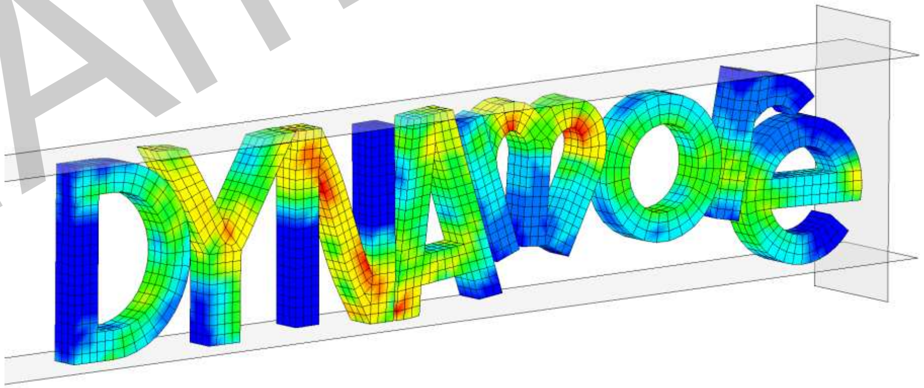
Webinar
LS-DYNA Implicit

Dr. Tobias Erhart - 27 February 2015



Overview

- LS-DYNA / LSTC / Dynamore GmbH
- Motivation for implicit analyses
- Application examples
- Differences between explicit and implicit → consequences
- Implicit analysis with LS-DYNA: main keywords
- Guidelines: best practice

- **Exercise:**

  Going through different settings by means of a practical example

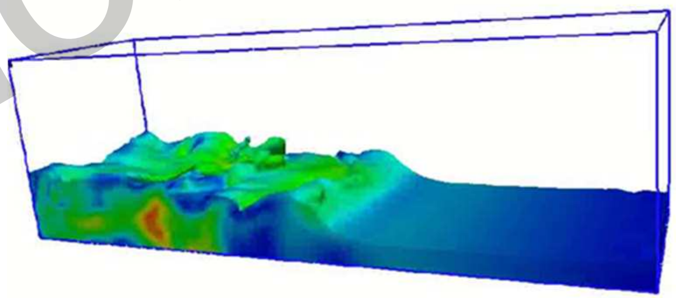
What is LS-DYNA ?

A Finite Element software originally developed for nonlinear stress analysis of structures subjected to impact loading

- Automotive, Defense, Consumer products, …

Now, a multiphysics code…

- Fluid-structure interaction (ALE, CFD, EFG, SPH)
- Coupled thermomechanics
- Electromagnetism
- Discrete Elements
- …



…with well developed parallel processing capabilities

- SMP (Shared Memory Parallel)
- MPP (Distributed Memory Multiple Parallel Processing)

Introduction

Livermore Software Technology Corp.

- Founded in 1987
  by Dr. John Hallquist
- Located in Livermore, California
- Worldwide distribution
- 60+ full-time employees and numerous consultants

- Current products:

LS-DYNA®	(nonlinear explicit and implicit)
LS-PrePost ®	(postprocessor)
LS-OPT ®	(optimization)

Dynamore GmbH

- Founded in 2001 by Prof. Dr. Schweizerhof
- Headquarter in Stuttgart, Germany
- European master distributor (w/o UK and France)
- 60+ employees

- Core products: LS-DYNA® (nonlinear explicit and implicit)
  LS-PrePost ® (postprocessor)
  LS-OPT ® (optimization)
- Related products: dummy models, impact models, barrier models, Primer, D-SPEX, DYNAtools, ...
- Consulting, support, maintenance, training

One Model vision of LS-DYNA

One model for all applications

- Analysts work in parallel to reduce the time to produce the initial model
- Only one model to revise for design changes
- Only one model to check for errors
- Multi-physics problems can be addressed
- Simplified database management

One code strategy of LSTC

- Multi-physics, Multi-stage, Multi-formulations, Multi-processing, …

One of the goals of LSTC is to provide a state-of-the-art implicit solver to complement our explicit solver.

The development of the implicit solver is in line with the "One Model" vision.

Motivation: Why implicit ?

prestressed, quasi statically loaded structures

long duration analysis > 500 ms

different time scales in process

e.g. static loading followed by transient loading
or transient loading followed by static loading

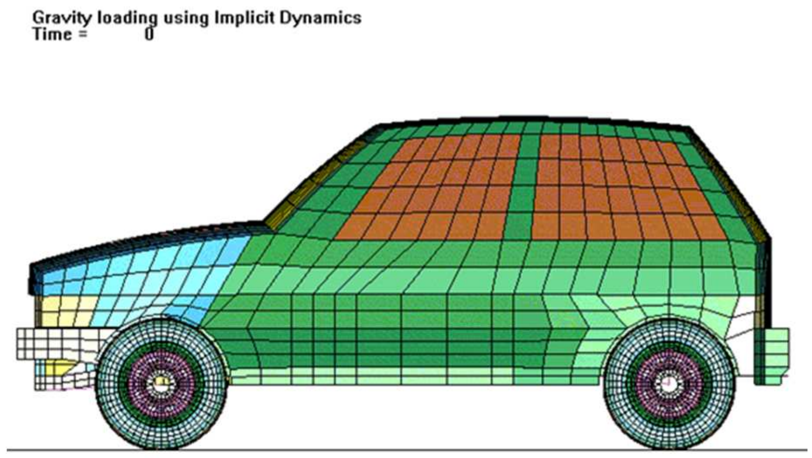
applications

e.g. metalforming, roof crush, door sag, dummy seating, strength analysis, ...

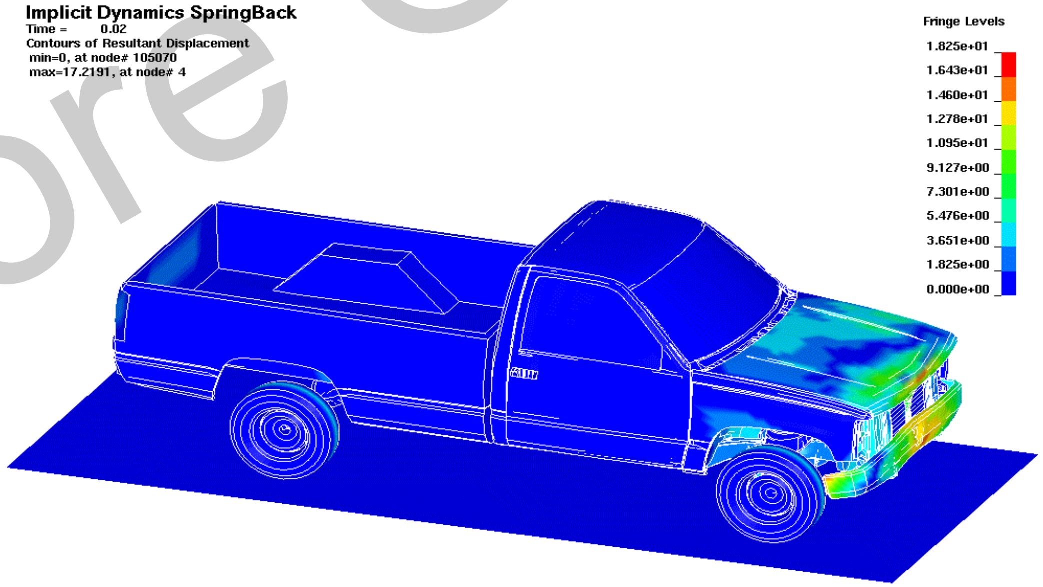
➡ **LS-DYNA provides explicit and implicit solution schemes**

one code – one license - one data structure
one input / output

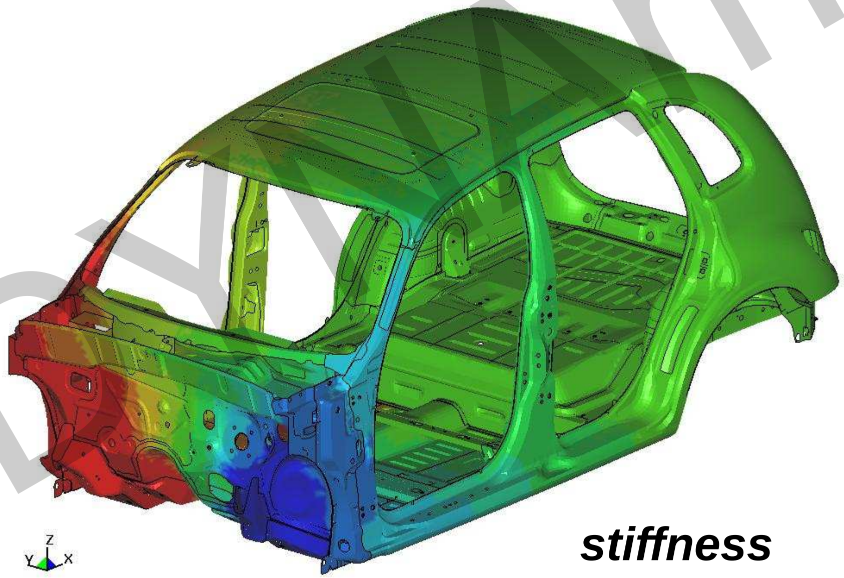
Examples



gravity

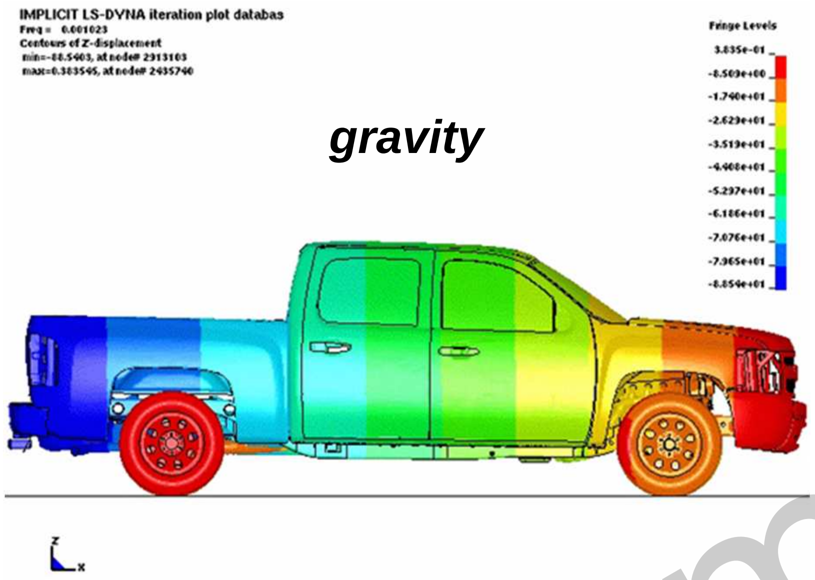


springback

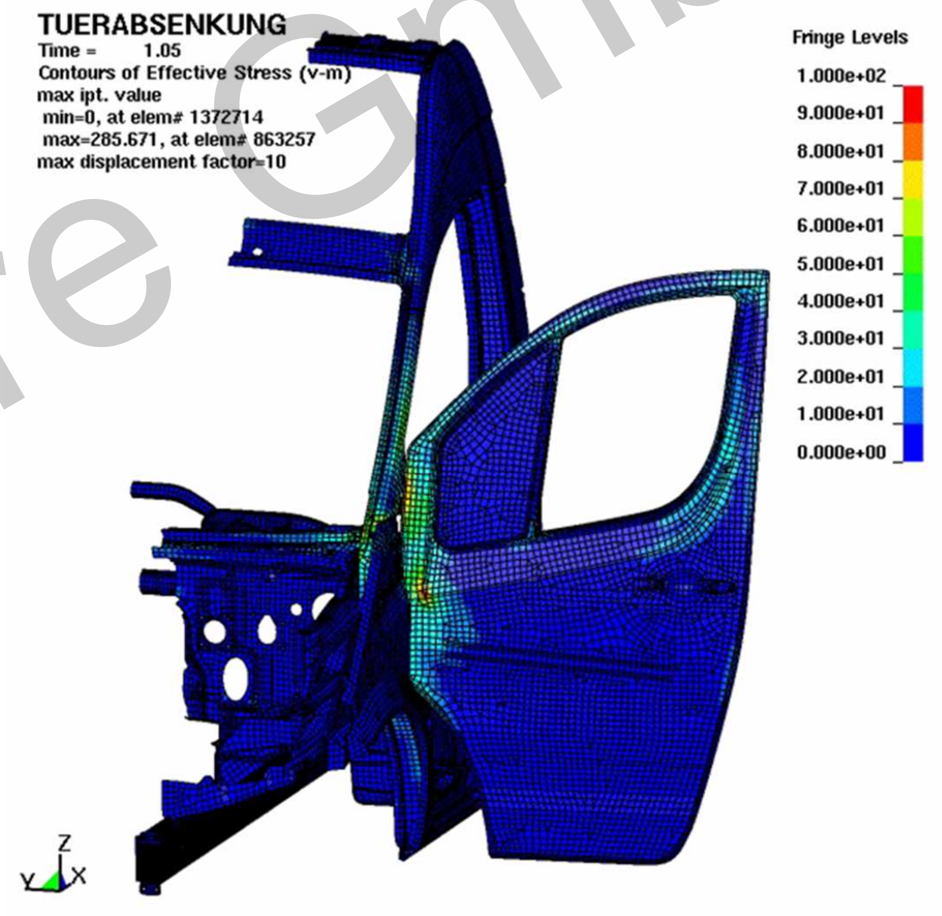


stiffness

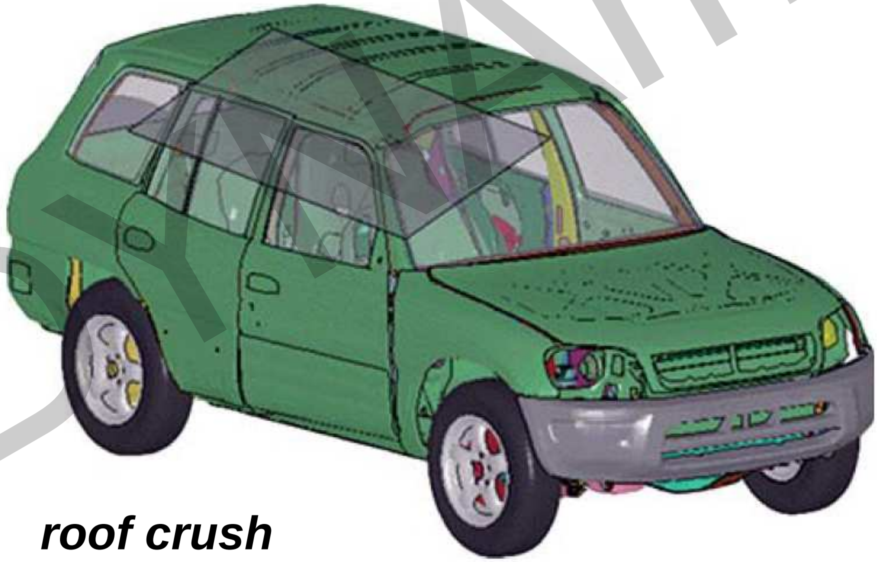
Examples



gravity

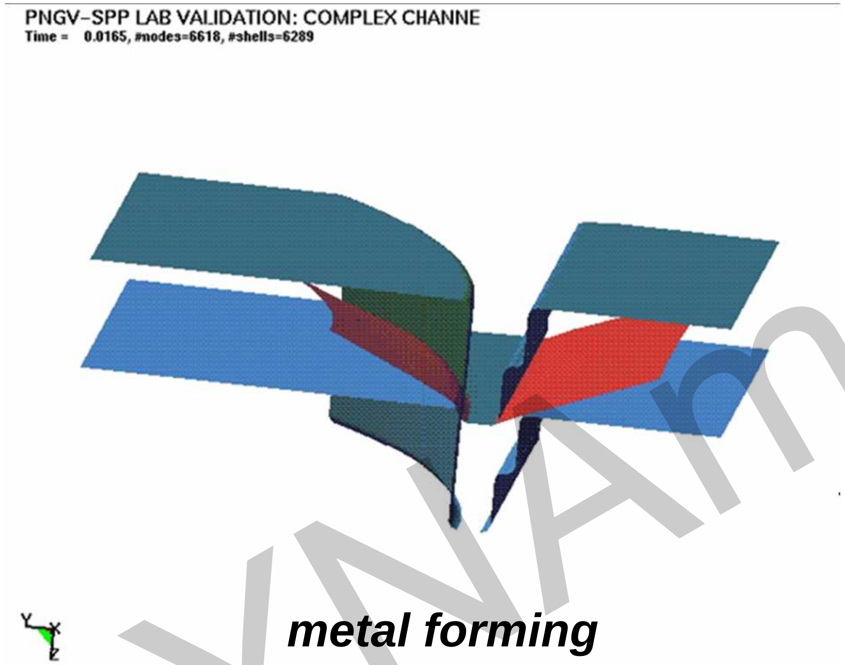


door sag



roof crush

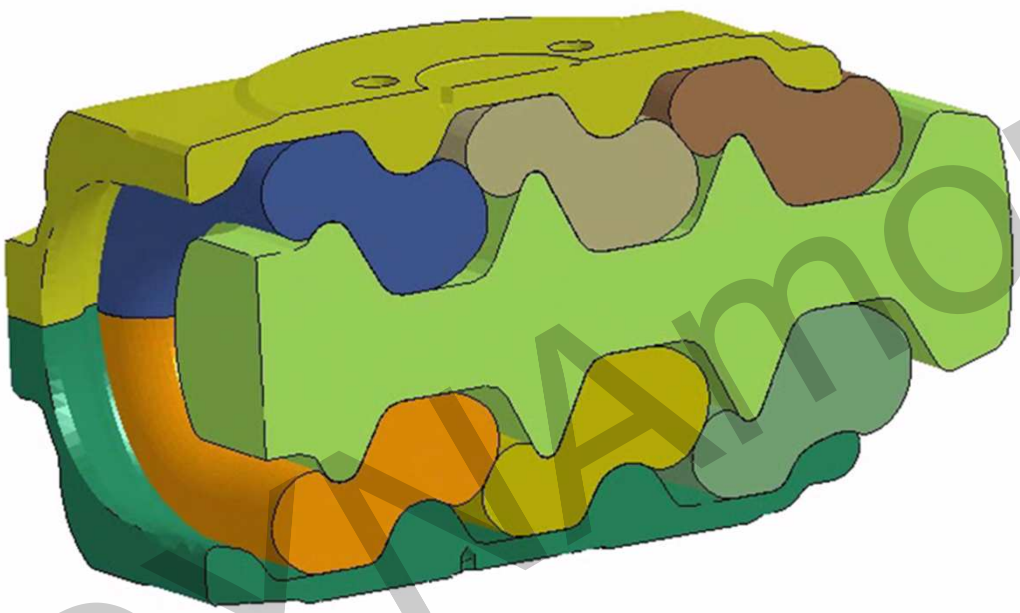
Examples

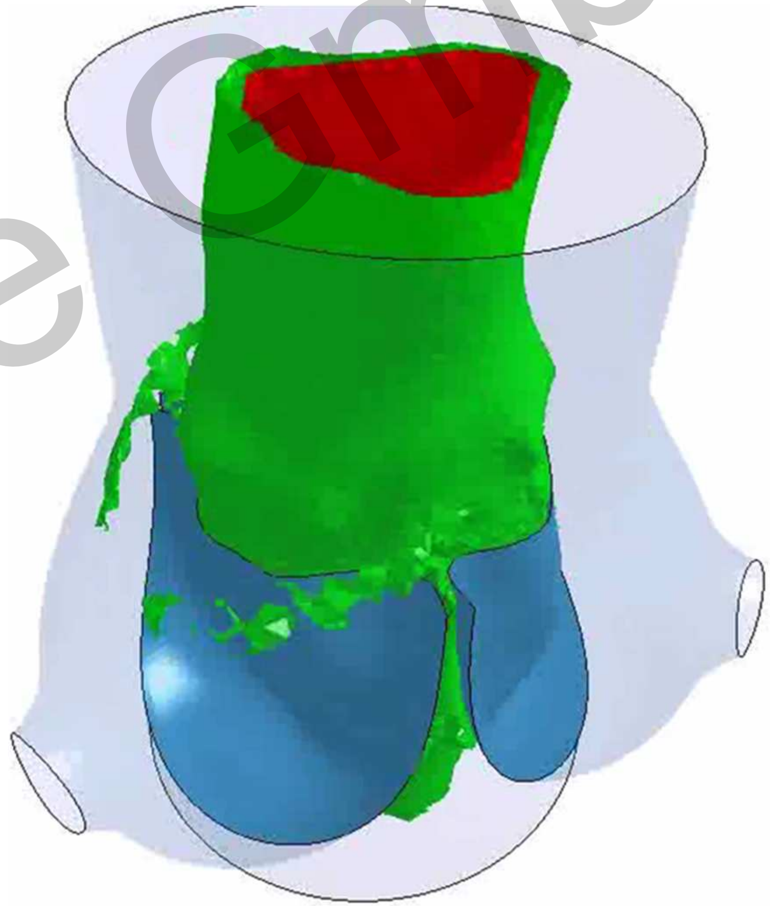


metal forming

Examples

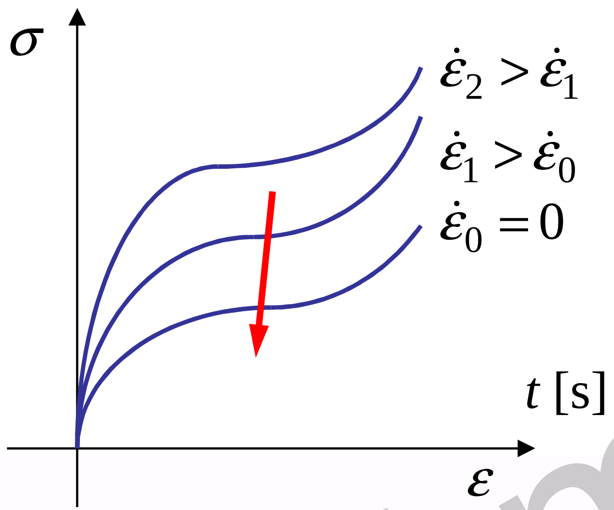
rubber bushing

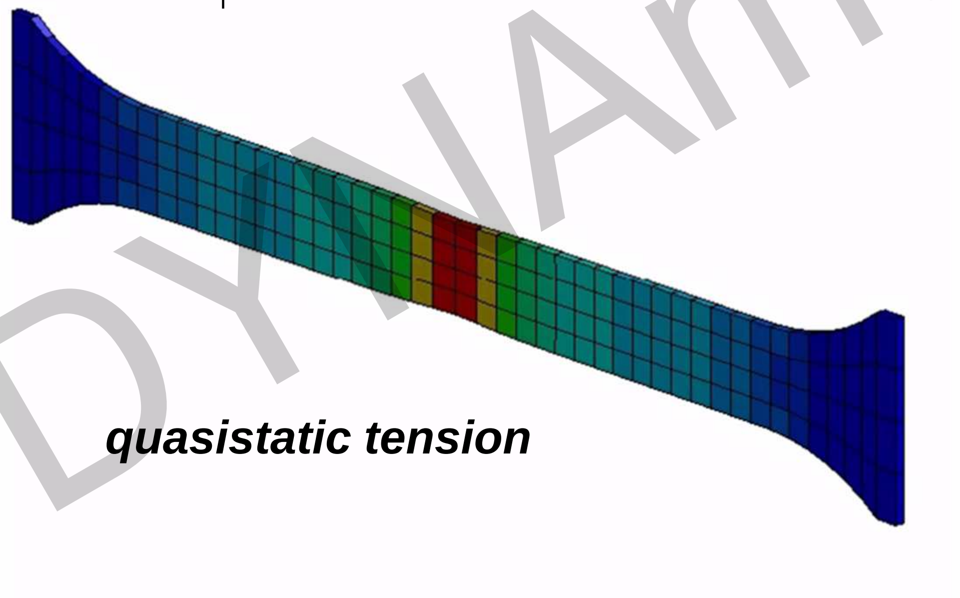




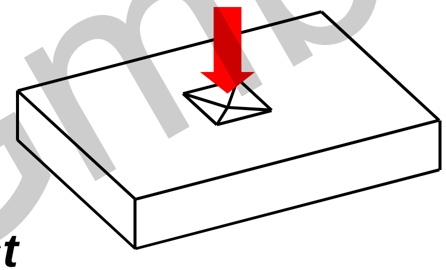
heart valve with FSI

Examples

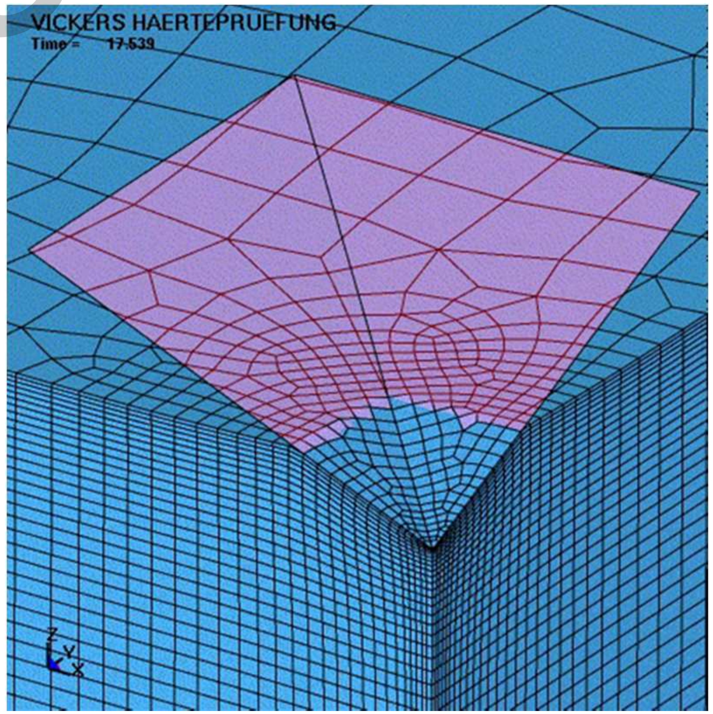




quasistatic tension



Vickers hardness test



Explicit vs. Implicit

Explicit: $\mathbf{x}_{n+1} = \mathbf{f}(\mathbf{x}_n, \ldots)$

$$\mathbf{M}\mathbf{a}_n = \mathbf{f}_n^{ext} - \mathbf{f}_n^{\text{int}}$$

+ solution: directly
+ decoupled: efficient, fast
- many small time steps
- conditionally stable (Courant)

equilibrium? energy balance!

short time dynamics:
high frequency response,
wave propagation

➡ impact, crash, ...

Implicit: $\mathbf{f}(\mathbf{x}_{n+1}, \mathbf{x}_n, \ldots) = \mathbf{0}$

$$\mathbf{M}\mathbf{a}_{n+1} + \mathbf{K}\mathbf{u}_{n+1} = \mathbf{f}_{n+1}^{ext} - \mathbf{f}_n^{\text{int}} - \mathbf{M}\mathbf{a}_n$$

- solution: iteratively
- linearization necessary
+ few large time/load steps
+ unconditionally stable

equilibrium! convergence?

structural dynamics:
low frequency response,
vibration, oscillation

➡ static/dynamic strength, ...

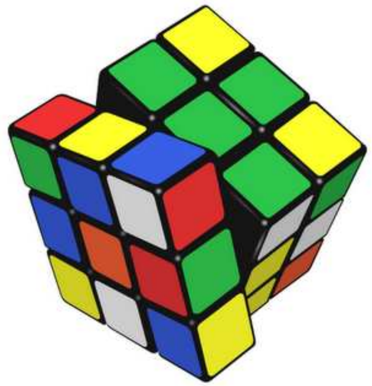
Explicit vs. Implicit

- Explicit inevitably includes inertial effects and resolves high frequencies whether you want it or not
- Implicit can neglect inertial effects and the selected time step size determines the resolved frequency spectrum

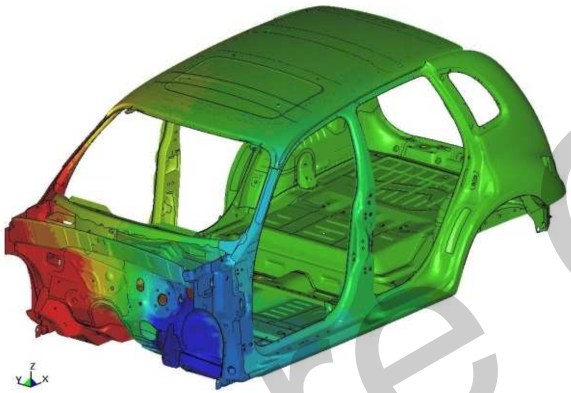
Consequences for FE models

- "cleaner" models in implicit for the sake of convergence, e.g. no initial penetrations, smooth material curves, contact, …
- expensive features are not so expensive anymore
- no resctriction on element size (time step size) in implicit
- often more work to get "normal termination" in implicit

"Explicit is handcraft - Implicit is art"

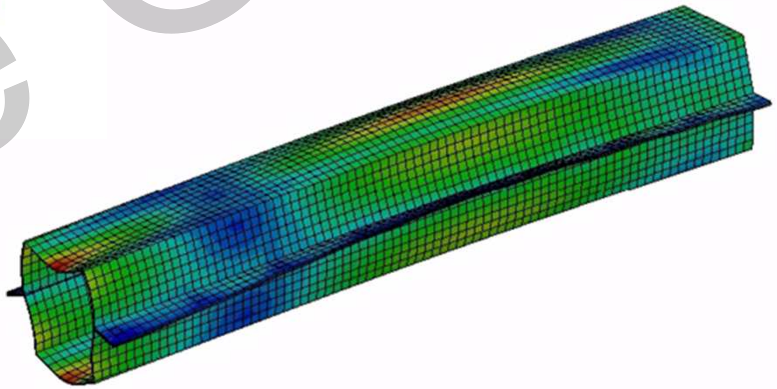


Types of Implicit Analyses



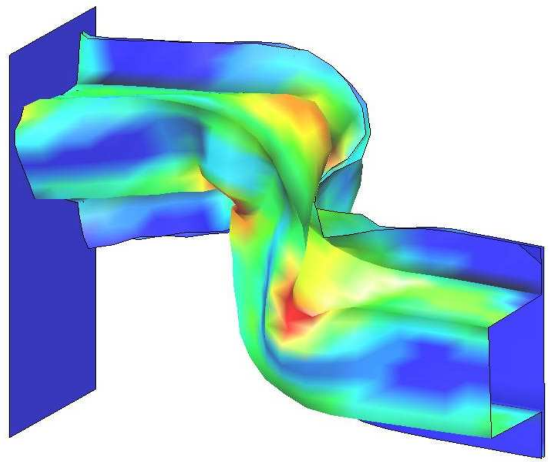
Linear Analysis

- static or dynamic
- single, multi-step



Eigenvalue Analysis

- frequencies and mode shapes
- linear buckling loads and modes
- modal analysis: extraction and superposition
- dynamic analysis by modal superposition



Nonlinear Analysis

- Newton, Quasi-Newton, Arclength solution
- static or dynamic
- default LS-DYNA: **static and nonlinear!**

Activating implicit analysis in LS-DYNA

Use *CONTROL_IMPLICIT_GENERAL to activate implicit

- specify time step size
- all other *CONTROL_IMPLICIT keywords are optional
- default is nonlinear, static analysis

Use a double precision executable for implicit analysis

- better convergence for nonlinear
- mandatory for linear, eigenvalue accuracy
- mandatory for MPP

```
lsdyna i=inp.k memory=500m
```

500,000,000 words:

2000 Mbytes in single precision
4000 Mbytes in double precision

Stiffness matrix requires lots of memory

huge speed penalty for out-of-core jobs

Most keywords apply to explicit and implicit

easy to run a model with either method, but: carefully inspect input deck

Activating implicit analysis

Three types of analyses can be performed

fully explicit (default), fully implicit, or switching (explicit - implicit, implicit - explicit)

All keywords for implicit

*CONTROL_IMPLICIT_GENERAL
*CONTROL_IMPLICIT_SOLUTION
*CONTROL_IMPLICIT_STABILIZATION
*CONTROL_IMPLICIT_MODES
*CONTROL_IMPLICIT_BUCKLE
*CONTROL_IMPLICIT_JOINTS
*CONTROL_IMPLICIT_FORMING
*CONTROL_IMPLICIT_STATIC_CONDENSATION
*CONTROL_IMPLICIT_SOLVER
*CONTROL_IMPLICIT_AUTO
*CONTROL_IMPLICIT_DYNAMICS
*CONTROL_IMPLICIT_EIGENVALUE
*CONTROL_IMPLICIT_INERTIA_RELIEF
*CONTROL_IMPLICIT_TERMINATION
*CONTROL_IMPLICIT_CONSISTENT_MASS

Proper selection of LS-DYNA features

- only few features are not available in implicit mode
- warning & error messages, feature substitution

Main Implicit Keywords

***CONTROL_IMPLICIT_GENERAL** (required for implicit)

- activates implicit mode, explicit-implicit switching
- defines implicit time step size
  (standard LS-DYNA termination time is used too)

***CONTROL_IMPLICIT_SOLUTION** (optional)

- parameters for nonlinear equation solver (Newton-based methods)
- controls iterative equilibrium search, convergence
- "linear" analysis selected here
  (a special case where no iterations are performed)

***CONTROL_IMPLICIT_AUTO** (optional)

- activates automatic time step control
- default is fixed time step size, error termination if any steps fail to converge

Main Implicit Keywords

***CONTROL_IMPLICIT_DYNAMICS** (optional)

- include inertia terms
- problem "time" must now be real, physical time
- can improve convergence, especially when rigid body modes are present

***CONTROL_IMPLICIT_EIGENVALUE** (optional)

- signals LS-DYNA to perform eigenvalue analysis, then stop
- number of eigenvalues/vectors, optional frequency shift
- great for debugging/model checking
- optional: intermittent eigenvalue analysis

Two-day implicit class

Day 1

- Introduction
- Linear static analysis
- Dynamic implicit analysis
- Nonlinear implicit analysis
- Eigenvalue analysis
- Modal analysis
- Buckling analysis
- Frequency response function

+ Live Demos ...

Next opportunity: March 17, 2015, Stuttgart, Germany

Two-day implicit class

Day 2

- Implicit/Explicit switching
- Element types
- Material types
- Contact types
- Arclength method
- Troubleshooting convergence problems
- Miscellaneous
- Final guidelines & references

+ Live Demos ...

Next opportunity: March 18, 2015, Stuttgart, Germany

General

In general an explicit input deck can easily be transformed into an implicit input deck. However, in practice the implicit technique can give (convergence) problems since it is more sensitive to e.g. boundary conditions and non-linear behavior. Some general remarks and tips are given in the following in order to get started using the implicit solver in LS-DYNA

The following card is added to the deck

*CONTROL_IMPLICIT_GENERAL: set IMFLAG=1 and DTO=STEPSIZE

Default is a static analysis but that can be changed to dynamic

*CONTROL_IMPLICIT_DYNAMICS: set IMASS=1

Default is a non-linear analysis but that can be changed to linear

*CONTROL_IMPLICIT_SOLUTION: set NSOLVR=1

Recommendations

Use double precision of the code (_d_ in the name)

- required for accurate linear analysis
- improved convergence behavior in nonlinear analysis

Use the most recent LS-DYNA version possible

implicit functionality is rapidly improving

Use MPP version

now well established for implicit

e.g. **ls-dyna_mpp_d_r8_0_0_...**

Use command line option "memory=" to run job in-core

verify using LPRINT=1 on *CONTROL_IMPLICIT_SOLVER or "<ctrl-c> lprint". The CPU penalty for out-of-core can be as high as 100 times the in-core simulation!!

Read Appendix P in the User's manual

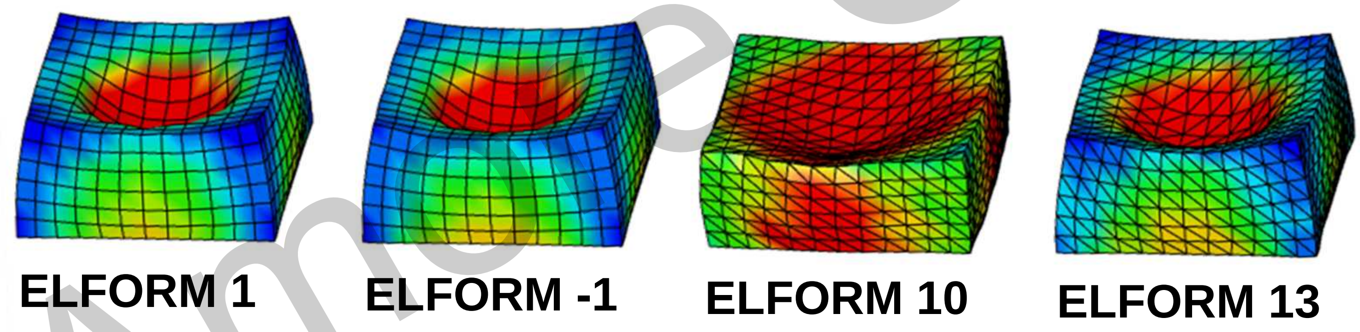
nice summary about LS-DYNA's Implicit Solver

Recommendations

Element types

- for solids use type 1, -1, 13, or 16 elements for non-linear analysis
- for shells use type 6 or 16 elements for non-linear analysis

rubber block compression



Contact

- try to avoid initial penetrations or try IGNORE=1
- for tied connections use penalty based tied contact (_offset option)
- try IGAP=2 on Additional card C or try the new Mortar contact
- contact often requires small time steps in implicit, too
- use soft part as slave

Recommendations

General

- apply 2nd order stress update by setting OSU=1 at *CONTROL_ACCURACY
- try to model displacement driven simulation instead of force driven simulation
- try to use IGS=1 (not default) on *CONTROL_IMPLICIT_GENERAL in case of convergence problems
- set DNORM=1 on *CONTROL_IMPLICIT_SOLUTION, displacement tolerance can often be increased in that case, e.g. DCTOL=0.005
- (try ABSTOL=1.e-20 on *CONTROL_SOLUTION to improve accuracy)
- often dynamic solution more robust than static solution
  → if static implicit fails to converge, try dynamic implicit
- try to avoid discontinuities, e.g. in material curves, geometry, ...

Recommendations

General

- in case of convergence problems, dump iteration states via "<ctrl-c> iter" (residual forces in d3iter via RESPLT=1 on *DATABASE_EXTENT_BINARY)
- in general, if problems occur when running an implicit model, then try to check the model using *CONTROL_IMPLICIT_EIGENVALUE
- in problems where there is much rigid body motion the displacement tolerance DCTOL may be insufficient, and it may be advisable, in some problems, to tighten the energy tolerance to 0.001.
- element size in implicit is not as important as in explicit (time step)
- CPU cost implicit is roughly proportional to ndof^2
- CPU cost explicit is roughly proportional to ndof

Recommendations

For "typical" implicit analysis, start with the following keyword settings:

```
*CONTROL_ACCURACY
$      osu       inn
         1         2
*CONTROL_IMPLICIT_GENERAL
$   imflag       dt0
         1       ...
*CONTROL_IMPLICIT_SOLUTION
$   nsolvr    ilimit    maxref     dctol     ectol     rctol   lstol   abstol
    (2/12)         6             (0.005)              (0.01)         (1.e-20)
$    dnorm    diverg     istif   nlprint    nlnorm   d3itctl
         1                             1                 (1)
$

$    lsmtd
     (3/5)
*CONTROL_IMPLICIT_AUTO
$    iauto    iteopt    itewin     dtmin     dtmax
         1        30        10           (term/20)
*CONTROL_IMPLICIT_DYNAMICS
$    imass
       (1)
```

Guidelines and Examples

New package on www.dynasupport.com:

http://www.dynasupport.com/howtos/implicit/some-guidelines-for-implicit-analyses-using-ls-dyna/ImplicitPackage.zip

… provided by Dynamore Nordic.

In this document, some basic control card settings suitable 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. The package also includes a document about Implicit Mortar Contact Problems.

Also, the draft version of the Theory Manual contains revised implicit sections:

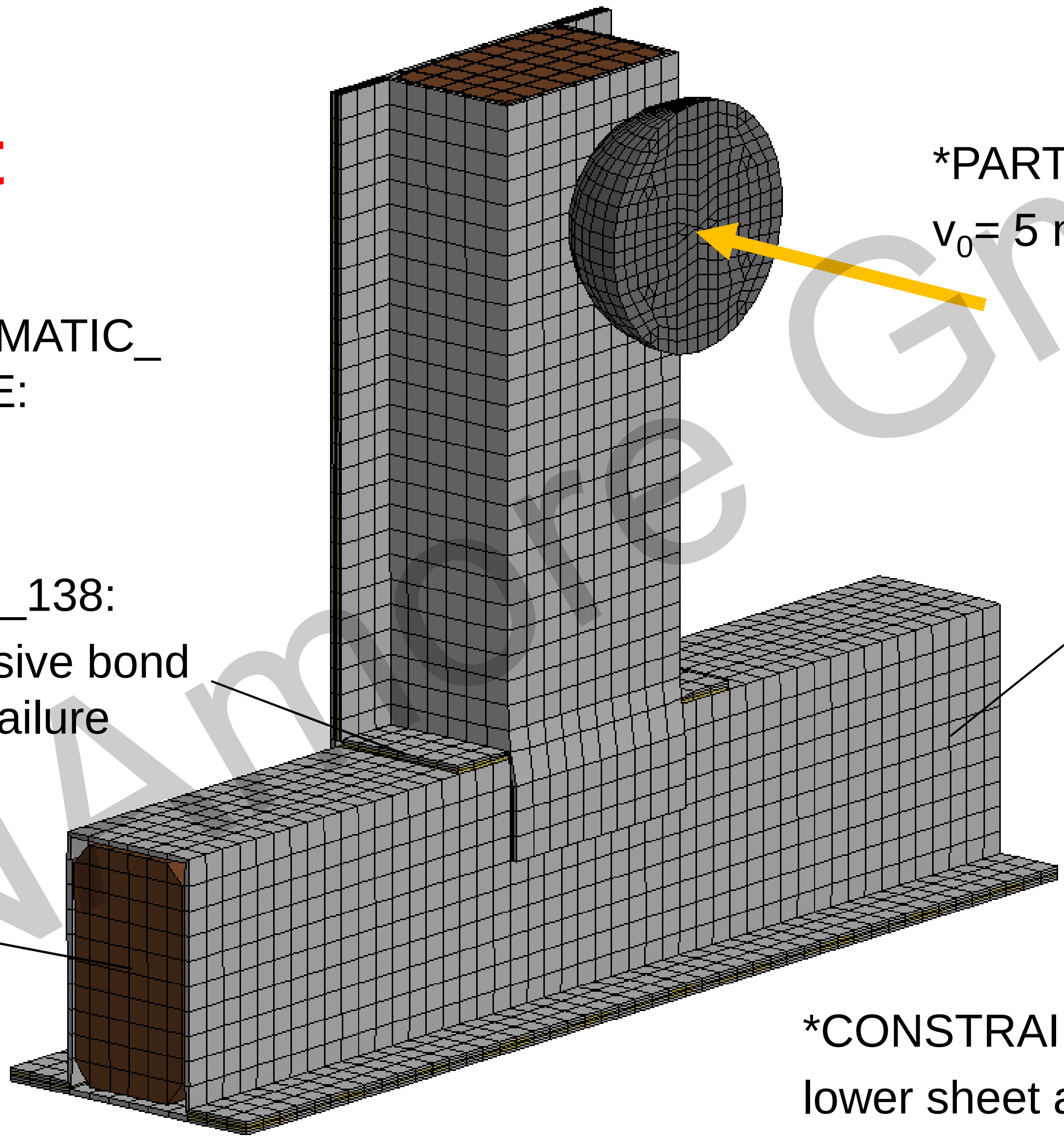
http://ftp.lstc.com/anonymous/outgoing/jday/manuals/DRAFT_Theory.pdf

T-joint component

*CONTACT_AUTOMATIC_SINGLE_SURFACE: overall contact

*MAT_138: adhesive bond with failure

*MAT_024: wooden blocks



*PART_INERTIA: v_0= 5 m/s

*MAT_024: DP 800

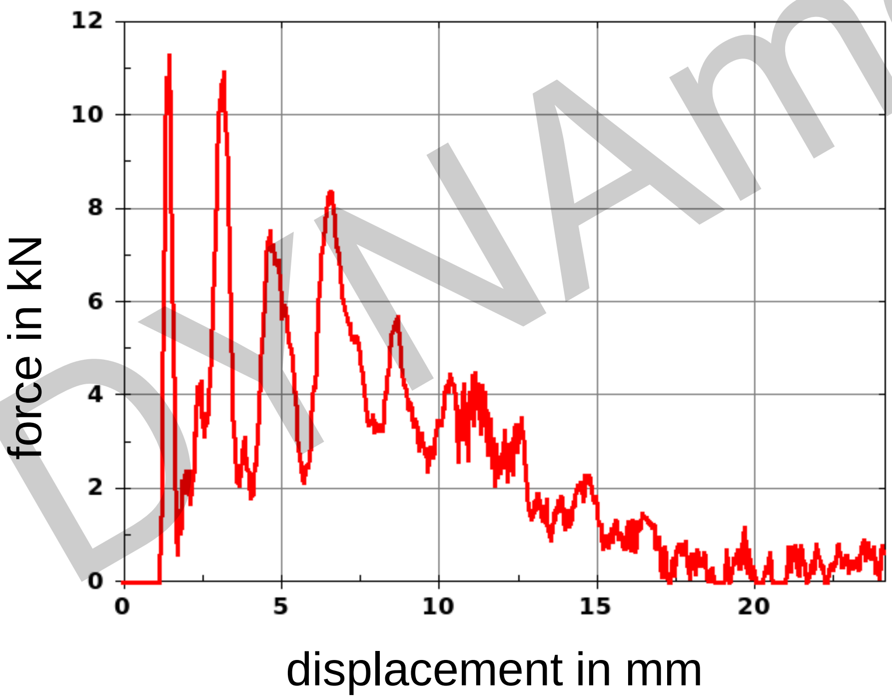
5 mm mesh for steel parts

*CONSTRAINED_RIGID_BODY: lower sheet and wooden block

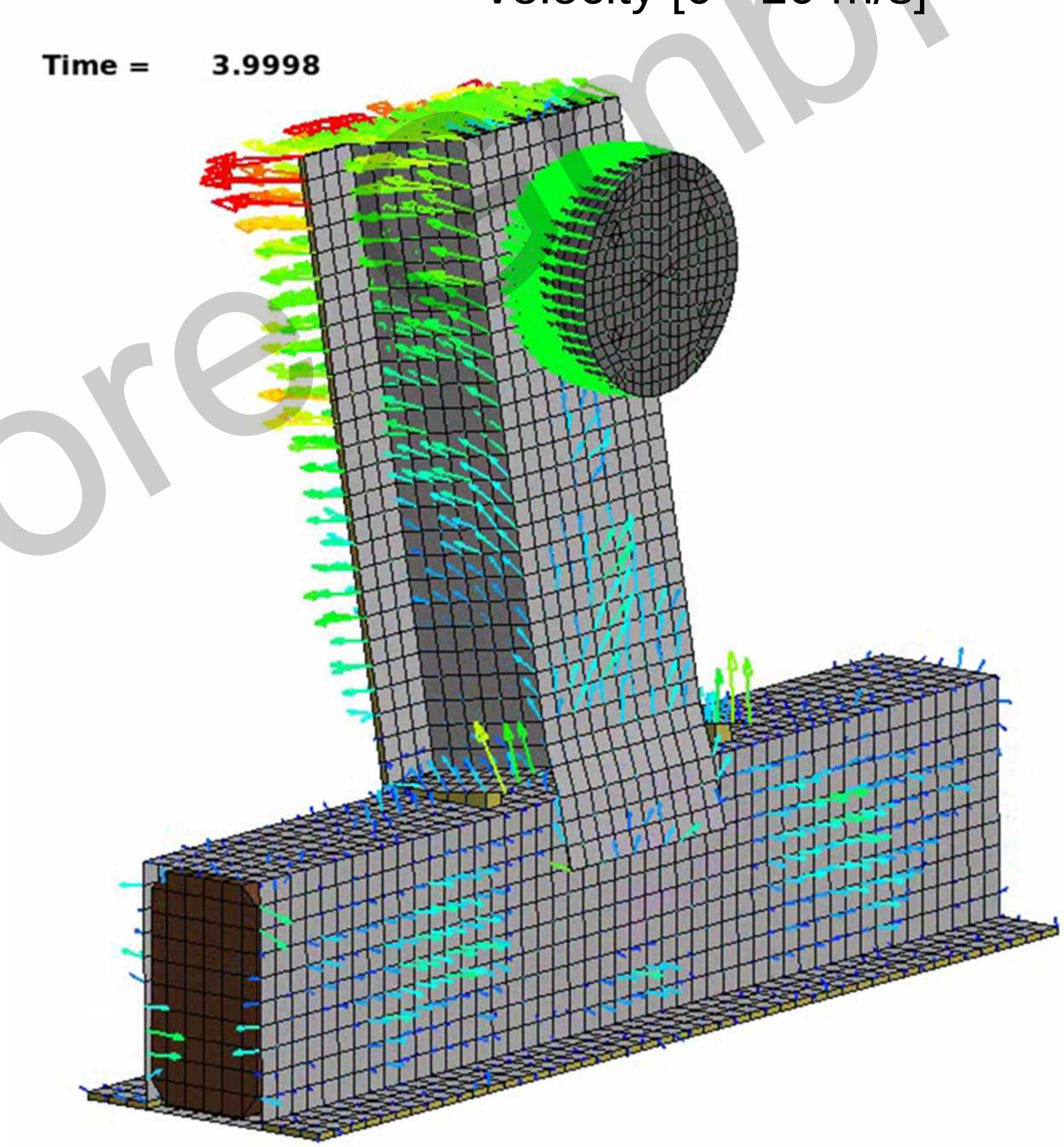
Exercise

Dynamic explicit

- Process time = 5 ms
- ~10,000 time steps
- 52 cohesive elements fail
- Low-frequency vibration and high-frequency response (wave propagation)



velocity [0 - 10 m/s]



Exercise

Now, we want to do a static analysis of that process.

But instead of directly going from dynamic explicit to static implicit, intermediate steps will be made to illustrate several interesting phenomena:

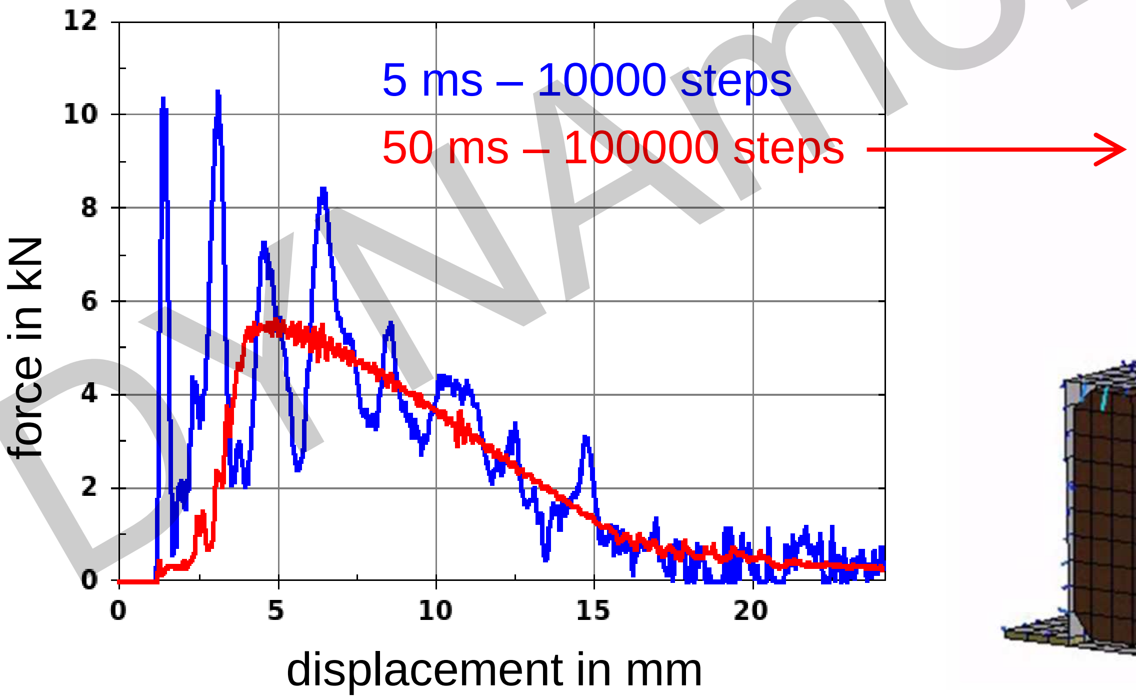
1. Start with explicit using a larger time period (“slow“ loading)

2. Add implicit cards needed for dynamic implicit analysis (“fast“ and “slow“ loading)

3. Finally, remove dynamics and perform pure static analysis

Static (??) explicit

- Process time = 5 / 50 ms
- ~ 10,000 / 100,000 time steps
- No initial velocity, but prescr. motion
- 52 cohesive elements fail
- Response still dynamic
- Damping… ??



velocity [0 - 3 m/s]

Exercise

Dynamic implicit (default)

- Process time = 5 ms (“fast“)
- *CONTROL_IMPLICIT_GENERAL: DT0 = 0.05 (100 steps)
- *CONTROL_IMPLICIT_DYNAMICS: IMASS = 1

→

- 100 steps
- 2779 problem cycles
- 58 failed cohesives

+ Recommendations

- *CONTROL_ACCURACY: OSU = 1
- *CONTROL_IMPLICIT_SOLUTION: NSOLVR = 12, ILIMIT = 6, DNORM = 1 (DCTOL=0.005)
- *CONTROL_IMPLICIT_AUTO: ITEOPT = 30, ITEWIN = 10, DTMAX = 0.1

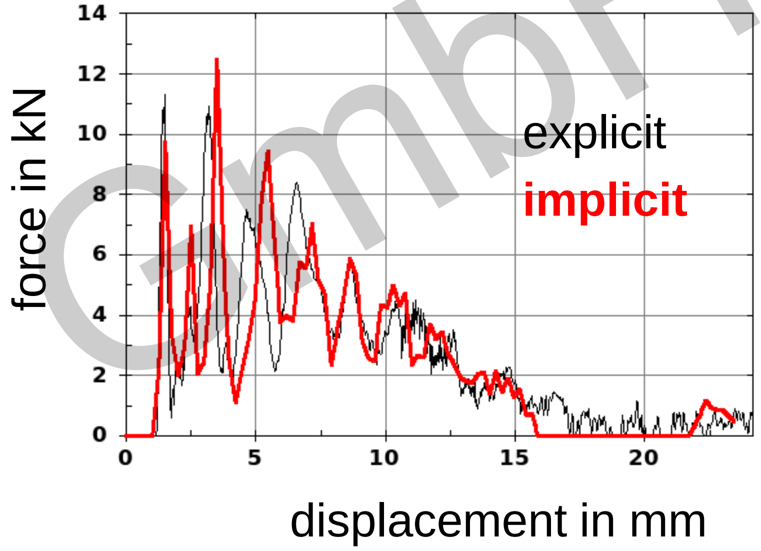
→

- 51 steps
- 1063 problem cycles
- 52 failed cohesives

Exercise

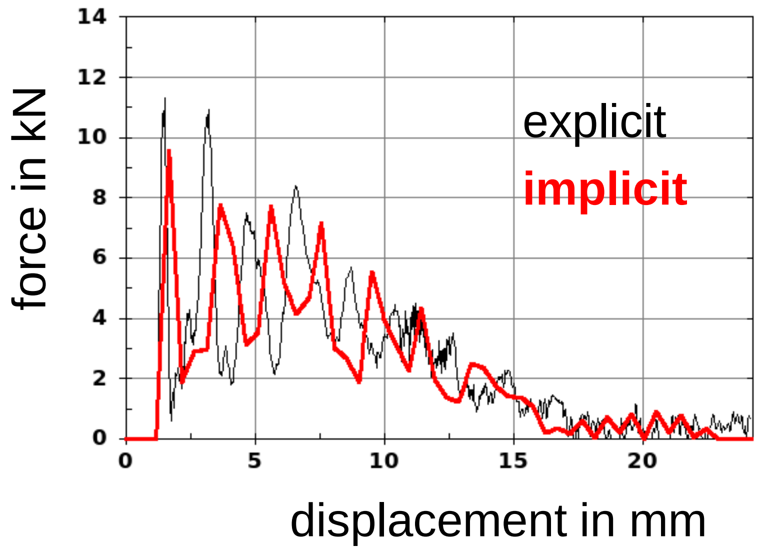
Dynamic implicit (default)

- Process time = 5 ms (“fast“)
- *CONTROL_IMPLICIT_GENERAL:
  DT0 = 0.05 (100 steps)
- *CONTROL_IMPLICIT_DYNAMICS:
  IMASS = 1



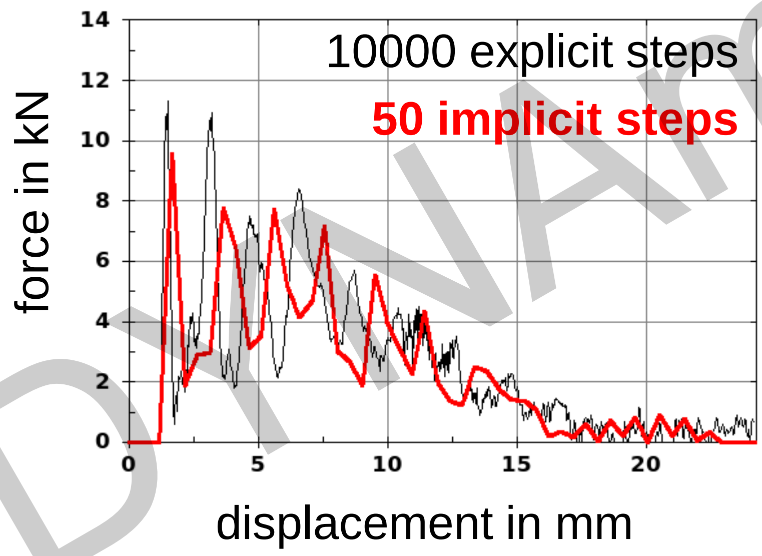
+ Recommendations

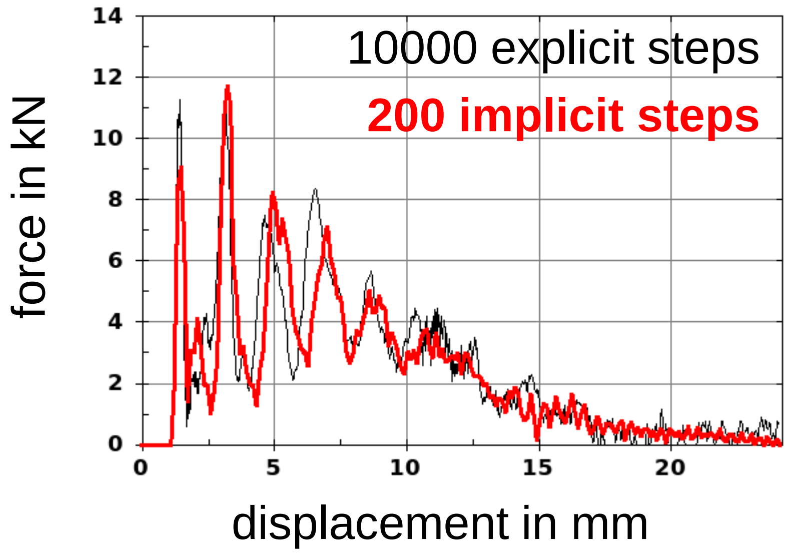
- *CONTROL_ACCURACY: OSU = 1
- *CONTROL_IMPLICIT_SOLUTION:
  NSOLVR = 12, ILIMIT = 6,
  DNORM = 1 (DCTOL=0.005)
- *CONTROL_IMPLICIT_AUTO:
  ITEOPT = 30, ITEWIN = 10, DTMAX = 0.1



Dynamic implicit

- What time step size is necessary to resolve the dynamic process?
- User needs good knowledge about the problem at hand
- User has to decide about the solution frequency
- Contact dominated problems need small time steps

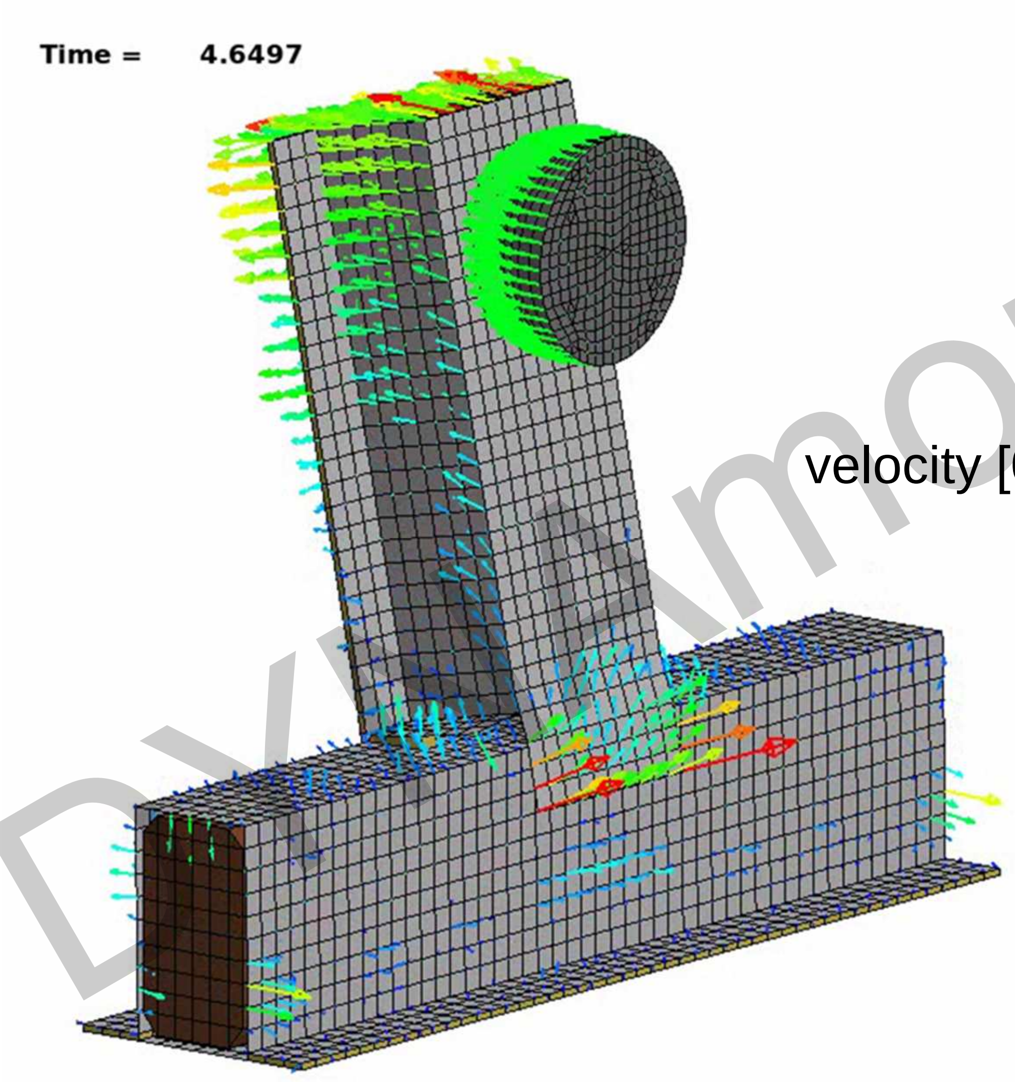




Exercise

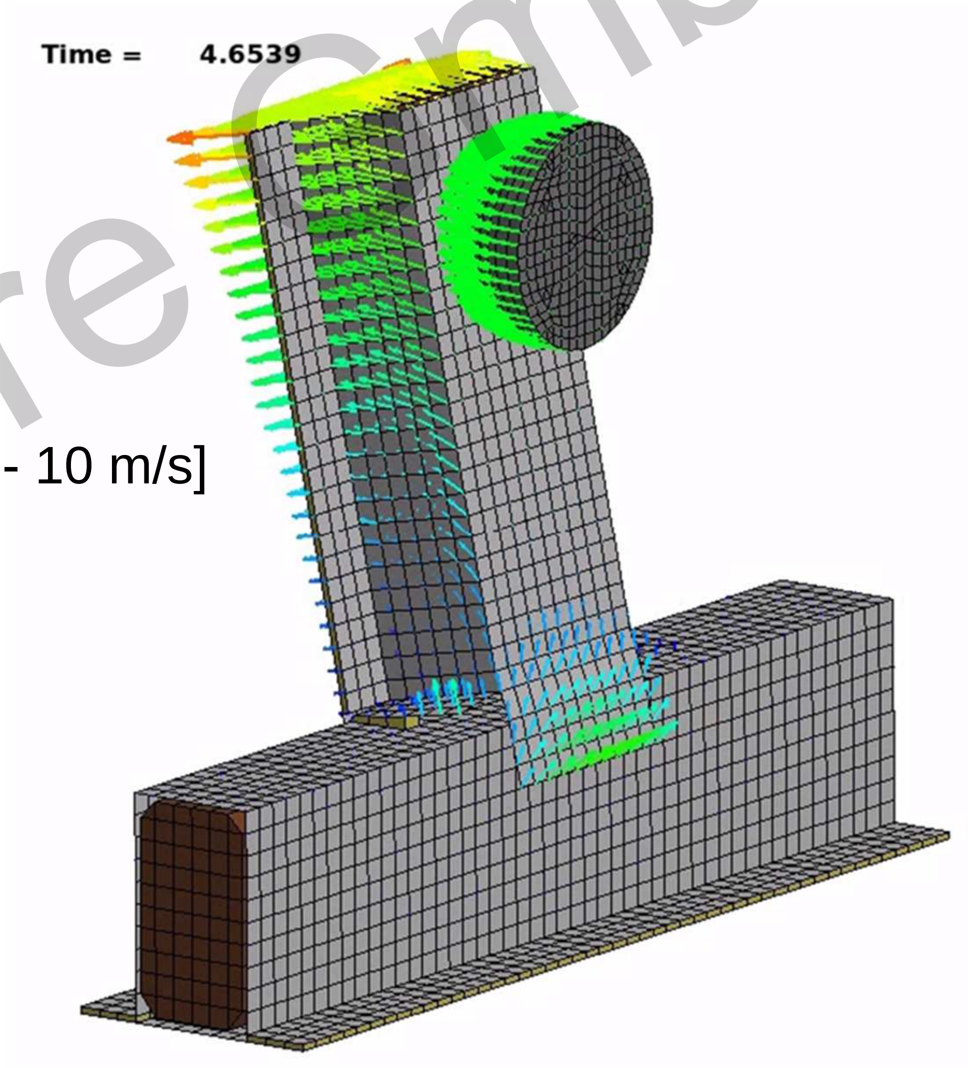
Dynamic explicit

- Low- and high-frequency response



Dynamic implicit

- Low-frequency response

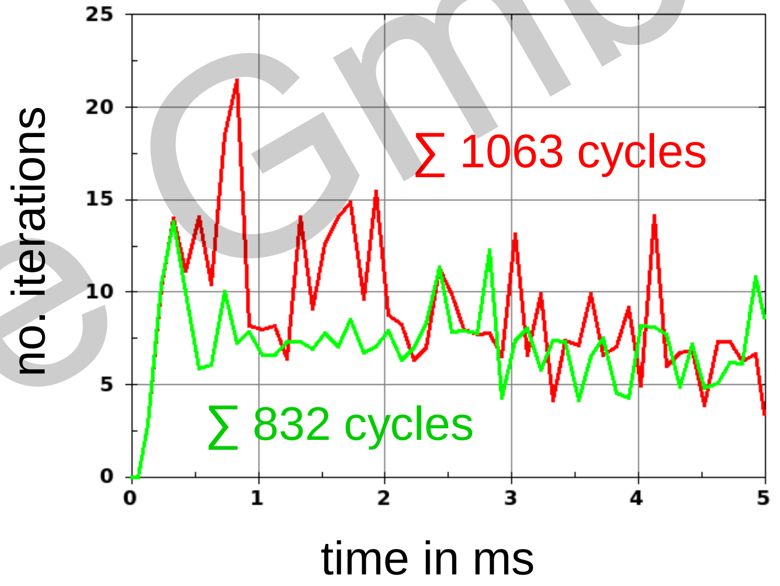


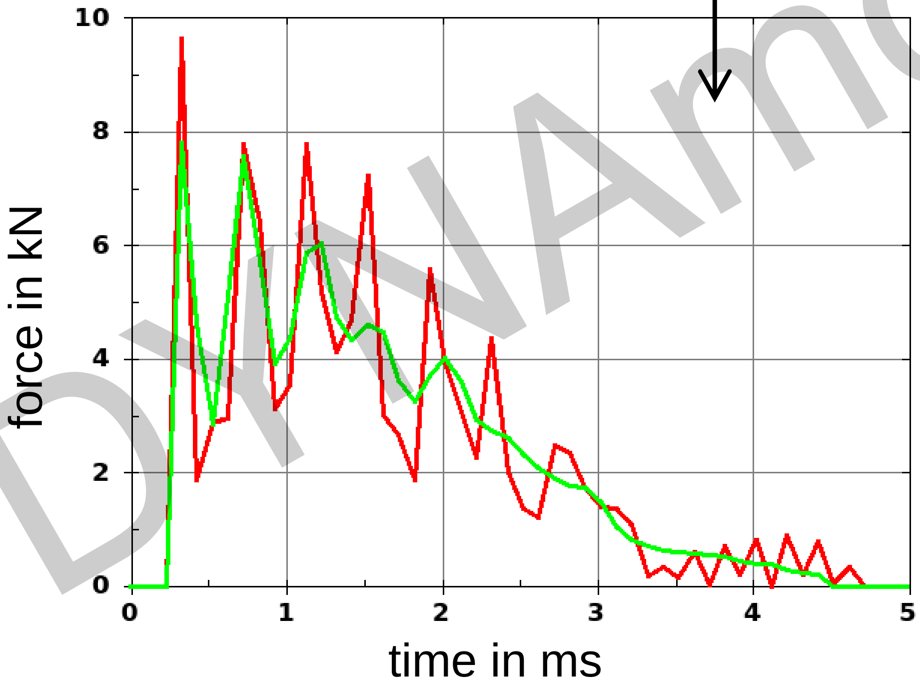
velocity [0 - 10 m/s]

Exercise

Dynamic implicit

- Check influence of Newmark parameters GAMMA and BETA
- Default: GAMMA=0.5, BETA=0.25
- Larger GAMMA and BETA values introduce numerical damping
- Often helps convergence
- But: affects solution!





GAMMA=0.5, BETA=0.25

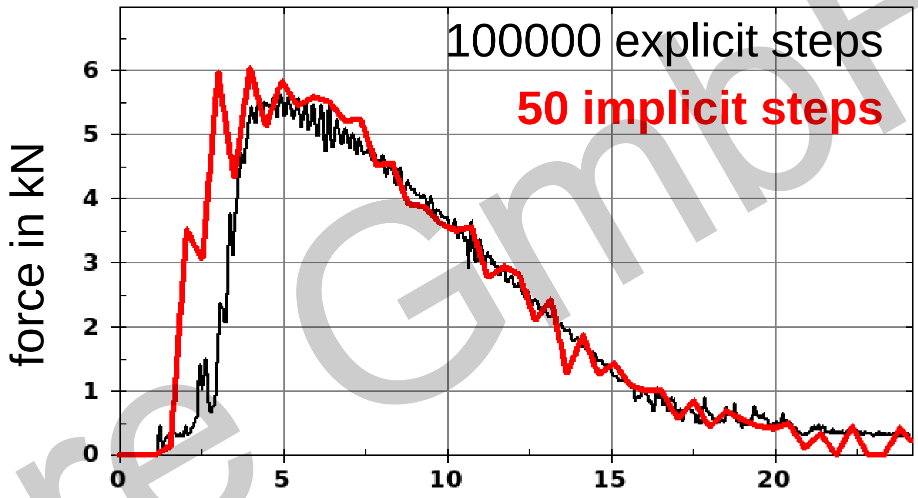
GAMMA=0.6, BETA=0.38

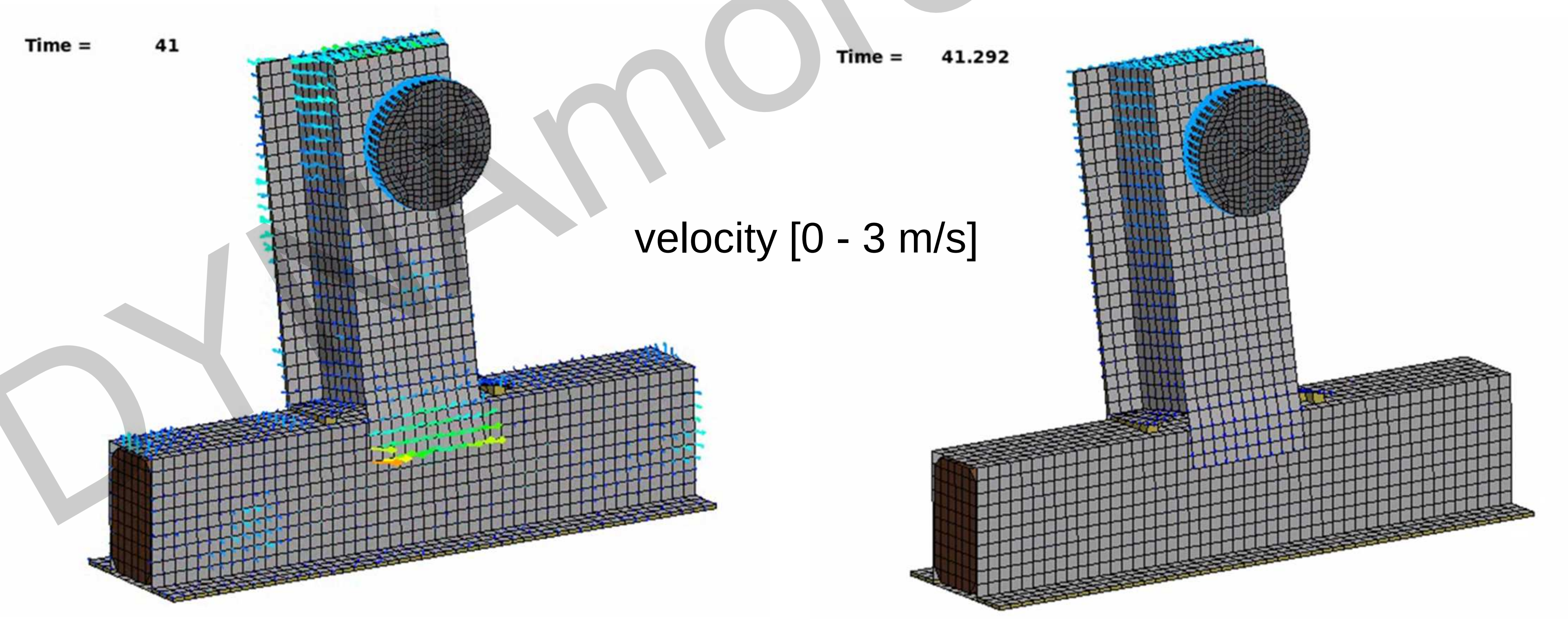
$$\beta > \frac{1}{4}\left(\frac{1}{2} + \gamma\right)^2$$

Exercise

Dynamic implicit

- Process time = 50 ms (“slow“)
- Compare to “slow“ explicit run

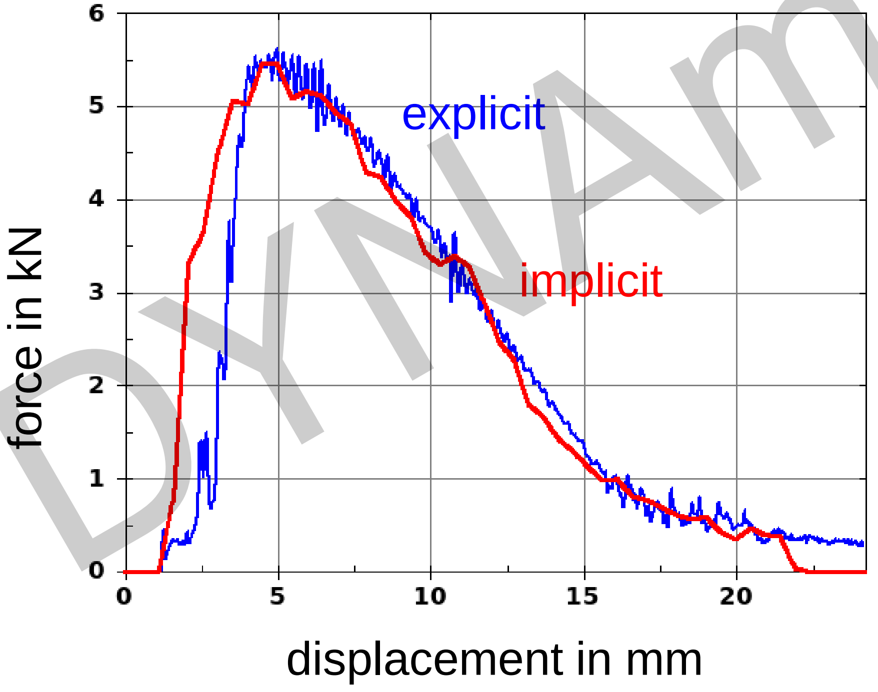


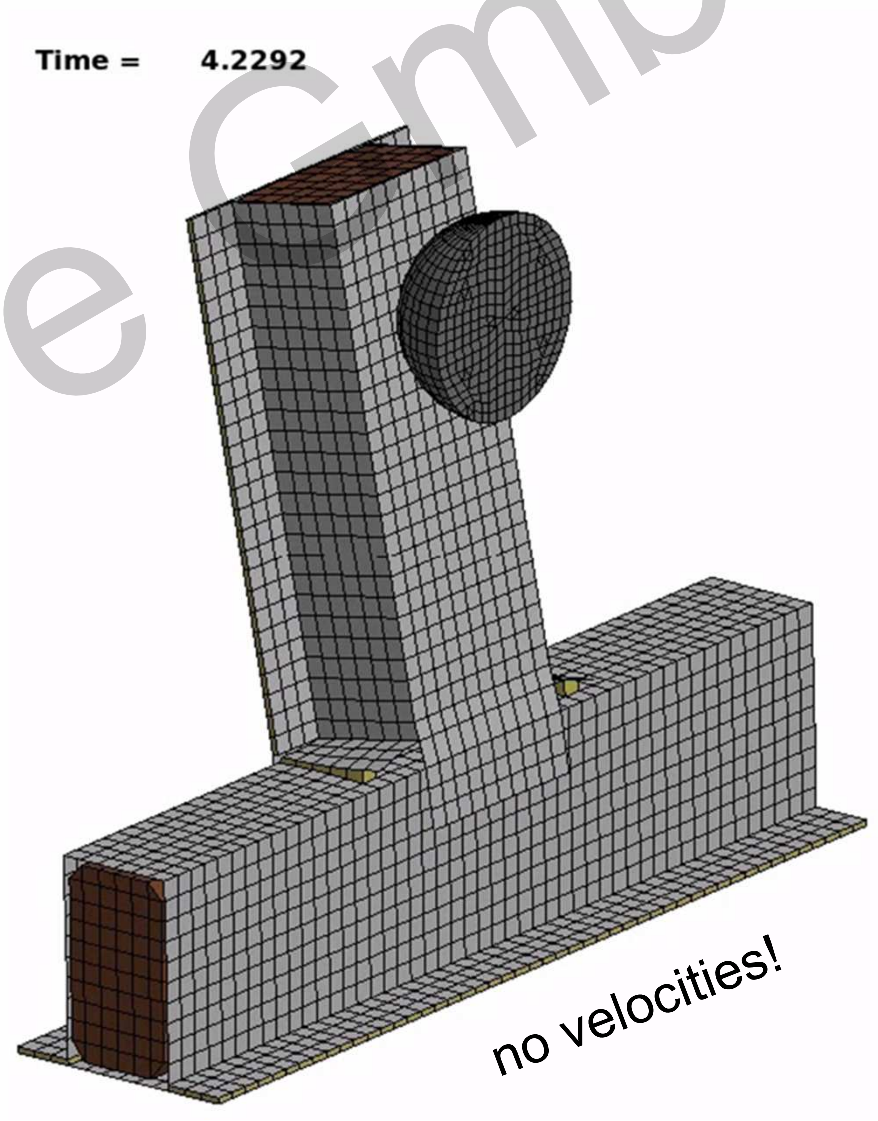


Exercise

Static implicit

- Remove *CONTROL_IMPLICIT_DYNAMICS
- No initial velocity, but prescr. motion
- "time" not physical anymore
- Real static response
- statically defined !?!

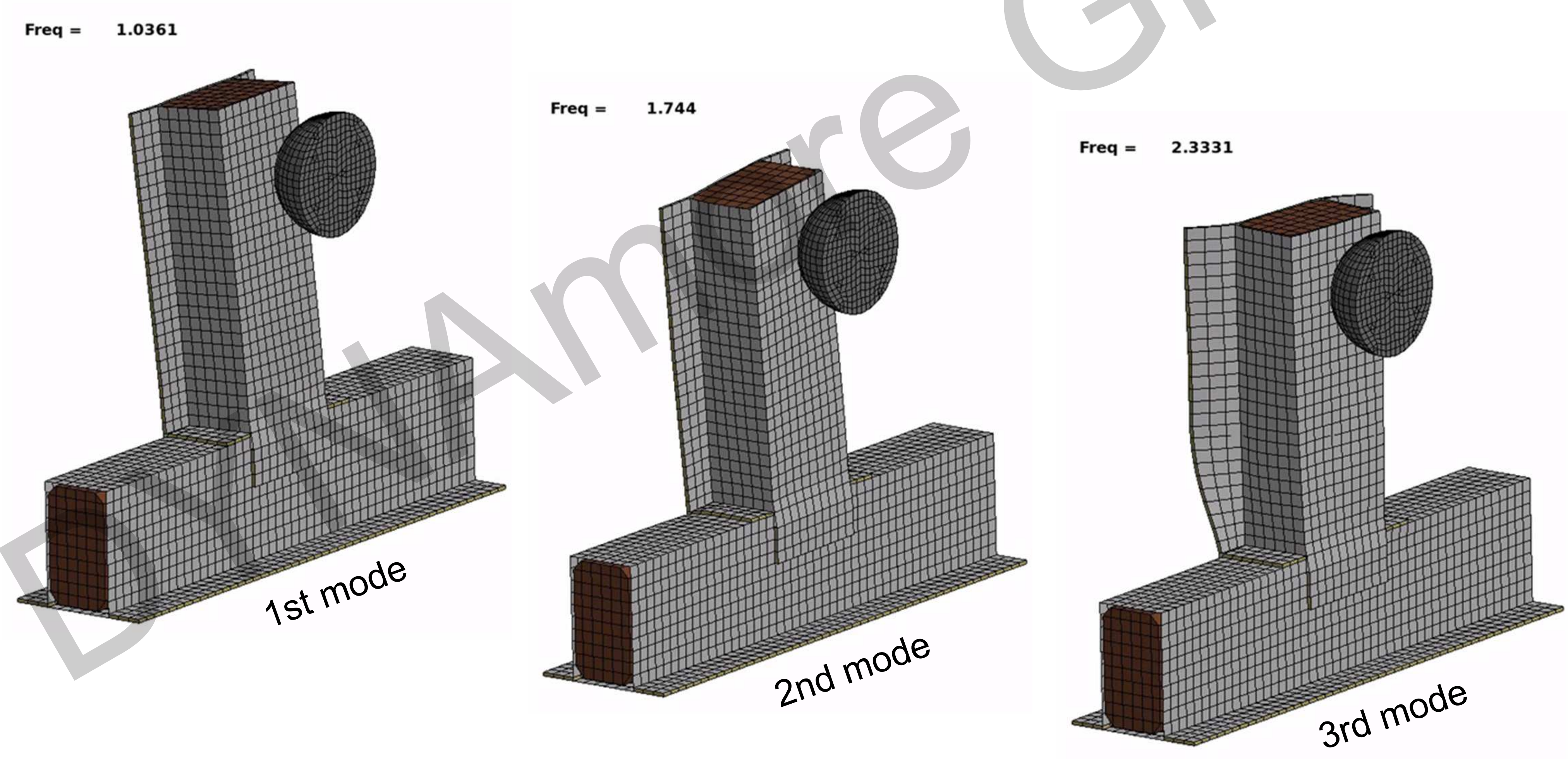




Exercise

Eigenvalue analysis

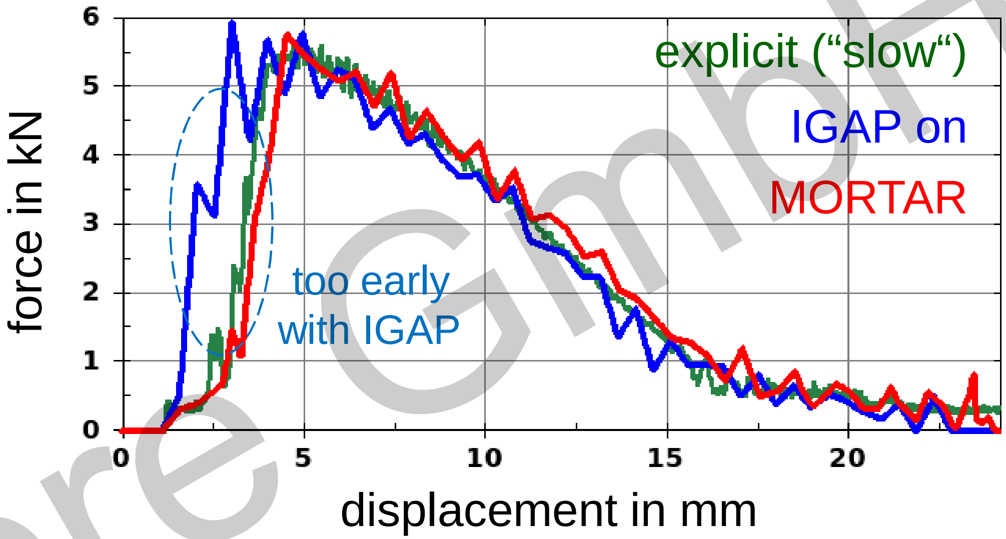
- *CONTROL_IMPLICIT_EIGENVALUE
- Reveals possible rigid body modes
- Superelevated deformations in d3eigv database

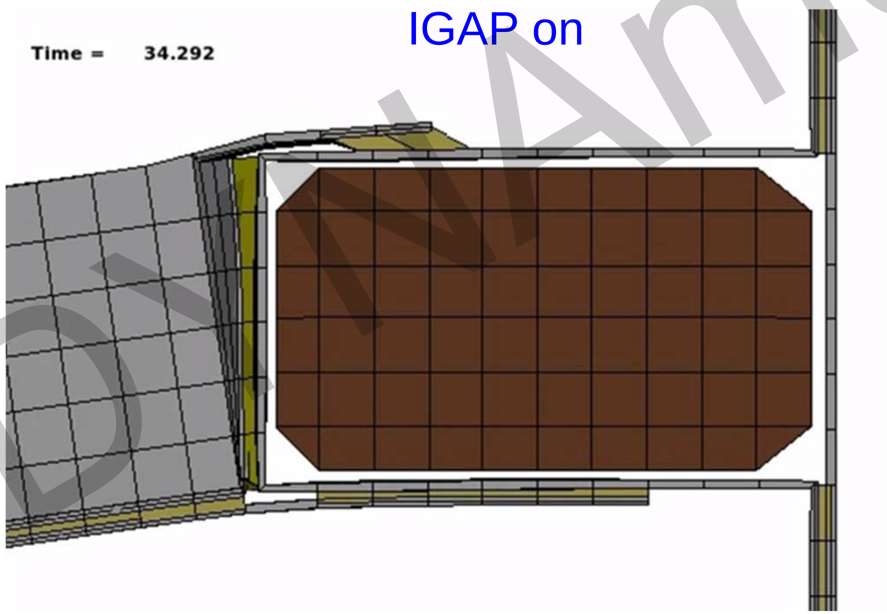


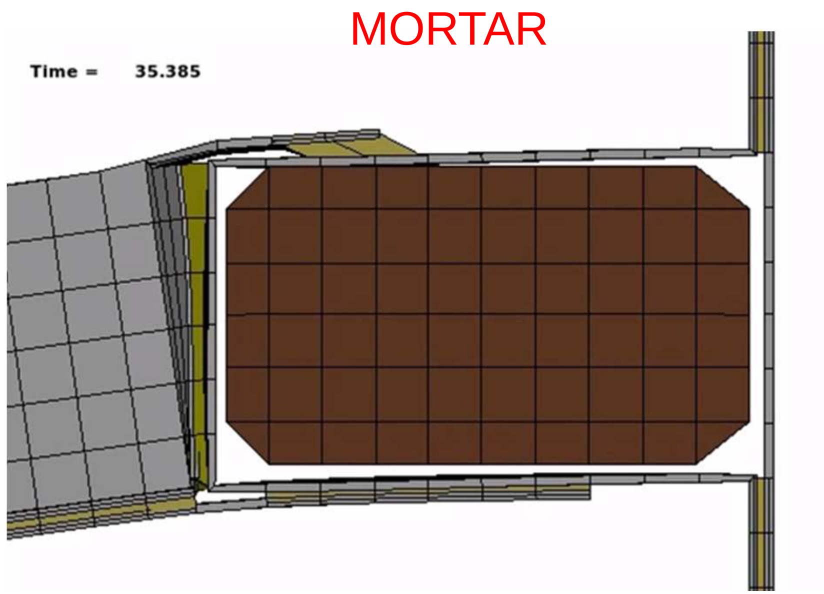
Exercise

Implicit contact

- Contact is very important issue (especially) in implicit analysis
- User should know about IGAP options ("sticky contact") and Mortar contact (continuous tangent)
- Dynamic implicit shown here







IGAP option

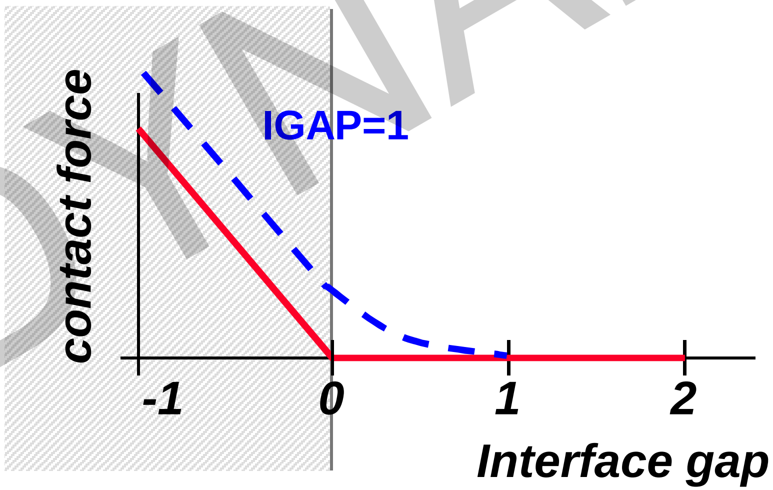
The IGAP option can significantly improve the convergence behavior but can also produce a "sticky" contact, that will resist opening of the contact gap

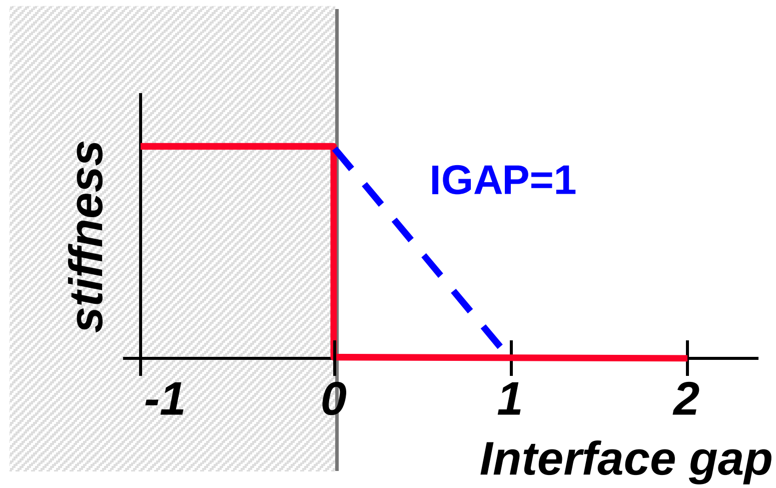
Contact gap flag controls treatment of nodes which are "close"

IGAP on optional contact interface card "C"

- IGAP = 2: no sticky contact
- IGAP = 1: improved convergence behavior (default in implicit)

Stiffness terms are added before there actually is contact. When contact is established the stiffness is 100%.





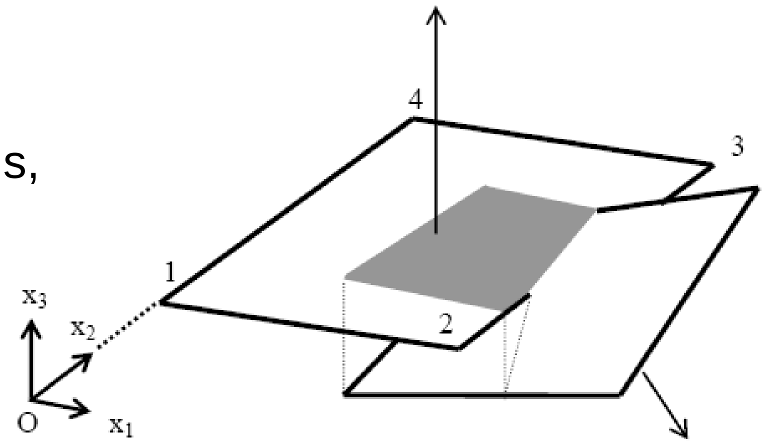
Mortar contact

Standard contact algorithms in LS-DYNA

- penalty based, double sided node to surface contacts
- in implicit, nodes tend to oscillate in and out of the contact
- often leads to convergence problems
- stiffness smoothing (IGAP=1) can help but accuracy suffers

New method: segment-to-segment mortar contact

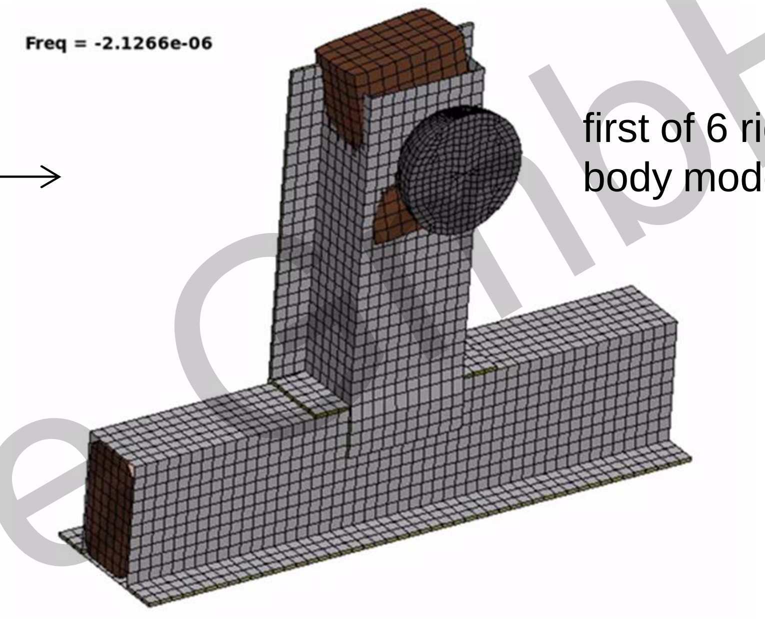
- penalty based segment to segment contact intended for implicit analysis
- Smooth force transition for penetration and segment to segment sliding
- Contact stress consistent with element shape functions, even for 10-noded tetrahedron type 16
- well suited for implicit: continuous tangent stiffness
- append optional suffix to contact keyword: *CONTACT_..._MORTAR
- IGAP option inactive

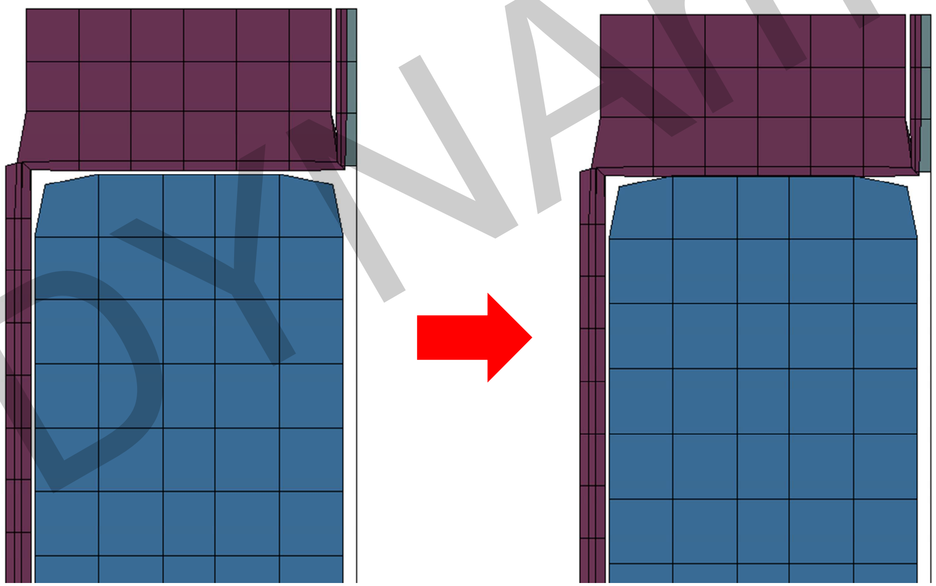


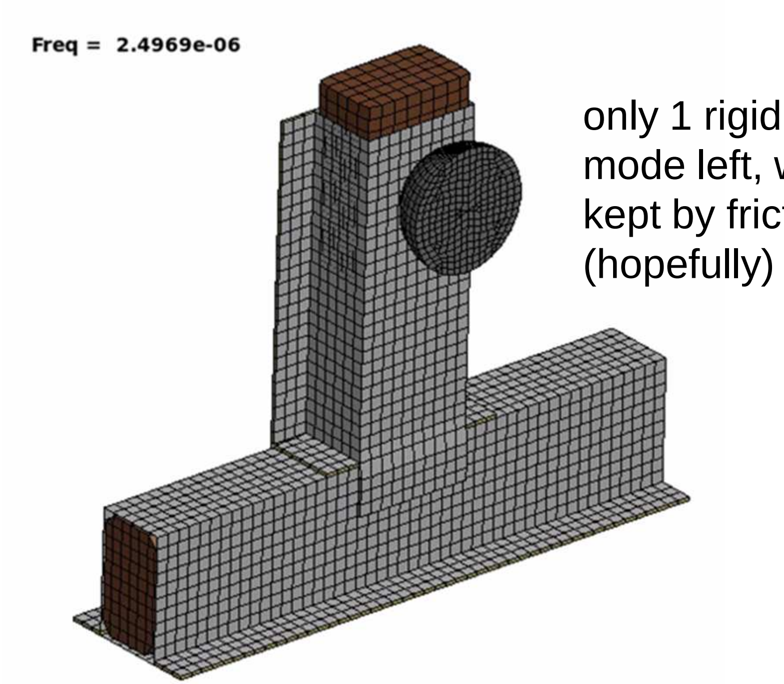
Exercise

Static implicit with Mortar contact

- "Missing" contact gap now reveals 6 rigid body modes (wooden block)
- Additional action(s) needed to allow for static analysis
- Slight **scaling** of wooden block's size causes initial contact penetration to get statically determined system
- +IGNORE=1 to avoid initial contact forces



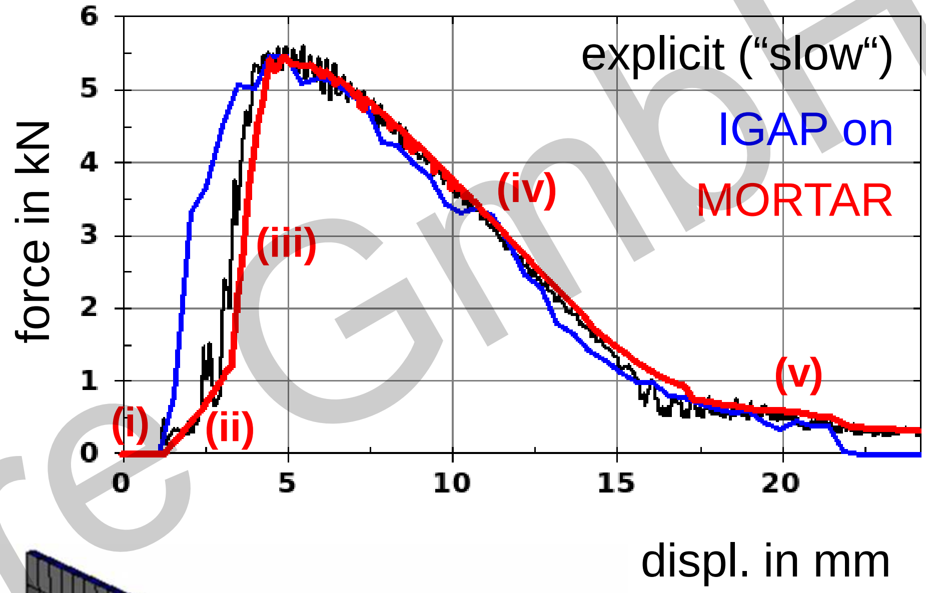


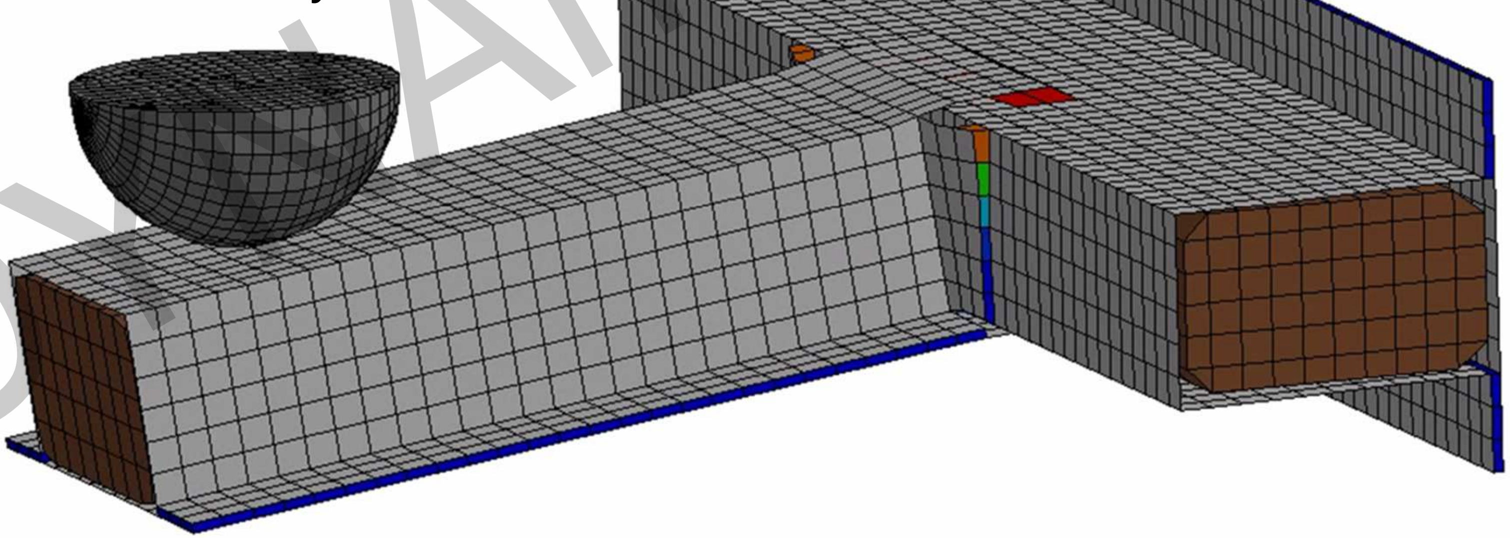


Exercise

Static implicit with Mortar contact

- More realistic results with Mortar contact
- 5 different phases can be observed: no contact (i), tipping (ii), elastic bending (iii), adhesive softening (iv), and glue failure (v)



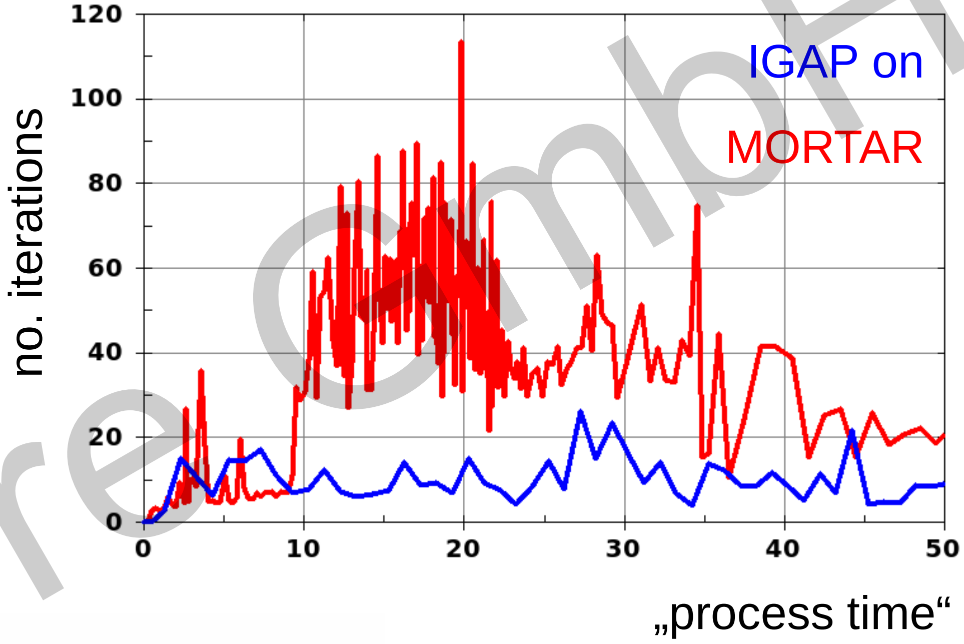


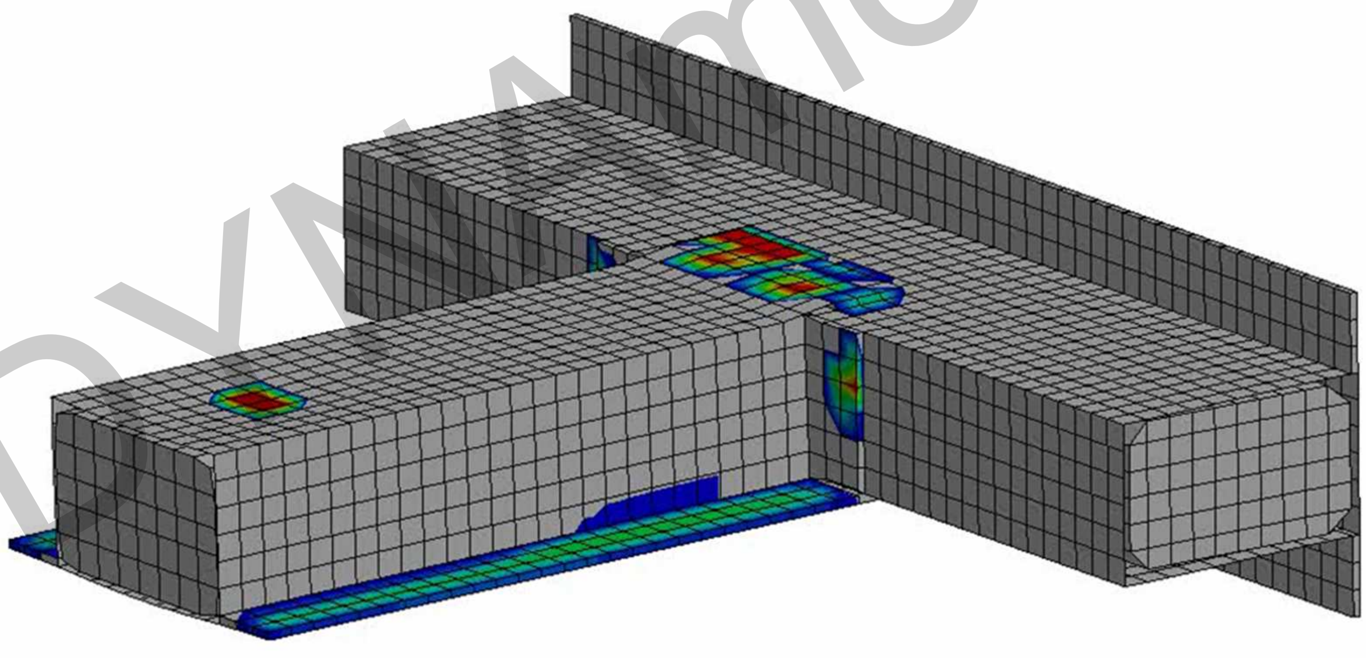
deformation and damage in adhesive layer

Exercise

Static implicit with Mortar contact

- Convergence becomes more difficult
- Reason(s) for difficulties can be detected with special "iteration plot database" d3iter
- Evolution of out-of-balance forces during iteration process shows critical areas



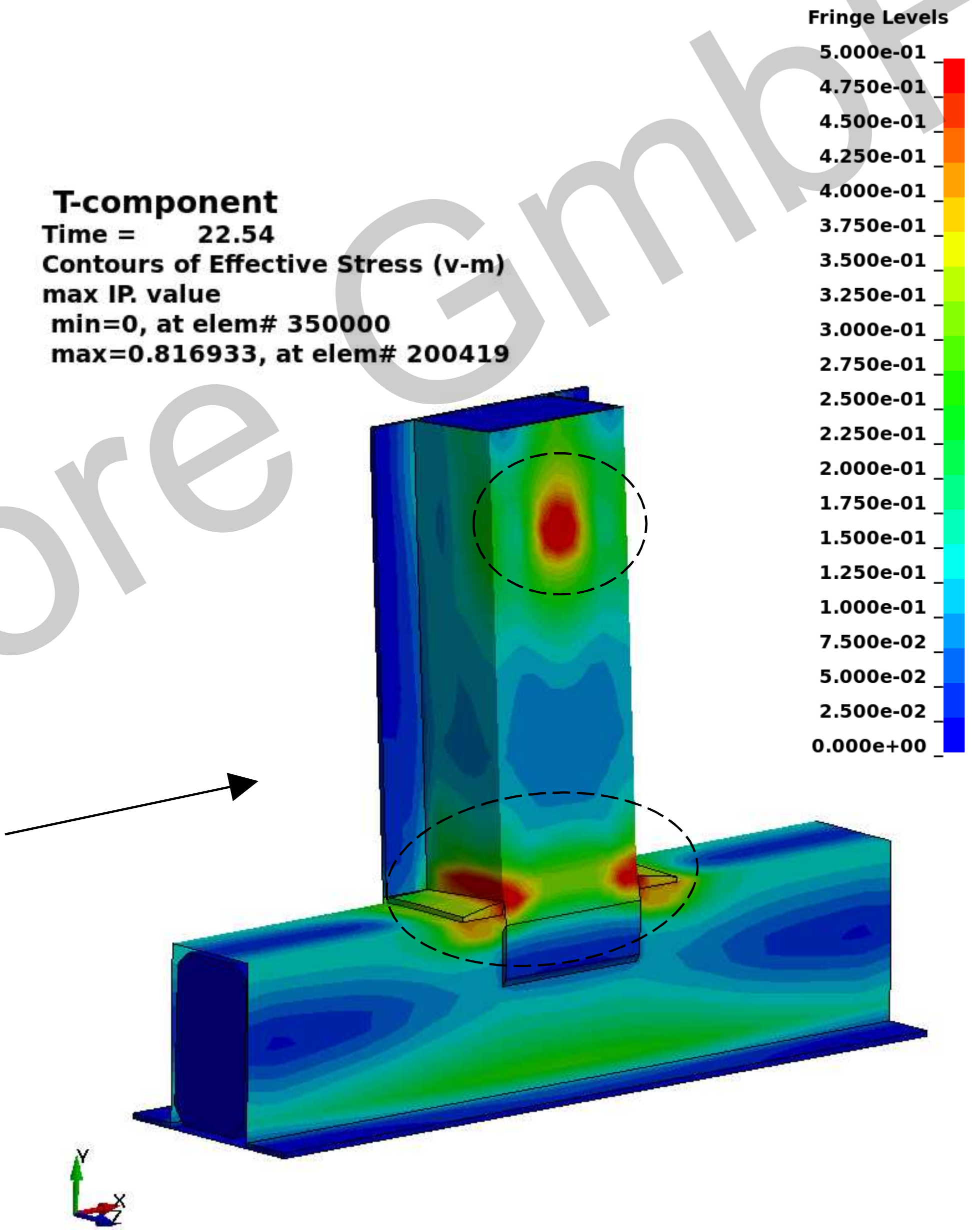


Troubles from damage evolution in cohesive material and contact to impactor

Exercise

Ideas for improvement

- Perhaps Full Newton better suited for this problem (ILIMIT=1)
- Modify other implicit settings (timestep size, tolerances, …) or contact parameters (IGAP, …)
- But maybe better to improve the model itself:
- Replacement for cohesive material (MAT_186 with smooth curve?)
- Mesh refinement in critical areas?
- Dynamic implicit – very slow
- …



Conclusion

- Explicit analysis runs into its limits for long duration processes or even real static load cases.
- Therefore, implicit analysis is often preferrable.
Actually, computation time can be decreased in many cases.
- But: more demanding to get a solution, especially if large deformations, contact, and nonlinear material behavior is involved.
- Users must be aware of crucial differences between explicit (e.g. time step size) and implicit (e.g. "smooth" model)
- Often greater effort is needed to obtain a functional model in implicit, but also the feeling of success is greater in the end