

LS-DYNA implicit Workshop

nonlinear Solver

Alexander Gromer, Dr. Tobias Erhart, Dr. Thomas Borrvall
Bamberg, 11.10.2016

Copyright: Dynamore GmbH, Industriestr. 2, 70565 Stuttgart

- Intro
- R9 Solver
- Walkthrough: NCAC Toyota Yaris model conversion to implicit
- LS-DYNA implicit with AVX2
- Convergence behavior monitoring
- Summary

Motivation: Why implicit?

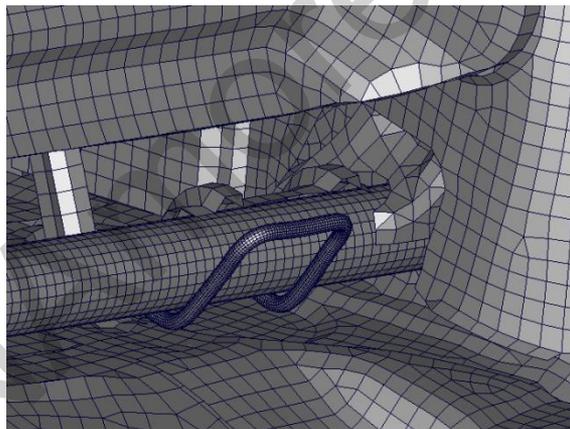
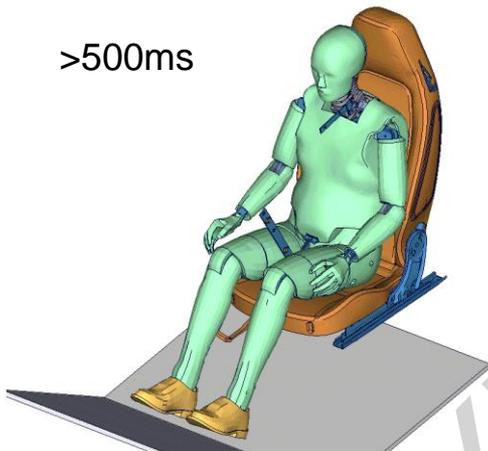
pre-stressed, quasi statically loaded structures

long duration analysis

different scales in
discretization

different time scales in
process

>500ms



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



LS-DYNA provides explicit and implicit solution schemes

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

LS-DYNA implicit features

Basic equipment

- Newton, Quasi-Newton, arclength methods
- direct and iterative solvers
- automatic step size adjustment
- Newmark methods with consistent mass matrix
- ...



LS-DYNA implicit features

Outstanding features

- one code – one license – one input – one output
- switching between implicit and explicit in one run
- high scalability through MPP
- mortar contact
- post-processing of residual (out-of-balance) forces



- Intro
- **R9 Solver**
- Walkthrough: NCAC Toyota Yaris model conversion to implicit
- LS-DYNA implicit with AVX2
- Convergence behavior monitoring
- Summary

Latest official release: R9.0.1

NEW

- Release R9.0.1 from August 2016
- Contains several new features in the areas of solid mechanics, multiphysics, and implicit
- Details: <http://www.dynasupport.com/news/ls-dyna-r9.0.1-r9.109912-released>
- **Highly recommended for implicit analyses**

General philosophy of LS-DYNA implicit

Increased accuracy implies better convergence

New feature: IACC **NEW**

*CONTROL_ACCURACY

Card 1	1	2	3	4	5	6	7	8
Variable	OSU	INN	PIDOSU	IACC				
Type	I	I	I	I				
Default	1	2	0	0				

Implicit accuracy option IACC=1

- Higher accuracy in selected material models (24, 123)
 - Fully iterative plasticity, tightened tolerances, smooth failure
- Strong objectivity and consistency in selected tied contacts
 - Physical (only ties to degrees of freedoms that are "real")
 - Finite rotation
- Strong objectivity in selected element types
 - Finite rotation support for hypoelasticity

- Intro
- R9 Solver
- **Walkthrough: NCAC Toyota Yaris model conversion to implicit**
- LS-DYNA implicit with AVX2
- Convergence behavior monitoring
- Summary

CCSA (former NCAC) Toyota Yaris model



~ 1.2 mio nodes

~ 1.2 mio elements

1 global contact

1 global tied contact (spotweld)

Preparing the Yaris model for LS-DYNA implicit

- The model has a typical car crash model setup
- Idea: Do as less modifications as necessary to make the model „implicit ready“ and keep the explicit model structure/philosophy
- 3 step approach:
 - (1) Eigenvalue analysis
 - (2) “No load run”
 - (3) Small test load (e.g. gravity load)

General model modifications:

- Changed control cards (for explicit analysis) to crash model recommendation
- Added *PART_CONTACT with appropriate OPTT to all parts in contact definition
 - $OPTT = 0.9 * \text{true shell thickness (secant errors)}$
 - Some volume parts are only represented by shell surfaces. In order to match the part masses the shells have high thickness values.
- Depenetration of the model

- Removed the seat foam parts (not needed for the following studies)

Implicit subset of control cards

*CONTROL_ACCURACY

```
$      osu      inn      pidosu      iacc
      1         4
      1
```

*CONTROL_IMPLICIT_GENERAL

```
$      imflag      dt0
      1         0.01
```

*CONTROL_IMPLICIT_SOLUTION

```
$      nsolvr      ilimit      maxref      dctol      ectol      rctol      lstol      abstol
      12           6           15       2.5e-3
$      dnorm      diverg      istif      nlprint      nlnorm      d3itctl
      1           2           2           1
```

*CONTROL_IMPLICIT_AUTO

```
$      iauto      iteopt      itewin
      1           25          5
```

*CONTROL_IMPLICIT_DYNAMICS

```
$      imass      gamma      beta
      1           0.55      0.28
```

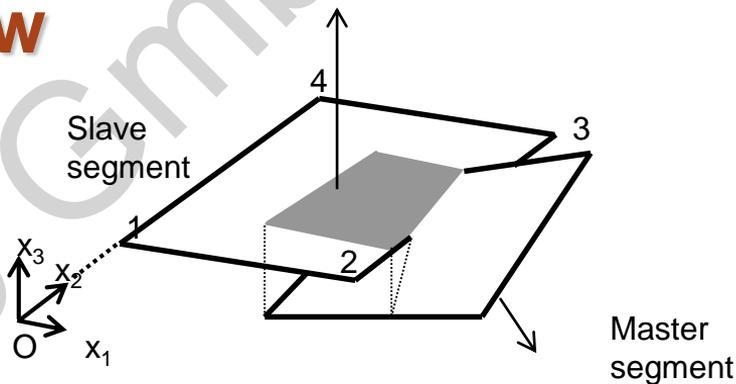
Contact definition

*CONTACT_AUTOMATIC_SINGLE_SURFACE_MORTAR

```
$#      ssid      msid      sstyp      mstyp      sboxid      mboxid      spr      mpr
      1000002      0      2
$#      fs      fd      dc      vc      vdc      penchk      bt      dt
      0.200000
$#      sfs      sfm      sst      mst      sfst      sfmt      fsf      vsf
$#      soft      sofsc1      lcidab      maxpar      sbopt      depth      bsort      frcfrq
$#      penmax      thkopt      shlthk      snlog      isym      i2d3d      sldthk      sldstf
$#      igap      ignore      dprfac      dtstif      unused      unused      flangl
      2
$
```

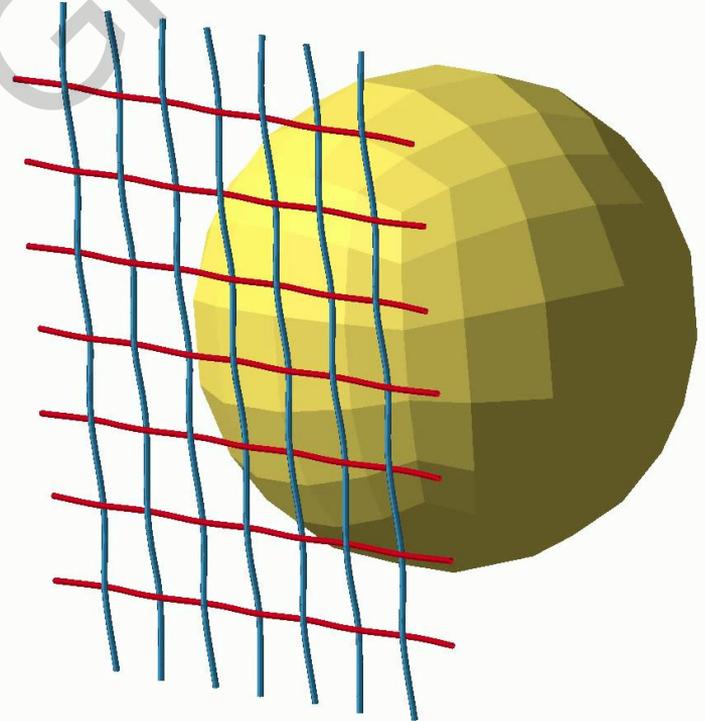
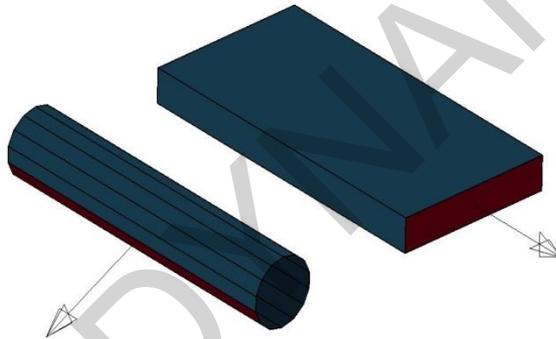
Mortar Contact: brief overview

- Penalty based segment to segment contact
 - Finite element consistent force
 - Continuous force displacement relation
- Parabolic constitutive law
 - Continuous stiffness displacement relation
- Relatively expensive
 - Intended for implicit analysis, slow in explicit analysis (at this time)
 - **To the best of our knowledge best total implicit contact algorithm**



Mortar Contact: beams and shell edges

- Flat edge contact always apply in automatic contact
- Beam lateral surfaces are discretized into segments with mortar contact applied to each segment
- From R9: Support "rolling beams"



Mortar Contact: stiffness and release

$$\sigma_c = \alpha \varepsilon K_s f\left(\frac{d}{\varepsilon d_c}\right) \quad f(x) = \begin{cases} \frac{1}{4}x^2 & x < \frac{1}{4\varepsilon} \\ \text{cubic function that depends on IGAP} & \frac{1}{4\varepsilon} \leq x \end{cases}$$

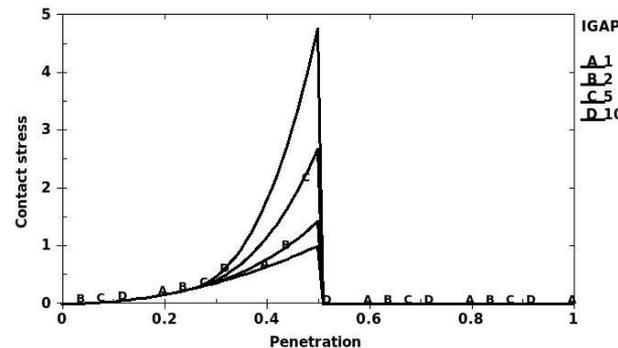
α = stiffness scaling factor (SFS*SLSFAC)

K_s = stiffness modulus of slave segment

d = penetration distance

$\varepsilon = 0.03$

d_c = characteristic length



- Contact is released if penetration is larger than half characteristic length *after* equilibrium
- Information of penetration may be requested

(1) Eigenvalue check

- Basically a check of the stiffness matrix
- only a linearized version of the model is considered
- Eigenvalues must be ≥ 0 (if we want to run a static analysis EVs > 0)

```
$  
*CONTROL_IMPLICIT_EIGENVALUE  
$      neig  
      20  
$
```

 eigout (ASCII)

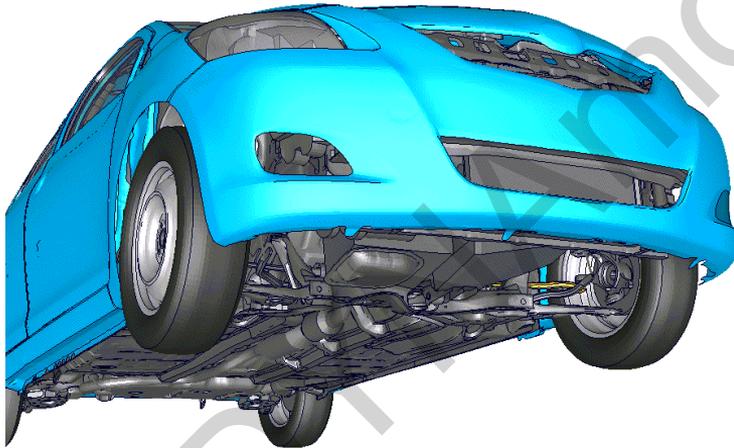
 d3eigv

```
ground  
ls-dyna mpp.109095 d          date 07/06/2016  
  
r e s u l t s   o f   e i g e n v a l u e   a n a l y s i s :  
  
problem time = 1.00000E-02  
  
(all frequencies de-shifted)  
  
|----- frequency -----|  
MODE   EIGENVALUE          RADIANS          CYCLES          PERIOD  
1   -2.439537E-03      4.939167E-02      7.860929E-03      1.272114E+02  
2   -8.982204E-04      2.997032E-02      4.769925E-03      2.096469E+02  
3   -7.700314E-06      2.774944E-03      4.416460E-04      2.264257E+03  
4    6.172319E-03      7.856411E-02      1.250387E-02      7.997527E+01  
5    1.321770E+02      1.149683E+01      1.829777E+00      5.465148E-01  
6    1.509247E+02      1.228514E+01      1.955241E+00      5.114460E-01  
7    2.477678E+02      1.574064E+01      2.505201E+00      3.991696E-01  
8    4.077650E+02      2.019319E+01      3.213846E+00      3.111536E-01
```

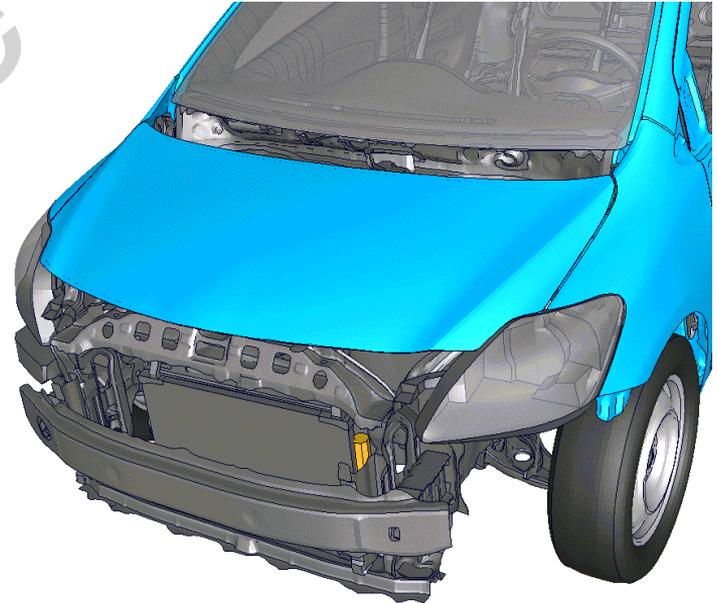
(1) Eigenvalue check

- Example of first eigen modes

≈ 0 Hz eigen mode



≈ 0 Hz eigen mode



(1) Model modifications:

- Removed unsupported rotation d-o-f of wheels by adding small frictional moment to the wheel bearings with *CONSTRAINT_JOINT_STIFFNESS
- Removed unsupported rotation d-o-f steering linkage by adding small frictional moment to the wheel bearings with *CONSTRAINT_JOINT_STIFFNESS
- Fixed some of the engine parts properly

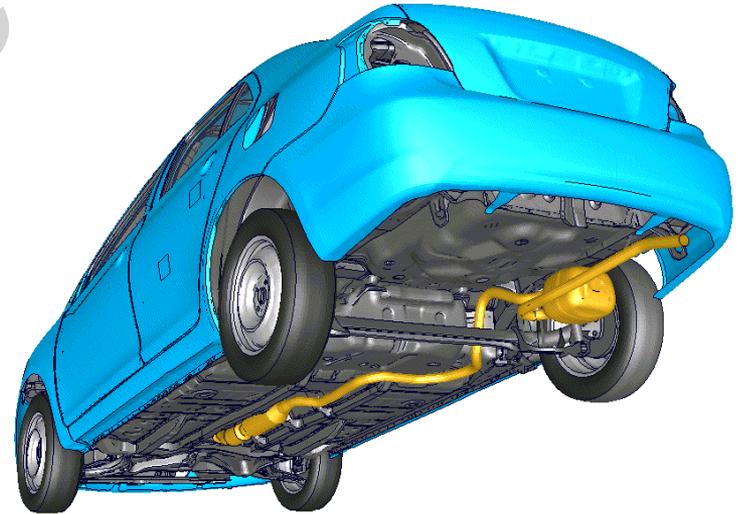
(1) Eigenvalue final check

- Lowest eigen modes after optimization

1.3Hz eigen mode



2.3Hz eigen mode



(2) No-load run

- All definitions in the model are considered
- For a well defined model model this means:

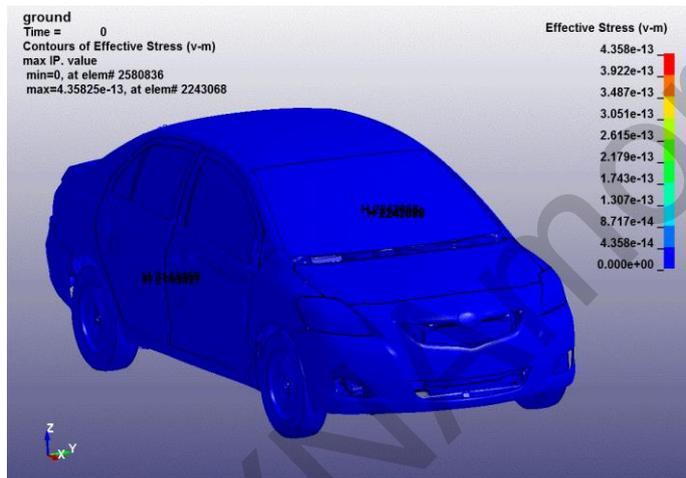


- In case of slow convergence there might be
 - Still penetrations
 - Bad defined materials
 - ...

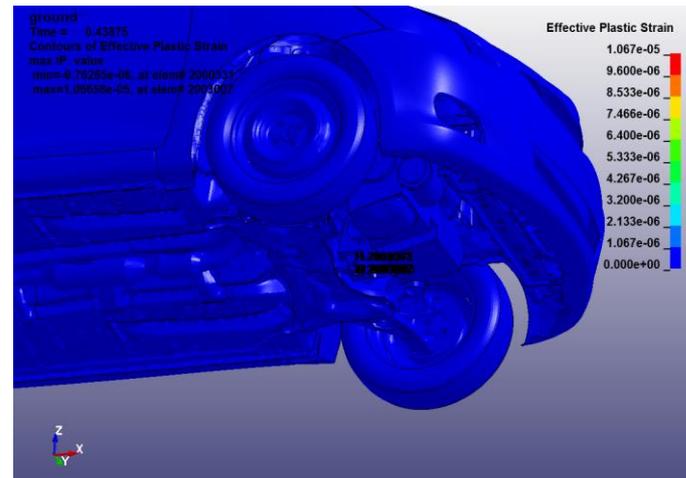
(2) No-load run

- Visualize stresses, plastic strains, residual forces, ...

V-M-stress



plastic strains



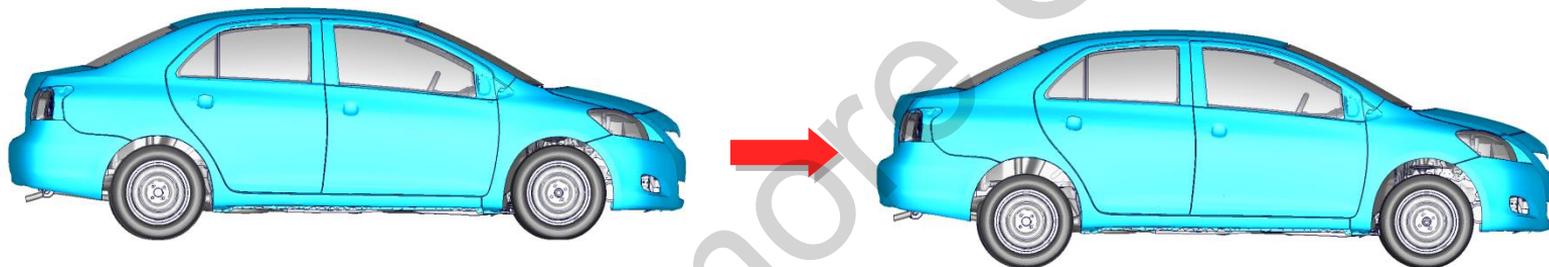
- Here: LS-DYNA struggles with rubber bearing material definition (Blatz/Ko rubber) → replaced by *MAT_ELASTIC with corresponding parameters

(3) Small test load

- Final quality check for the model
- Expect plausible results
- Expect “N o r m a l T e r m i n a t i o n”

Shock absorber loading setup

- Generated the geometry of an unloaded under-carriage



- Added a rigid ground model with

```
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_ID
```

```
100tires2ground
```

```
$#      ssid      msid      sstyp      mstyp      sboxid      mboxid      spr      mpr
      2000008    10000001      2          3
$#      fs        fd        dc        vc        vdc        penchk      bt        dt
      0.100000
$#      sfs        sfm        sst        mst        sfst        sfmt        fsf        vsf
```

Static shock absorber loading

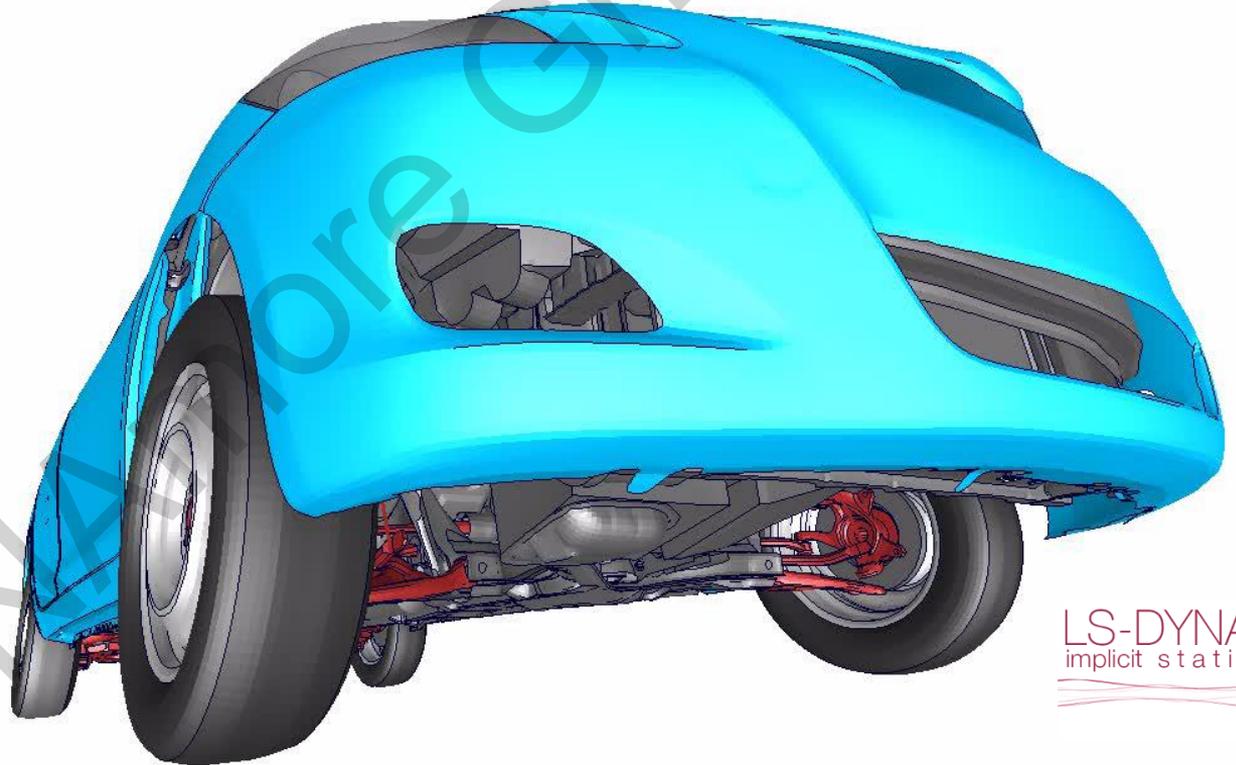
three load steps

- 1) inflate tires
- 2) Initiating contact
- 3) gravity load



Static shock absorber loading

Solution in 73 steps
5.5 hours on 16cores



Dynamic shock absorber loading setup

- Added ground with

*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_ID

100tires2ground

\$#	ssid	msid	sstyp	mstyp	sboxid	mboxid	spr	mpr
	2000008	10000001	2	3				
\$#	fs	fd	dc	vc	vdc	penchk	bt	dt
	0.100000							
\$#	sfs	sfm	sst	mst	sfst	sfmt	fsf	vsf

- Added implicit dynamics control card

*CONTROL_IMPLICIT_DYNAMICS

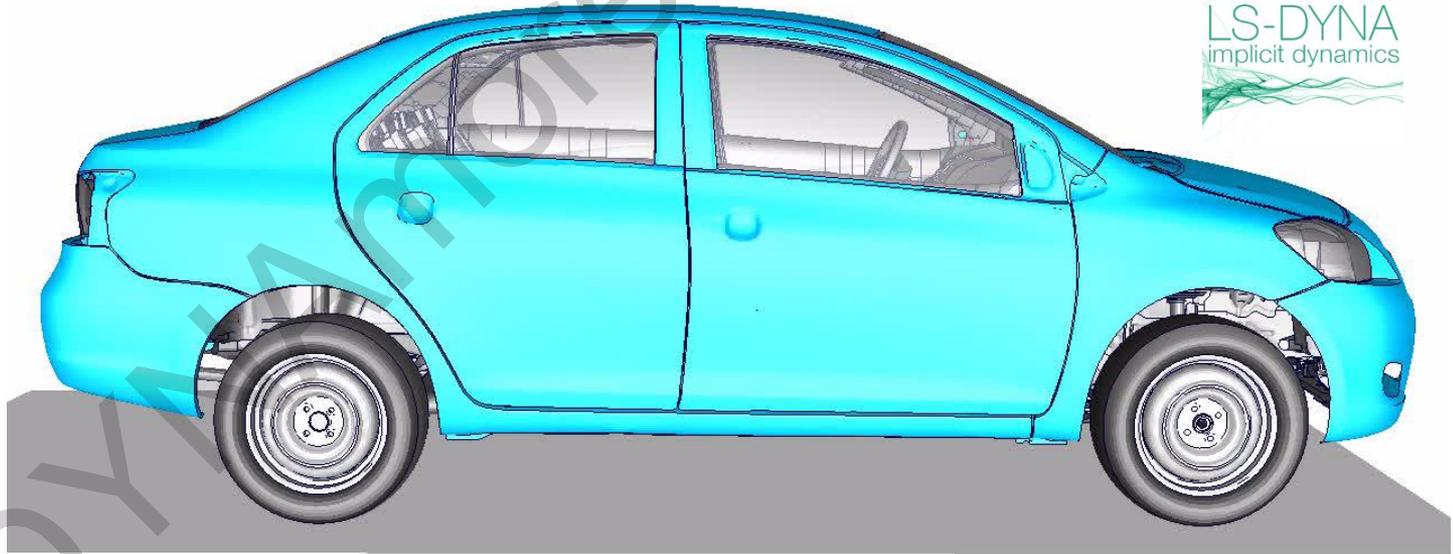
\$	imass	gamma	beta
	1	0.55	0.28

Dynamic shock absorber loading

3 seconds simulation time

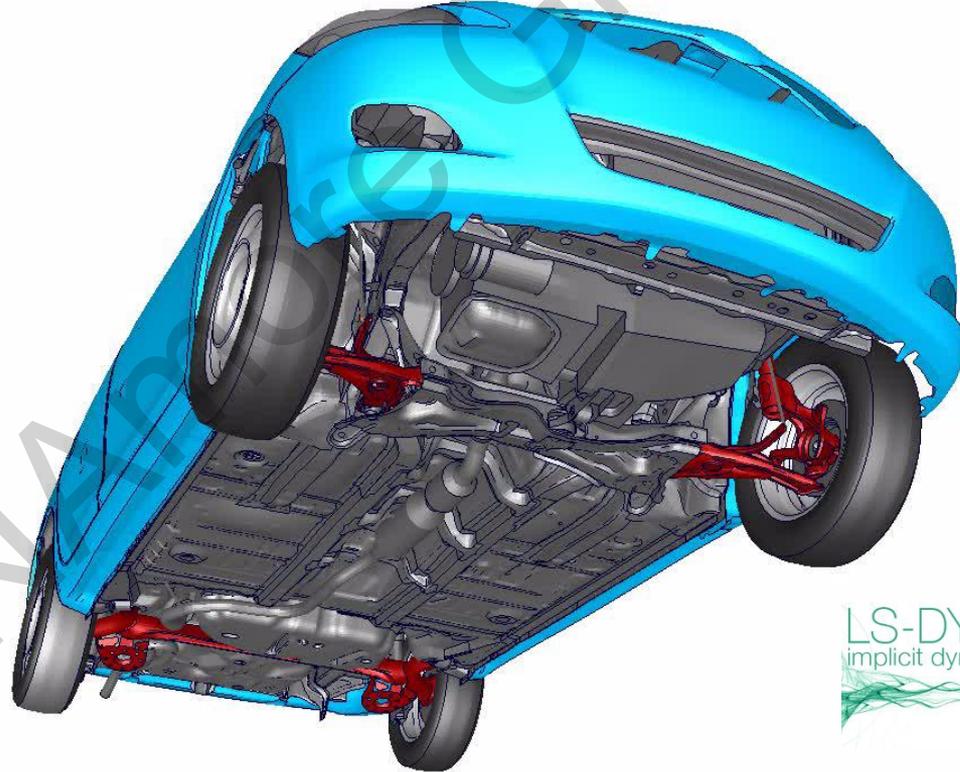
Slight numerical damping

- 1) inflate tires
- 2) gravity load

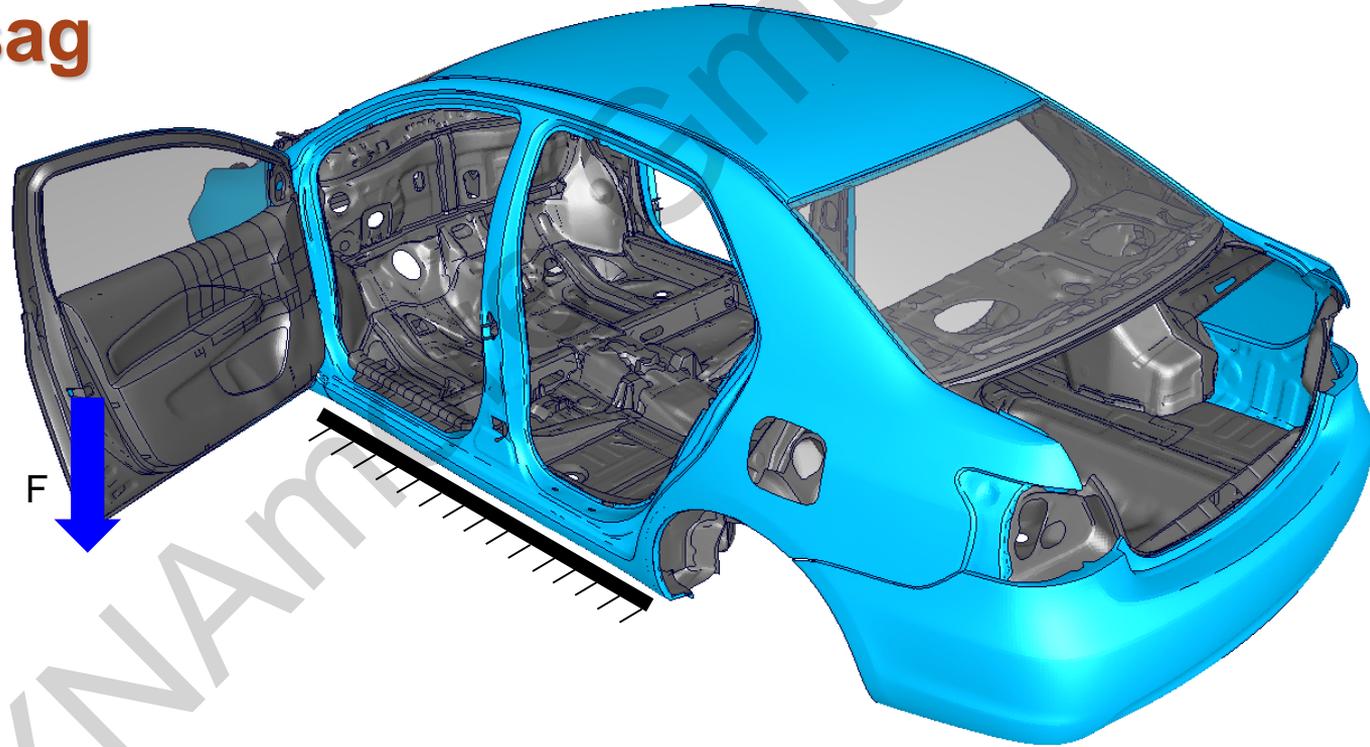


Dynamic shock absorber loading

Solution in 103 steps
7.5 hours on 16cores



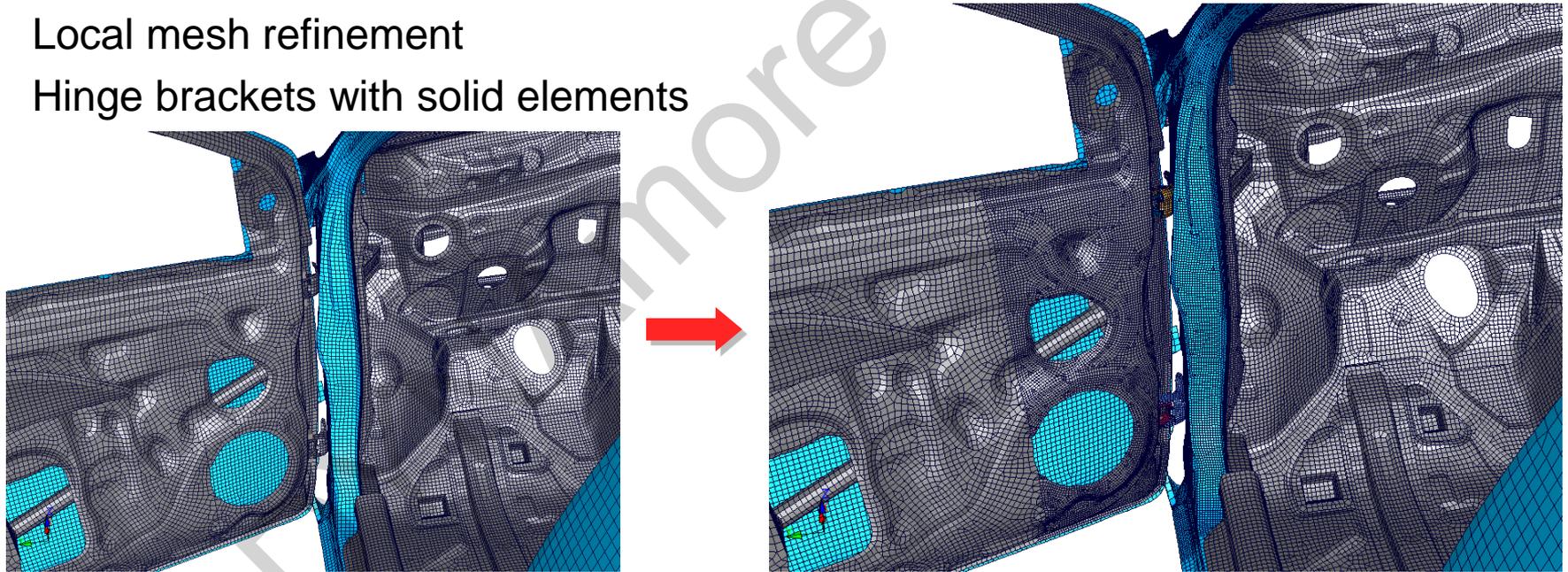
Static door sag



- two load steps
 - (1) gravity load
 - (2) Door load

Static door sag: modifications

- Removed non-necessary parts of the model
- Loadcase definitions
- Local mesh refinement
- Hinge brackets with solid elements

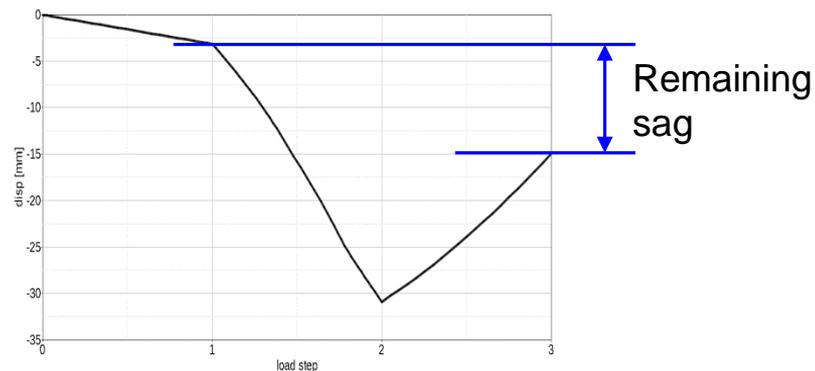
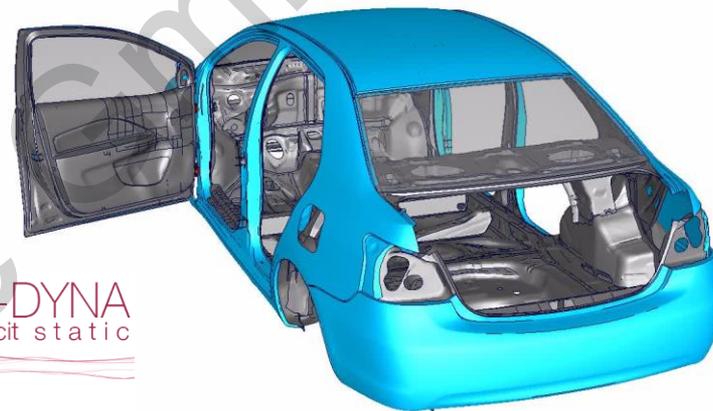
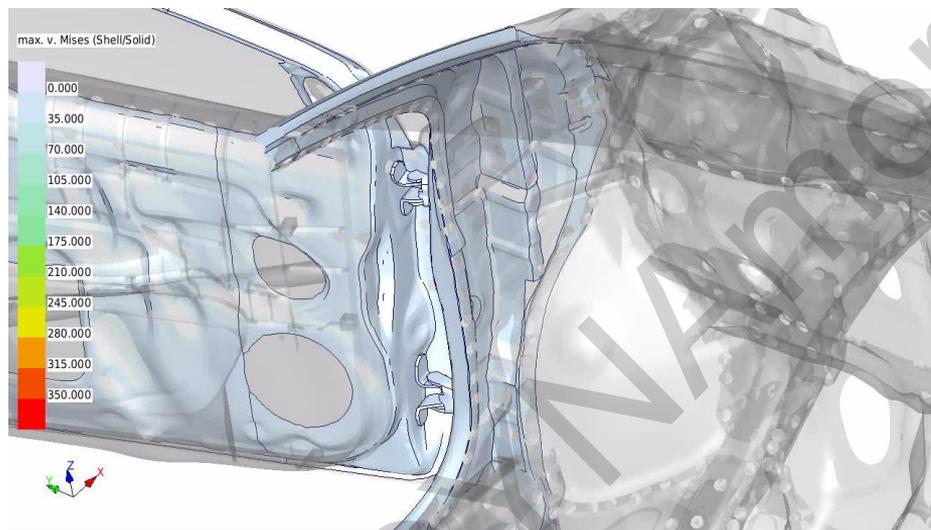


Conversion to implicit

Static door sag

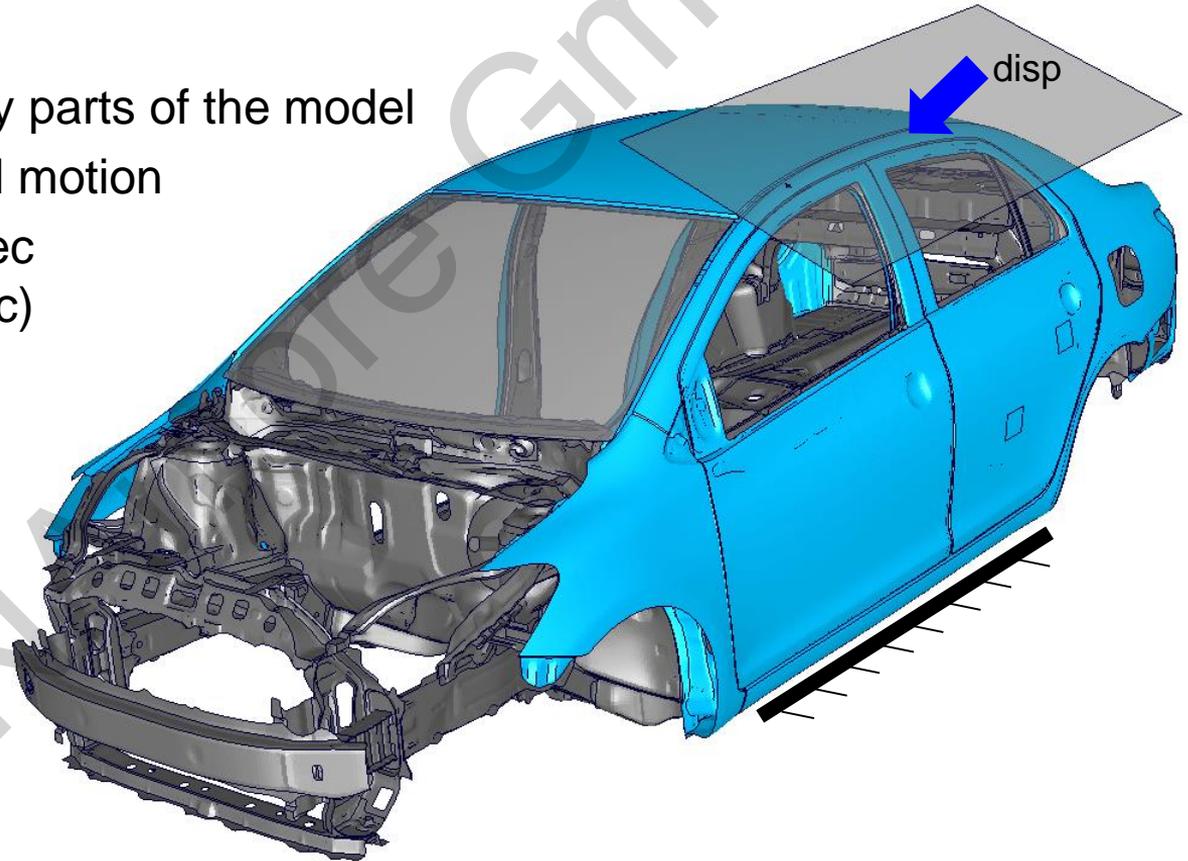
Solution in 94 steps (conservative)

2 h 50min on 16cores



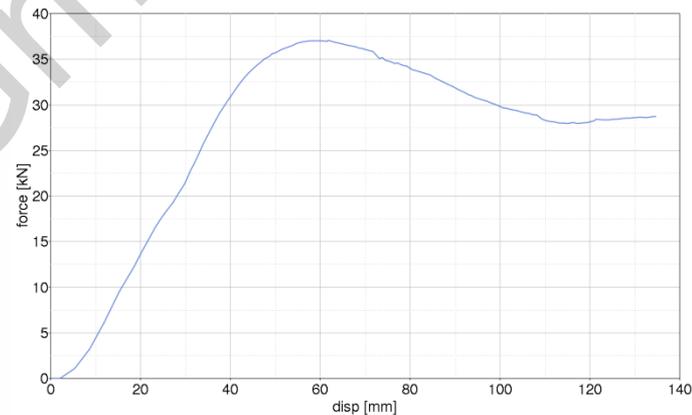
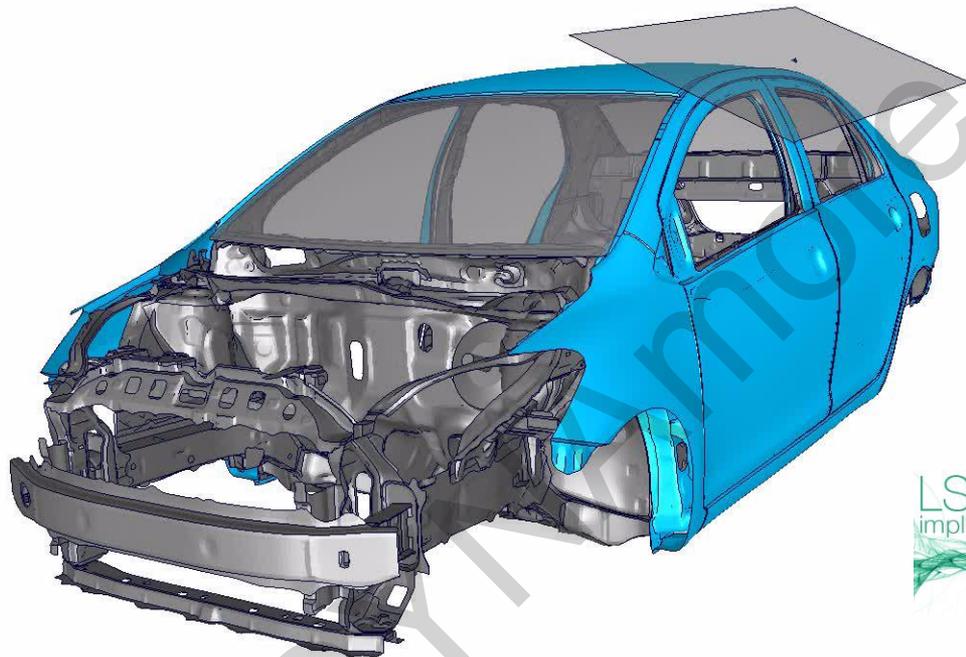
Roof crush

- Removed non-necessary parts of the model
- Impactor with prescribed motion
- Applying load within 2 sec
(Termination time 2.2 sec)



Roof crush

- Takes about 20h on 10cores



Remarks

- Material definitions and connection modelling is not on most OEMs state-of-the-art level. However, it has on a quite detailed level of modeling. For the investigations of this project all required parts and model functionality was present in the baseline model.
- Model size is adequate but not as large as OEMs current models (up to 7mio elements)
- Conversion process may look straight forward but in deed it is not.

- Intro
- R9 Solver
- Walkthrough: NCAC Toyota Yaris model conversion to implicit
- **LS-DYNA implicit with AVX2**
- Convergence behavior monitoring
- Summary

Advanced Vector eXtensions

- Extensions to the x86 instruction set architecture
- Introduced 2013 with the Haswell processor generation
- Includes for example FMA3: solves

$$(A*B)+C=D$$

in a single CPU-cycle

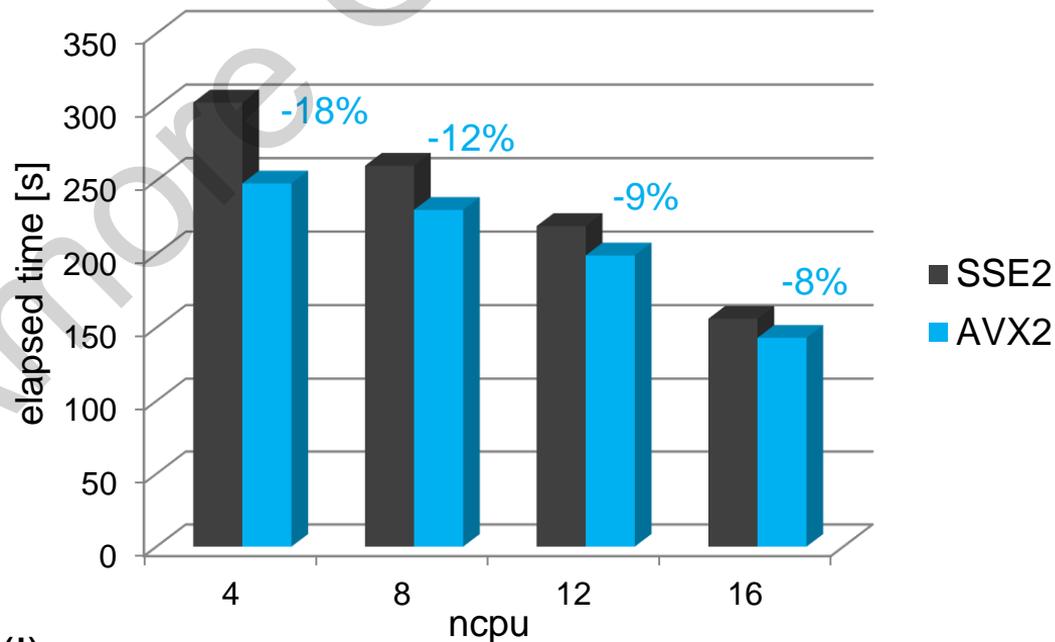
- AMD's counterpart: Carrizo with Excavator microarchitecutre (Released end of 2015)

ls-dyna_mpp_d_R9_107411_x64_redhat54_ifort131_sse2_platformmpi

ls-dyna_mpp_d_R9_107411_x64_redhat54_ifort131_avx2_platformmpi

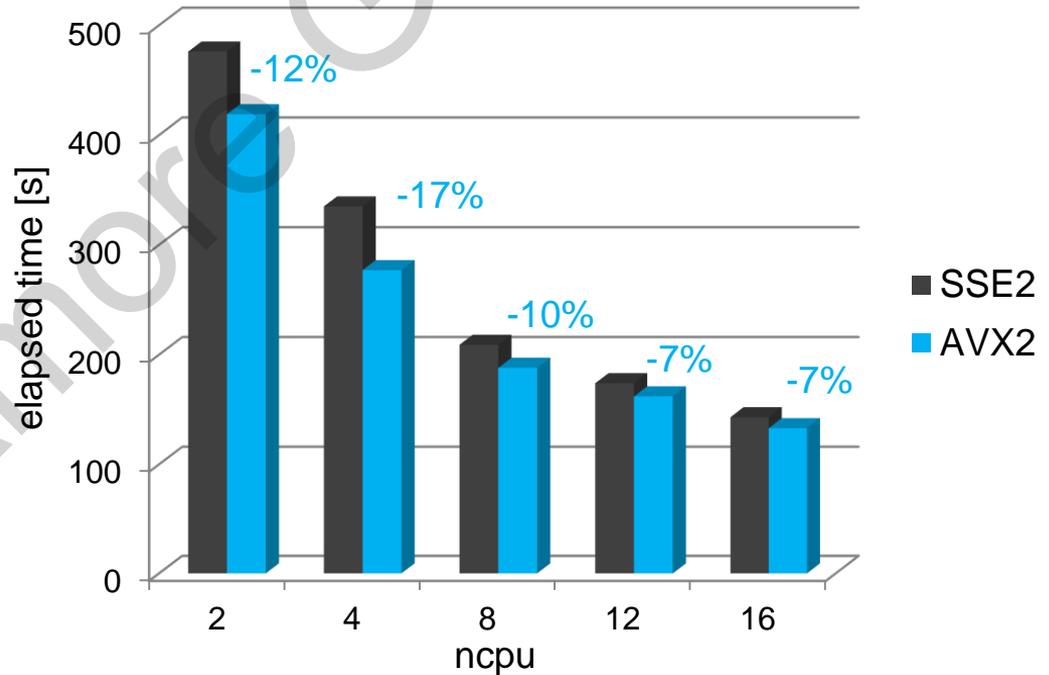


SSE2 vs AVX2: Eigenvalue analysis simple model



- Results are totally identical(!)

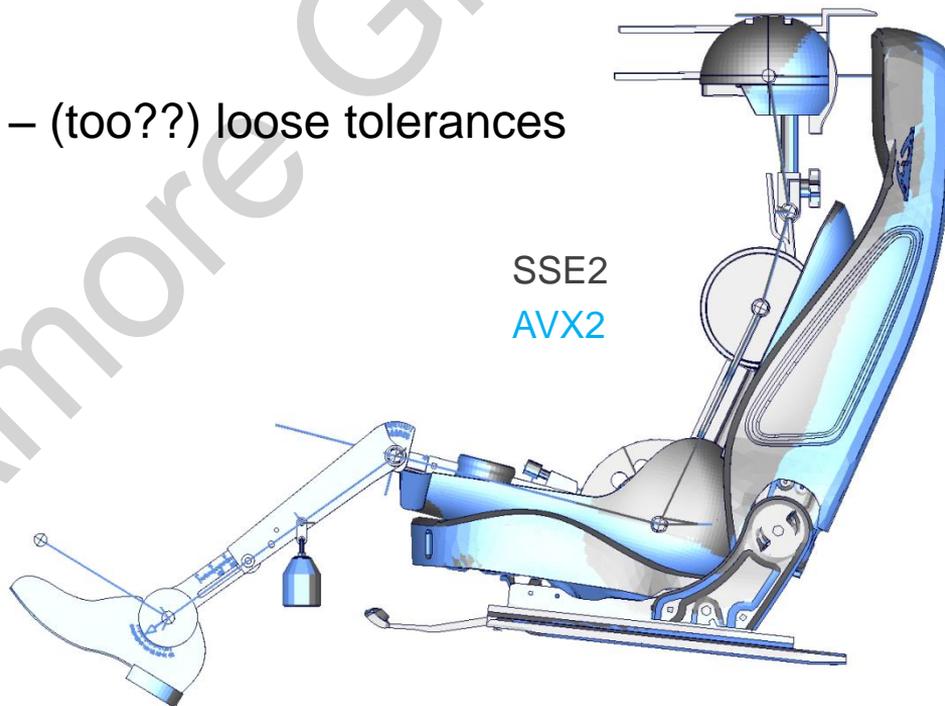
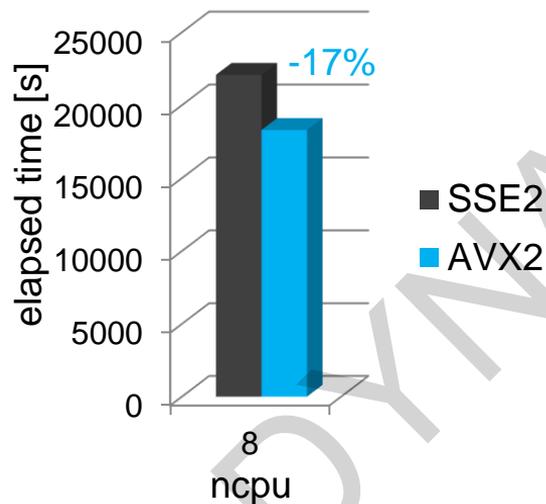
SSE2 vs AVX2: transient analysis simple model



- Results are totally identical(!)

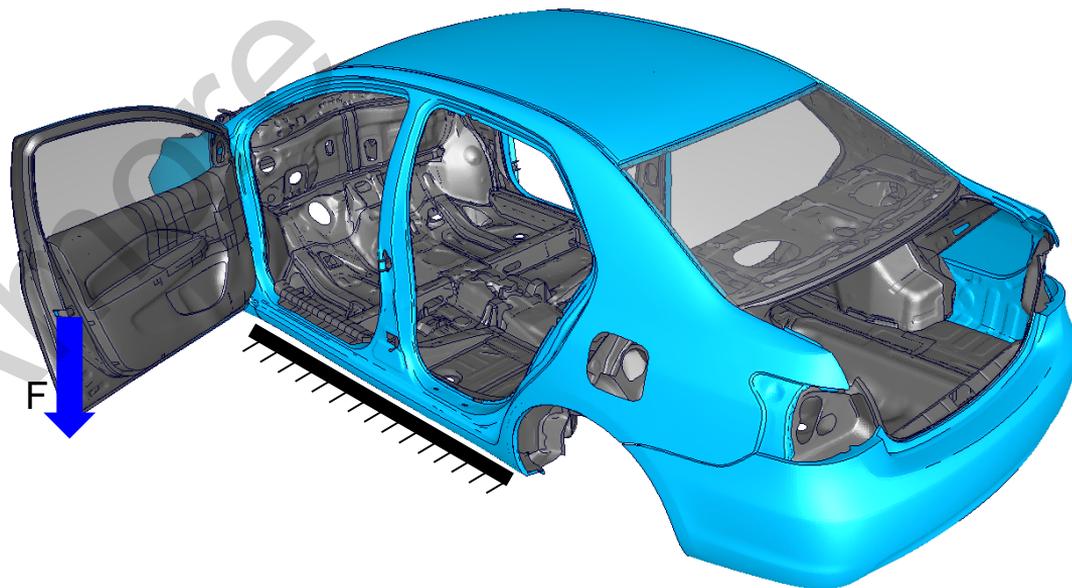
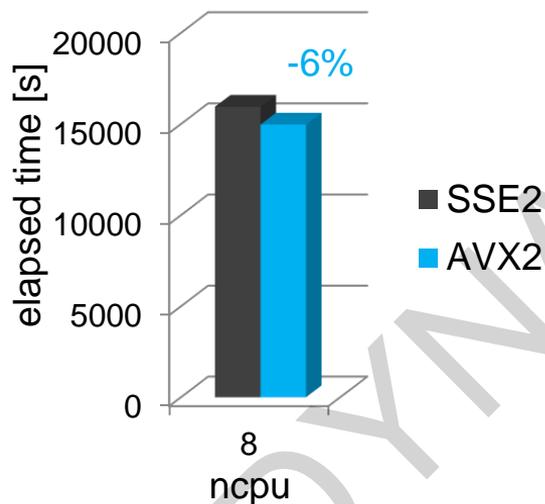
SSE2 vs AVX2: transient analysis HPM settling

- Benchmark on 8 cores
- Results have slight differences – (too??) loose tolerances
- Different steps during solution



SSE2 vs AVX2: door sag

- Benchmark on 8 cores
- Results are totally identical(!)



- Intro
- R9 Solver
- Walkthrough: NCAC Toyota Yaris model conversion to implicit
- LS-DYNA implicit with AVX2
- **Convergence behavior monitoring**
- Summary

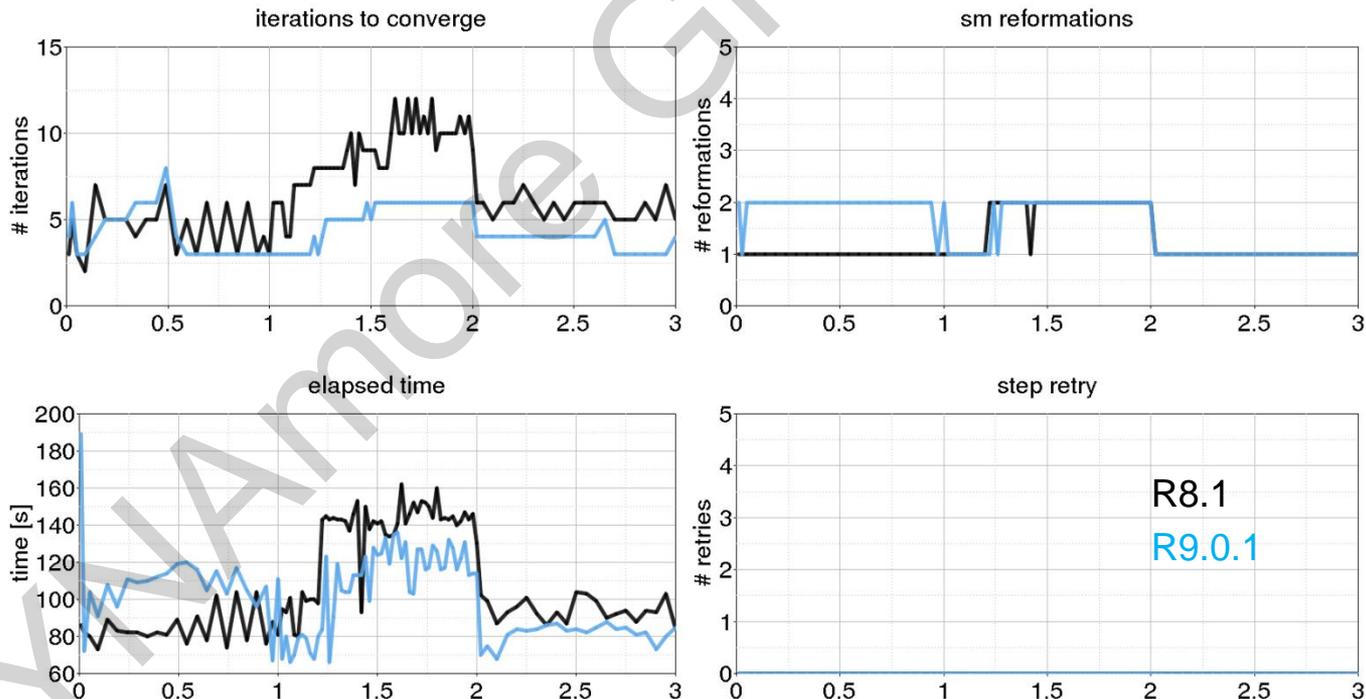
Providing convergence related information

- Included in the DYNA-tools package
- Greps information about convergence behavior from d3hsp
- Prints out table view of information like iterations, retry, ...
- Generates csv-file for post-processing with EXCEL, HG, ...

```
scanning d3hsp
```

STEP	TIME	step size	#iter	#SM Reform	#RETRY	elaTime[s]		
1	1.0000E+00	1.0000	2	0	0	18		
2	2.5849E+00	1.5849	2	0	0	12		
3	5.0968E+00	2.5119	2	0	0	13		
4	9.0779E+00	3.9811	12	1	0	100		
5	1.5387E+01	6.3091	7	0	0	41		
....								
39	1.5516E+03	271.8000	16	3	0	112		
40	1.9825E+03	430.9000	37	5	0	247		
41	2.0000E+03	17.5000	23	2	0	144		
42	2.0928E+03	92.8000	14	1	2	209		
43	2.2400E+03	147.2000	24	2	0	763		
44	2.3482E+03	108.2000	26	2	1	588		
45	2.4278E+03	79.6000	30	2	1	506		
46	2.5540E+03	126.2000	39	3	0	917		
47	2.6802E+03	126.2000	26	2	0	671		
48	2.7233E+03	43.1000	30	2	2	246		
49	2.7550E+03	31.7000	59	5	1	428		
50	2.7709E+03	15.9000	84	8	0	536		
...								
63	3.7730E+03	316.9000	2	0	0	17		
64	4.0000E+03	227.0000	2	0	0	16		
65	4.5024E+03	502.4000	28	5	0	189		
66	5.0000E+03	497.6000	15	4	0	131		
			1579	202	11	14633	TOTAL	
			75.7576	23.9	3.1	0.2	221	AVERAGE

Example: Yaris door sag R8.1 vs R9.0.1



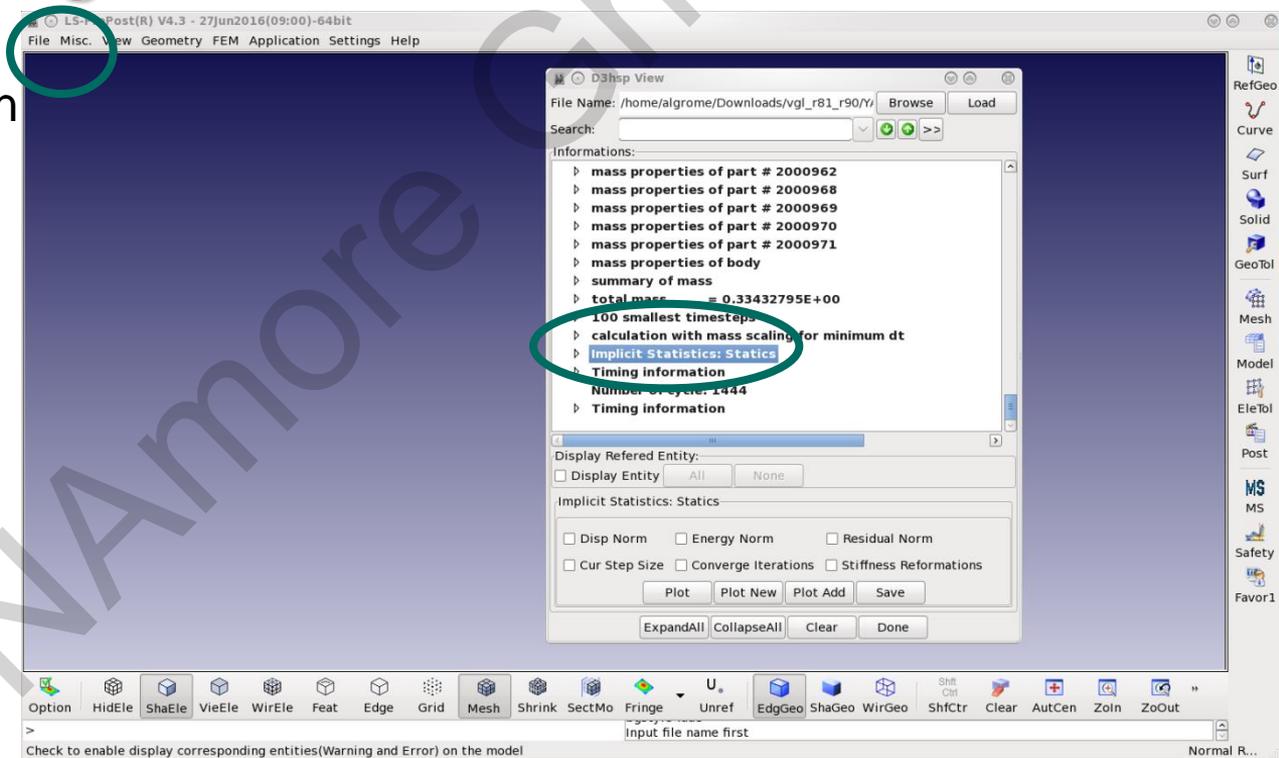
R8.1 94 steps in 3h 7min

R9.0.1 with IACC=1

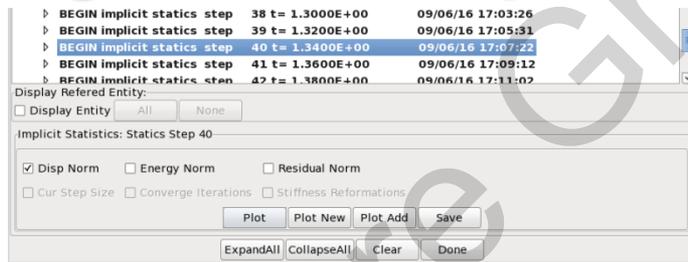
94 steps in 2h 49min

Providing convergence related information

- Organized presentation of d3hsp's content
- For implicit runs:
Display of each step's norms
- MISC – d3hsp view

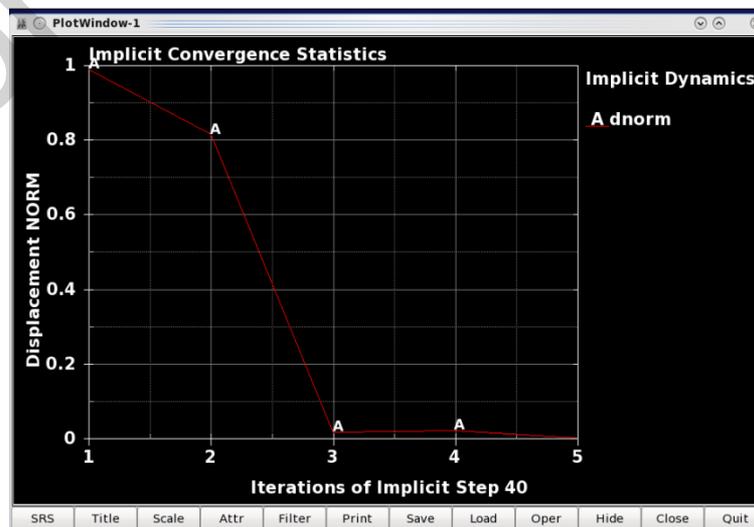
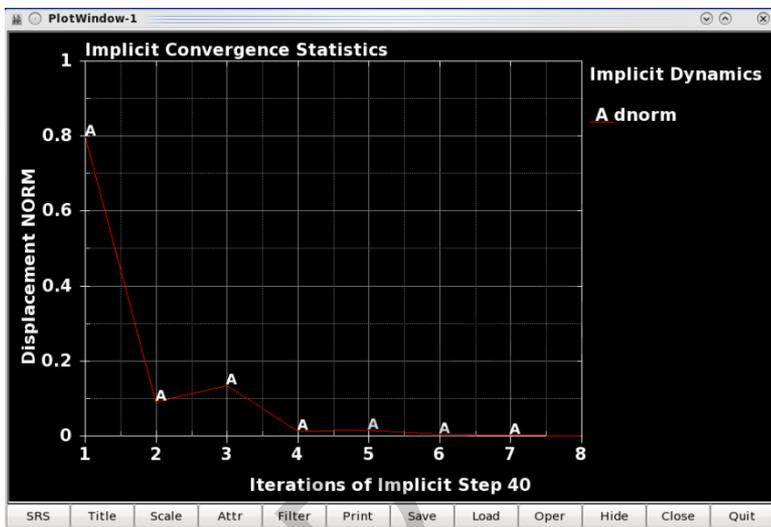


Example: Viewing the development of a disp norm



R8.1

R9.0.1



- Intro
- R9 Solver
- Walkthrough: NCAC Toyota Yaris model conversion to implicit
- LS-DYNA implicit with AVX2
- Convergence behavior monitoring
- **Summary**

When you need information, help, inspiration, ...

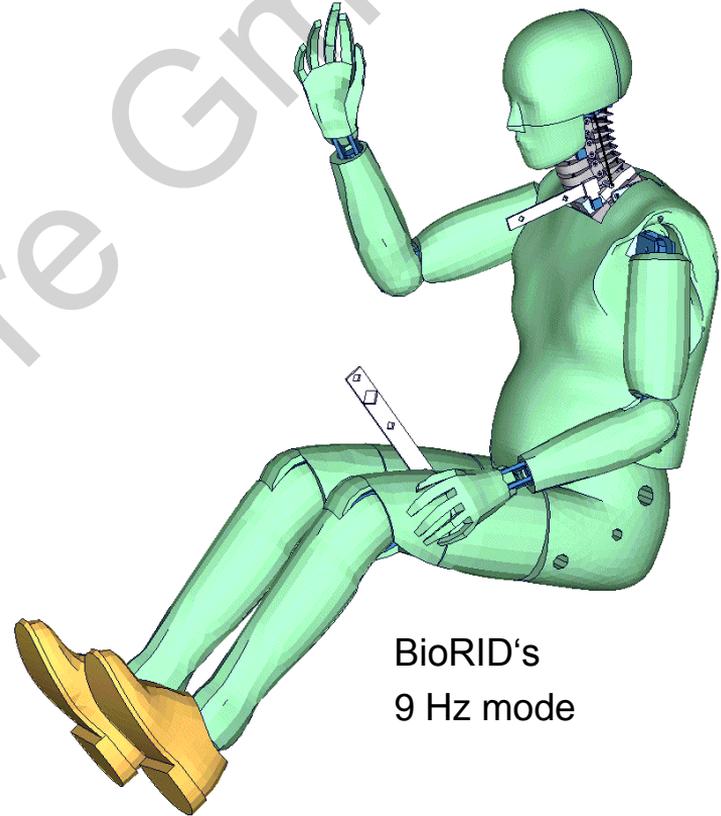
- <https://www.dynamore.se/en/resources/tips-and-tricks>
 - Implicit starter kit including guideline
- Appendix P: LS-DYNA DRAFT Manual
 - A lot of information about LS-DYNA implicit
- www.dynasupport.com
 - Further guidelines, checklists
- www.dynaexamples.com/implicit
 - Application examples (free download)
 - Includes the Yaris models
- support@dynamore.de

NEW

NEW

- LS-DYNA R9 solver is a successive enhancement. For running nonlinear implicit problems the R9 solver should be definitely the user first choice. Within the last years the LS-DYNA solver has grown to a powerful tool and it has reached a competitive grade.
- The successful conversion of the CCSA Yaris model demonstrates the capability of the implicit solver. The total effort of bringing the model to a “implicit ready” grade was manageable.
- On current hardware architectures implicit jobs turn around time can be reduced about 10% by using the avx2 executables of LS-DYNA. Considering robust models there is no effect on the results.
- With LSPPs d3hsp and the DYNATool check-convergence powerful tools can help user to learn more about the convergence behavior. A comparison between different versions of a model can be easily made.

THANK YOU
for listening



BioRID's
9 Hz mode