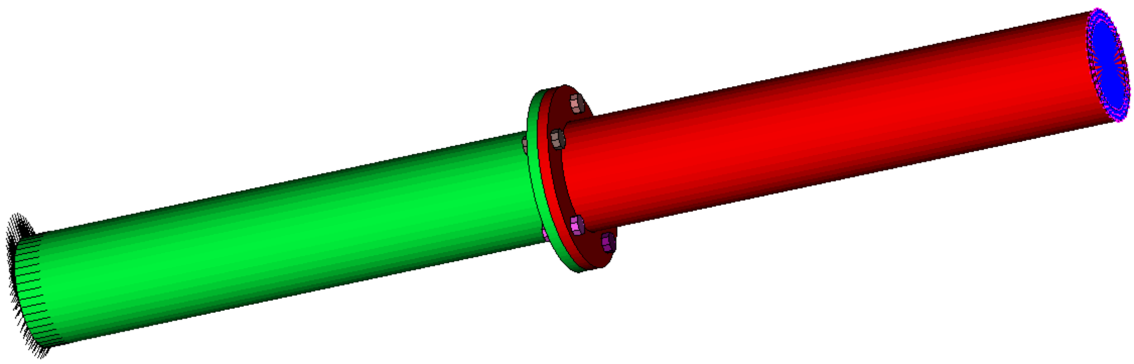
Guideline for implicit analyses using LS-DYNA



2018-01-13

LS-DYNA / LS-PrePost

Summary

In this document, some basic control card settings for different implicit analysis types are presented. The analysis types are also accompanied by some basic examples. The purpose is to reduce the effort of getting started with implicit analysis in LS-DYNA.

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

Contents

1 Background

The purpose of the present document is to provide a starting point[1] for setting up implicit analyses in LS-DYNA, 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 R9.1 [1], or later. The control card settings have been developed and tested for every-day use at Dynamore Nordic, and found to work well for many different 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. A guideline document for NVH specific analyses is under development.

For thorough details regarding LS-DYNA keywords and material models, the reader is referred to Ref [1][2]. From R9.0, Appendix P of the Keyword manual [1] has been added, with 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. [8].

See also www.dynasupport.com for general support. Useful publications may be found on www.dynalook.com. Example keywordfiles can be found at www.dynaexamples.com/implicit. For further questions, or when errors are found in this document, please contact support@dynamore.se.

Dynamore Nordic provides training courses and webinars in implicit analyses using LS-DYNA, see www.dynamore.se for further details. The present document is based on the course notes [4] developed by Dr. Thomas Borrvall and others. Courses in contacts and material modelling are also available.

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

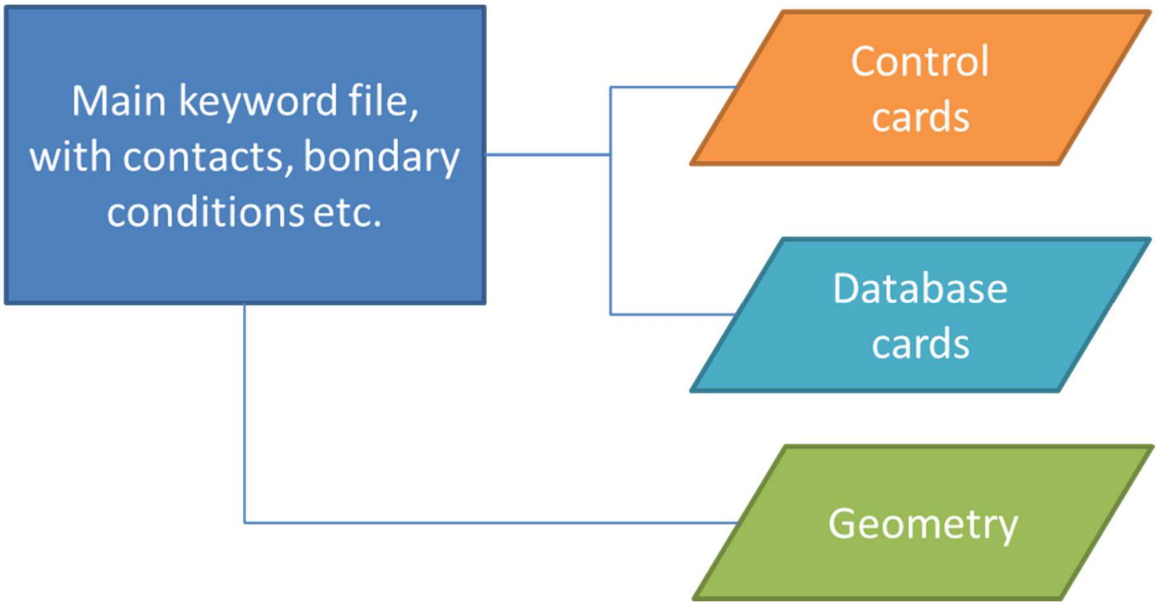


Figure 1. Recommended include-file structure for convenient use of the provided control cards files.

2 Overview

In Section 3, output options are briefly discussed. In Section 4, control card settings for different implicit analysis types are presented. 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.

Appended keywordfiles are listed in Table 1 (see page 2).

3 LS-DYNA database cards for different analysis types

Two different include files for controlling the output from implicit analyses were developed, one for static analyses and one for transient dynamic analyses. 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 that 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.

Output of element integration point quantities is discussed in Section 5.4. Contact output is discussed in Section 6.3.

Table 1. List of some of the appended files

Filename	Comment
database_cards_static.key	Output request for static analyses
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_arc.key	Control cards for non-linear static analyses using the arc length method
Filename	**Example of**
doorstiff001.key	Linear static analysis
bend001.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

4 LS-DYNA control cards for different implicit analysis types

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 types, is presented in Table 2. Example keyword files are also presented in the following Sections.

It is in general recommended to use the latest available version of LS-DYNA for implicit analysis. In general, the mpp version of LS-DYNA is recommended, but all features discussed, except for the linear transient modal dynamics (Section 4.6), are available in both mpp and smp versions of LS-DYNA. Linear transient modal dynamics is only available in smp. In mpp, only double precision can be used. Also for smp, the use of double precision is warmly recommended, although some analysis types may be performed using single 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 seems reasonable for a particular application. It might very well be the case that other settings give better results.

In some cases, it might be useful to switch to the explicit solver (see Appendix D), and for this purpose also a keyword file with control cards for explicit analyses is provided: `control_cards_expl.key`, and accompanying database file: `database_cards_expl.key`.

Table 2. Guide for quick selection of include files for different analysis types (see also Table 1)

	Include files		Add keywords:(*)
Analysis type	`control_cards_`	`database_cards_`	`*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.

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. Call material routines in order to compute the stresses and strains from the displacements.

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 routine, as a post-processing step. Since no equilibrium iterations are performed, the obtained stresses and strains from non-linear materials in a linear analysis may be more or less non-sense. If consistent estimates of stresses and strains from a linear analysis are of interest, it is highly recommended to use linear elastic material models (for example `MAT_ELASTIC`, see also Section 7).

In LS-DYNA, 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.

Tied contacts are valid in linear analyses, see also Section 6.2. Other contacts should be used with 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 keywordfile `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
```

By using these settings, it may (with some restrictions) be possible to solve for a sequence of load cases, see Figure 2, without re-forming the stiffness matrix. The limitation is that each load case should cause truly small displacements and strains (which implies for example that unit loads cannot be chosen arbitrarily for this analysis procedure to be valid, they must be "small"). The result from each load case will then be output as a new d3plot state in an individual file.

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

From R10.1, it is possible to perform linear solutions for a sequence of arbitrary load cases, without re-forming the stiffness matrix. This option is activated by setting *NSOLVR* = -1 on *CONTROL_IMPLICIT_SOLUTION. 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 linear stiffness of a door panel is to be determined. The geometry of the example is shown in Figure 3. A force of 100 N is applied, distributed over a node set. The example keywordfile is `doorstiff001.key`. The resulting displacement magnitude is shown in Figure 4.

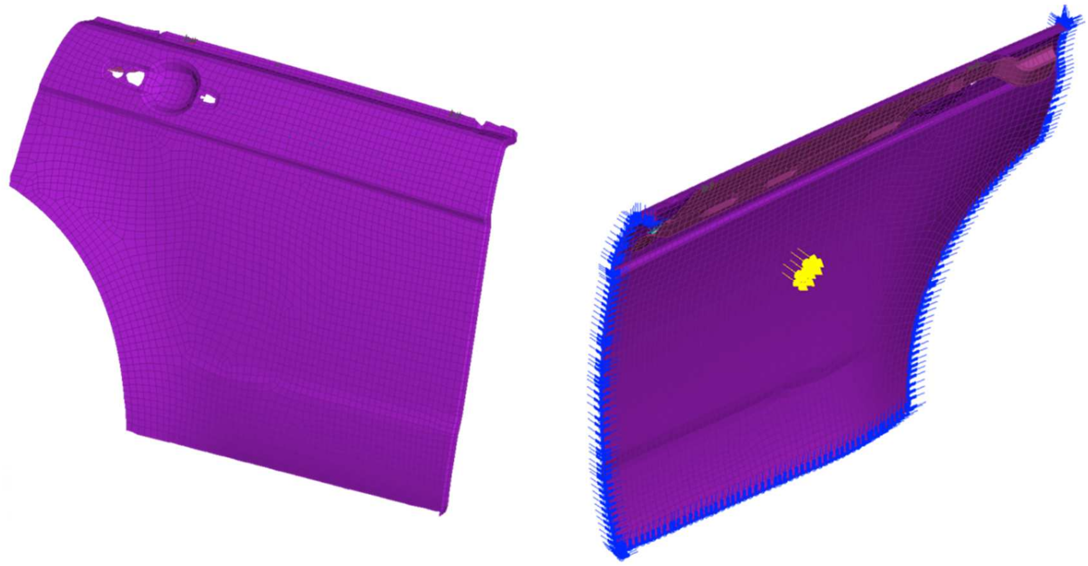


Figure 3. The left image shows the FE-model of the door panel. The right image shows the constrained boundary conditions as blue lines and the loading as yellow arrows.

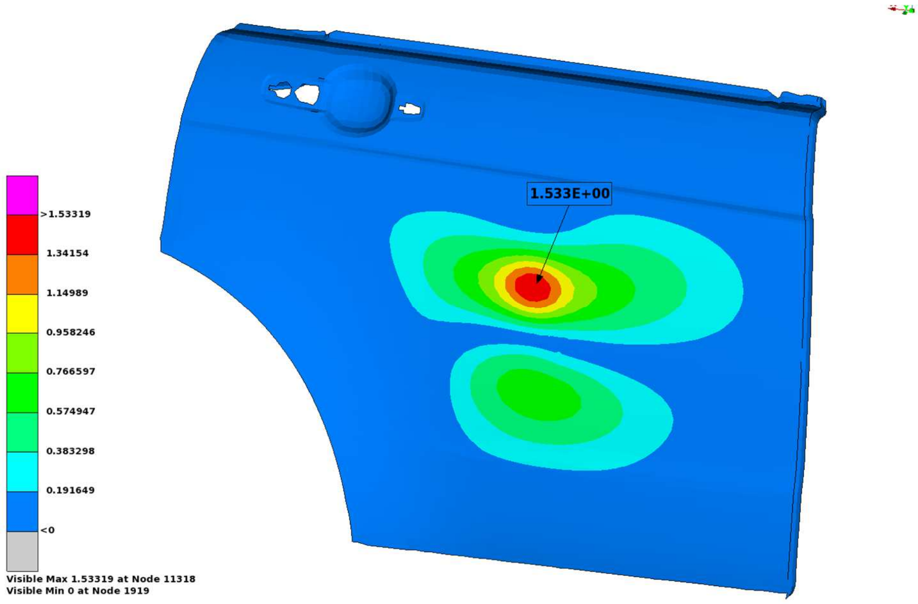


Figure 4. Magnitude of displacement for the door stiffness example.

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 4.5). 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].

4.2 Non-linear static analysis

In LS-DYNA implicit, fully non-linear static analysis is the default, that is 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), or
- 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 LS-DYNA implicit, an analysis is either fully non-linear or fully linear. It is not possible to mix for example non-linear materials and linear (small) deformation theory.

The keywordfile `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 5.

---

[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.
[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 4.4).

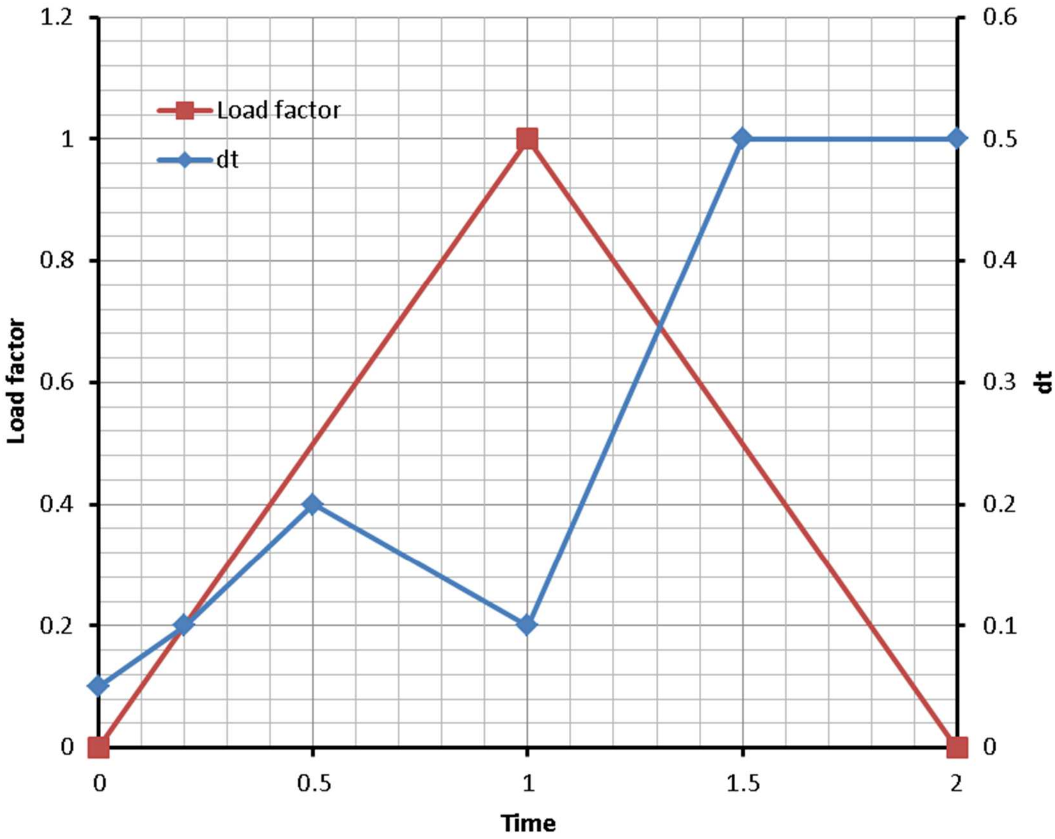


Figure 5. Illustration of the use of the time incrementation curve (blue) to synchronize the simulation with the applied loading (red curve). By specifying a value of (1.0, 0.2) for the time incrementation curve, a step at t = 1.0 (coincident with the maximum loading) is obtained.

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. Many analyses involve an initial stage where rigid body modes exist in the structure, which are eliminated (by for example bolt pre-tensioning) at later stages. This type of situations can be handled by performing the initial stage, until contacts are established etc., using implicit dynamics (LS-DYNA keyword `*CONTROL_IMPLICIT_DYNAMICS`) which is discussed further in Section 4.2.2.

4.2.1 Non-linear static example

An aluminium cantilever beam is bent over a rigid support. The geometry for the example is shown in Figure 6. 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 keywordfile is `bend001.key`. Results are shown in Figure 7 and Figure 8.

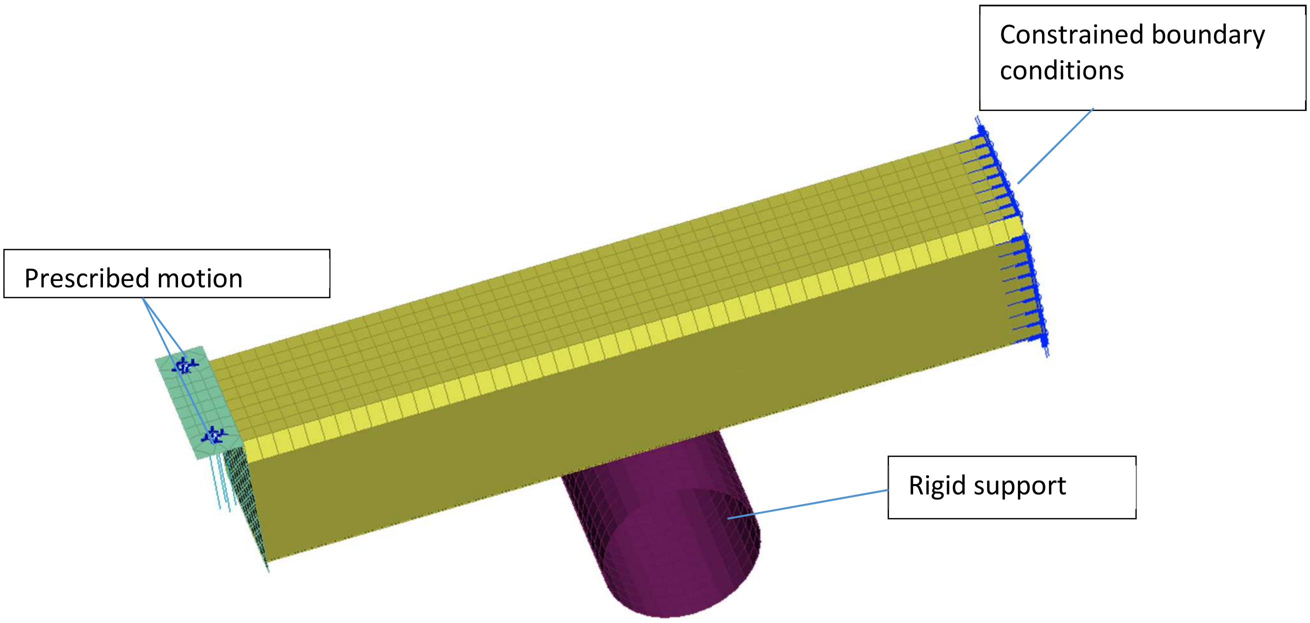


Figure 6. A cantilever beam is bent over a rigid support.

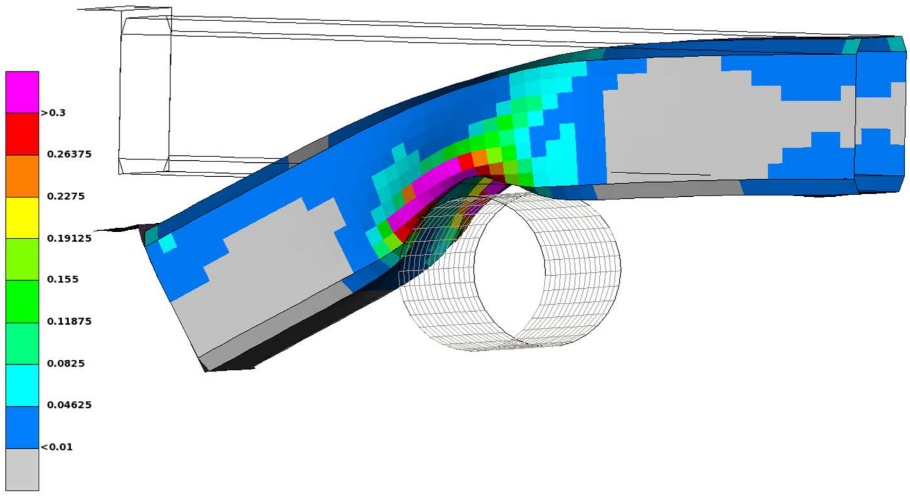


Figure 7. Plastic strains and deformation at the final stage.

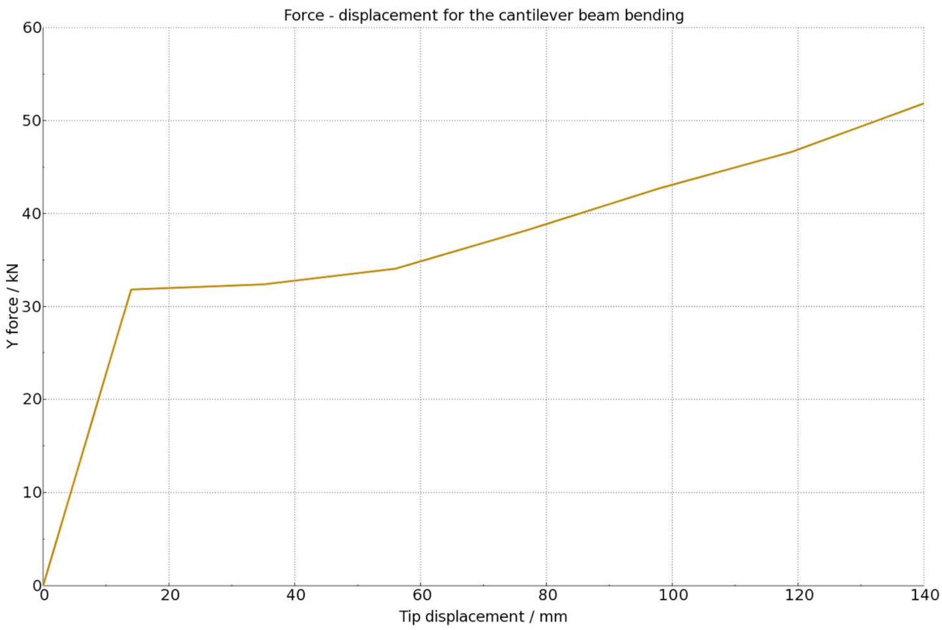


Figure 8. Required Y-force vs. tip displacement.

4.2.2 Bolt pre-tensioning

A bracket is connected to a base plate by four bolts, see Figure 9. 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 force. 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 keywordfile is `bolts001.key`.

Implicit dynamics is used during the bolt pre-tensioning in order to overcome 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 4.8) control the time integration. Normally *GAMMA* = 0.6 and *BETA* = 0.38 are used in order to introduce some numerical damping. 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 10. Setting the parameter *IRATE* = 1 will switch off the rate effects in material models, see further Section 7. Since the simulation time in a static analysis most commonly is a parameter that does not correspond to physical time, computed strain rates will be unphysical, and switching off the rate effects is then a sensible choice.

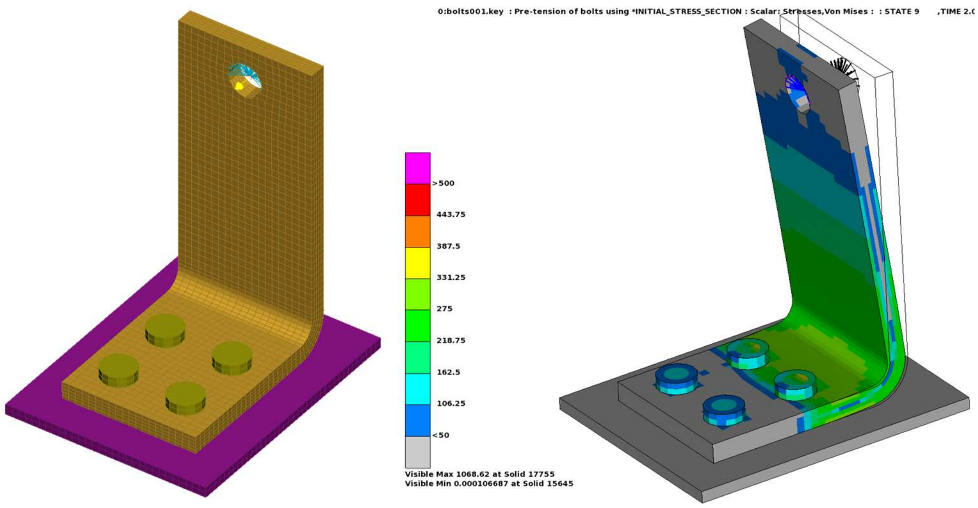


Figure 9. A bracket is attached to a base plate by four bolts. A force (shown as a yellow arrow) is applied by a distributing coupling at the centre of the hole in the flange. The left image shows the initial geometry. The right image shows the final state with a fringe plot of von Mises effective stress.

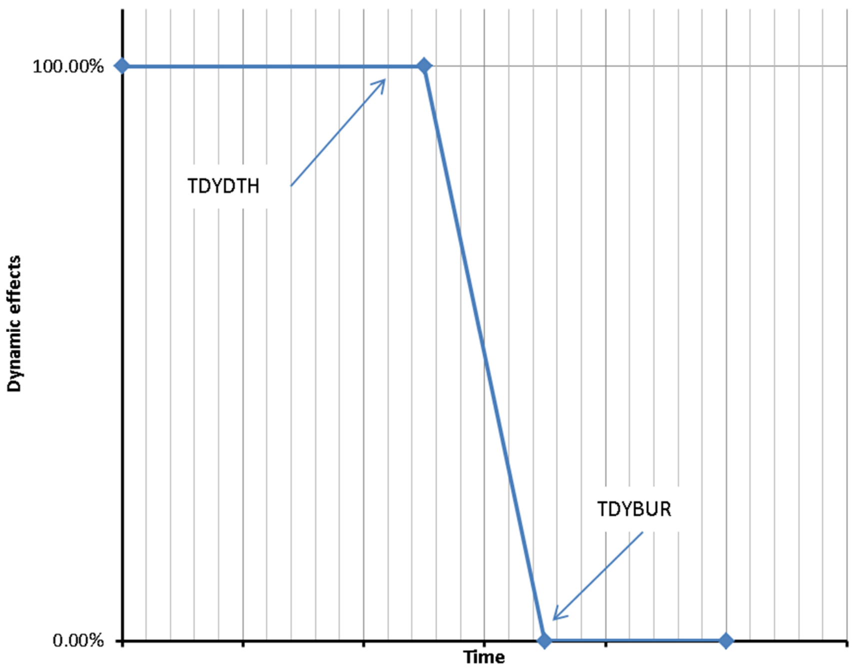


Figure 10. Death and burial time for implicit dynamics.

4.3 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 eigenprobelm

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

where $\mathbf{K}_g$ is the geometric stiffness matrix (due to stress stiffening effects). In order to compute $\mathbf{K}_g$, some initial loading of the structure must be done so that stresses develop. The pre-loading can be applied by a static implicit analysis. If the pre-loading is done in a non-linear analysis, it is recommended that the loading be such that the structural response is still within the linear regime (no major plastic deformation).

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 load curve ID can be specified by entering the LCID as a negative number of buckling modes. The x-values of the load 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 4.3.1.

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).

4.3.1 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 11. The edge at the flange is constrained in the transverse direction. The example keywordfile 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 12.

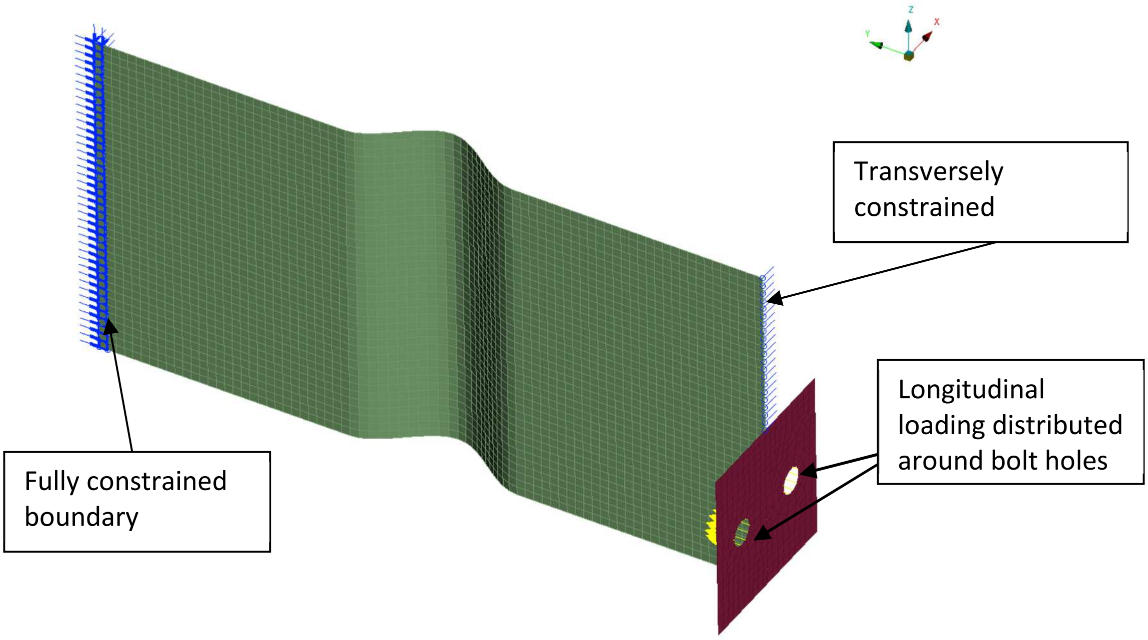


Figure 11. A panel with a bead under compression loading.

Figure 12. The lowest buckling mode shape for the panel.

4.4 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 13. 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.

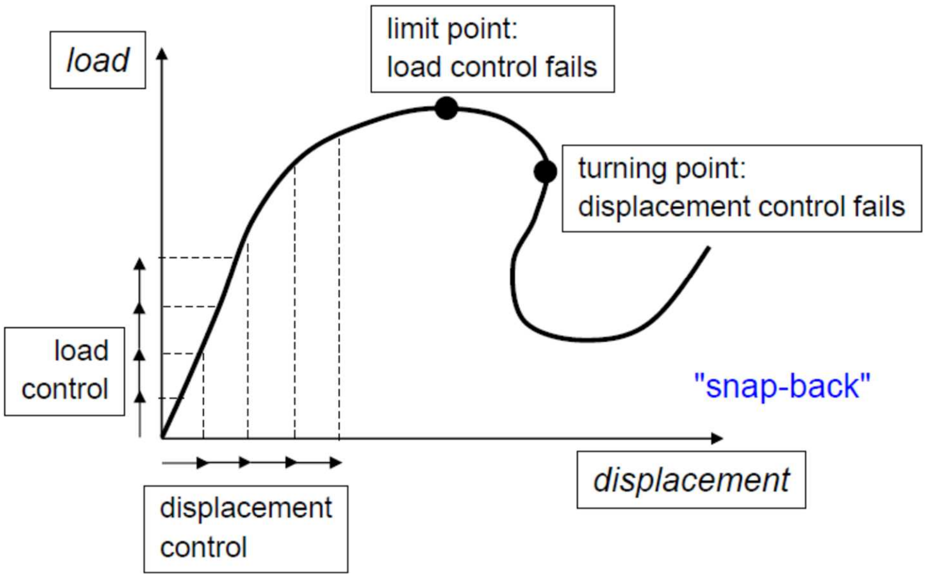


Figure 13. Arc-length control is required in order to follow a non-monotonically increasing (generalized) load path. From Ref. [4].

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 includefile `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`. A combination of (reasonably scaled) buckling mode shapes (see Section 4.3) is often a suitable choice in cases where the shape of the imperfection is not known exactly.

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 14. If multiple loads exist in the model (including thermal loading), all loads will vary simultaneously.

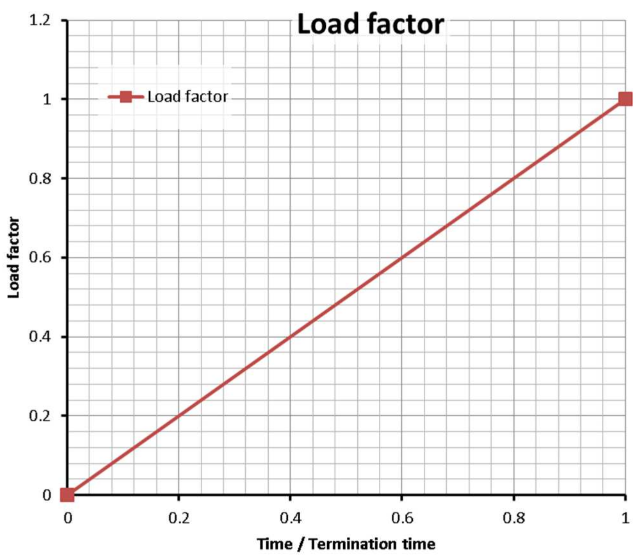


Figure 14. Typical load curve for arc-length analysis.

The keywordfile `control_cards_nonlin_arc.key` contains control card settings which are generally well suited for non-linear postbuckling analyses. In this file, *ARCTIM* is set to zero.

The simulation time in this case takes a clear roll of a load parameter, and using `*CONTROL_TERMINATION` 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` 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 4.4.1.

4.4.1 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 11. The edge at the flange is constrained in the transverse direction. The geometry, boundary conditions and loadings are the same as in Section 4.3.1. The material in the panel is Domex 355. The example keywordfile is `limitload001.key`. The force-deflection curve is shown in Figure 15. The peak load is 100 kN.

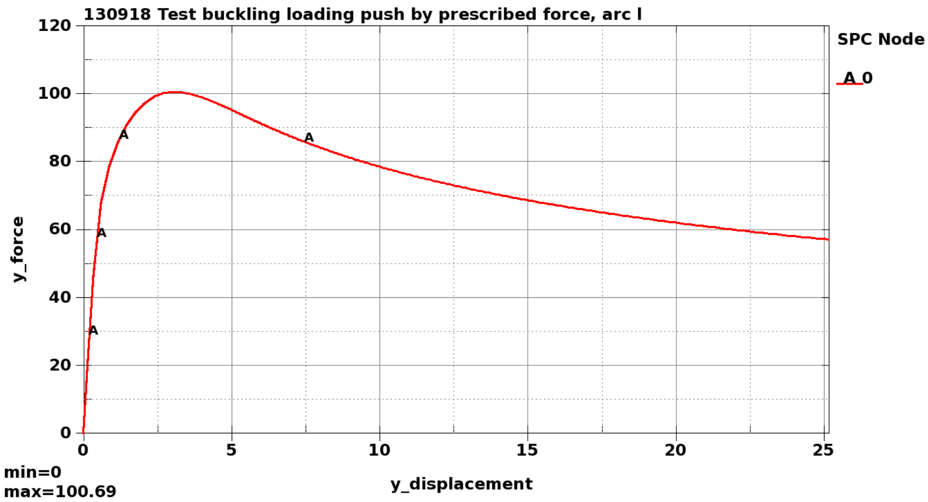


Figure 15. Force-displacement curve for the postbuckling analysis of a panel with a bead.

4.5 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

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

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

$$\left(\mathbf{K}-\omega^2\mathbf{M}\right)\mathbf{\Phi}=0$$

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 4.6 and 8.

In LS-DYNA, eigenfrequency analysis is activated by using the keyword `*CONTROL_IMPLICIT_EIGENVALUE`. An eigenfrequency analysis may be part of a linear or non-linear static analysis. 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 4.5.2). It is also possible to compute modal stresses by setting the parameter *MSTRES* `= 1`. Note that 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
Define number of eigenmodes etc.
*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 eigenfrequency analysis will be performed at the beginning of the simulation, which then terminates. A basic eigenfrequency example is presented in Section 4.5.1. It is possible to perform intermittent eigenfrequency analyses at different stages of the simulation. A load curve ID can be specified by entering the LCID as a negative number of eigenmodes. The x-values of the load 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. See also the example of Section 4.5.2.

The eigenfrequencies are printed in the `eigout` - file. The mode shapes are output in the `d3eigv` file(s).

4.5.1 Eigenfrequency analysis of a panel with a bead

A panel is fully constrained at one edge, see Figure 16. The example keywordfile is `eigen001.key`. The lowest eigenfrequency is 44.3 Hz, and the mode shape is a side-to-side swinging motion.

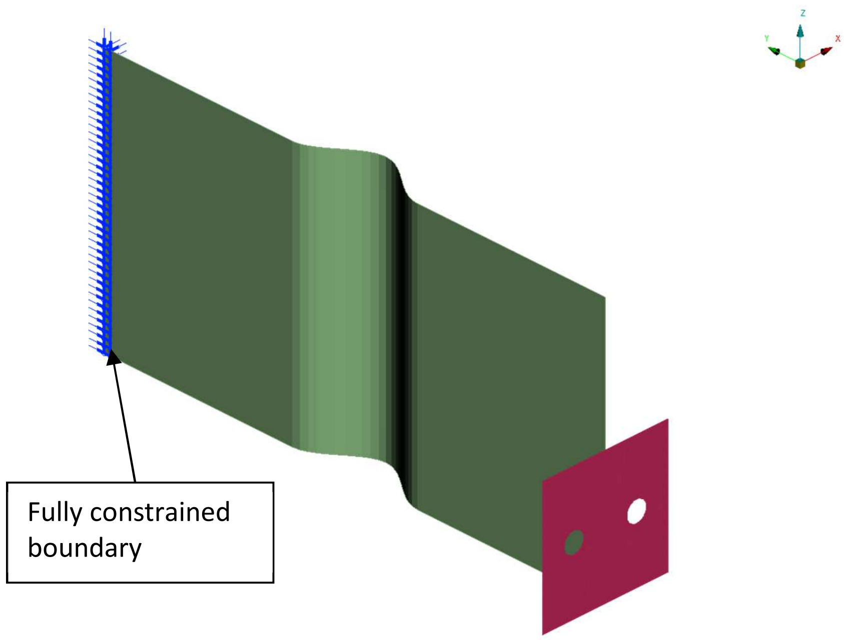


Figure 16. Eigenfrequency analysis of a panel with a bead.

4.5.2 Intermittent eigenfrequency analysis of a bolted L-bracket

Eigenfrequency analyses of the bolted assembly of Section 4.2.2 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 keywordfile 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.

4.6 Linear transient modal dynamic analysis

Transient linear dynamic analyses can often be performed very efficiently using a modal basis, denoted as $\mathbf{\Phi}$**,**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: $\mathbf{x} = \mathbf{\Phi z}$. Also, by the orthogonality properties of $\mathbf{\Phi}$, 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 smp-version 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. 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

- 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, 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 possible from R8.0.0, 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
.
.
.
```

4.6.1 Transient loading of an L-beam

An L-shaped beam is loaded by a transverse, transient (triangular) loading, see Figure 17. A constant modal damping of 3 % is applied. The transient displacement response of the loaded node is shown in Figure 18.

The example keywordfile is `transient001.key`.

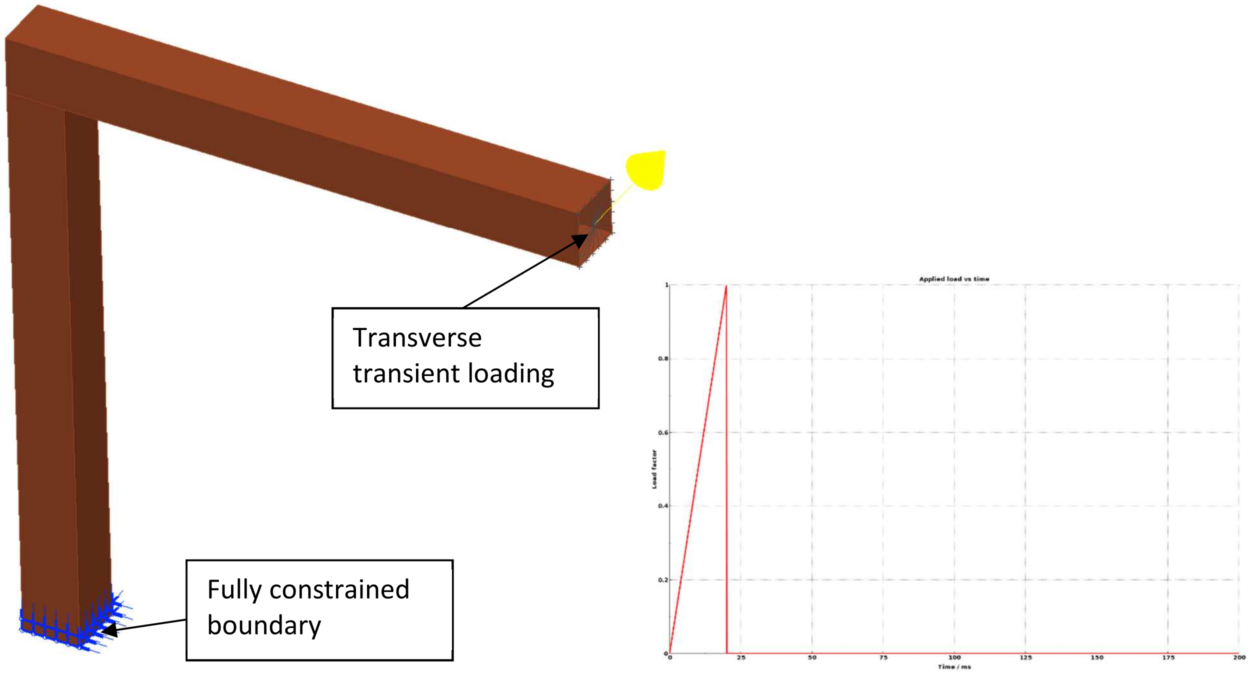


Figure 17. Transient loading of an L-shaped beam. The left image shows the geometry, with boundary conditions and loading. The right image shows the loading history.

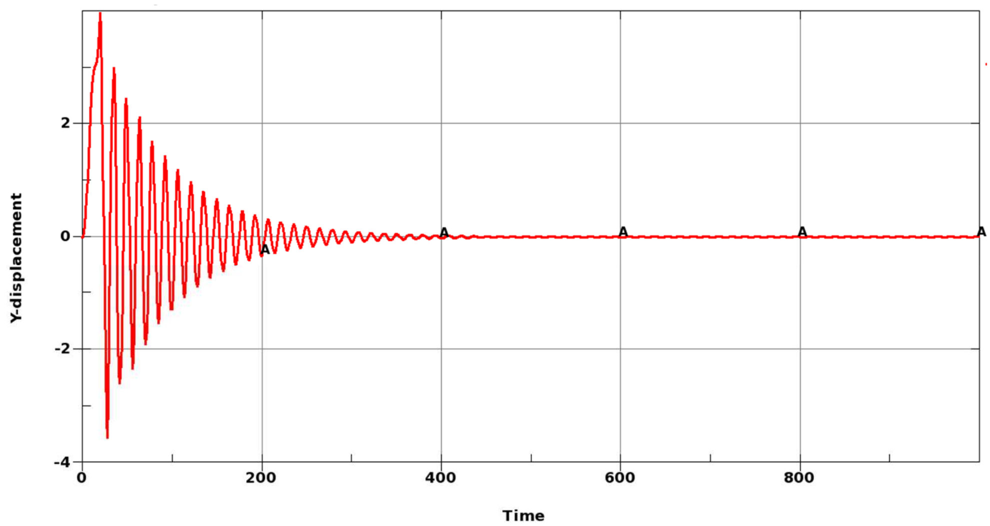


Figure 18. Displacement history in the transverse direction of the loaded node.

4.7 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, and
- random vibration analysis.

Currently (2018-02-13) these analyses are preformed using a modal basis, which implies that an initial eigenfrequency analysis, see Section 4.5, 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.

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

An overview of the frequency domain analysis capabilities is presented in Ref. [9], and a thorough description is given in Ref. [10].

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 datalines. Output for the different frequency domain analysis types can be requested by use of 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 19.

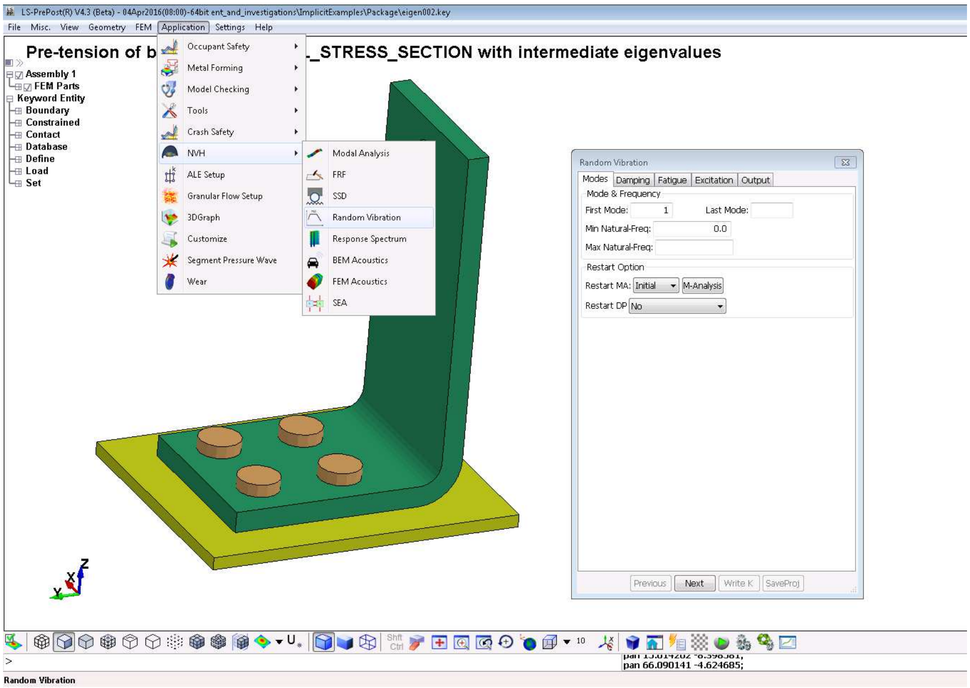
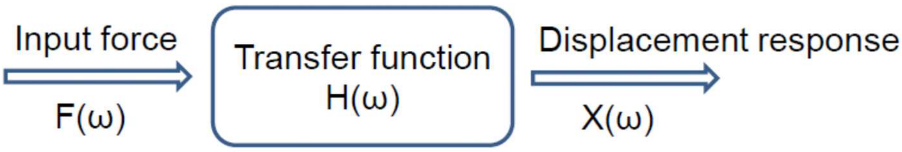


Figure 19. Wizard for frequency analysis set-up in LS-PrePost 4.3.

4.7.1 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 20.



$$X(\omega) = H(\omega) \cdot F(\omega) \qquad H(\omega) = \frac{X(\omega)}{F(\omega)}$$

Figure 20. The frequency response functions are the transfer functions of a dynamic system. From Ref. [10].

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
Dataline 1: Define excitation and select modes
Dataline 2: Define damping
Dateline 3: Define the response
Dataline 4: Define output parameters
```

4.7.2 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 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`) shall be applied to the nodes where an enforced motion is to be applied by use of the large mass method. Note also that the application of the large masses may alter the eigenfrequencies or mode order, which means that care shall be taken when selecting the eigenmodes.

In 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 3.

Table 3. Overview of excitation input types (*VAD*) for `*FREQUENCY_DOMAIN_SSD`

VAD	LS-DYNA R9	LS-DYNA R10
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
Dataline 1: Select modes to be used in the analysis
Dataline 2: Define damping
Dateline 3: Define output for subsequent acoustic analysis
Dataline 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
```

From R9.0.1[4], 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 *LC3* (position 7 of Dataline 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. Note also that `*DATABASE_FREQUENCY_BINARY_D3FTG` must be specified in order to get 3D – binary plot files for visualization of fatigue analysis results.

[4] Available in R9.0.1 smp only, but in coming versions both smp and mpp implementations will be available.

A basic example of a steady state dynamics analysis of a bracket, including fatigue evaluation, has been attached, see `ssd_fatigue001.key`, and Figure 21.

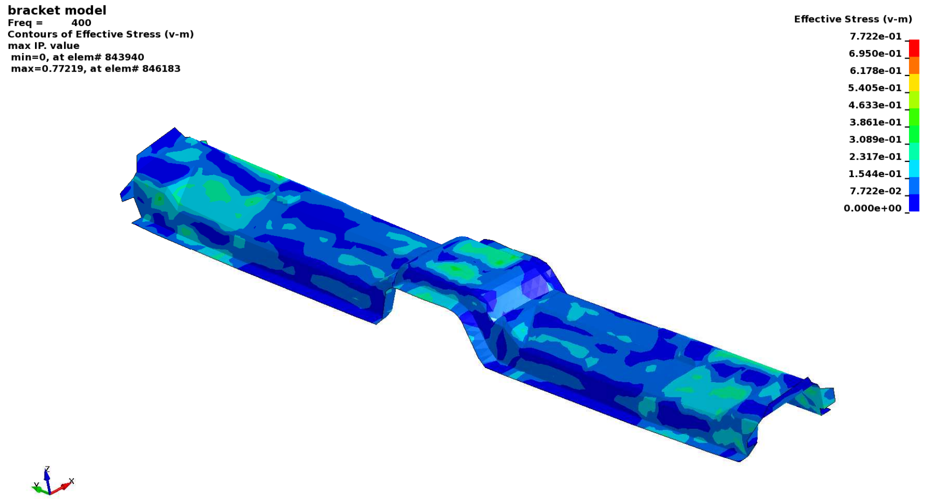


Figure 21. Steady state dynamics analysis of a bracket. The image shows the effective stress (von Mises) response at 400 Hz.

4.7.3 Random vibration and fatigue analyses

Random vibration is motion which is non-deterministic, for example a vehicle riding on a rough road. Structures subjected to random vibrations is usually studied using statistical or probabilistic approaches. The power spectral density (PSD) is commonly used to specify a random vibration process [10]. 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. In general [10], Dirlik's method is considered 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.

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 ASCII files `elout_psd` and `nodout_psd`. Fatigue results are output as stress components in the 3D-binary database `d3ftg`, see Table 4. The irregularity factor is a real number from 0 to 1, where a sine wave has irregularity factor of 1 and white noise has 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 4. Fatigue analysis results are output as stress components in the d3ftg 3D binary database.

Stress component	Fatigue analysis results
X	Cumulative damage ratio
Y	Expected fatigue life
Z	Zero-crossing frequency
XY	Peak-crossing frequency
YZ	Irregularity factor

A template for a random vibration fatigue analysis follows:

```
*FREQUENCY_DOMAIN_RANDOM_VIBRATION_FATIGUE
Dataline 1: Select modes
Dataline 2: Define damping
Dateline 3: Define loading type and the number of PSD loadings and fatigue
data (SN-curves) definitions
Dataline 4: Define exposure time and type of SN-curves
Dataline 5: Define the excitation PSDs (repeat if multiple excitations are
present)
Dataline 6: Define the fatigue data for each PID
*DATABASE_FREQUENCY_BINARY_D3PSD
Define frequency range, spacing etc. for output
*DATABASE_FREQUENCY_BINARY_D3PSD
1,
*DATABASE_FREQUENCY_BINARY_D3FTG
1,
```

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 flag *NFTG* (position 8 of Dataline 3) is set to -999.

In random vibration analysis, special care must be taken when computing the effective stress; it cannot be obtained directly from the stress components. In LS-DYNA, the method of Ref.[13] 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[5].

A basic example of random vibration analysis of a link arm, including fatigue evaluation, has been attached, see keywordfile `set_4.key` and Figure 22.

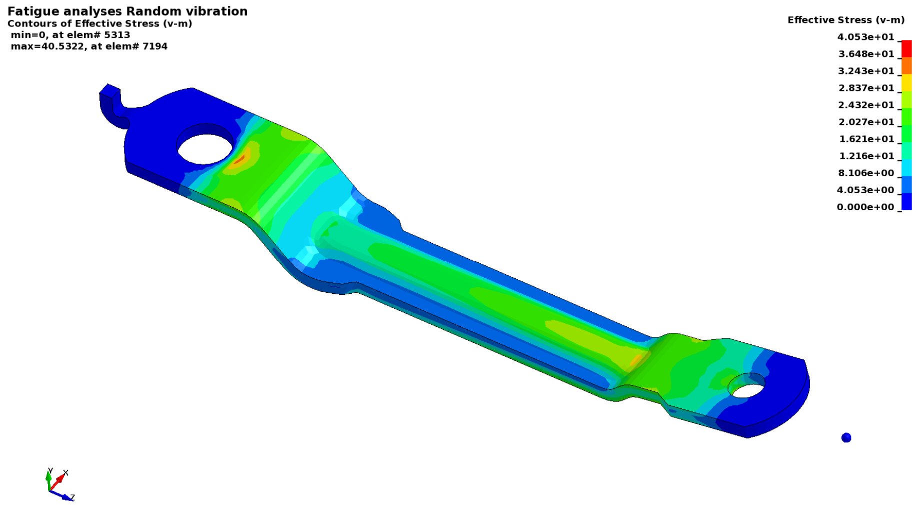


Figure 22. RMS effective stress from random vibration analysis of a link arm.

4.8 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 nonlinear, transient dynamic analyses, but the procedure can also be useful for the initial phase of analyses with "loose" parts, see Section 4.2.2. 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
*INCLUDE
database_cards_dynamic.key
*CONTROL_TERMINATION
Define end time of the simulation
*INCLUDE
Include file defining geometry, materials etc.
```

[5] 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.

```
*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 that energy is conserved. Even for purely transient dynamics simulations, a small amount numerical damping may be very beneficial for convergence reasons, for example by using *GAMMA* = 0.55 and *BETA* = 0.27563. 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, see Table 5.

Table 5. Overview of `GAMMA` and `BETA` settings for different applications.

GAMA	BETA	Comment
0.5	0.25	The default settings. No numerical damping
0.55	0.27563	Moderate numerical damping for transient analysis
0.6	0.38	Suited for quasi-static analyses

4.8.1 Transient loading of a L-beam with contact

This example is very similar to the one of Section 4.6.1, but a fully constrained, rigid transverse support (green in Figure 23 is added). Loads and boundary conditions are the same as in Section 4.6.1. 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 24. The response is quite chaotic at the beginning, but decays quickly due to the combined effect of the damping definitions.

The example keywordfile is `transient_nonlin001.key`.

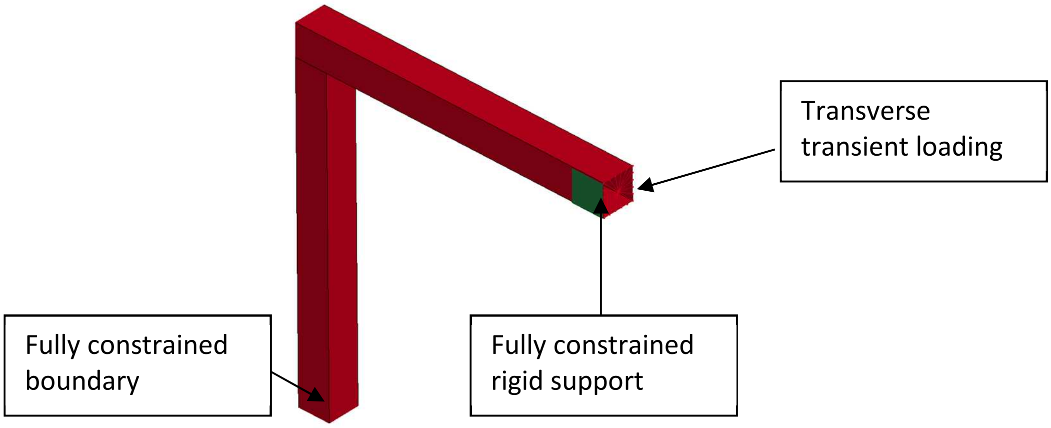


Figure 23. Geometry for the L-beam (red) with rigid support (green).

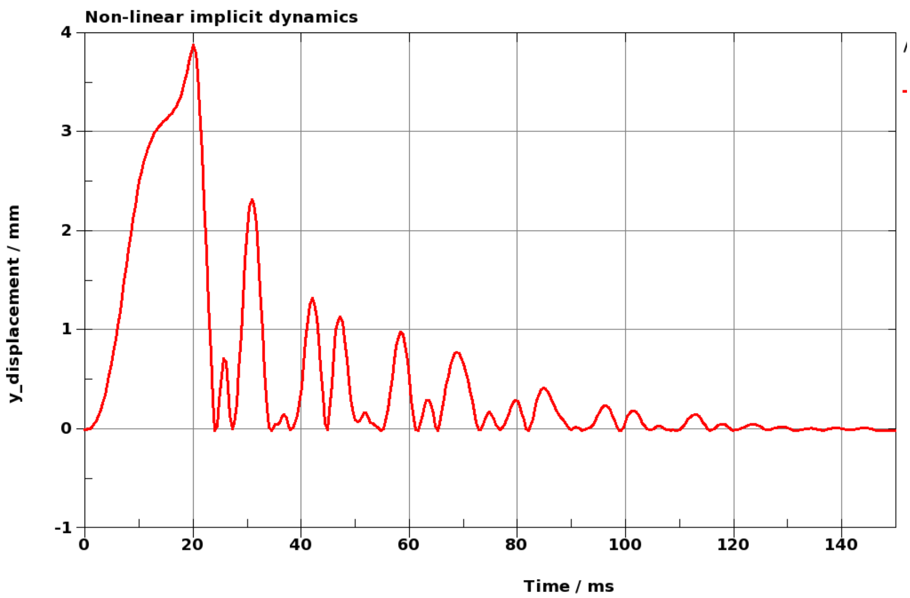


Figure 24. Displacement history in the transverse direction of the loaded node.

5 Element types

In LS-DYNA, many different element types and element techniques are available. This section presents a brief overview of elements that are well suited to implicit analyses, see Table 6. There are several special beam, shell and solid element formulations for linear analyses, see Table 7.

Table 6. Recommended element types for non-linear implicit analyses

Element type	Comment	LS-DYNA keyword	Element formulation
Beam		*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
	2nd order tet		16 (17)
	2nd order hex		23

Table 7. Specialized element types for linear implicit analyses

Element type	Comment	LS-DYNA keyword	Element formulation
Beam	For linear static analyses	*SECTION_BEAM	13(1)
Shell	1st order	*SECTION_SHELL	18 (20, 21, 99)
Solid	1st order hex	*SECTION_SOLID	18 (99)

Notes: (1) This is the only allowed beam type for linear static analyses. All shell and solid element formulations of Table 6 can also be applied in linear static analyses.

An additional functionality of *CONTROL_IMPLICIT_EIGENVALUE (see also Section 4.5) is to specify element types for implicit analyses. This makes it possible to very quickly switch element formulations 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[6]
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.

---

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

Element formulation 9 can also be used for applying pre-tensioning using `*INITIAL_AXIAL_FORCE_BEAM`.

For linear static analyses, beam element formulation 13 is the only available option.

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 to 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`).

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 4.2.1 and 4.4.1. From R9 of LS-DYNA, a strongly objective version of this shell element is introduced[7].

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, which makes it suited for benchmarking.

Also 2nd order shell elements are available. Element formulation 23 is an 8-node quadratic quadrilateral shell, and element formulation 24 is a 6-node quadratic triangular shell.

---

[7] Element formulation -16. For implicit analyses, element formulation 16 is automatically switched to -16 if *IACC* = 1 on `*CONTROL_SOLUTION` (active in the attached include files),

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[8].

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 $NIP \geq 3$, and activating Lobbato 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 ($NIP \geq 5$).

A general review of shell element formulations in LS-DYNA is presented in Ref. [15].

5.3 Solid elements

For linear analyses, the linear solid hexahedron element formulation 18 is recommended.

Recommended first order elements hexahedra for non-linear analyses is solid element formulation -2. 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 [6]. It is possible to mix first 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 also available as element formulation 10 and 13. It shall 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 [11], 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 in both parts (the remedy may then be to switch to elform 10 in the rigid part).

Two different types of 2nd - order, 10-node tetrahedra are available: element formulation 16 and 17. Elform 16 is the current recommendation for 2nd order tet elements. Elform 17 is slightly costlier and may be more sensitive to negative volume under large deformations but has the advantage that applied positive nodal forces corresponds to positive pressure.

Solid element formulation 23 provides a 20-node, 2nd – order hex element (but there is no matching 2nd – order penta element).

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

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 post-processing. For shells, the average element stresses and strains at the top, mid and bottom surface 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`. 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`.

6 Contacts for implicit analyses

Sliding contacts in LS-DYNA are of penalty type. This makes it important to avoid (unintentional) initial penetrations between parts or elements of the FE-model, which would otherwise introduce large contact forces. It is strongly recommended to use the capabilities of for example ANSA or LS-PrePost to check for and fix unintended initial penetrations. It is also recommended to check that the tied contacts properly connect the involved parts as intended.

For a general overview of contact definitions in LS-DYNA, the course "Contacts in LS-DYNA" (or similar) is recommended.

6.1 Sliding interface contact

The purpose of sliding contact is to allow movement between parts without the parts intersecting each other. There is no "small sliding" option on contacts in LS-DYNA, all contacts are always finite sliding. The Mortar [7] contacts are recommended for modelling sliding contact in implicit analyses. Note that the input syntax for the Mortar contacts has been slightly changed from R10 of LS-DYNA, see Section 6.1.4. Note that 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 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, with a clear master-slave relation, and self-contact need not be considered, the contact type `*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_ID` [7] is recommended. The mortar - type contacts are segment based, using a penalty formulation, and specially developed for implicit analyses. It is a very general contact, which can handle edge-to-edge contacts for shells and solids, as well as beam-to-beam and beam-to-edge situations. It is generally recommended to use the "softer" part as slave. A template for using the Mortar surface-to-surface contact (versions before R10) follows:

```
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_ID
Define contact ID, Heading
Dataline 1: Define what shall be in contact
Dataline 2: Define friction
Dateline 3: Define alternative penalty stiffness (SFS, SFM) and
thicknesses (SST) or leave blank to get defaults
Dataline 4: blank / optional
Dataline 5: blank / optional
Dataline 6: optional / Define IGAP and IGNORE (MPAR1, MPAR2)
```

If only solid elements are involved in the contacts, a “contact thickness” for these may be specified using the *SST* - parameter on dataline 3. 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 25. The relative penetration distance, used for calculating the penalty force, is related to this distance (see Figure 26). This means that by reducing *SST*, a stiffer contact (between solid parts) can be obtained, but at the same time the risk that a slave node will be released from the contact increases. It is important to note that the *SST* – parameter for solids specify 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. It shall be noted that if both shell and solid parts are involved in a surface-to-surface contact, the solid parts must be on the slave side, if the intension is to use *SST* for specifying the contact thickness of the solids, otherwise the contact thickness of the shell parts will be modified.

If *SST* is left blank, LS-DYNA automatically calculates a value for the contact thickness for solids. This normally is a good starting point, and works well, provided that the mesh quality in the contact surfaces is reasonably good.

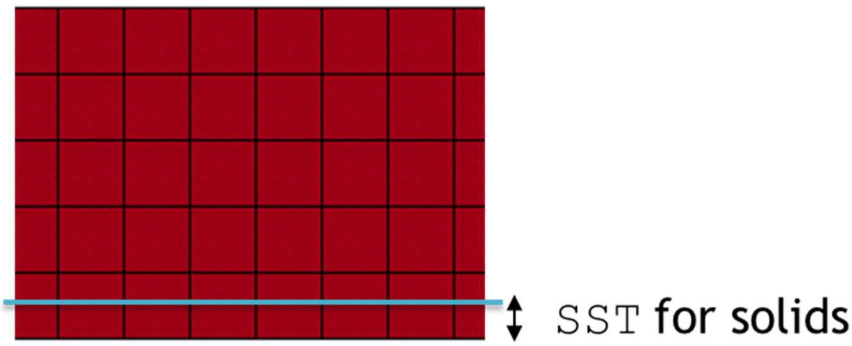


Figure 25. Use of the *SST* – parameter to define the contact thickness for solid parts.

Datalines 1 - 3 must be defined, while datalines 4 - 6 are optional. Note that 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. It is up to the user to verify that the penetration is of acceptable magnitude. In cases involving solids-to-solids contact, the default settings of the mortar – contact may give noticeably large penetrations. Increasing the penalty stiffness by setting (for example) *SFS* = 5.0 and *IGAP* = 5 (see below) may remedy this, but also note that an increased penalty stiffness may make convergence harder.

For the mortar contact, the *IGAP* - parameter determines how steeply the penalty force is ramped up for “large” penetrations, see Figure 26.

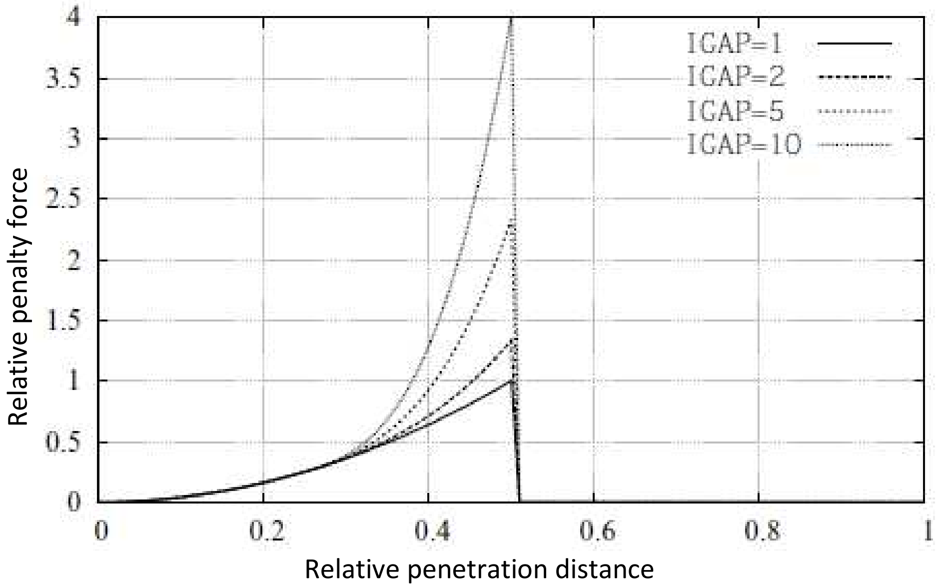


Figure 26. Mortar penalty force as a function of penetration (schematic). Image from Ref. [1].

The *IGNORE* parameter can to some extent be used to allow initial 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, first contact will occur at the physical surface of the parts, see Figure 27. The option *IGNORE* = 2 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 28.

The options *IGNORE* = 3, 4 can be used for resolving initial interferences, such as press-fit. The penetrations will then be resolved linearly between t = 0 and t = *MPAR1*, as specified on dataline 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 dataline 6), which shall be at least as large as, and on the order of, the maximum initial penetration.

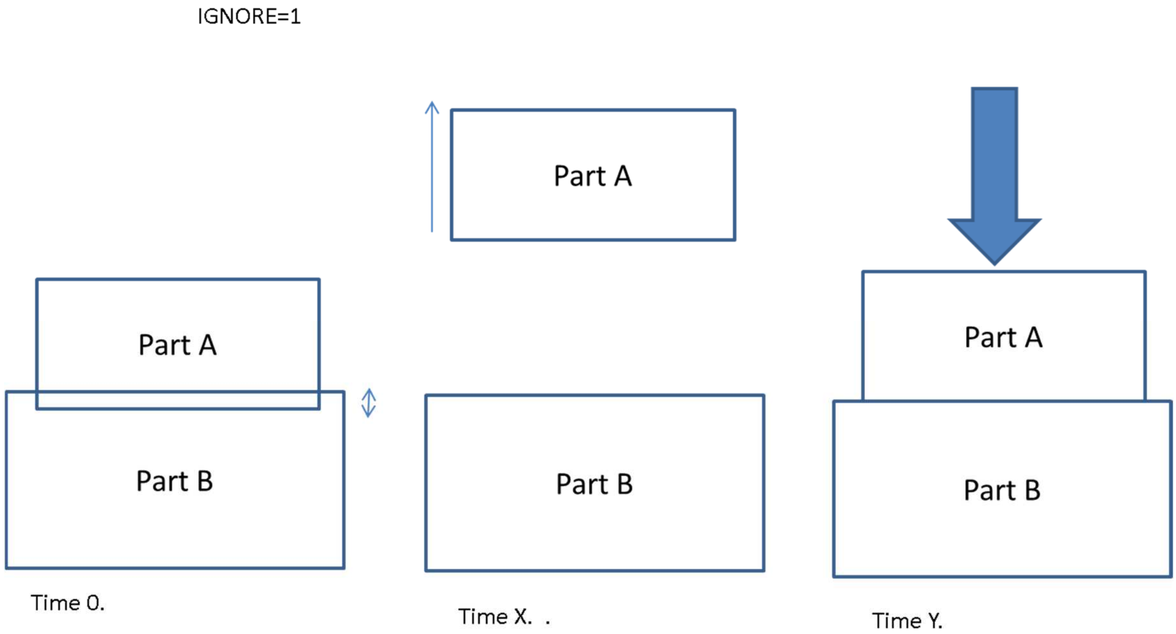


Figure 27. Schematic illustration of IGNORE = 1. Parts A and B are initially penetrating at Time 0. At Time X, the parts are separated, for example by prescribed displacements. At Time Y, the parts are brought back into contact. Contact then occurs at the physical surface of the parts.

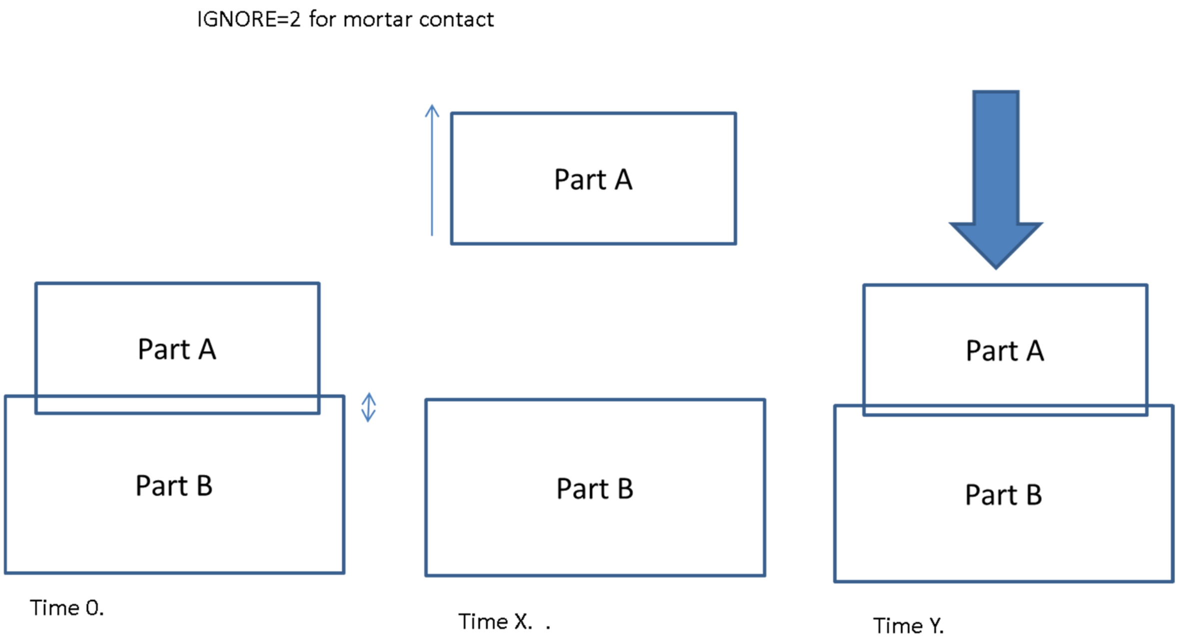


Figure 28. Schematic illustration of IGNORE = 2 for mortar contacts. Parts A and B are initially penetrating at Time 0. At Time X, the parts are separated, for example by prescribed displacements. At Time Y, the parts are brought back into contact. Contact then occurs at the original intersecting surface 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`.

Contact forces between certain parts of a 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.

A template for using the automatic single-surface mortar contact (versions before R10) follows:

```
*CONTACT_AUTOMATIC_SINGLE_SURFACE_MORTAR_ID
Define contact ID, Heading
Dataline 1: Define what shall be in contact (only slave)
Dataline 2: Define friction
Dateline 3: Define alternative penalty stiffness (SFS)and thicknesses
(SST) or leave blank to get defaults
Dataline 4: blank / optional
Dataline 5: blank / optional
Dataline 6: optional / Define IGAP and IGNORE
```

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

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

It is in general 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 Changes in the Mortar contacts from R10

From R10 (rev. 118243) of LS-DYNA, the input for the Mortar contact has been modified. The "contact thickness" for solids (cf. Figure 25) is now defined using *PENMAX* on optional card B. This means that from R10, it is possible to define the contact thickness for shells and solids separately, which makes it possible to mix shells and solids (in the same part set, for example for a single surface contact definition) and still give a "contact thickness" for the solid elements. A template for using the automatic single-surface mortar contact (versions R10) follows:

```
*CONTACT_AUTOMATIC_SINGLE_SURFACE_MORTAR_ID
Define contact ID, Heading
Dataline 1: Define what shall be in contact (only slave)
Dataline 2: Define friction
Dateline 3: Define alternative penalty stiffness (SFS)and thicknesses
(SST) for shells, or leave blank to get defaults
Dataline 4: blank / optional
Dataline 5: Define contact thickness for solids (PENMAX)
```

Dataline 6: optional / Define IGAP and IGNORE

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 29. This behavior is controlled by the *IGAP* - parameter on dataline 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. Setting *SOFT* = 1 is not recommended for implicit analyses.

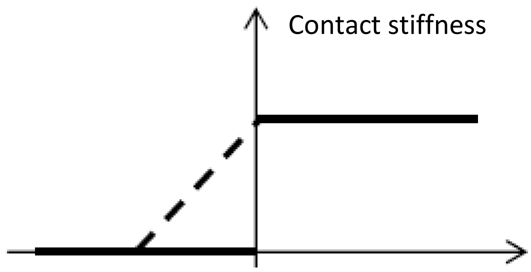


Figure 29. Illustration of contact stiffness as a function of penetration for IGAP = 1 (dashed line) and IGAP = 2 (solid line).

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[9] 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 linear implicit analyses

Non-mortar sliding contacts will (in some cases) be linearized in linear 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 are initially open, no contact forces will be transferred, see Figure 30.

[9] Shell structures involving metallic materials may work reasonably well with *RIGIDWALL in implicit, while for example plastic components meshed with solid elements most likely will end up with unrealistically large penetrations.

The linearization procedure for non-mortar contacts can lead to overestimation of eigenvalues, if for example an FE-model is taken from an automotive crash simulation, where typically initial penetrations may exist, and non-mortar contacts are used, which then introduce linearized contacts in (some parts of) the model. For cases like this, it is recommended to remove non-mortar contacts from the eigenvalue analysis. Single-surface contacts may be transformed into a tied contact, using `*CONTACT_AUTOMATIC_SINGLE_SURFACE_TIED`, when desired.

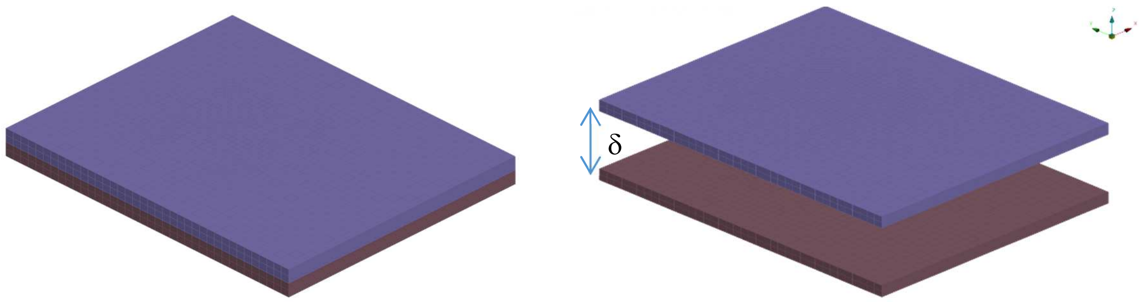


Figure 30. The left image shows two contact surfaces that are initially touching. For this case, the surfaces will be tied in the normal direction but free to move in the tangential direction. The right image shows two contact surfaces with an initial gap. For this case, no contact forces will be transferred in a linear analysis.

The Mortar sliding contacts do not transfer forces during purely linear analyses.

All sliding contact types will be respected during intermittent eigenvalue analyses and intermittent linear buckling analyses, see also Section 4.5.2 for an example.

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

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_...`[10]) 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.

By adding the option `OFFSET` to these contacts (for example `*CONTACT_TIED_NODES_TO_SURFACE_OFFSET`), a penalty-based tie formulation is

[10] By setting *IACC* = 1 on `*CONTROL_ACCURACY` (active in the attached include files), a strongly objective formulation of these contacts is activated, see §29.10 of Ref. [12] for further details.

obtained. This is required when rigid nodes/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 tie contact for the rigid nodes/segments by setting *IPBACK* > 0 on the optional card E (Dataline 8). By this, LS-DYNA will first try to use kinematic constraints to tie the slave nodes, and for those nodes who fail due to "rigid segments", a penalty-based constraint is created.

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

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

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

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

It is possible for the user to explicitly specify the tie distance by entering a negative value for the *SST* and *MST* - parameters on dataline 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 pre-processor such as ANSA or LS-PrePost, in order to assure that the nodes get tied as intended. It is also recommended to check in the messag (mes00*) file(s) for warning messages for nodes that are not being tied.

If a slave 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 master 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. If the OFFSET option is active, the slave nodes will not be moved. If a constrained-based contact is desired, that does not move the slave nodes, the can option CONSTRAINED_OFFSET be used. In addition, this option will also account for the moment created due to separation between slave nodes and master segments. An overview of the most common tied contact options is presented in Table 8.

An alternative to *CONTACT_TIED_NODES_TO_SURFACE is the SURFACE_TO_SURFACE – type. For this later type, segment sets may be specified for the slave side, which is not allowed for the NODES_TO_SURFACE – type. For the NODES_TO_SURFACE – type, node sets may be specified for the slave side, which is not allowed for the SURFACE_TO_SURFACE - type.

Note that if the model consists solely of solid elements, *CONTACT_TIED_SHELL_EDGE_TO_SURFACE_{*OPTION*} cannot be used.

Table 8. 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 slave nodes within the search distance will be moved. (2) Due to separation between slave nodes and master segments. Only applies to the `_OFFSET_` option. (3) All available degrees of freedom, depending on element type of the slave node.

An example of a tied contact definition follows:

```
*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET_ID
Define contact ID, Heading
Dataline 1: Define what shall be tied
Dataline 2: Blank
Dateline 3: Define alternative tie distance via SST, MST < 0 (optional,
can be blank line)
Dataline 4-7: Optional / blank
Dataline 8: Optional / define IPBACK (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.

6.2.1 Making contact surfaces stick

In some situations, it is desired to have two contact surfaces stick to each other as soon as they come in to contact: initially the contact surfaces are separate 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 using the contact option `*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK_ID with` *OPTION* = 1 on Dataline 5. 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
Dataline 1: Define what shall be tied
Dataline 2: Define friction (optional)
```

```
Dataline 3: Blank
Dateline 4: Set OPTION = 1
Dateline 5 - 6: Blank / optional
Dateline 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 31.

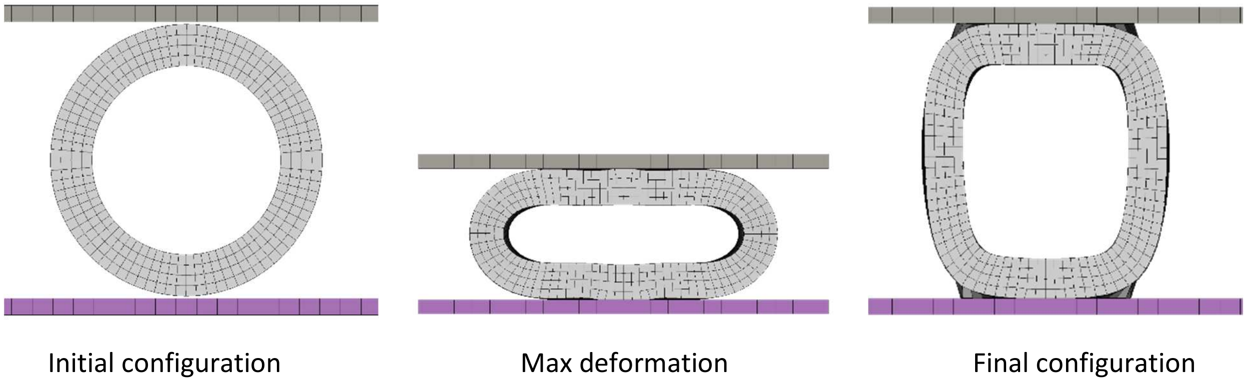


Figure 31. Example of a rubber tube being compressed by two rigid plate using tiebreak contact with option = 1.

6.3 Contact output

Contact forces between master and slave surfaces are output in the `rcforc` file. This is a tabular result, which can be used for 2D - post-processing. It is also possible to obtain output of contact gaps and contact pressure for 3D - visualization. Then, the following steps must be followed:

- On the dataline 1 of the contact definition, set the parameters `SPR` and `MPR` to one.
- When submitting the LS-DYNA job, the command line option **`s=d3iff`** must be added. (Any arbitrary unused filename can be chosen).
- Also, the keywords `*DATABSE_BINARY_INTFOR` and `*DATABSE_EXTENT_INTFOR` are required to specify the 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`.

Constraint-based tied contacts will not output contact pressures into the interface force file, 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 de-bugging convergence problems related to contacts.

7 Material models

This Section contains 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. Unsupported materials include for example `*MAT_ARUP_ADHESIVE`. If unsupported materials are used, LS-DYNA will print an error message in the `d3hsp`-file and terminate.

Since the simulation time in a static analysis most commonly is a parameter that does not correspond to physical time, computed strain rates will be unphysical, and switching off the rate effects is then a sensible choice. This can be achieved by setting *`IRATE`* = 1 on `*CONTROL_IMPLICIT_DYNAMICS`. This flag will have effect, even if the dynamic analysis option is inactive (*`IMASS`* = 0).

The material model `*MAT_PIECEWISE_LINEAR_PLASTICITY` (or `MAT_24`) is commonly used in LS-DYNA explicit analyses to characterize metallic materials with plasticity. In releases of LS-DYNA prior to R9.0.0, this material model is not so well suited, since it is optimized for performance in explicit analyses: only one radial return iteration was performed. But from R9.0.0, also `MAT_24` performs well for implicit analyses[11].

A more general alternative to using `*MAT_24`, if anisotropy or kinematic hardening is of interest, is `*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.

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.

---

[11] With *IACC* = 1 on *CONTROL_SOLUTION (active in the attached include files).

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 stresses are of interest in a linear analysis, it is recommended to use an elastic material model, such as *MAT_ELASTIC.

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 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 further Appendix A. 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, *MAT_THERMO_ELASTO_VISCOPLASTIC_CREEP (MAT_188) is recommended.

8 Other implicit analysis types

There are many additional implicit analysis types available in LS-DYNA which are not currently described in the present document. In coming revisions of this document, rotational dynamics (by *CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS [16]) will also be treated. Thermal analyses will also be described.

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.

9 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.
- The inclusion of geometric stiffness (*IGS* = 1 on *CONTROL_IMPLICIT_GENERAL) can sometimes inhibit convergence, especially if (nearly) incompressible materials are present. If convergence problems occur, it might help to turn off the geometric stiffness effect by setting *IGS* = 2 on *CONTROL_IMPLICIT_GENERAL.
- 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.

- For large deformation (relative to the model dimensions) the provided convergence settings may be too relaxed. Indications of this may be for example spurious oscillations in contact forces or unbalance between applied forces and reaction forces. One remedy in such cases may be to set *DNORM* = 1 and increasing *DCTOL* to for example 0.01 on `*CONTROL_IMPLICIT_SOLUTION`.
- 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 a more extensive troubleshooting guide, see Appendix C.

10 References

[1] Livermore Software Technology Corporation, LS-DYNA keyword user's manual Volume I, Livermore 2017 (see also http://lstc.com/download/manuals).
[2] Livermore Software Technology Corporation, LS-DYNA keyword user's manual Volume II, Livermore 2017.
[3] Livermore Software Technology Corporation, LS-DYNA keyword user's manual Volume III, Livermore 2017.
[4] Borrvall, T., *et al.*, Implicit analysis in LS-DYNA (course material), Dynamore Nordic AB, Linköping 2013.
[5] Huang, Y., *et al.*, Development of frequency domain dynamic and acoustic capabilities in LS-DYNA, 8th European LS-DYNA Users Conference, Strasbourg, 2011, http://www.dynalook.com/8th-european-ls-dyna-conference/session-23/Session23_Paper3.pdf
[6] 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
[7] 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
[8] Lilja, M., Benchmark of LS-DYNA for offshore applications according to DNV Recommended Practice C208, 13th International LS-DYNA Users conference, 2014.
[9] Huang, Y., Cui, Z., Frequency domain analysis in LS-DYNA, Oasys LS-DYNA UK User's Meeting, 2013-01-16.
[10] Huang, Y., NVH and frequency domain analysis with LS-DYNA (Course material), LSTC, 2017.
[11] Erhart, T., Review of solid element formulations in LS-DYNA, LS-DYNA Forum 2011, Internet source: http://www.dynamore.de/de/download/papers/forum11/entwicklerforum-2011/erhart.pdf
[12] Livermore Software Technology Corporation, LS-DYNA Theory manual, Livermore 2016.
[13] 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
[14] Forsberg, J., LS-DYNA recommended settings explicit analyses, Dynamore Nordic Document, 2016-06-17
[15] 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

[16] 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

11 Revision record

Rev. no	Release date	Author	Description
1	2014-02-17	Anders Jonsson	First revision
2	2014-04-02	Anders Jonsson	Added frequency domain analyses (not complete)
3	2014-04-11	Anders Jonsson	Added frequency domain analyses
4	2014-06-26	Anders Jonsson	Added info on press-fit / interference contacts
5	2014-09-11	Anders Jonsson	Minor updates to elements section
6	2014-11-24	Marcus Lilja	Appendix A: Rubber modelling added to document
7	2015-06-03	Anders Jonsson	Appendix B: Restart of analyses added to document
8	2016-03-30	Anders Jonsson	Removed failure from restart examples, minor modifications of freq. domain
9	2016-11-17	Anders Jonsson	Updates for LS-DYNA R9.0
10	2017-02-01	Anders Jonsson	Changes to contact and material section
11	2017-06-28	Anders Jonsson	Added Section 6.2.1, Appendices C, D and E.
12	2017-10-02	Anders Jonsson	Added Section 6.1.4 and other modifications for R10.
13	2018-02-13	Anders Jonsson	Minor corrections and modifications

12 Appendix A: Rubber modeling for implicit analysis

12.1 Background

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

In many cases, the default or general recommended settings do not apply for simulation of rubber materials. The purpose of this document is that it shall be used 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.

12.2 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 experience 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.

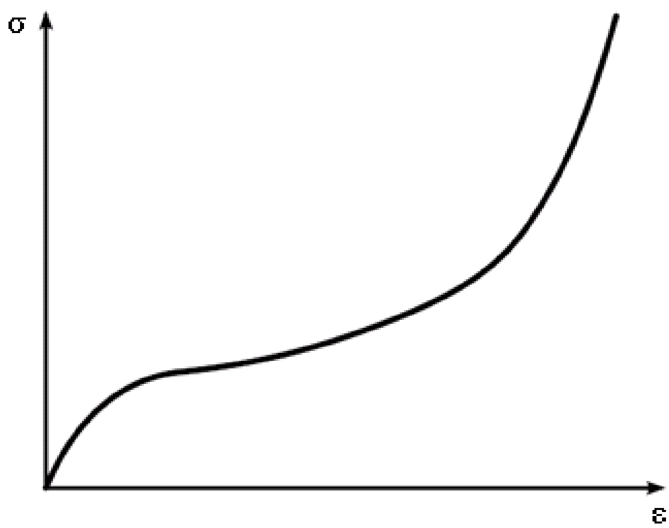


Figure 32. General rubber response curve.

12.2.1 *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 33. 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 such case only the stability of the material model.

The user may also use data from tests to fit the parameters (C_{nn}) to the test curve. This is available when n>0, see Figure 34. The user then specifies sgl, sw, st and lcid according to Ref. [2].

```
*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 33. 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 34. 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 :

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

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

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

12.2.2 *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. 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 and other, so correct test data are crucial to get the model to converge.

```
*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 35. Example of simplified rubber/foam material card.

12.2.3 *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 36. 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 32. 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 of 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 36. Mooney-Rivlin rubber.

The strain energy function is defined as:

$$
\begin{aligned}
&\mathrm{W} = \mathrm{A}(\mathrm{I}-3) + \mathrm{B}(\mathrm{II}-3) + \mathrm{C}(\mathrm{III}^{-2}-1) + \mathrm{D}(\mathrm{III}-1)^2 \\
&\text{Where:} \\
&\mathrm{C} = 0.5\mathrm{A} + \mathrm{B} \\
&\mathrm{D} = \frac{\mathrm{A}(5\nu - 2) + \mathrm{B}(11\nu - 5)}{2(1 - 2\nu)} \\
&\nu = \text{Poisson's ratio} \\
&2(\mathrm{A}+\mathrm{B}) = \text{Shear modulus of linear elasticity} \\
&\mathrm{I}, \mathrm{II}, \mathrm{III} = \text{Invariants of right Cauchy - Green Tensor C}
\end{aligned}
$$

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

12.3 Elements

12.3.1 Structural elements

The recommended element types for rubber modeling are:

Hexahedral: -1, -2

Tetrahedral: 13 (10)

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 has shown to be more efficient than elform -2 in most rubber cases.

Elform 13 was developed to avoid volumetric locking by applying nodal pressure averaging. The tangent stiffness may in some cases cause problems when elform 13 is used in implicit analyses. If so, use elform 10 instead.

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

12.3.2 Element free methods

If the mesh becomes very distorted it may be hard to get a converged solution using structural solid elements. In such 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 for using this method are that it may be computational expensive and that more work may be needed in order to get contact regions to converge.

In current versions of LS-DYNA (R9.2, R10.0) only the tet form (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
```

12.4 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.

12.5 Solver settings

The solver set-up 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.

- In some cases, it may be necessary to de-activate the geometric stiffness (set IGS = 2 on `*CONTROL_IMPLICIT_GENERAL`) in order to obtain convergence.

12.6 Examples

The following example problems are included in this document:

- Insert_and_interference.key
  Initial penetrations are removed 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.

If these keywordfiles are missing, please contact DYNAmore Nordic AB.

13 Appendix B: 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 (static) load cases. For these situations, it may 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 three different possibilities for restart analysis, se Appendix "Restart input data" of Ref. [1]. In the following, only the "small restart" and "full restart" options will be discussed.

It is possible to get restart files (`d3dump` and `d3full`) 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.

13.1 Small restart

A "small restart" means that only a limited amount 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 load curves are defined in the keywordfile for the pre-loading, but with zero magnitudes. Then, in the following "small restart" analyses, the curves are re-defined 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 keywordfile (`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 stat 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
```

13.1.1 Bolt pre-tensioning followed by prescribed loading

In the first stage, bolt pre-tensioning is applied to a bolted flange joint, see Figure 37. At one end, the pipe (green in the figure) 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 figure). Non-linear material properties are used both in the pipes and the bolts. The example keywordfile is `pretens001.key`. This keywordfile also includes the definitions of the applied loadings.

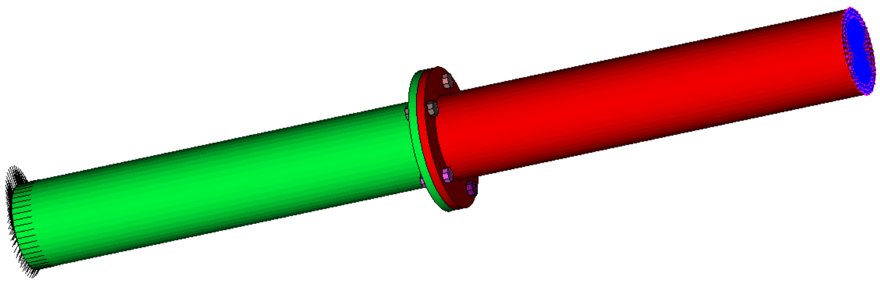


Figure 37. Pre-tensioning of a bolted flange joint.

In the following analysis (`small_restart.key`), loading is applied in the Y-direction, by re-definition of the load curves using the keyword *CHANGE_CURVE_DEFINITION. The result is shown in Figure 38.

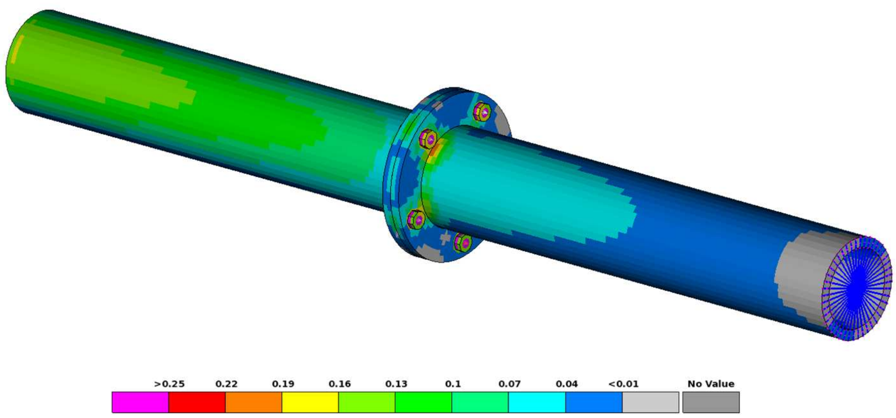


Figure 38. Effective stress due to 10 kN applied transverse loading.

13.2 Full restart

In a "full restart", a complete re-definition of the simulation model is possible, including loads, boundary conditions, contacts etc. 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 is 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 "full restart" feature works from R8.0.0 of LS-DYNA. 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
```

13.2.1 Bolt pre-tensioning followed by prescribed displacement

The same bolted joint as in Section 13.1.1 is studied again, 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 40, and the moment vs. tip displacement curve is shown in Figure 39.

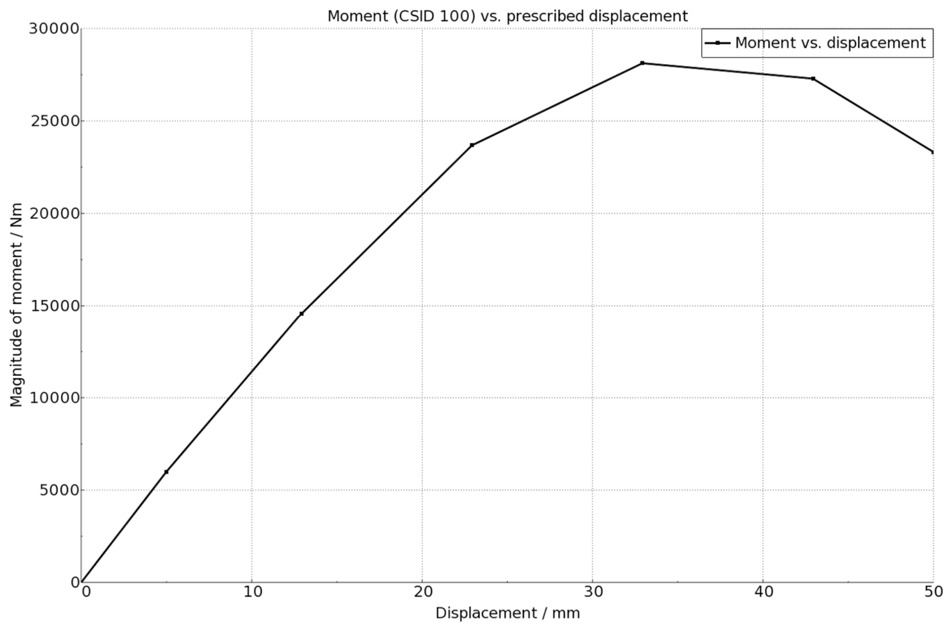


Figure 39. Cross-sectional moment in the lower pipe as a function of applied tip displacement.

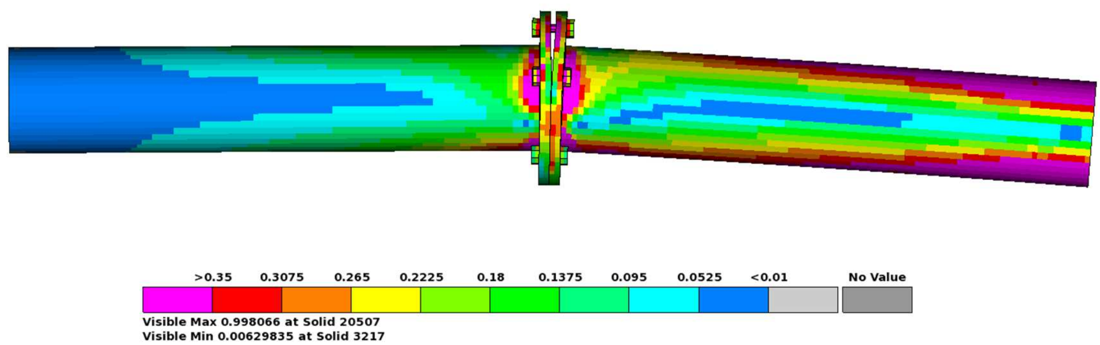


Figure 40. Effective stress due to 50 mm prescribed tip displacement.

14 Appendix C: 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.

The purpose of this Appendix is to present some tips on how to identify and possibly circumvent convergence problems in implicit analyses in LS-DYNA.

- **Start with the recommended control card settings** (attached to this document, or from other sources, or your private preferences). The many options and possible combinations of control card settings in LS-DYNA in general makes it tempting to modify these in order to obtain convergence, or assume that the convergence problem is caused by some LS-DYNA control card setting. However, before making any modifications of the standard control card settings, it is strongly recommended to first investigate other possible reasons for the convergence problems.
- Check the model integrity.
    - Inspect the model carefully with respect to element quality; small (or even negative) initial volume of solid elements, and small Jacobian values.
    - Check that no unintended cracks in the mesh are present.
    - Check that the element connectivity is valid (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).
    - Most pre-processors, like Primer, LS-PrePost or ANSA, have built in tools for basic model checking – use these tools (even though no perfect tool for automatic model checking exist, some useful hints may still be obtained).
- Check that consistent units are used in the model, for materials, loadings, accelerations, frequencies etc.
- 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 DOFs are required, use joints (`*CONSTRAINED_JOINT_...`) instead.
- Check the material models.
    - Avoid using `*MAT_ELASTIC` with $\nu \approx 0.5$ for modelling rubber, at least for finite deformations. Use `*MAT_HYPERELASTIC_RUBBER` or `*MAT_SIMPLIFIED_RUBBER_FOAM` instead (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. 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.
    - Note! Some material models, like `*MAT_ARUP_ADHESIVE`, are currently not supported in implicit.

  - Avoid involving parts with *MAT_NULL in tied contacts.
  - If *MAT_NONLINEAR_ELASTIC_DISCRETE_BEAM (*MAT_067) is used in the model, try switching to *ELEMENT_DISCRETE with *MAT_SPRING_NONLINEAR_ELASTIC (*MAT_S04).

- Check and correct unintended initial penetrations in contacts.
  - Most pre-processors have built-in tools for checking and correcting initial penetrations.
  - The Mortar contacts will report initial penetrations in the mes0000 – file.
  - Use *IGNORE* = 3 or 4 of the Mortar contact[12] to resolve press-fits or other intended initial penetrations.
- Check tied contacts.
  - Make sure that the intended slave nodes get tied. Not too many (which may result in a surprisingly rigid behaviour), not too few (which may for example cause unintended rigid-body modes).
  - Most pre-processors have built-in tools for checking tied contacts.
  - *CONTACT_TIED_... will report nodes that are not tied in the d3hsp – file.
- 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 re-try the step with a smaller step size if convergence fails (due to a too big initial step size, or other reason). It cannot increase step size (even though this in some cases resolves the convergence problem).
  - Should a very small initial time step be desired, remember to also decrease *DTMIN* on *CONTROL_IMPLICIT_AUTO, so that LS-DYNA is given the possibility to do at least one halving of the time step if a re-try is necessary.
  - In some cases, for example transient dynamic analyses, a very small time-step size may be desired in order to resolve a rapid event. Then also decrease *DTMIN* on *CONTROL_IMPLICIT_AUTO, to perhaps 1E-2 times the smallest desired time step.
- Negative eigenvalue warnings (look for

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

  in the d3hsp file) are most likely due to rigid-body modes or elements that get severely distorted (for example rubber) during deformation.

  - Check for rigid-body modes by performing an eigenvalue analysis (*CONTROL_IMPLICT_EIGENVALUE), see Section 4.5.
  - Add appropriate boundary conditions, or
  - use implicit dynamics, *CONTROL_IMPLICIT_DYNAMICS, (or in some cases inertia relief, *CONTROL_IMPLICIT_INERTIA_RELIEF) to properly handle rigid body modes. Note that dynamic effects can be ramped down, in case the rigid body modes are only present initially.

---

[12] Or alternatively *CONTACT_SURFACE_TO_SURFACE_INTERFERENCE.

    - In many cases LS-DYNA may still find static equilibrium, even when rigid-body modes are initially present in the model. Then it is important to remember that this solution is not unique.
- The `d3iter` - file in many cases provides very useful (and visual) information regarding 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.
    - By setting *RESPLT* = 1 on Card 4 `*DATABASE_EXTENT_BINARY`, it is possible to fringe plot the force residual from the `d3iter` binary database. This can also be very useful for pin-pointing the areas of the model where convergence is the hardest.
- Relaxing the penalty stiffness of the (Mortar) contacts in the model is normally beneficial for convergence, since it will give a smoother response from the model.
    - The Mortar contact will report both relative as well as absolute values of the maximum penetrations during the simulation in the `mes0*` - files. Based on this the user can judge what is acceptable in terms of penetration distance.
- Is the physical problem in fact unstable? Will collapse / buckling / bifurcation occur?
    - Try switching from load to displacement control.
    - Activate the arc-length solver (see Section 4.4)
    - Activate implicit dynamics (see Section 4.8)
- For analyses involving rubber (or other incompressible materials) try disregarding the initial geometric stiffness effect by setting *IGS* = 2 on `*CONTROL_IMPLICIT_GENERAL`.
- Switching to a full-Newton solution scheme may be efficient for solving highly non-linear problems.
    - Set *ILIMIT* = 1 and increase *MAXREF* to 30 – 60 on `*CONTROL_IMPLICIT_SOLUTION`.
- 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.
    - Set *LCPAK* = 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 (see Figure 41), or unsatisfied global equilibrium (for example if `spcforc` doesn't add up to the external forces).
    - Try tightening the tolerances slightly on `*CONTROL_IMPLICIT_SOLUTION`, for example by setting *DCTOL* = 5.E-4.
    - In case of large global displacement, try switching to *DNORM* = 1 on `*CONTROL_IMPLICIT_SOLUTION`, possibly in combination with increasing *DCTOL*.

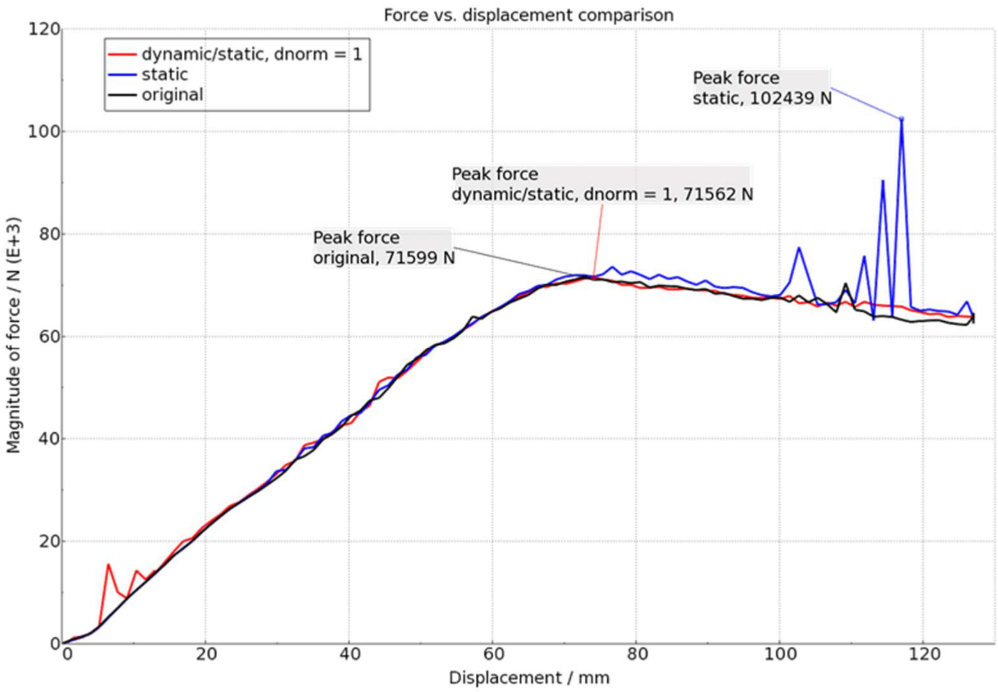


Figure 41. The blue line shows an example of contact force history from "too easy" convergence (in this case Normal termination was reached, but the obtained force-displacement curve was of little use). Since the global displacement in this case was large, the remedy was to switch to *DNORM* = 1 (red line).

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

The main information source for tracking the implicit convergence are the `d3hsp` and `mes0*` files. In the `mes0*`– files, the convergence history (displacement, energy and residual norms), line search information, and before each time step, penetration information from the Mortar contacts, are output. The Mortar contacts will issue warnings in the `mes0*`-files, search for

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

The warning means that the slave nodes have penetrated too much and may be released, which may make convergence impossible in the following time step. To remedy this, check and fix initial penetrations, and then increase penalty stiffness (set for example *SFS* = 5, *IGAP* =5).

Some clues may be found from tracking the convergence info, see Figure 42. Ideally, the norms should decrease monotonically. In cases when the norms oscillate from high to low to high again ("jump up and down") between the iterations, this is an indication that it might be the contacts that are causing the convergence problems. In such cases, it might help to relax the Mortar contact stiffness (remember to monitor the penetrations!).

If the norms decrease, but slowly, it might be worth trying full Newton, by setting *ILIMIT* = 1 and *MAXREF* to 30 – 60 on `*CONTROL_IMPLICIT_SOLUTION`.

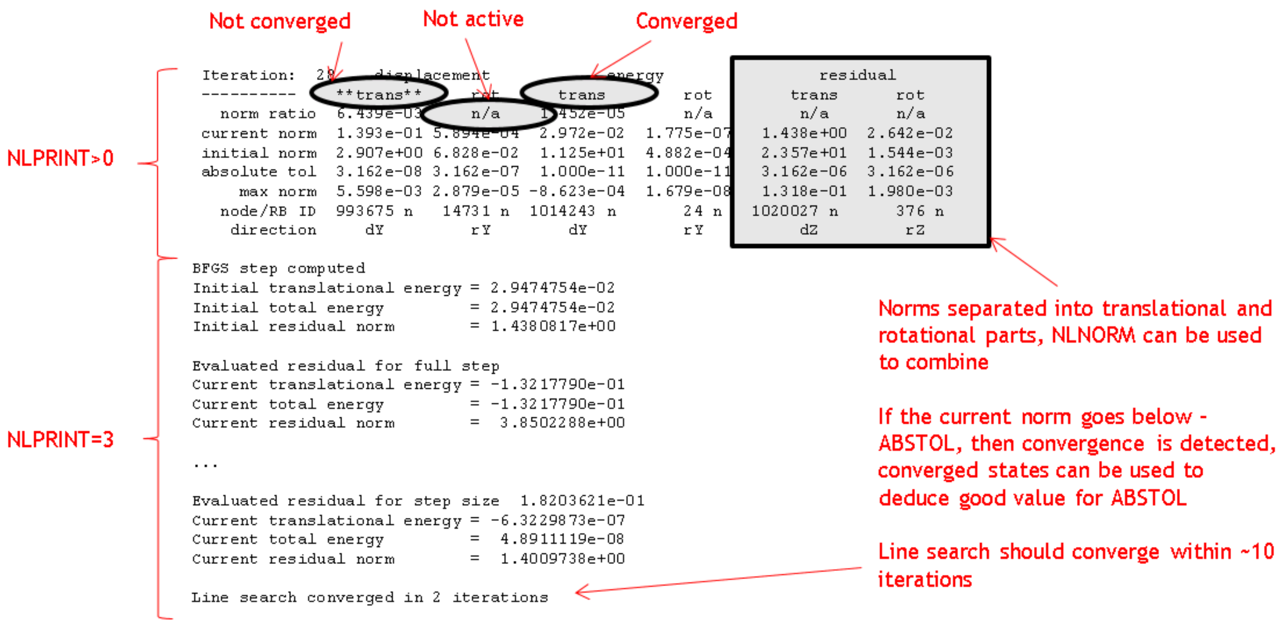


Figure 42. Tracking convergence information in the mes0* - files. If the norms in the table header are surrounded by "***" this means that the norm is not converged.

Typically, the line search should converge in less than 10 iterations. In cases when the line search takes significantly more than 10 iterations, and the residuals remain high even with very small step sizes, this may be due to that the loading is unreasonable 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 reducing the time step size. Note that the node ID/rigid body ID with the maximum residuals are listed in the iteration information. Inspecting the model in at these ID:s (and perhaps a surrounding) might give clues to what is causing the convergence problems (for example poor element quality or lose parts) – especially if the same ID:s show up as critical for several consecutive iterations.

Residual histories, iterations to converge, etc. may also be plotted in LS-PrePost 4.5 using Misc>D3hsp view, see Figure 43.

If the problems (convergence related or other) occur during a restart, it might help testing other options.

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

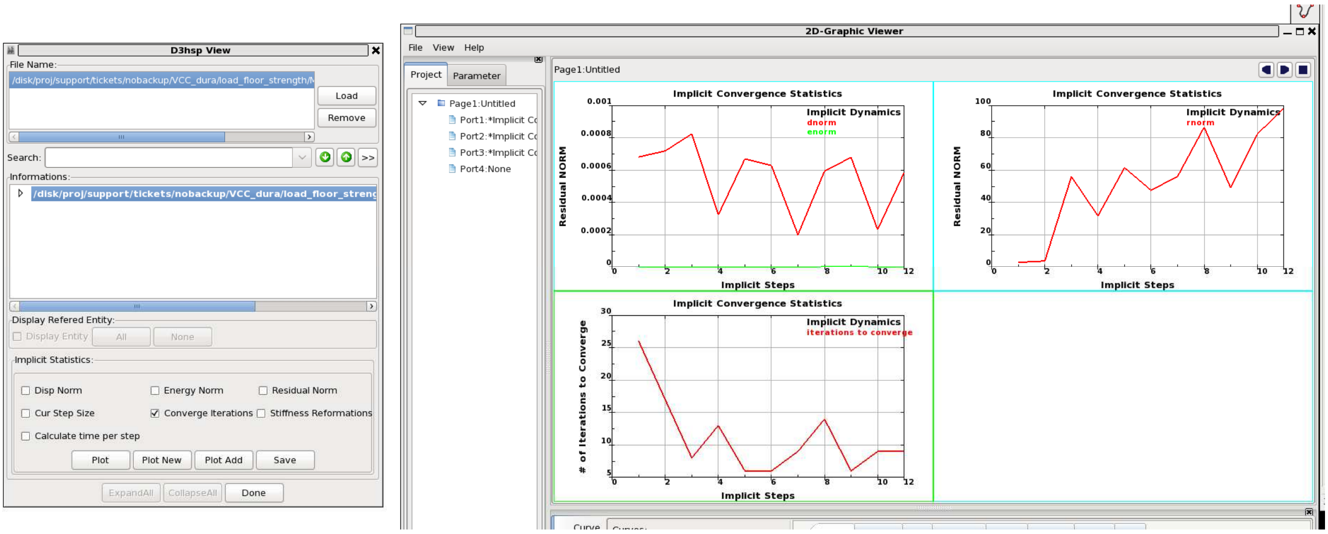


Figure 43. Tracking the convergence histories using the D3hsp view – tool in LS-PrePost 4.5.

If convergence problems prevail, don't hesitate to contact your LS-DYNA supplier (support@dynamore.se).

15 Appendix D: Converting an implicit model to explicit

This Appendix gives a very brief description of how to convert an implicit model for running in LS-DYNA. Dynamore Nordic provides training courses and webinars in explicit analyses using LS-DYNA, see www.dynamore.se for further details. The control cards for explicit analyses, based on Ref. [14], are provided in the keyword file `control_cards_expl.key`, with accompanying output requests in `database_cards_expl.key`. Note that these control card settings are merely recommendations, and the user must verify the correctness for the current application. The user will have to add keywords `*CONTROL_TIMESTEP` for reasonable mass-scaling (see further Section 15.1) and `*CONTOL_TERMINATION`. The provided database cards depend on the definition of a `*PARAMETER` for controlling the output frequency (`dbparu`). A template for explicit analysis follows:

```
*KEYWORD
*INCLUDE
control_cards_expl.key
*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 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 efficient 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 effect and are performed in physical time, which means that very long load application time may be required for structures with low eigenvalues. To some extent, mass scaling (see Section 15.1) and damping (see Section 15.5) can be used to overcome these difficulties.

15.1 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[13] l_e, material stiffness E_e, density ρ_e, and damping d as

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

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

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, buy comes with the cost that the now non-diagonal mass matrix must 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 forming simulations can be run with over 100 % mass scaling.

15.2 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 will work also in an explicit analysis, but the computational cost / time may be high, and it might be an idea to switch to `*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_ID` or `*CONTACT_AUTOMATIC_SINGLE_SURFACE_ID`, in order to save solution time. It is recommended to use contact damping in explicit analyses (set *VDC* = 20 – 40 on Card 2 of the contact). The basic recommendation is to use *SOFT* = 1 on Optional Card A. It shall also be noted that by default, the automatic single surface contact will not always use the actual shell thickness as contact thickness (unless the contact thickness is specified using `*PART_CONTACT` and *OPTT*) but instead the smallest value of

- shell thickness, or
- 0.4 · characteristic element length.

---

[13] 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. [11].

This means that if a "thick" part is finely meshed, the contact thickness will be reduced. This can be avoided by setting *SSTHK* = 1 on `*CONTROL_CONTACT`, or by defining an optional contact thickness *OPTT* for each part, using `*PART_CONTACT`.

For further details on contacts in explicit, the course "Contacts in LS-DYNA" is warmly recommended. A template for using the automatic single surface contact follows.

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

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.

15.3 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 take a (relatively speaking) big part of the solution time. 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 9 for an overview of recommended elements for general structural analyses in explicit.

Table 9. 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
	2nd order tet		16 (17)	
	2nd order hex		23	

Fully integrated first order elements (for example shell elform 16 and solid elform -2) can also be used in explicit analyses, but may give (a factor of 8 -10 times) longer solution time. Under-integrated elements require hourglass control. It is recommended to use the LS-DYNA keyword `*HOURGLASS` and assign hourglass control per part. Hourglass type 4 with an hourglass coefficient of 0.05 is often recommended for shells, and type 6 with a coefficient of 0.1 is recommended for solids. But there are many situations where other hourglass parameters should be used. Note that under-integrated elements cannot guarantee mesh convergence in a strict mathematical sense.

The use of second order elements will increase solution time, both due to increased computational cost in the element routine and due to the need to reduce the explicit time step [11].

15.4 Load curves

In order to introduce 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 a lot of 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 inertia 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 44. Also for forces and other loadings, smooth curves (at least linear ramps) should be used.

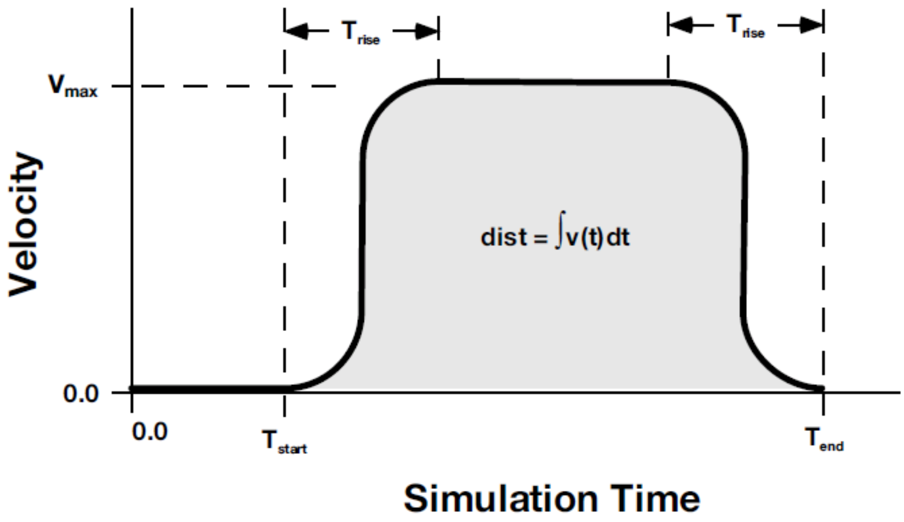


Figure 44. The keyword *DEFINE_CURVE_SMOOTH can be used for creating a smooth velocity curve based on a desired displacement distance.

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

15.5 Damping

It is in general recommended [14] to apply stiffness damping, *DAMPING_PART_STIFFNESS_SET, with a value of 2.E-2 – 5.E-2 (corresponding to 2 – 5 %) in order to reduce high frequency noise in the solution. Mass damping (*DAMPING_GLOBAL or *DAMPING_PART_MASS_{SET}) can in some situations be applied to reduce unwanted oscillations in the solutions due to the dynamics that is introduced when the loading is ramped up. This is illustrated[14] by an example in Figure 45, 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, and after that 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 off mass damping can “freeze” rigid body motions).

---

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

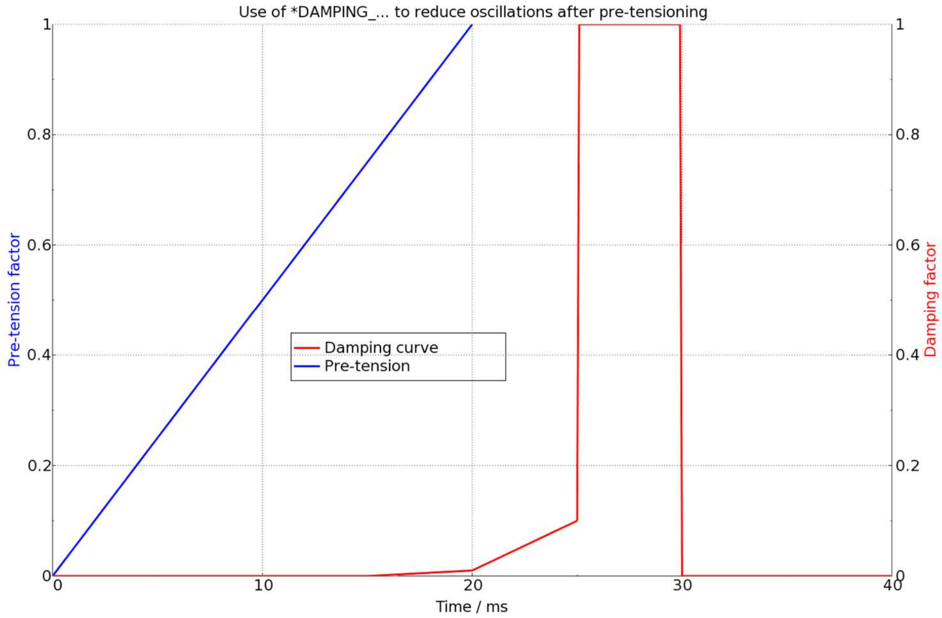


Figure 45. The use of mass damping to reduce oscillations after load application.

15.6 Global energy balance

The global energy balance from the `glstat` – file, see Figure 46 for an example, can be used as a basic check of the model validity. Negative sliding energy in general 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 very negative sliding energy is found in a contact, some part has a corresponding increase in internal energy. Internal energies per part can be studied per part in the `matsum` file. By looking in `matsum` for parts which have a noticeable increase in internal energy at the time when the sliding energy decreases, problematic parts or areas may be identified.

From under-integrated element, a small amount (5 – 10 % of the total energy) of hourglass energy is acceptable. For quasi-static analyses, check that the kinetic energy is low (5 – 10 %) compared to the total energy.

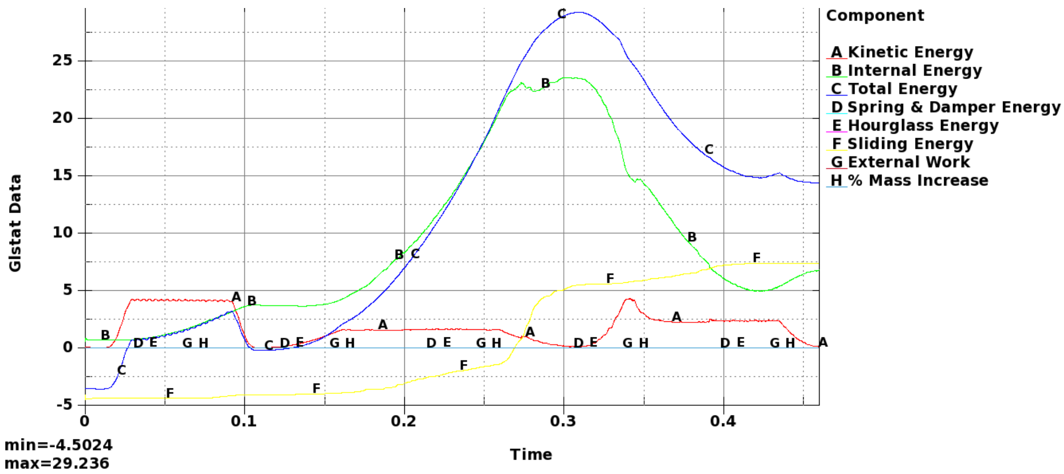


Figure 46. Example of global energy balance plot from `glstat`.

Indications of an invalid 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.

15.7 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 pre-processors have built-in checks for identifying massless nodes in the model. In explicit, great care must be taken when applying loads or boundary conditions to massless nodes (for example centre nodes of CNRB “spiders”).

15.8 Recommended LS-DYNA versions

In general, it is recommended to use a single precision mpp version of LS-DYNA, since it will save solution time. In some cases, with very many (> 10^6) explicit time steps, round-off errors in single precision may cause instabilities, and it is in those cases recommended to switch to double precision. At the time when this Appendix is written (2017), mpp LS-DYNA R7.1.3 is the main version used for crash analyses.

16 Appendix E: Converting an explicit model to implicit

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. Some material models, like `*MAT_ARUP_ADHESIVE`, are not supported in implicit. 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, but may make convergence very hard. It is recommended to de-activate 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.
- Switch control cards to the appropriate implicit settings. See Table 2 for an overview.
- Proper boundary conditions, suppressing rigid body modes, are required in implicit static analyses.
- Rigid body modes should be avoided in implicit statics. Apply appropriate boundary conditions. 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 4.8).
- Parts or assemblies connected by joints (`*CONSTRAINED_JOINT_{OPTION}`) may cause rigid body modes or mechanisms. Additional constraints, or adding a joint stiffness (`*CONSTRAINED_JOINT_STIFFNESS_{OPTION}`) may be a remedy.
- Switch contacts to Mortar, see Section 6.1.
- 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 required for these element formulations. In most cases, the hourglass control can be kept from the explicit set-up (hourglass control type will be automatically switched to a type which is valid in implicit). Still, in many cases it might be worth switching to fully-integrated elements, see Table 6 for an overview.

Should convergence problems arise, see Appendix C for some trouble shooting tips. Don't hesitate to contact your LS-DYNA supplier (support@dynamore.se) for further help with converting explicit models to run in implicit.