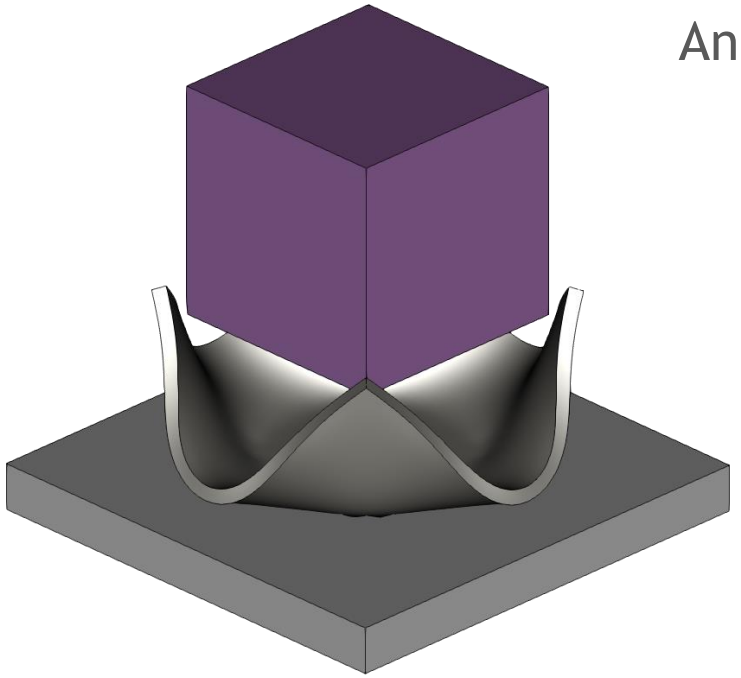
User-defined features in LS-DYNA

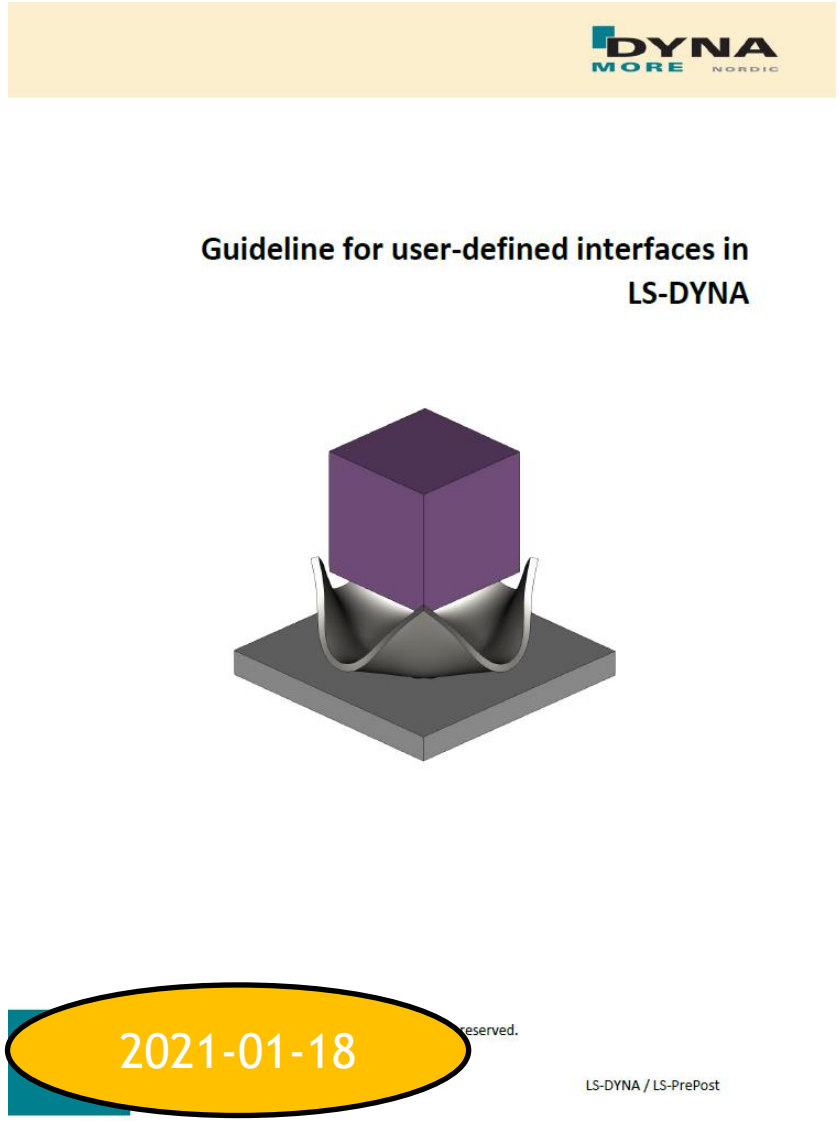


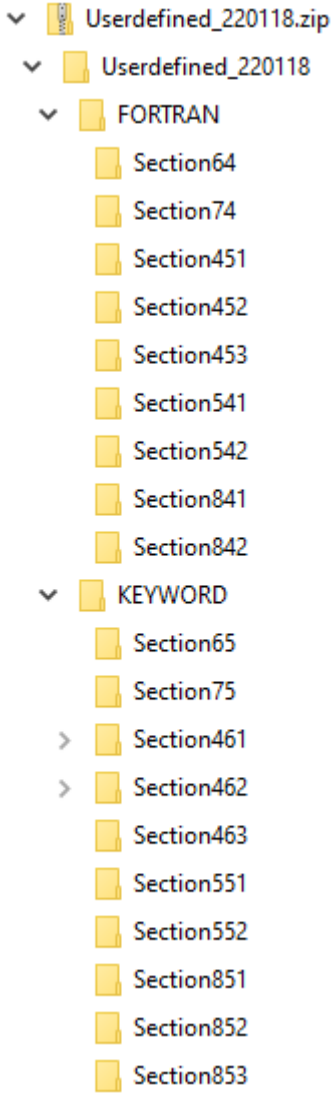
Anders Jonsson, DYNAmore Nordic AB

Overview

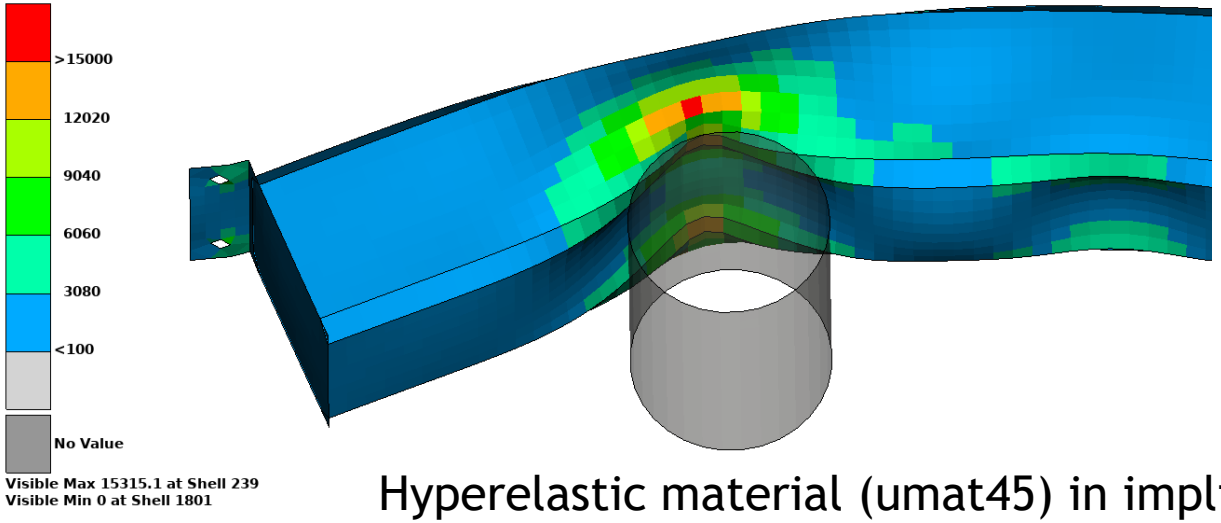
- Guideline for user-defined interfaces in LS-DYNA
- Seminar – 16/6: User defined materials in LS-DYNA
  - Register today!
- Overview of available user-defined interfaces
  - Possibilities for customization and extended functionality
- Getting started with user-defined feature development
  - Download the LS-DYNA usermat package
  - Fortran compilers and environment
  - Fortran programming
- Incorporating user-defined features into LS-DYNA
  - Static vs. dynamic linking

Guideline for user-defined interfaces in LS-DYNA

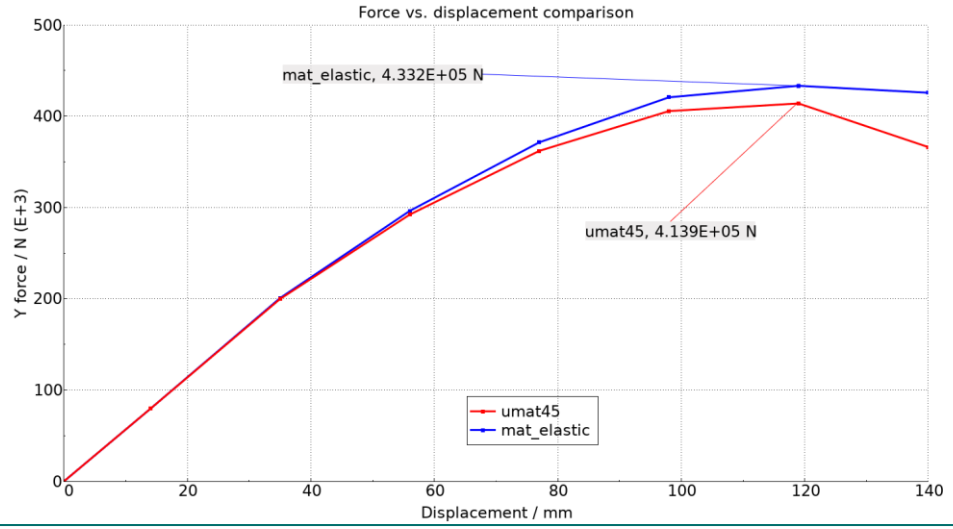


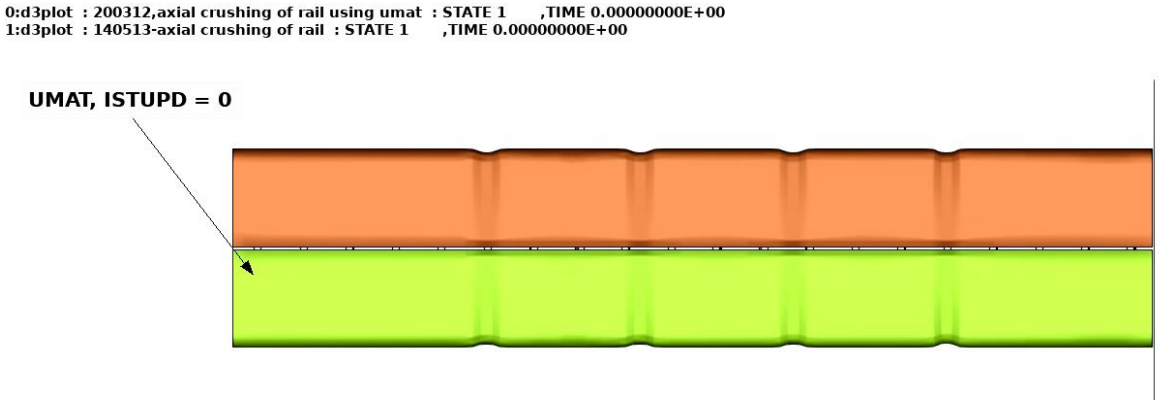


The Guideline contains examples of ...

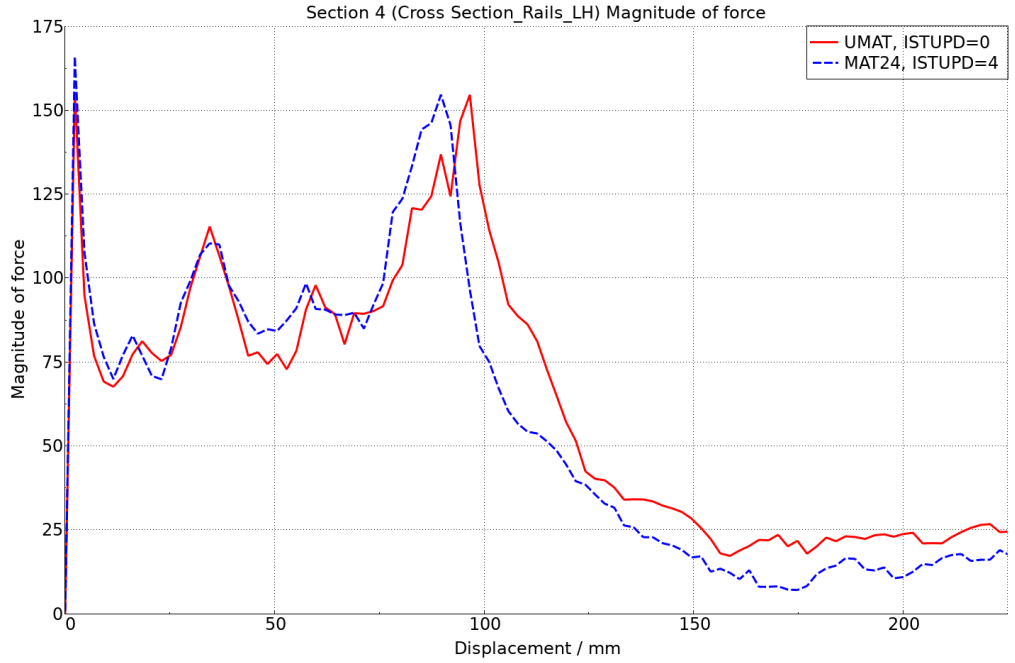


Hyperelastic material (umat45) in implicit



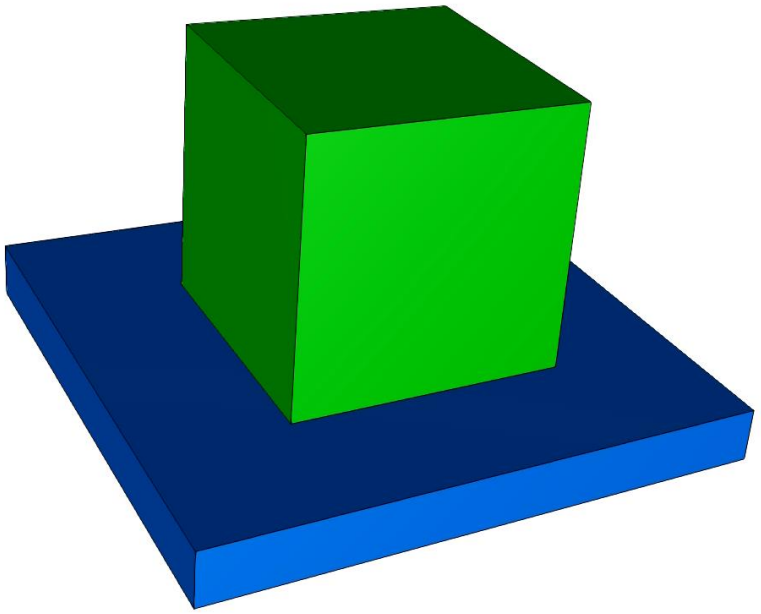


J2 Plasticity in explicit

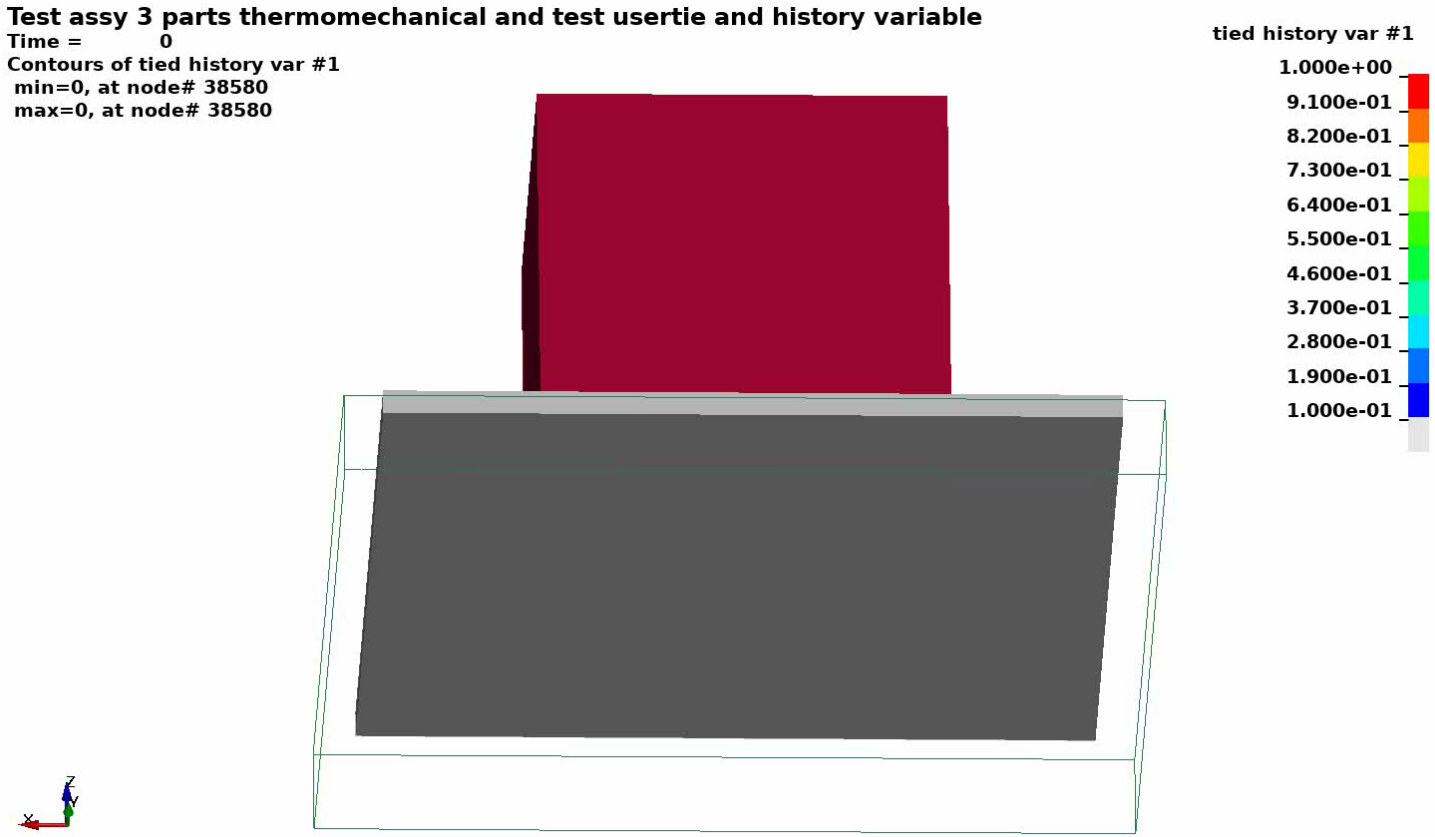


The Guideline contains examples of ...

User defined friction (`mortar_usrfrc`)

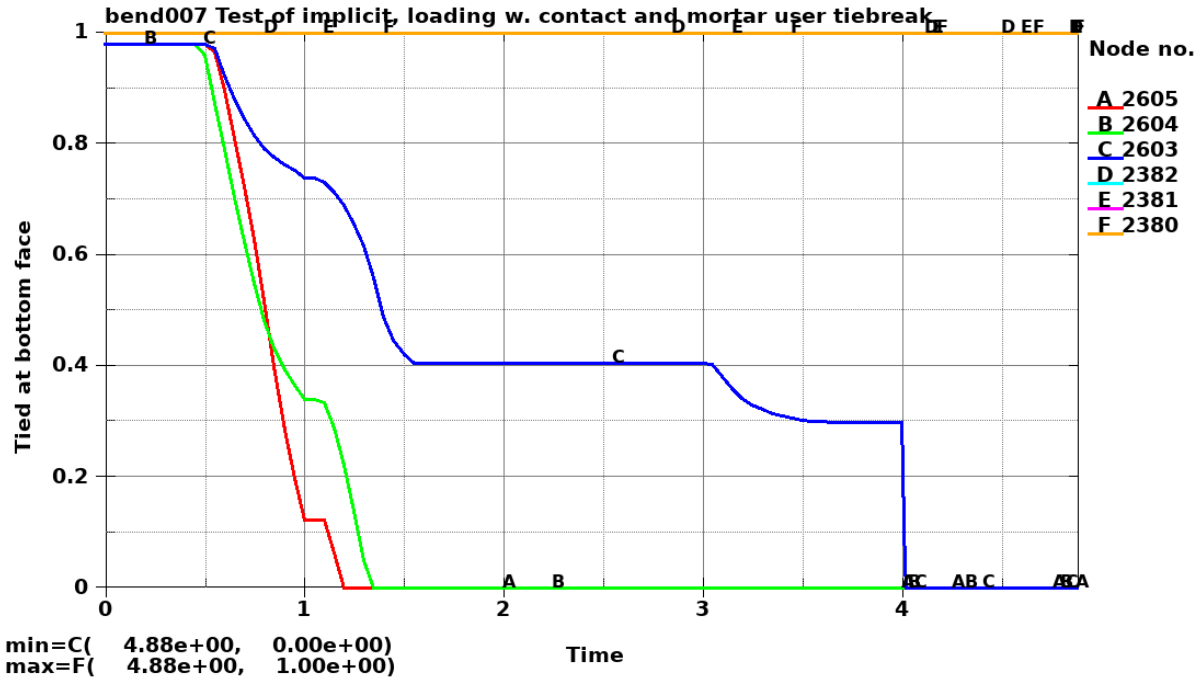


User defined tied condition (`mortar_usrtie`)

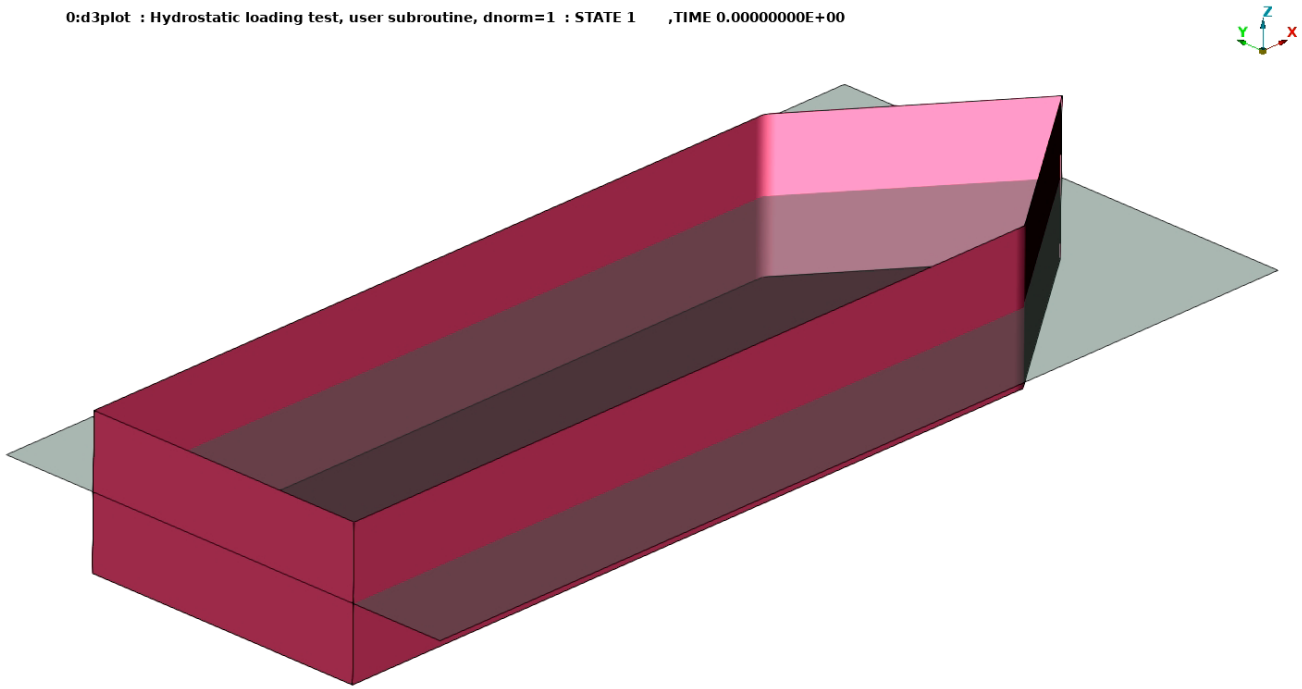


The Guideline contains examples of ...

User defined tiebreak contact (`mortar_usrtbrk`)



User defined loading (`loadsetud`)



The Guideline contains examples of ...

```
            call prludparm(1,parm,i,min(i+7,n))
     10     continue
            endif
*LOAD_BODY_Z
       100       9.81
*USER_LOADING
$     model       dof         rho           g         ref
         4.         3.      1000.        9.81        0.75
*USER_LOADING_SET
         2PRESSS             100                                                1
$=========================================================================
$ Simulation tilte
$=========================================================================
*TITLE
Hydrostatic loading test, user subroutine, dnorm=1
*END
```

```
        nid =lqfe(nodext,
                                return
                                end
                                subroutine loadud(fnod,dt1,time,ires,x,d,v,a,ixs,
                               . numels,ixb,numelb,idrflg,tfail,isf,p,npc,fval,iob,iadd64,numelh,
                               . ixh,nhex_del,nbeam_del,nshell_del,hexarray,hextim,bemarray,
```

For going from the user-defined n
keyword to the node ID used by L
mpp implementation. In case the particular node ID is not accessible by the current mpi

[13] In Table 7 some other useful functions for converting between internal and keyword-input numbering are listed.

Guideline for user-defined interfaces in LS-DYNA

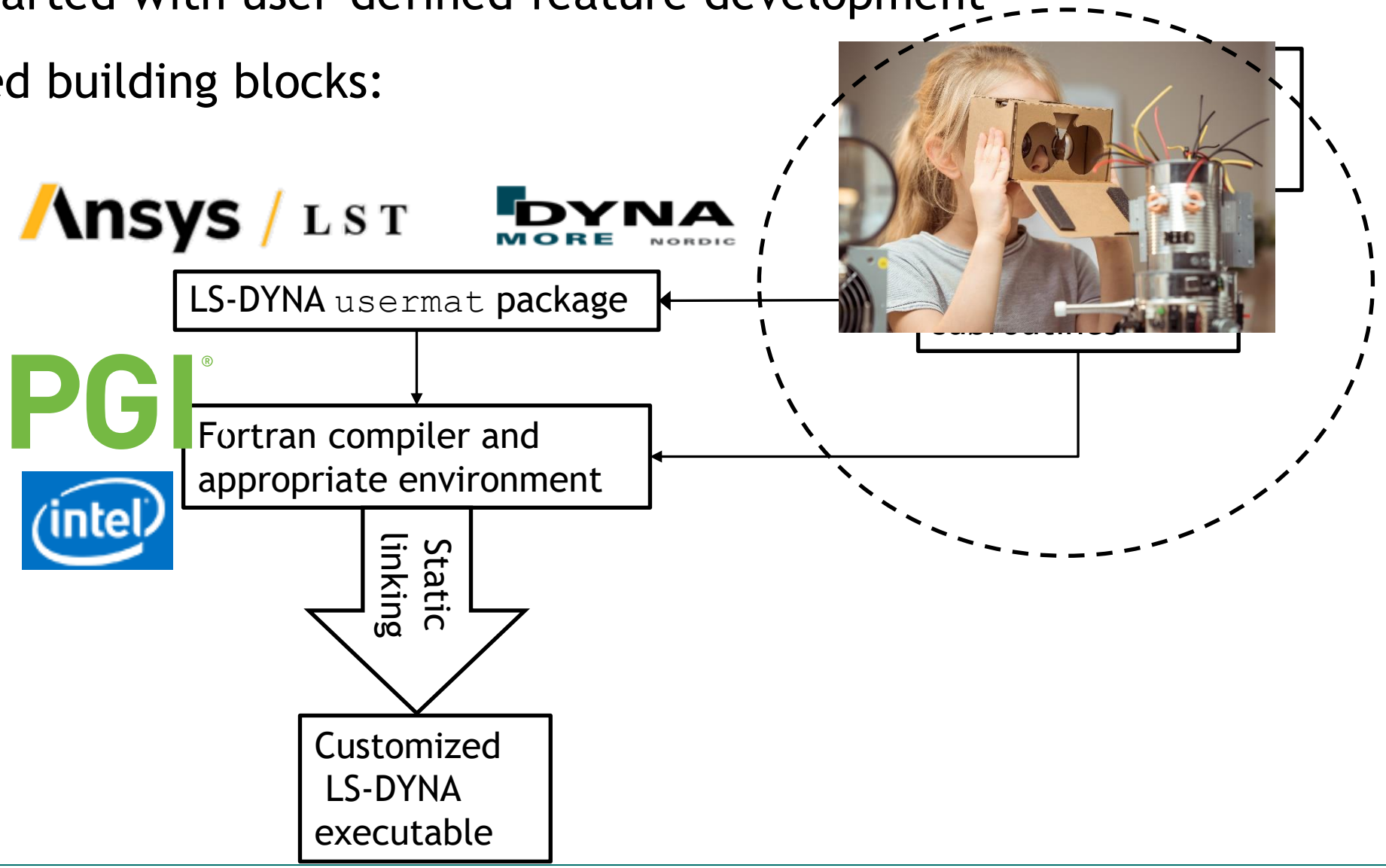
- Help for experienced LS-DYNA users to get started with creating user-defined features
- Detailed descriptions of some of the most commonly used features:
  - Material models
  - Friction models
  - Mortar weld tied contact
  - Mortar tiebreak contact
  - Loadings
- Contains
  - Guideline in PDF format
  - Examples in the form of Fortran subroutines and
  - LS-DYNA keyword files
- Available for download for DYNAMore Nordic customers, from files.dynamore.se > Client Area > 1_LS-DYNA_Guidelines
- To be extended and improved in coming releases
  - Your feedback is highly valuable!

Overview of available user-defined interfaces

- Material models
  - Complete user-defined material models
  - User-defined failure models for some of the built-in materials: MAT24, MAT103, MAT123, etc.
  - User-defined weld failure (MAT_SPOTWELD)
- Elements
- Friction
  - Contact conductance
  - User defined wear law
- Tie condition for Mortar tied weld contact
- Tiebreak contact (damage/failure model)
- Loading
  - Boundary flux
- Solution control
- Interfaces control
- Airbag sensor
- ...

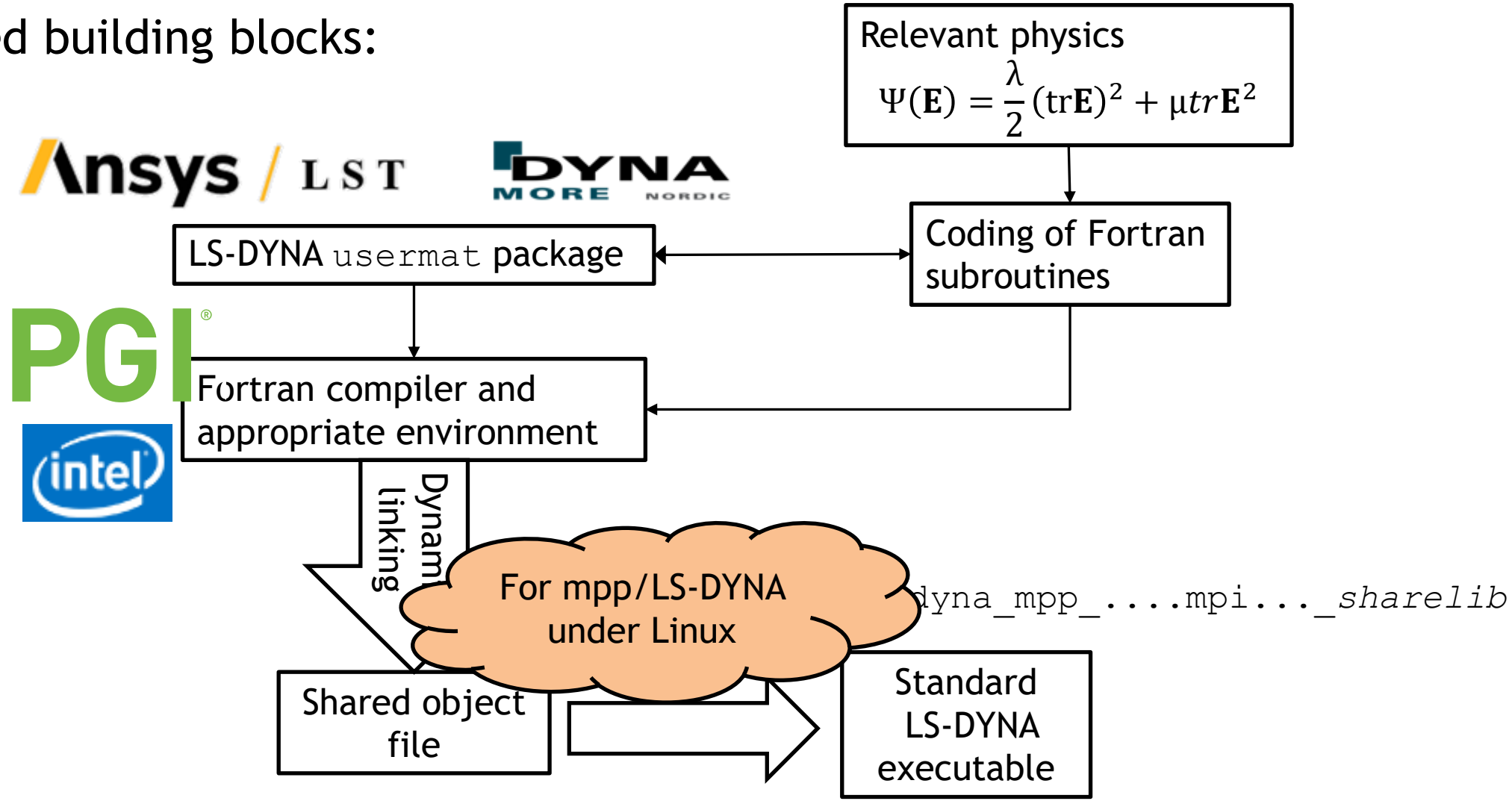
Getting started with user-defined feature development

- Required building blocks:



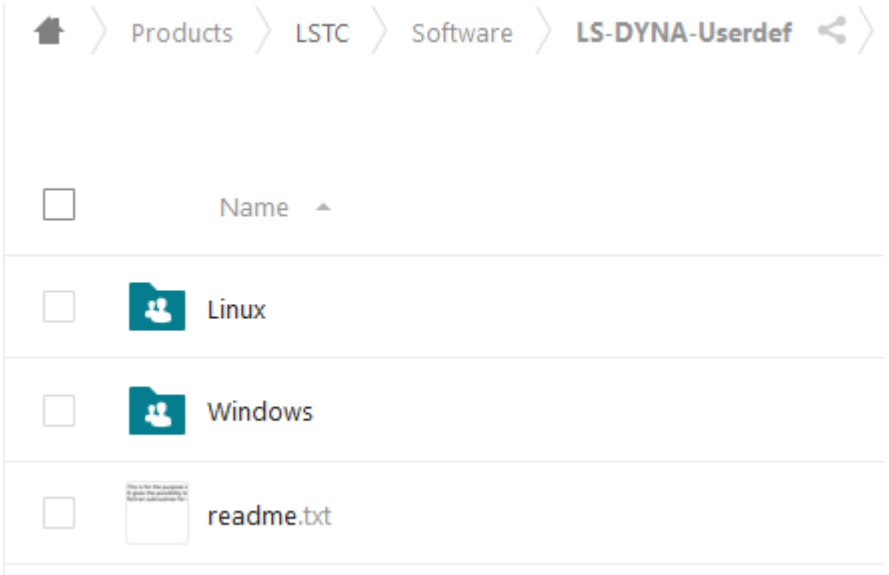
Getting started with user-defined feature development

- Required building blocks:



Getting started with user-defined feature development

- Download the LS-DYNA `usermat` package
  - Different packages for mpp, smp, double or single precision, of each version
  - Also the sse2, avx2 etc. extensions
  - For mpp, Linux: shared object or static linking
- From your local LS-DYNA provider
- For DYNAMore Nordic customers:
  - files.dynamore.se



The LS-DYNA `usermat` package (Linux)

- Fortran files: `dyn21.f`, `dyn21cnt.f` ...
- Object files, required for linking: `userinterface.mod` (`libdyna.lib` ...)
- Include files: `iounits.inc`, ...

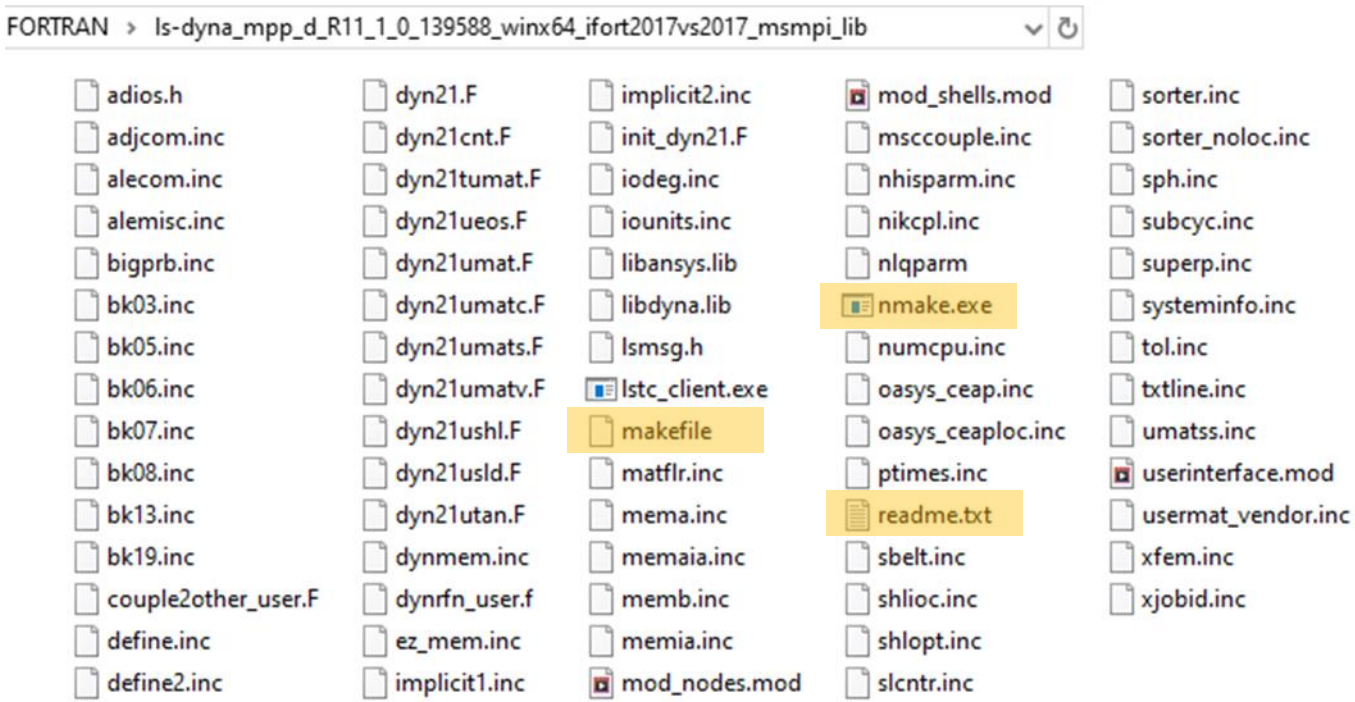
Linux

```
adjcom.inc    bk13.inc             dyn21umat.f    iodeg.inc      oasys_ceap.inc  superp.inc
alemisc.inc   bk19.inc             dyn21umats.f   iounits.inc    ptimes.inc      txtline.inc
bigprb.inc    couple2other_user.f  dyn21umatv.f   Makefile       sbelt.inc       umatss.inc
bk03.inc      dyn21cnt.f           dyn21ushl.f    matflr.inc     shlioc.inc      userinterface.f90
bk05.inc      dyn21.f              dyn21usld.f    memaia.inc     shlopt.inc      userinterface.mod
bk06.inc      dyn21tumat.f         dyn21utan.f    nhisparm.inc   slcntr.inc      xjobid.inc
bk07.inc      dyn21ueos.f          dynrfn_user.f  nikcpl.inc     sph.inc
bk08.inc      dyn21umatc.f         implicit1.inc  nlqparm        subcyc.inc
```

The LS-DYNA `usermat` package (Windows)

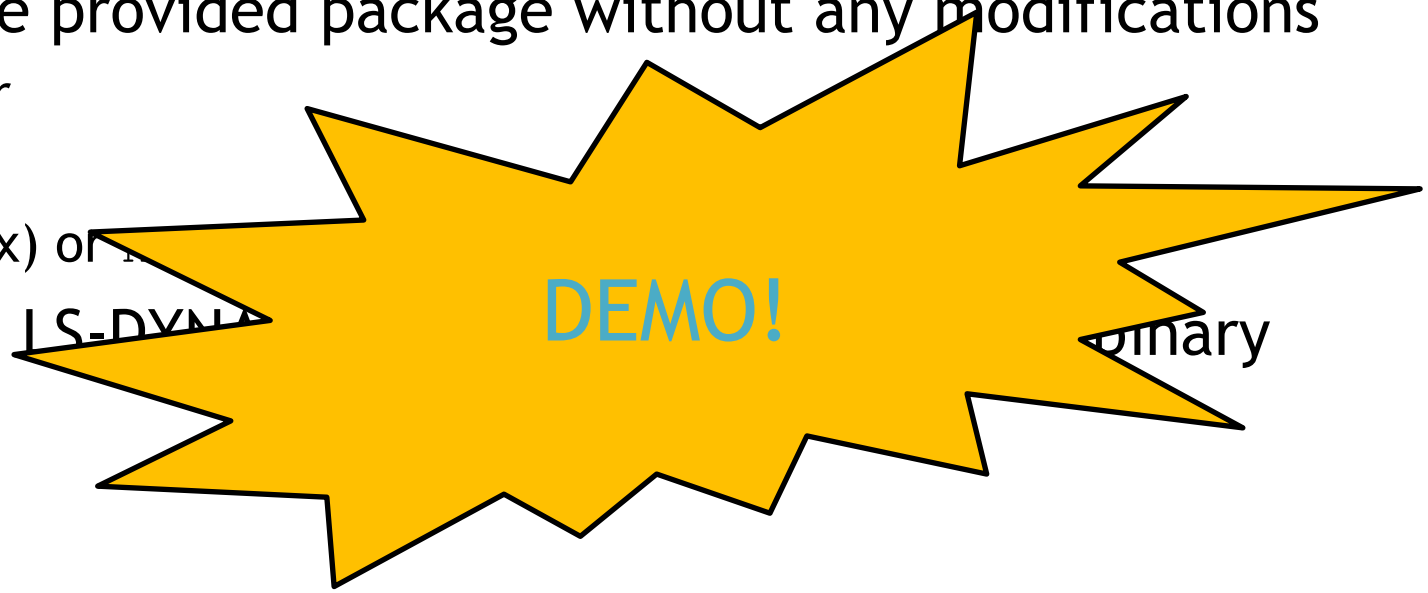
- Fortran files: `dyn21.f`, `dyn21cnt.f` ...
- Object files, required for linking: `userinterface.mod` (`libdyna.lib` ...)
- Include files: `iounits.inc`, ...

Windows



Getting started with user-defined feature development

- Download the LS-DYNA `usermat` package
- Unpack and inspect
- Test-compile the provided package without any modifications
  - Set up compiler
  - Edit Makefile
  - run `make` (Linux) or
- Test run a small LS-DYNA binary



Files - DYNAmore Nordic Files — Mozilla Firefox

Files - DYNAmore Nordic

https://files.dynamore.se/index.php/apps/files/?dir=/&fileid=101987

Most Visited | LS-PrePost Docume... | tyda.se | Dynamore | LS-DYNA Support | LS-OPT Support | Intranet - Home | webCRM | WebPlotDigitizer | Python - Convert N... | Other Bookmarks

- All files
- Recent
- Favorites
- Shares
- Tags
- External storage

Add notes, lists or links ...

delningar.txt
Recently edited

20220208_mesh_gänga.pptx
Recently edited

exp20221031_500crs_... .txt
Recently edited

Lic_ES-2re_v8.0_ES-2_... .zip
Recently edited

MasterThesis2022___0... .lic
Recently edited

Name	Size	Modified
Client Area	2 GB	4 days ago
Clients	569.8 GB	an hour ago
DmN Shared	2 GB	a month ago
Documents	3.4 GB	4 days ago
Products	566.3 GB	5 days ago
Nextcloud Manual.pdf	6.5 MB	2 years ago
5 folders and 1 file	1.1 TB	

Files of the LS-DYNA `usermat` package (R11.1 and later ...)

User-defined feature	Subroutine	Fortran source code file
Material models	**umat*XX***, umat*XX*v **utan*XX***, utan*XX*v	dyn21umats.f, dyn21umatv.f dyn21utan.f
Damage/failure for some materials	matusr_24, matusr_103	(dyn21umat.f) dyn21.f
Thermal materials	thumat*XX*	dyn21tumat.f
Elements (shells, solids)	u*XXX*_b*YYY*, u*XXX*_e*YYY*	dyn21ushl.f, dyn21usld.f
Friction	**usrfrc, mortar_usrfrc**	dyn21cnt.f
Tie condition for Mortar tie weld	**mortar_usrtie**	
User defined wear law	userwear	
Tiebreak contact	**mortar_usrtbrk**, utb10*X*	
Contact conductance	usrhcon	dyn21.f
Loading	**loadud, loadsetud**	dyn21.f
Solution control	uctrl1	
Interfaces control	uctrl2	
Airbag sensor	airusr	

Fortran compilers and environment

- Recommended compiler depends on
  - LS-DYNA version
  - Operating system
- For Redhat Linux, use Intel Fortran compiler
- For Suse Linux, use PGI Fortran compiler
- For Windows 10, use
  - Intel Parallel Studio
  - Microsoft Visual C++ x64 Cross Tools (for linking and access to standard libraries)
  - Microsoft MPI Software Development Kit (SDK) for mpp/LS-DYNA under Windows

Fortran compilers and environment

LS-DYNA version	Linux	Windows
R11	Intel Fortran 2016	Intel Parallel Studio XE 2017 Microsoft Visual C++ 2017 x64 Cross Tools
R12(1)		
R13	Intel Fortran 2019	Intel Parallel Studio XE 2019 Microsoft Visual C++ 2019 x64 Cross Tools

Notes: (1) R12 with avx512 extension requires Intel Fortran 2018.

Getting started with user-defined feature development

- Fortran programming will be needed
  - Translate physics into user-defined subroutines
- Normally only basic functionality is required
  - Assignment and arithmetic operations
  - Conditional statements
  - Loops
- Sample code is provided with the ...
  - LS-DYNA `usermat` package, look into the `dyn21...f` files,
  - Guideline package, look in the example subroutines
- Internet resources
  - https://www.tutorialspoint.com/fortran/fortran_basic_syntax.htm
  - ...

Fortran

- Fortran statements are input in position 7 – 72 on a line
- Case-insensitive input
- Comments: put a C in position 1
- Continuation line: put a character in position 6
- The basic structure is

```
PROGRAM name
declarations
executable statements
END
```

Fortran - Variables and arrays

- Basic types: `integer`, `real` (floating point number), `real*8` (double precision floating point number), `char`, `logical` (`.true.` or `.false.`)
- Arrays are fields of variables, and declared as (for example)

```
integer ii(20)
real*8 sigma(6), sdev(6), mat(3,3)
```

- Declaration is not mandatory but strongly recommended! Use

```
implicit none
```

- Assignment examples

```
kk = 10
ii(kk) = 5
sdev(1:3) = sigma(1:3) - sum(sigma(1:3))/3.0
```

"Fortran 90" style

Fortran – Useful statements

- Loop

```
do varaible=first,last
  statements ...
enddo
```

- Conditional execution

```
if condition then
  statements
elseif condition then
  statements
else
  statmenets
endif
```

Incorporating user-defined features in LS-DYNA

- Static linking
  - Currently the only option for smp, and Windows
  - In Linux, download the LS-DYNA `usermat` package called

    `ls-dyna_.....usermat.tar.gz`
  - A customized LS-DYNA executable is built
  - Contains all built-in functions plus the additional user-defined features
- Dynamic linking
  - Available in Linux, for mpp/LS-DYNA only
  - Download the LS-DYNA `usermat` package called

    `ls-dyna_mpp....sharelib.usermat.tar.gz`
  - A shared object (dynamic library) is built, containing the user-defined features
  - The shared object is dynamically linked to a standard LS-DYNA executable, for example

```
ls-dyna_mpp_d_R11_1_0_x64_centos65_ifort160_avx2_intelmpi-2018_sharelib
```

Incorporating user-defined features in LS-DYNA

- Static linking
  - Currently the only option for Windows
  - A customized LS-DYNA executable is built
- Dynamic linking
  - Available in Linux for mpp/LS-DYNA
  - A shared object (dynamic library) is built, containing the user-defined features
  - The shared object is dynamically linked to a standard LS-DYNA executable
- Both approaches require that user-defined features be re-compiled for new LS-DYNA versions
  - Download the corresponding LS-DYNA `usermat` package
  - Transfer previous user-defined subroutines
  - Shared objects need to match a specific LS-DYNA version.
    NOTE! Also single precision vs. double precision! sse2, avx2, avx512 must also match!

Dynamic linking, shared objects and *MODULE

- The shared object is “plugged in” by the *MODULE_{*OPTION*} keywords.
- *MODULE_PATH
  - Specify where LS-DYNA should search for the shared object
- *MODULE_LOAD
  - For loading shared objects, and assigning an ID
- *MODULE_USE
  - Specify which feature to use from a specific shared object
- By the use of shared objects, customized features from different sources (external 3rd party companies, internal developments, etc.) can be included in the same LS-DYNA analysis.

- An example follows:
  - An internally developed material model (umat41) has been developed and built as a shared object: sharedobje01.so
  - Another company has developed a material model (umat41) and delivered that as a shared object: sharedobje02.so

Dynamic linking, shared objects and *MODULE

```
*MODULE_PATH
 … path_to_modules …
*MODULE_LOAD
$#1             MDLID                                           TITLE
                    1 Library 1
$#2 FILENAME
sharedobje01.so
*MODULE_LOAD
$#1             MDLID                                           TITLE
                    2 Library 2
$#2 FILENAME
sharedobje02.so
*MODULE_USE
$#1             MDLID
                    1
$#2              TYPE              PARAM1                PARAM2
UMAT                                   41                  1001
*MODULE_USE
$#1             MDLID
                    2
$#2              TYPE              PARAM1                PARAM2
UMAT                                   41                  1002
```

“From shared object 1, use UMAT 41 as User defined material model ID 1001”

“From shared object 2, use UMAT 41 as User defined material model ID 1002”

Summary

- LS-DYNA offers many possibilities for the user to create customized functionality
- User defined
  - material models
    - damage/failure models for some built-in materials
  - elements
  - friction models
  - loading
  - etc.
- A Guideline has been developed in order to make it easier getting started with user-defined features in LS-DYNA
  - Including examples in the form of Fortran code and LS-DYNA keyword files
  - Download it from files.dynamore.se (DYNAMore Nordic Customers)
- Fortran compiler is required from external supplier

Thank you!

DYNA MORE

Your LS-DYNA distributor and more