPLICIT_SOLVER
$#   lsolvr    lprint     negev     order      drcm    drcprm   autospc   autotol
                    3
$#   lcpack    mtxdmp    iparm1    rparm1    rparm2
          3
$#   emxdmp    rdcmem    absmem
               0.85
```

- *CONTROL_IMPLICIT_ORDERING, for selecting equation ordering scheme. In most cases this keyword is not required since the default settings work fine. In many cases, involving large-scale models on “many” (>8) cores in mpp, activating ParMetis (available from R14) by setting *ORDER* = 3 can significantly increase performance. Or, as an alternative, the option to use LS-GPart [41] (available from R13) by setting *ORDER* = 4 may be of interest.

Attachments

Filename	Description
database_cards_static.key	Output request for static analyses, R10.2 and R11.X
database_cards_before_R102.key	Output request for static analyses before R10.2
database_cards_dynamic.key	Output request for dynamic analyses
control_cards_linear.key	Control cards for linear static analyses
control_cards_nonlin.key	Control cards for non-linear static analyses
control_cards_nonlin_R15.key	Control cards for non-linear static analyses, with some additional features available from R15 activated
control_cards_arc.key	Control cards for non-linear static analyses using the arc length method
control_thermal_steady.key	Additional control cards for steady-state thermal only analysis
control_thermal_transient.key	Additional control cards for transient thermal analysis
ROPS_linear_stiffness.key	Linear static analysis
bend001.key, bend002.key	Non-linear static analysis
bolts001.key	Non-linear dynamic/static analysis.
buckle001.key	Linear buckling analysis
limitload001.key	Non-linear limit load analysis
eigen001.key	Eigenfrequency analysis
eigen002.key	Intermittent eigenfrequency analysis
transient001.key	Linear transient modal dynamics
transient_nonlin001.key	Non-linear transient dynamics
ssd_fatigue001.key	Steady state dynamics with fatigue evaluation
Set_4.key	Random vibration analysis with fatigue evaluation
sticking_contact.key	Tiebreak contact to make contact surfaces stick once contact is established
run_2d.key	Axisymmetric analysis

thermal.key	Coupled thermal structural analysis
bolts002.key	Bolt pre-tensioning using explicit dynamic relaxation
run_rotodyn.key	Rotational dynamics example
run_pressfit.key	Linear analysis with nonlinear contact, for resolving a press-fit

References

[1] Livermore Software Technology, LS-DYNA keyword user's manual Volume I, Livermore 2023 (see http://lstc.com/download/manuals).
[2] Livermore Software Technology, LS-DYNA keyword user's manual Volume II, Livermore 2023.
[3] Livermore Software Technology, LS-DYNA keyword user's manual Volume III, Livermore 2023.
[4] Borrvall, T., et al., Implicit analysis in LS-DYNA (course material), DYNAmore Nordic AB, Linköping 2019.
[5] Borrvall, T., A heuristic attempt to reduce transverse shear locking in fully integrated hexahedra with poor aspect ratio, 7th European LS-DYNA Users conference, 2009, http://www.dynalook.com/european-conf-2009/G-I-02.pdf
[6] Borrvall, T., Mortar contact algorithm for implicit stamping analyses in LS-DYNA, 10th International LS-DYNA Users conference, 2008, http://www.dynalook.com/international-conf-2008/MetalForming2-3.pdf
[7] Lilja, M., Benchmark of LS-DYNA for offshore applications according to DNV Recommended Practice C208, 13th International LS-DYNA Users conference, 2014.
[8] Huang, Y., Cui, Z., Frequency domain analysis in LS-DYNA, Oasys LS-DYNA UK User's Meeting, 2013-01-16.
[9] Huang, Y., NVH and frequency domain analysis with LS-DYNA (Course material), Ansys, 2025.
[10] Erhart, T. and Schmied, C., Updated review of solid element formulations in LS-DYNA, LS-DYNA Forum 2018, Internet source: http://www.dynamore.de/de/download/papers/forum11/entwicklerforum-2011/erhart.pdf https://www.dynamore.de/de/download/papers/dynamore/de/download/papers/2018-ls-dyna-forum/papers-2018/mittwoch-17.-oktober-2018/simulation-isogeometric-and-fe-technology/a-study-on-the-new-higher-order-solid-elements-in-ls-dyna
[11] Livermore Software Technology Corporation, LS-DYNA Theory manual, Livermore 2023.
[12] Segalman, D. J., et al., An efficient method for calculating RMS von Mises stress in a random vibration environment, Internet source: https://sem.org/wp-content/uploads/2016/01/sem.org-IMAC-XVI-16th-Int-160502-An-Efficient-Method-Calculating-RMS-von-Mises-Stress-Random.pdf
[13] Forsberg, J., LS-DYNA recommended settings explicit analyses, DYNAmore Nordic Document, 2016-06-17
[14] Haufe, A., et al., Properties & limits: Review of shell element formulations, Internet source: https://www.dynamore.de/de/download/papers/2013-ls-dyna-forum/documents/review-of-shell-element-formulations-in-ls-dyna-properties-limits-advantages-disadvantages
[15] Li, L., et al., Introduction of rotor dynamics using implicit method in LS-DYNA, 14th international LS-DYNA Users conference, 2016, http://www.dynalook.com/14th-international-ls-dyna-conference/simulation/introduction-of-rotor-dynamics-using-implicit-method-in-ls-dyna-r
[16] Fyhrman, D., 2D-simulations in LS-DYNA, 2018, Webinar available from files.dynamore.se > Client Area.
[17] Gromer, A., Erhardt, T., and Borrvall, T., LS-DYNA implicit Workshop – nonlinear Solver, Bamberg 2016. Internet source: https://www.dynamore.de/de/download/papers/2016-ls-dyna-forum/Papers%202016/dienstag-11.10.16/workshops/tips-und-tricks-in-ls-dyna-implicit
[18] Karajan, N., et al., Modelling bolts in LS-DYNA using explicit and implicit time integration, 15th International LS-DYNA users conference, Dearborn 2018, internet source: https://www.dynalook.com/15th-international-ls-dyna-conference/implicit/modeling-bolts-in-ls-dyna-c-using-explicit-and-implicit-time-integration
[19] Borrvall, T., et al., Solution explorer in LS-PrePost – a GUI for non-linear implicit FE, 12th European LS-DYNA Conference, Koblenz 2019, internet source: https://www.dynalook.com/conferences/12th-european-ls-dyna-conference-2019/workshops/workshop_solution_explorer_in_ls_prepost.pdf
[20] Cui, Z., Hunag, Y., and Grimes, R., Discussion on NVH analyses with various eigensolvers in LS-DYNA, 15th International LS-DYNA conference, Dearborn 2018, internet source: https://www.dynalook.com/conferences/15th-international-ls-dyna-conference/nvh/discussion-on-nvh-analysis-with-various-eigensolvers-in-ls-dyna
[21] Borrvall, T., Rubin MR, and Jabareen M., Cosserat Point elements in LS-DYNA, LS-DYNA User's forum 2013, internet source: http://www.dynamore.de\de\download\papers\2013-ls-dyna-forum\documents\cosserat-point-elements-in-ls-dyna
[22] Engstrand, K., LS-DYNA Explicit Technical Guide – Reference models for explicit analyses using LS-DYNA (v 1.5), DYNAmore Nordic 2024 (internet source: https://lsdyna.ansys.com/knowledge-base/general/).

[23] Bernhardsson, A., Damping in explicit LS-DYNA, DYNAmore Nordic document (Webinar 2019-12-10) available from files.dynamore.se > Client Area.
[24] Lilja, M., Troubleshooting convergence problems in LS-DYNA implicit, DYNAmore Nordic document (Webinar 2021-12-16) available from files.dynamore.se > Client Area.
[25] Aspenberg, D., Model checking in LS-DYNA – from pre to post simulation, Webinar, DYNAmore Nordic AB, 2020.
[26] Jonsson, A., Ultimate capacity analyses in LS-DYNA, Webinar, DYNAmore Nordic AB, 2017.
[27] Engstrand, K., and Jonsson, A., LS-DYNA Memory management, Webinar, DYNAmore Nordic AB, 2022.
[28] Zienkiewicz, O. C. and Zhu, J. Z., The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique, Int. J. Numerical Methods in Engineering, 33:7 (1992), 1331 – 1364
[29] Jonsson, A., Pre-tensioning techniques in LS-DYNA, DYNAmore Nordic document (Webinar 2021-01-28) available from files.dynamore.se > Client Area.
[30] Engstrand, K., and Jonsson, A., A brief review of the continuation of analyses in LS-DYNA using restart, dynain-file and *CASE, Dynamore Nordic document (Webinar 2020-03-10) available from files.dynamore.se > Client Area.
[31] Huang, Y., Cui, Z., New options in frequency domain analysis and fatigue analysis with LS-DYNA, 12th European LS-DYNA Conference, Koblenz 2019 (internet source: https://www.dynalook.com/conferences/12th-european-ls-dyna-conference-2019/implicit/huang_lstc-paper.pdf/view)
[32] Noh, G., and Bathe, K.-J., For direct time integration: A comparison of the Newmark and ρ∞-Bathe schemes, Computes and Structures 225 (2019)
[33] Hallén, A., Pre-tensioning and dynamic relaxation, Dynamore Nordic document (Webinar 2022-12-01) available from files.dynamore.se > Client Area.
[34] Jonsson, A., Load history management using *CASE – with dynain.lsda and Mortar contact, Dynamore Nordic document (Webinar 2021-05-27) available from files.dynamore.se > Client Area.
[35] Engstrand, K., Tied and tiebreak contacts in LS-DYNA - Explicit simulation (MPP), Dynamore Nordic document (Webinar 2022-03-10) available from files.dynamore.se > Client Area.
[36] Jonsson, A., Memory management in LS-DYNA implicit, 2022-05-16, available from files.dynamore.se > Client Area.
[37] Schill, M., and Jonsson, A., Viscoelasticity in LS-DYNA, DYNAmore Nordic document (Webinar 2020-12-10) available from files.dynamore.se > Client Area.
[38] Jonsson, A., Implicit/explicit switching in LS-DYNA, DYNAmore Nordic document (Webinar 2023-03-09) available from files.dynamore.se > Client Area.
[39] Jonsson, A., Review of quadratic shell elements in LS-DYNA implicit, DYNAmore Nordic document (Webinar 2023-05-04) available from files.dynamore.se > Client Area.
[40] Jonsson, A., Contacts for implicit analyses in LS-DYNA, DYNAmore Nordic document (Webinar 2023-04-20) available from files.dynamore.se > Client Area.
[41] Ashcraft, C., et al., Increasing the scale of LS-DYNA, 15th International LS-DYNA Users Conference, Detroit 2018 (Internet source: https://www.dynalook.com/conferences/15th-international-ls-dyna-conference/implicit/increasing-the-scale-of-ls-dyna-r-implicit-analysis/view)
[42] Weckesser, T., and Karajan, N., A study of tied contacts in implicit stress analysis via a fire truck tip-over simulation, 14th European LS-DYNA User conference, Baden-Baden, 2023.
[43] Jonsson, A., Frequency domain analyses in LS-DYNA, DYNAmore Nordic document (Webinar 2022-11-16) available from files.dynamore.se > Client Area.
[44] Bengzon, F., Borrvall, T., and Basu, U., An enhanced assumed strain (EAS) solid element for nonlinear implicit analyses, 15th International LS-DYNA Users Conference, Detroit 2018 (Internet source: an-enhanced-assumed-strain-eas-solid-element-for-nonlinear-implicit-analyses (dynalook.com))
[45] Public Toyota Camry 2012 FE-model, from: https://lsdyna.ansys.com/implicit-roofcrush/, originally from NCAC FE model database (internet source https://www.ccsa.gmu.edu/models/2012-toyota-camry/)
[46] Borrvall, T., et *al.*, Implementation of u-P elements for incompressible hyperelastic materials, North American LS-DYNA Users Forum, Detroit 2023 (internet source https://lsdyna.ansys.com/wp-content/uploads/2023/12/Implementation-of-u-P-elements-for-incompressible-hyperelastic-materials-Thomas-Borrvall-Ansys.pdf)

[47] Larid, G., and Pathy, S., “A roadmap to linear and nonlinear implicit analysis in LS-DYNA”, 11th European LS-DYNA Conference, 2017 (internet source https://lsdyna.ansys.com/wp-content/uploads/attachments/a-roadmap-to-linear-and-nonlinear-implicit-analysis-with-ls-dyna.pdf)
[48] Craig, R.R., and Bampton, M.C.C, Coupling of substructures for dynamic analyses, AIAA Journal 1968, **6** (7), 1313-13139, Internet source: https://hal.science/hal-01537654/file/RCMB.pdf
[49] Lucas, R., et al., Speeding up LS-DYNA implicit with mixed precision, low rang approximations, and accelerators, 14th European LS-DYNA Conference, 2023.
[50] Friedrich, P., et al., Application of model order reduction techniques in LS-DYNA, 11th European LS-DYNA Conference, 2017.