FINITE ELEMENT ANALYSIS
Predictive Engineering

LS-DYNA® Handbook

Analysis Theory and Techniques for Structural Mechanics

An overview of the core analysis features used by LS-DYNA® to simulate highly nonlinear static (implicit) and dynamic (implicit/explicit) behavior in engineered structures and systems.

ANSYS LST

Acknowledgements

These notes were constructed from numerous sources, but special thanks should be given to the following people:

Technical Support Team at the ANSYS Livermore Software and Technology (LST)
With special mention to:
Satish Pathy, LST
Jim Day, LST
Todd Slavic, LST
Ushnish Basu, LST
Philip Ho, LST

And the invaluable team at:
DYNAmore, Gmbh, Germany
And, in particular, Dr. T Borrvall, DYNAmore Nordic AB, Linköping, Sweden

Trademarks:

Disclaimer:

The material presented in this text is intended for illustrative and educational purposes only. It is not intended to be exhaustive or to apply to any particular engineering design or problem. Predictive Engineering nor the organizations mentioned above, and their employees assumes no liability or responsibility whatsoever to any person or company for any direct or indirect damages resulting from the use of any information contained herein.

Special Acknowledgement:

These class notes stand on the shoulders of the outstanding team at the ANSYS LST and their colleagues who have provided suggestions and support through years of collaboration. I am sincerely indebted to them for their help and of course, any errors or omissions are due to me.

COURSE OUTLINE

1. INTRODUCTION

1.1 What The Student Can Expect

This class is directed toward the engineering professional simulating highly nonlinear, static and dynamic problems involving large deformations and contact between multiple bodies. What this means in layman terms, is that we will provide a realistic foundation toward the practical usage of LS-DYNA.

1.2 What we cover

- Nonlinear Explicit and Implicit FEA Mechanics
- The technology of creating accurate nonlinear, static and transient FEA models
- How to do your own research to create more advanced simulations
- Our condensed experience and that of our colleague's to help you *not* repeat our mistakes

1.3 How we do it

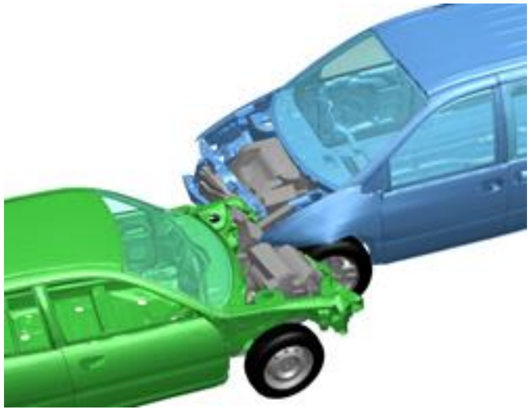
- The class covers the basics in a hands-on manner as taught by engineers that has had to live by what they have validated.
- Each day (four hour session) will have three to four Workshops. Each Workshop is part theory, part demonstration and part hands-on practice. Videos are provided for most Workshops thereby allowing the student to relax and follow along at their own pace. These videos cover the basics and also provide insight into the many tips and tricks that make LS-DYNA the world's most complete and accurate simulation code.
- A breaks is provided mid-way where students can pause, stretch and perhaps ask the instructor more detailed questions that might not be appropriate to involve the full class.
- Students are encouraged to turn off their email, text messaging and other forms of digital/social media during class time.

1.4 How To Be Successful with as a LS-DYNA Simulation Engineer (top-of-the-pack)

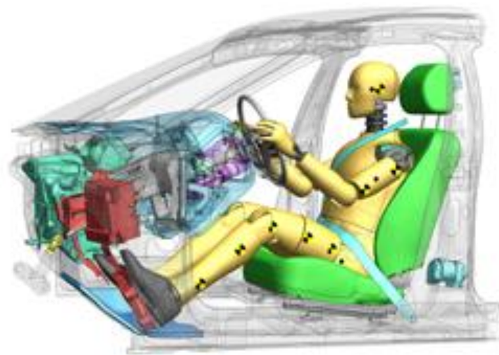
- You are already way ahead of the pack by simply attending this class. You have started on the journey of how to be more successful with LS-DYNA. It is this simple. To be successful, as far as we know, it requires:
    - Reading (very traditional but with LS-DYNA it is necessary to read the manual (RTM), read again and most likely for us normal people, read again;
    - Attend courses since it breaks up the learning process and opens doors to new avenues of learning and knowledge;
    - Be open to new ideas and then once again RTM and read some more;
    - After all this reading, one has to do some organic learning. That means building small models to explore options and mechanics and to suffer a bit prior to calling your colleagues for help;
    - Lastly, don't be hesitant to reach out for help once you have read, built small models to explore options, read some more until finally you are posed to ask questions that will lead you quickly toward the right solution for your project. Without this background, your questions will often be wild, untamed and often just not very constructive to you and your colleague.

1.5 General Applications

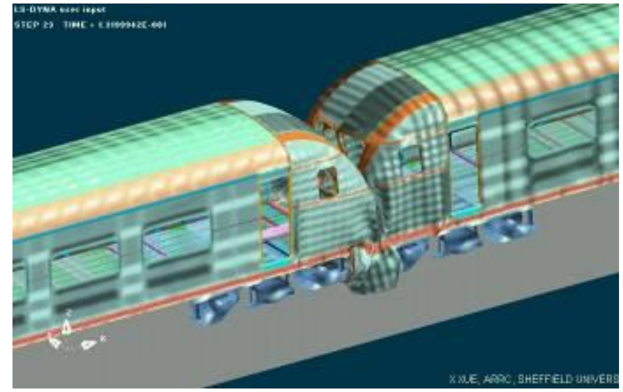
Crashworthiness



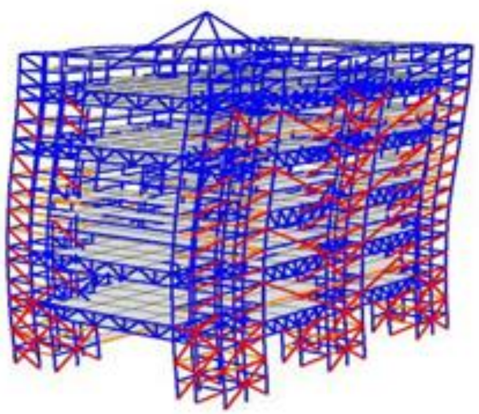
Driver Impact



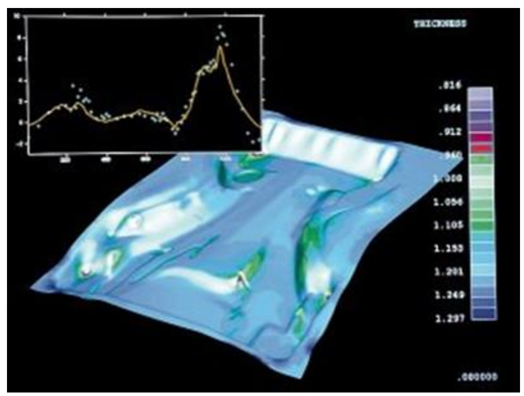
Train Collisions



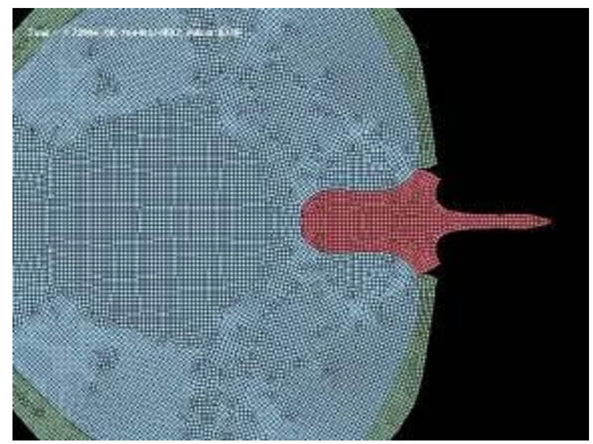
Earthquake Engineering



Metal Forming

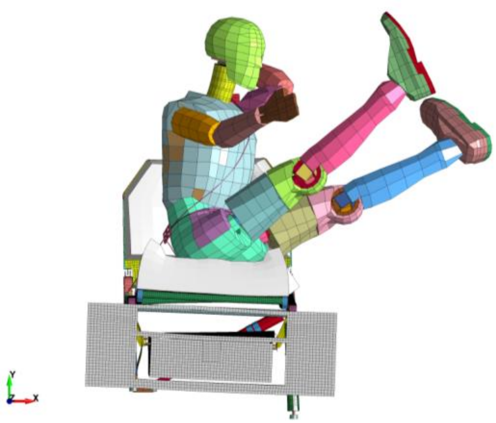


Military

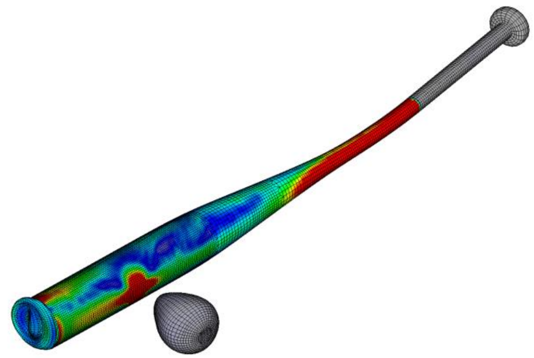


1.6 Specific Applications (Courtesy of Predictive Engineering)

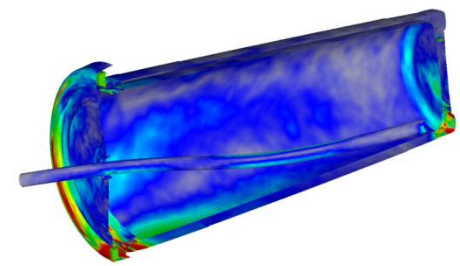
CS-FAR 25 16g Sled Analysis



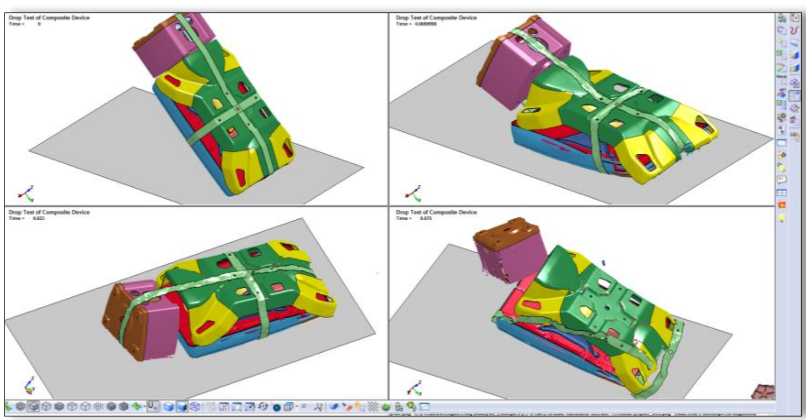
Sporting Goods Equipment



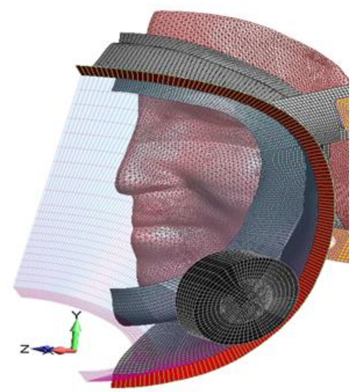
Drop Test Consumer Products



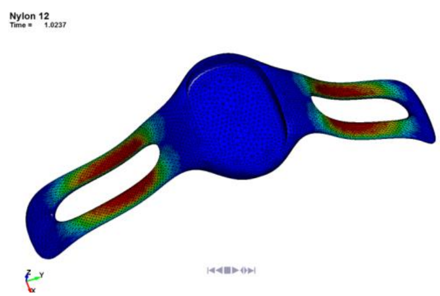
Drop Test of Composites / Electronics



Human Biometrics

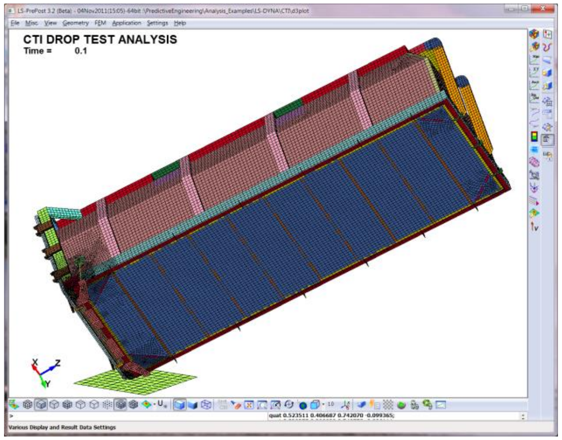


Large Deformation of Plastics

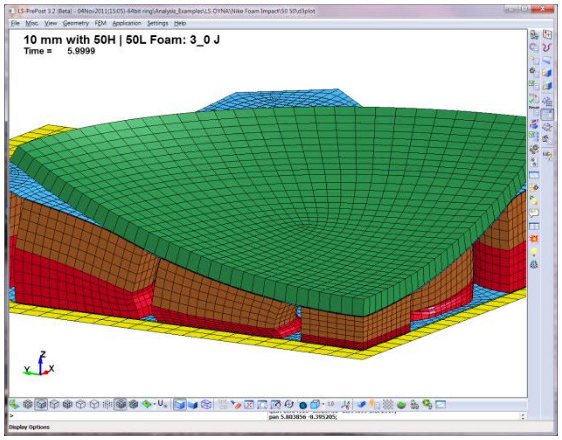


Crash Analysis of Cargo Net

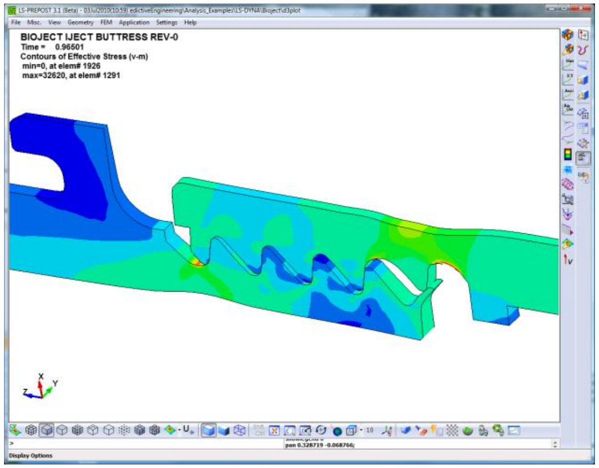
Drop Test of Nuclear Waste Container



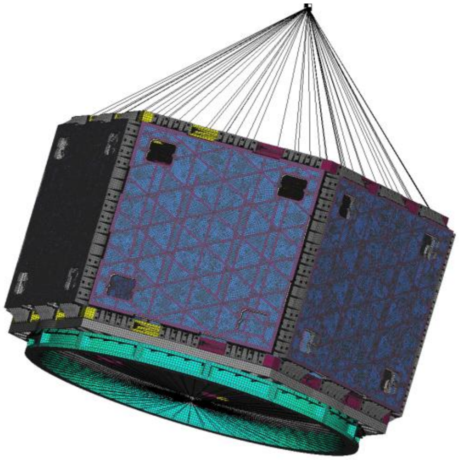
Impact Analysis of Foams



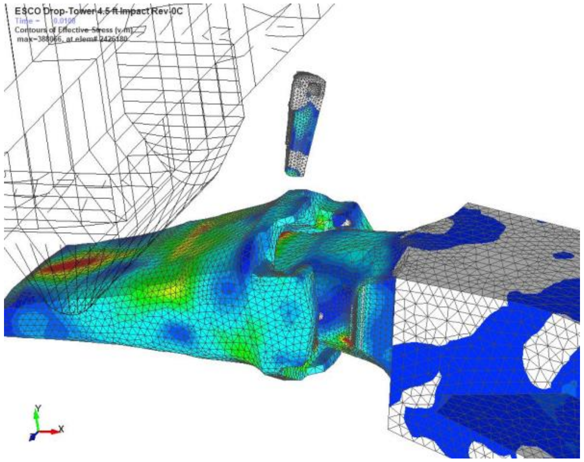
Plastic Thread Design



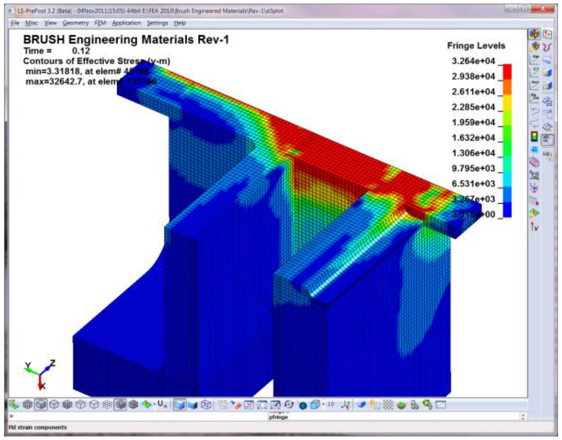
PSD / Modal Analysis



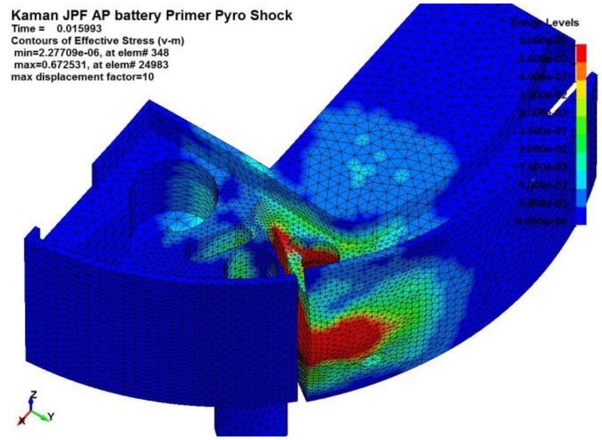
Digger Tooth Failure



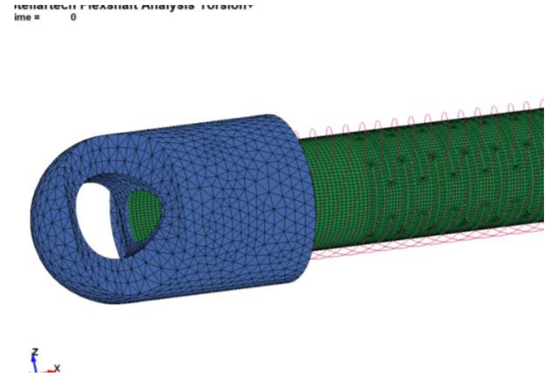
Electron Beam Welding



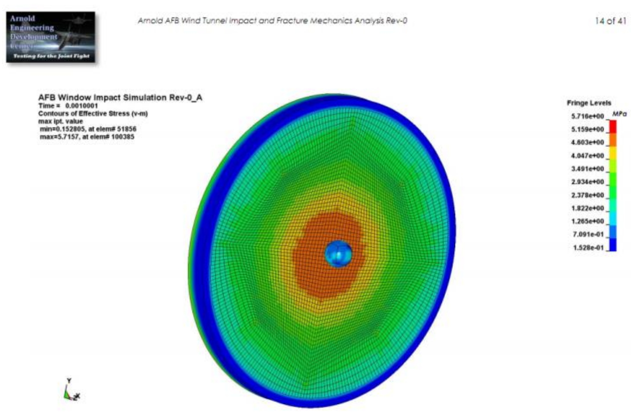
Pyro-Shock Analysis



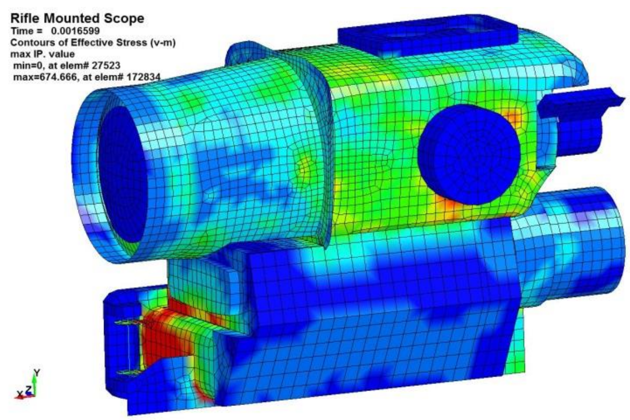
Medical Equipment



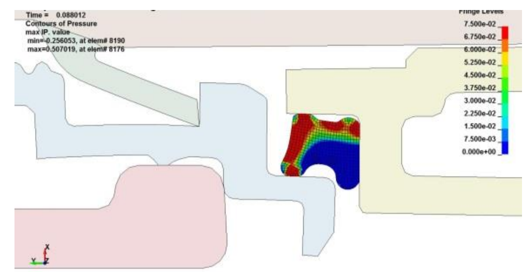
Fracture Mechanics of Glass



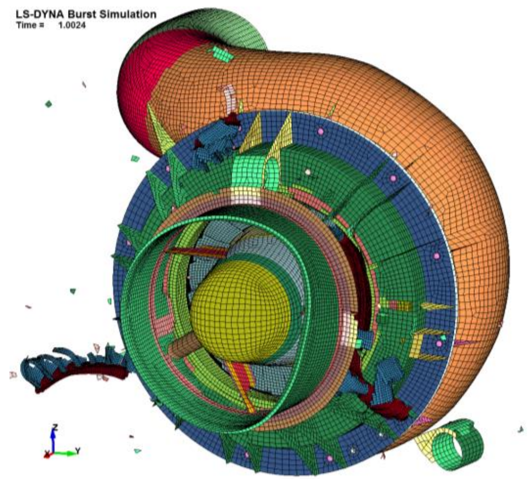
Ballistic Shock Loading of Optical Equipment



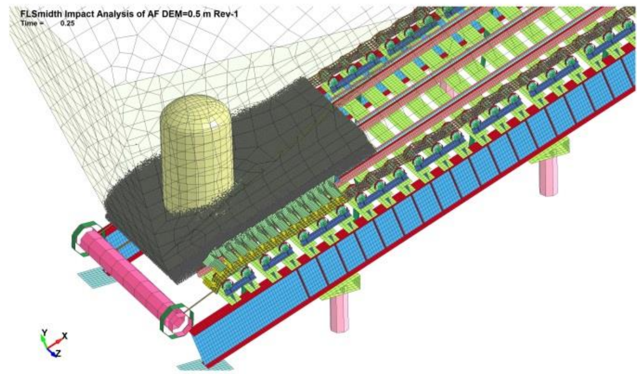
Hyperelastic Medical Seal Analysis



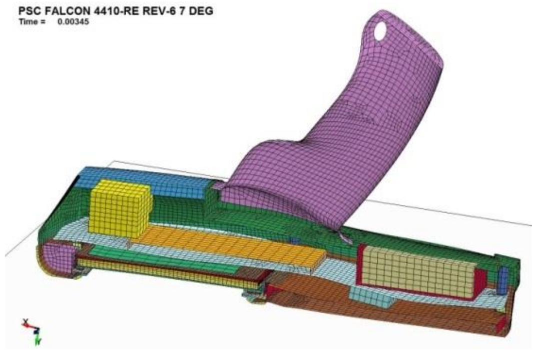
Blade-Out Analysis



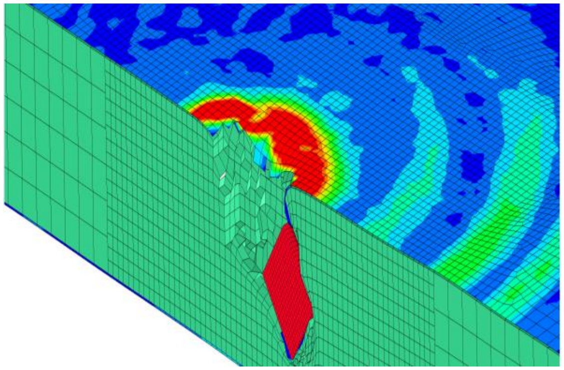
Discrete Element Method for the Mining Industry



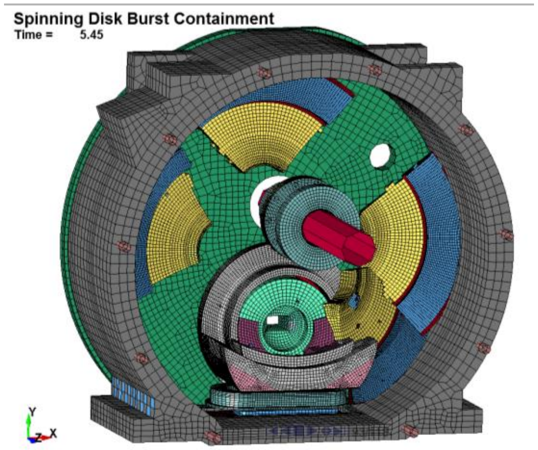
Drop-Test of Handheld Electronics



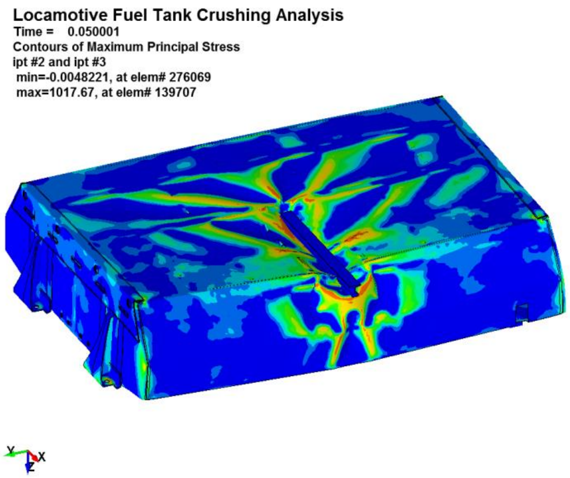
Ballistic Penetration



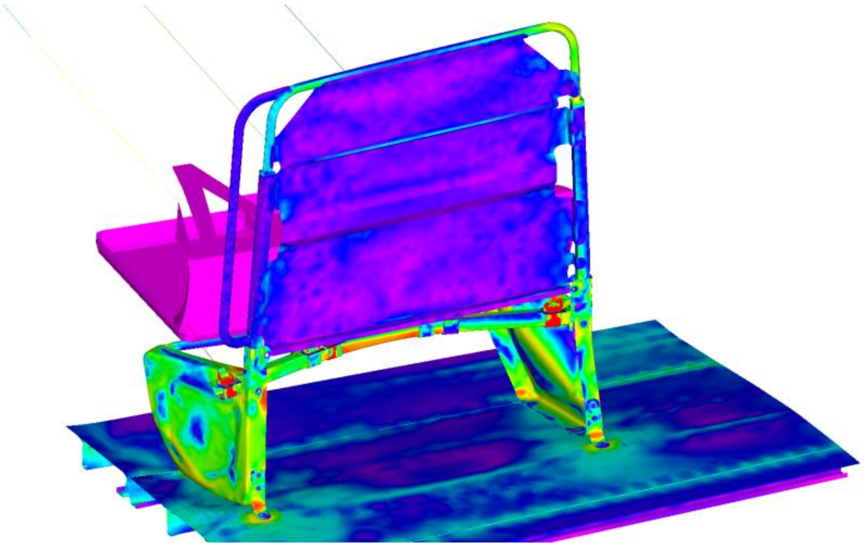
High-Speed Spinning Disk Containment



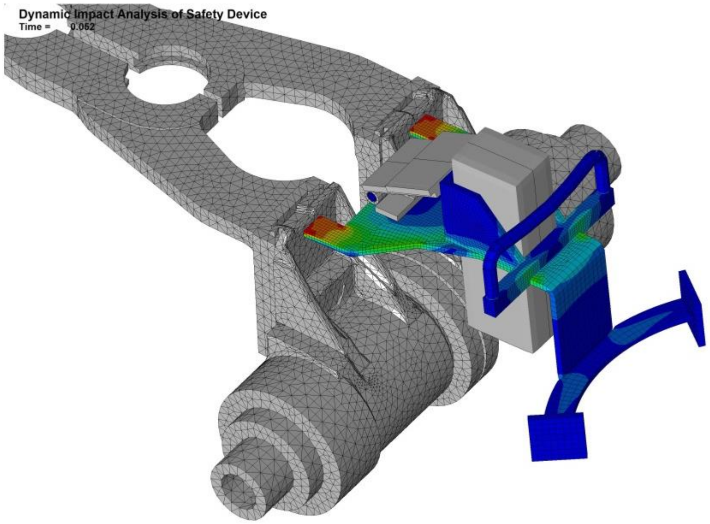
Locomotive Fuel Tank



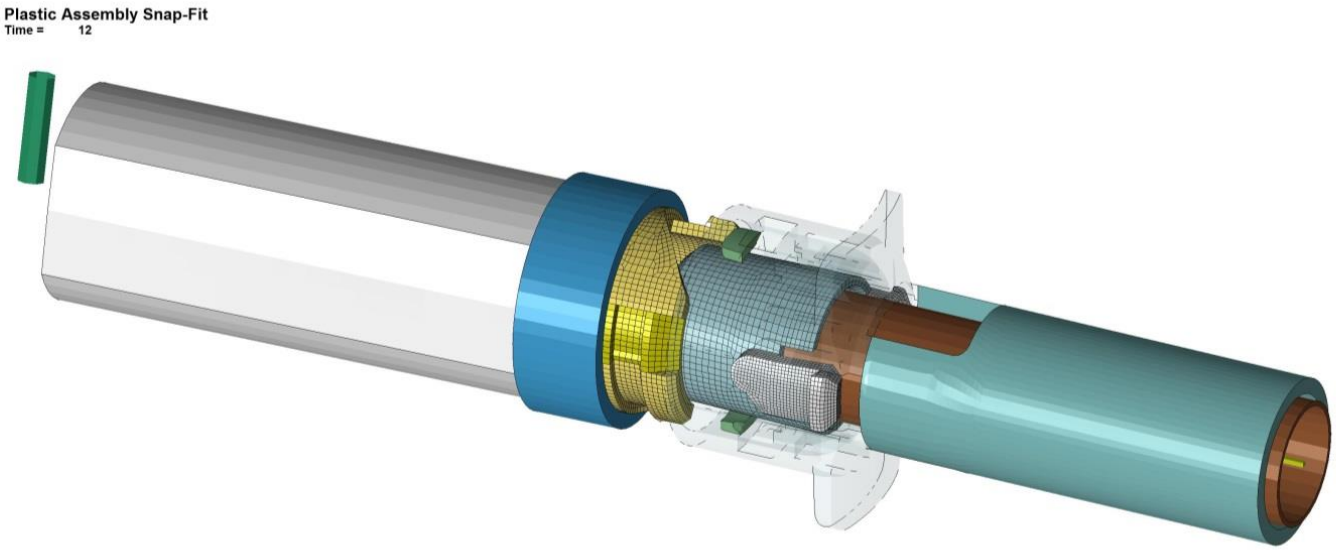
FMVSS Virtual Testing of Bus Seats



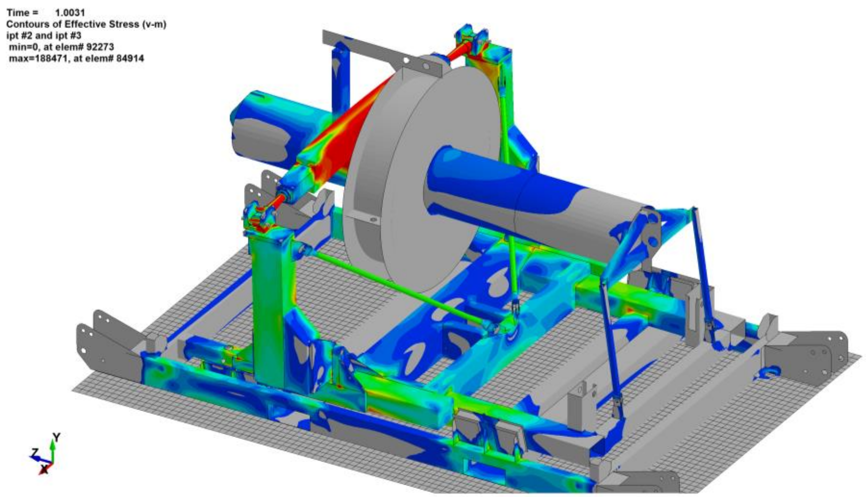
Impact Analysis of Safety Block Device



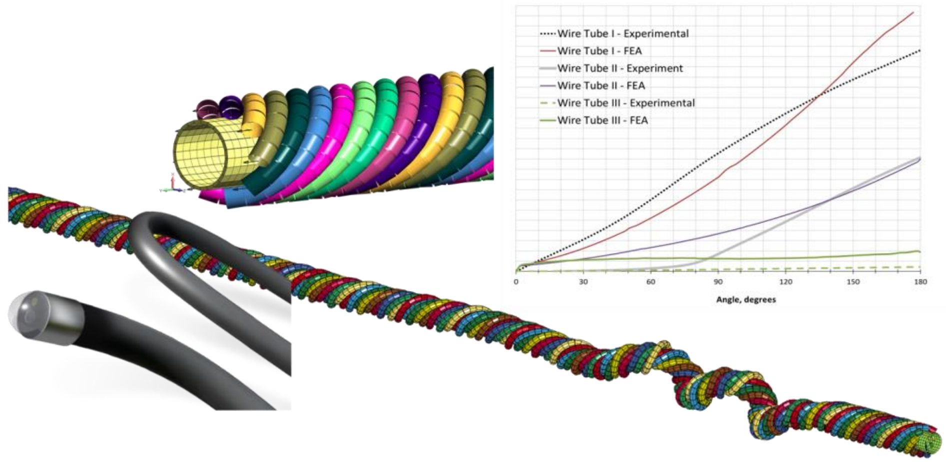
Snap-Fit Analysis – All Plastic Medical Device



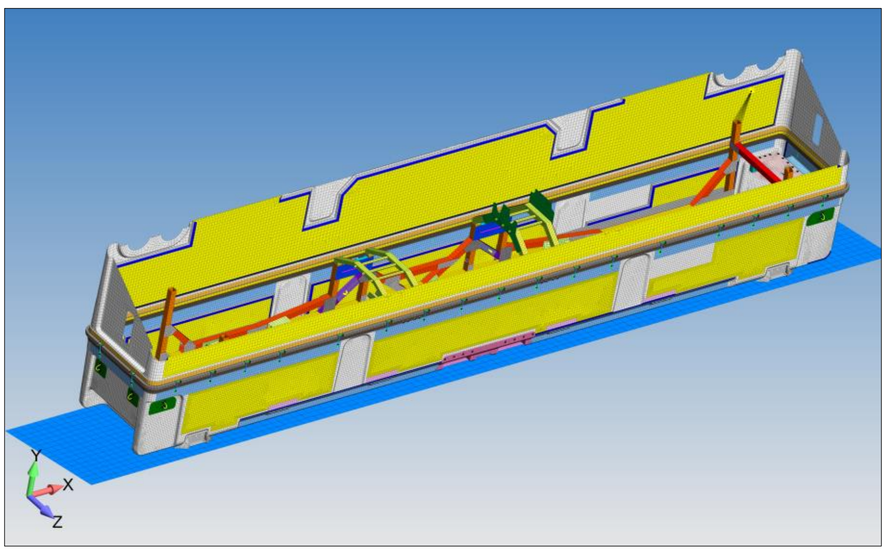
9g Crash Analysis of Jet Engine Stand

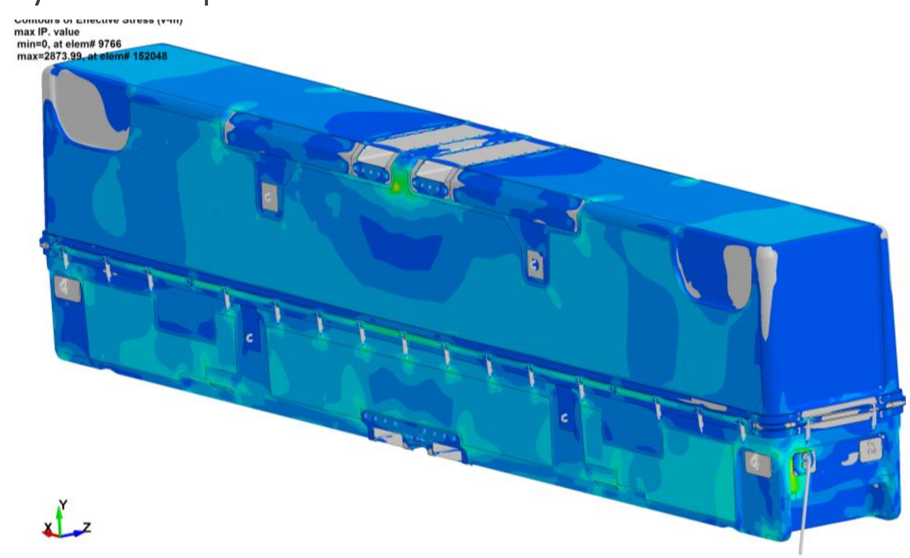


Torque Analysis of Endoscopic Medical Device

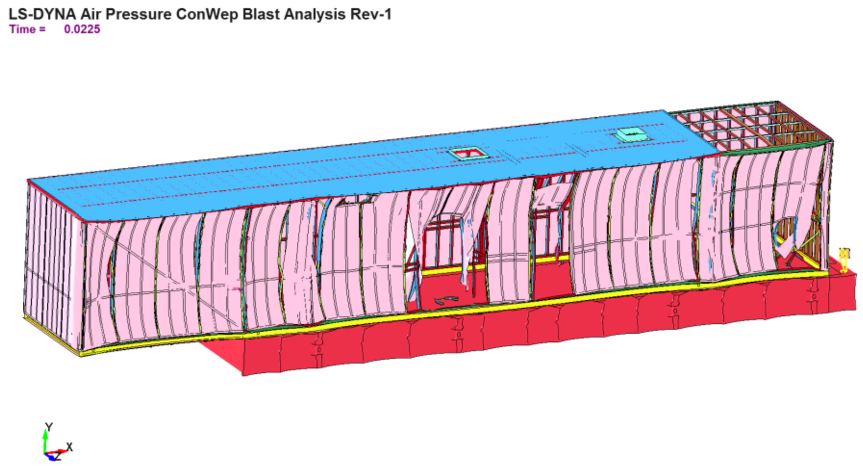


Drop, Rail Impact and PSD Analysis of Composite Container

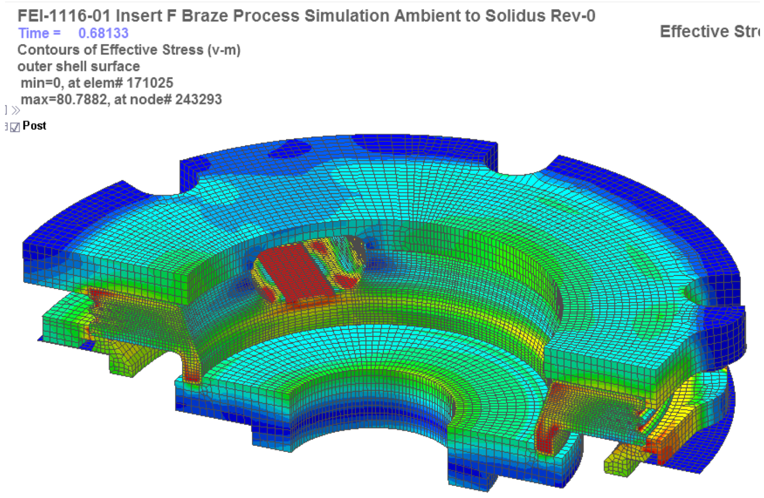




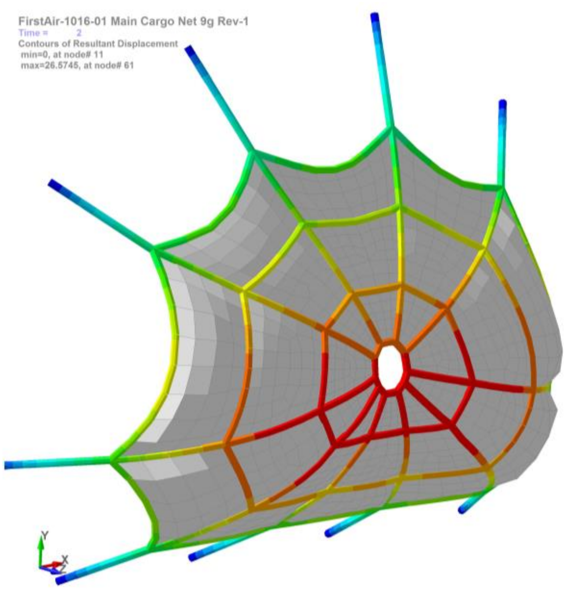
ConWep Air Pressure Blast Analysis of Generator Housing



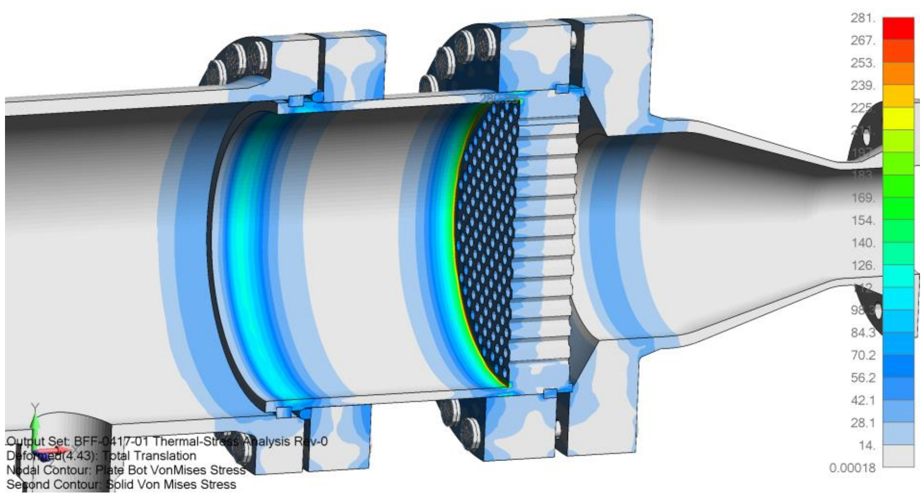
Alumina-Stainless Steel Braze Process Simulation



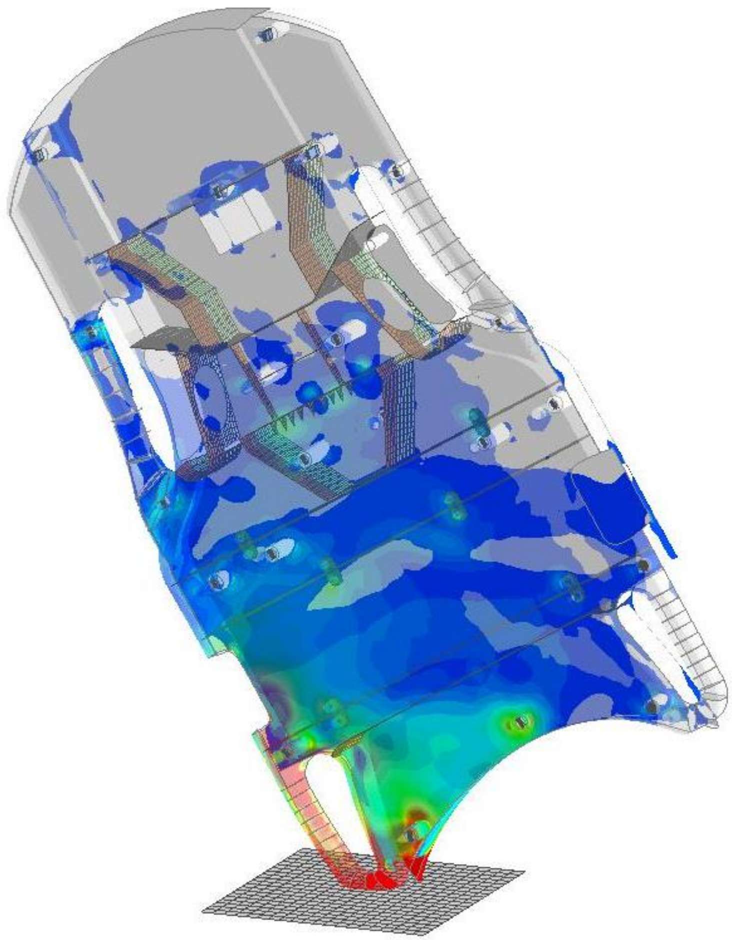
Air Freighter 9g Cargo Net Analysis



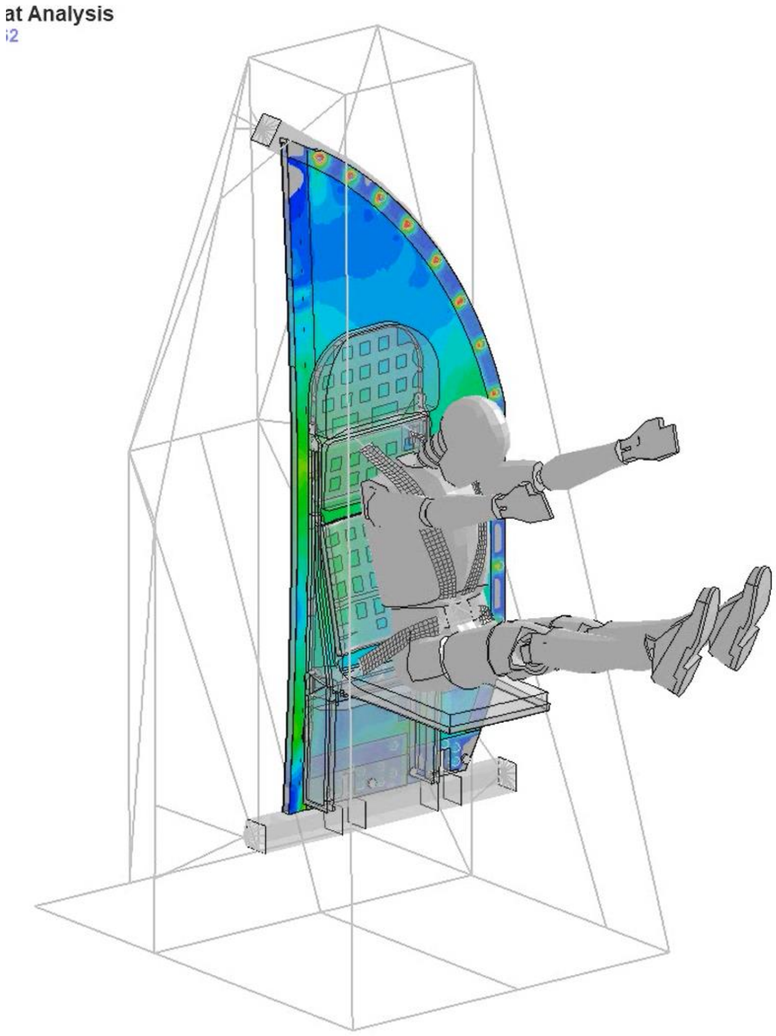
Thermal-Stress Fatigue Analysis of ASME Evaporator Vessel



Drop-Test of First-Responder Medical Equipment



Composite Lavatory Wall with Attendant Seat 16g Sled Test



2. WHAT IS LS-DYNA?

LS-DYNA is a finite element analysis (FEA) solver. It is the motor that generates results based on what the user provides as input. In other words, it is not a program that generates a mesh or that can create stress contour plots but the world's most sophisticated and complex FEA solver. The workflow is to provide LS-DYNA an ascii text based deck (with a suffix as *.k or *.dyn) with nodes, elements, loads, constraints, material laws, etc. and then LS-DYNA solves this input and generates another file (*.f06) with the requested results.

One can read an LS-DYNA analysis deck with any text editor. A lot of useful information about the LS-DYNA code and its structure can be found in the LS-DYNA Keyword Manual Vol. 1. For every new user, it is time well spent to read the Introduction and Getting Started sections. It provides some very nice background on the LS-DYNA code.

2.1 How We Visualize the LS-DYNA Analysis Process

No matter where you build your deck, LSTC's LS-PrePost (henceforth LSPP) is often an invaluable tool along the way to a validated FEA model. This course is focused on setting up a simulation model that is solvable by LS-DYNA, that will generate results that are verifiable and that will lead to a validated solution. We do not focus on how the nodes and elements are generated within a FEA tool but we do focus on their quality.

Files: Input and Output

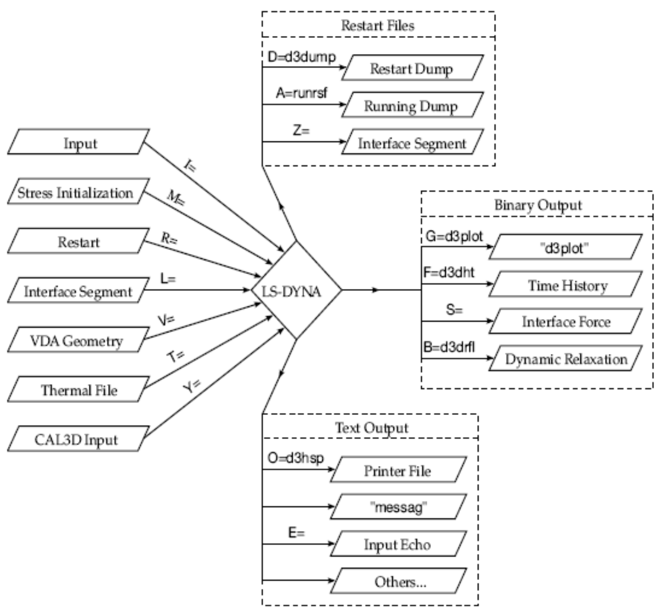


Figure 2-2. Files Input and Output.

LSPP

Pre-Processor
(Nodes, Elements, Etc.)

Commercial Software
{Ansys WP, Hypermesh, FEMAP, ANSA, etc.}

Pre-Processing to Create Keyword Deck

LS-DYNA

Post-Processing FEA Results

Commercial Software
{Ansys WB, Hypermesh, FEMAP, Oasys, etc.}

LSPP

Proprietary Software

3. IMPLICIT VERSUS EXPLICIT ANALYSIS

LS-DYNA is a non-linear transient dynamic finite element code with both explicit and implicit solvers.

3.1 What We Are Solving

Explicit only works when there is acceleration of ***mass*** (dynamic) whereas an implicit approach can solve the dynamic and the static problem (***no mass***). For dynamic problems, we are solving the following equation:

$$ma^n + cv^n + kd^n = f^n$$

where n=time step. A common terminology is to call the kd^n part the internal force in the structure. The basic problem is to determine the displacement at some future time or d^{n+1}, at time t^{n+1}. However, this is where it gets interesting, explicit is based on acceleration whereas implicit is displacement.

In conceptual terms, the difference between Explicit and Implicit dynamic solutions can be written as:

$$Explicit\!:\;\; a^{n+1} = f(d^n, v^n, a^n,\, d^{n-1},\, v^{n-1}, \dots)$$

All these terms are known at time state “n” and thus can be solved directly. This means that the solution marches forward regardless of the element deformation or contact behavior or whatever nonlinearities (*importantly, no residual – see below*). However, it doesn’t mean that it might not blow up if elements get too distorted and it doesn’t mean that contact will always contact.

For *Implicit*, the solution depends on nodal velocities and accelerations at state n+1, quantities which are unknown:

$$Implicit\!:\;\; d^{n+1} = f(v^{n+1}, a^{n+1},\, d^n,\, v^n, \dots.)$$

Given these unknowns, an iterative solution is required to calculate the displacement at this future time. If the nonlinearity is mild, the implicit approach allows one to use a comparably large time step as that compared to the explicit analysis and the run time can be advantageous. This is because an implicit solution must perform an iterative solution to reduce the residual within each time step:

$$ma^n + cv^n + kd^n - \; f^n \; = Residual$$

If the nonlinearity is severe, the implicit solution may require a very small time step and a large number of iterations within each step to reduce the residual to something reasonable (i.e., a converged solution). In contrast, an explicit solution has no residual and just solves but requires a small time step (more will be said about this later). Thus, when faced with large nonlinearities, an explicit solution is more robust whereas, if the nonlinearity is mild, an implicit solution is often more practical to get the job done quickly.

3.2 Explicit (dynamic) – One Must have "Mass" to make it go

Internal and external forces are summed at each node point, and a nodal acceleration is computed by dividing by nodal mass. The solution is advanced by integrating this acceleration in time. The maximum time step size is limited by the Courant-Friedrichs-Lewy (CFL) criterion (to be discussed). For now let's say that the solution marches forward in time using a fixed time step that is calculated based on the element size and the speed of sound in the material (i.e., CFL). Much more will be said about element size and the speed of sound in materials since execution speed for an explicit analysis is often of great importance given that careful meshing can mean the difference between a run time of days or hours. Just to keep this theme in the forefront of our discussion: an explicit analysis is all about mass since everything has a time step (e.g., contact, 1D spring elements, CNRB's, etc.).

3.3 Implicit (dynamic or Static)

A global stiffness matrix is computed, decomposed and applied to the nodal out-of-balance force to obtain a displacement increment. Equilibrium iterations are then required to arrive at an acceptable "force balance". The advantage of this approach is that time step size may be selected by the user. The disadvantage is the large numerical effort required to form, store, and factorize the stiffness matrix. Implicit simulations therefore typically involve a relatively small number of expensive time steps. The key point of this discussion is that the stiffness matrix (i.e., internal forces) has to be decomposed or inverted each time step whereas in the explicit method, it is a running analysis where the stiffness terms are re-computed each time step but no inversion is required. Since this numerical technique is independent of a time step approach, element size is not of direct concern only the size of the model (nodes/elements) directly affects the run time.

3.3.1 Pros and Cons of Explicit v Implicit

Explicit		Implicit	
Pros	**Cons**	**Pros**	**Cons**
It solves directly since the solution marches forward.	Solution time step controlled by wave speed and element mechanics.	Large time steps can be used since the solution is iterative.	Requires iterative process to converge.
Dynamic solution	Long run times for simulations that require long event times.	Static and Dynamic solutions	Requires iterative process to converge which can lead to long run times.
Extreme nonlinearity is easily handled.	Of course, solution can blow up due to twisted elements or contact problems.	Linear and Nonlinear solutions	Implicit struggles with extreme nonlinearity
Pretty much all physics can be solved.	W.R.T. multi-physics, no real cons since you are solving the impossible.	Provides the missing link in LS-DYNA to solve standard linear static and dynamic problems.	Focused on solid mechanics so don’t expect to see meshfree methods anytime soon.

4. LS-DYNA GETTING STARTED WITH THE FUNDAMENTALS

4.1 LS-DYNA KEYWORD MANUAL

LS-DYNA has perhaps one of the most basic learning methods. It is organic. One simply has to dig in and learn the basics and there is no substitute for doing it yourself. The Keyword Manual also provides recommended usage guidelines and examples on how to use the commands. It is your first and best resource. Given the frequency of program updates, the Keyword manuals are likewise being constantly updated. Fairly recent versions of the four Keyword manuals can be found in the *Class Reference Notes / Keyword Manuals.*

LS-DYNA®
KEYWORD USER'S MANUAL

VOLUME I

03/03/17 (r:8240)
LS-DYNA Dev

LIVERMORE SOFTWARE TECHNOLOGY CORPORATION (LSTC)

Analyst's Note: Please keep in mind that LS-DYNA is an analysis engine that runs off of an ascii deck (a text file) and that oftentimes the fastest path to an optimum solution is to edit the deck. It took me years to embrace the "deck" and I'm better for it.

4.2 KEYWORD SYNTAX

- Commands are strings of words separated by an underscore, e.g., *BOUNDARY_PRESCRIBED_MOTION_RIGID.
- Text can be uppercase or lowercase
- Commands are arranged alphabetically in User's Manual
- Order of commands in input deck is ***mostly*** unimportant (except *KEYWORD, *DEFINE_TABLE (but then one can use *DEFINE_TABLE_2D if this is a problem), *INCLUDE_TRANSFORM, ?)
- Keyword command must be left justified, starting with an asterisk
- A "**$**" in the first column indicates a comment
- If one would like to screen print out comments, use *COMMENT
- Input values (card data) can be *anywhere within* **fixed fields or/and comma-delimited** (Although one will notice that I like to right-justify values within fixed fields but it is not necessary.)
- A blank parameter indicates that the default value of the parameter will be used (or taken from *CONTROL_*option*)
- Please keep in mind that *every* Keyword starts with "*" and that each line below the Keyword is a "card" per the LST-ANSYS Keyword Manual.

Required Commands:

*KEYWORD
*CONTROL_TERMINATION
*NODE
*ELEMENT
*SECTION
*MAT
*PART
*DATABASE_BINARY_D3PLOT
*END

*Analyst's Note: Want more Keyword information – read Appendix V: How to Read Card Summaries. This Appendix explains the philosphy behind the *KEYWORD structure and its syntex. It should be required reading for any 'DYNA addict.*

4.3 Units

Many a fine analysis model has been brought down by bad units. Although one may wonder why in this modern age one still has to twiddle with units and not have it addressed by the interface is philosophical-like engineering debate between the ability to hand-edit the "deck" or be hand-cuffed to a GUI (pronounced "gooey") interface. Moving past this discussion, to use LS-DYNA effectively, one should have a rock-solid and un-shakable conviction in your chosen system of units.

Since the majority of LS-DYNA work is dynamic, the analyst will often be looking at the energies of the system or velocities, in addition to displacements and stresses. Hence, a consistent set of units that are easy to follow can provide significant relief in the debugging of an errant analysis. A general guide to units can be viewed within the Class Reference Notes / Units (see Consistent units — LS-DYNA Support.pdf). Saying all that, here are the five unit systems that I have standardized on for analysis work. It doesn't mean they are the best but at least they are generally accepted.

Consistent Unit Sets for LS-DYNA Analysis

Mass	Length	Time	Force	Stress	Energy	Density Steel	Young's	Gravity
kg	m	s	N	Pa	J	7,800	2.07e+9	9.806
g	mm	ms	N	MPa	N-mm	7.83e-03	2.07e+05	9.806e-03
kg	mm	ms	kN	GPa	kN-mm	7.83e-06	2.07e+02	9.806e-03
Ton (1,000 kg)	mm	s	N	MPa	N-mm	7.83e-09	2.07e+05	9.806e+03
lbf-s^2/in (slinch)	in	s	lbf	psi	lbf-in	7.33e-04	3.00e+07	386

4.4 Reference Materials and Program Download

The first site to visit: **www.lsdynasupport.com**
Another great site: **www.dynasupport.com**
LS-DYNA Examples: **www.DYNAExamples.com**
LS-DYNA Conference Papers: **www.dynalook.com**
Newsletter: **www.FEAInformation.com**
Newsletter and Seminars: **www.DYNAmore.com**
Yahoo Discussion Group: **LS-DYNA@yahoogroups.com**
Aerospace Working Group: **awg.lstc.com**
Varmit Al's Material Database (google'it)
Ed Wilson's Blog: http://www.edwilson.org/History/1Library.htm

LST-ANSYS Program Download Site

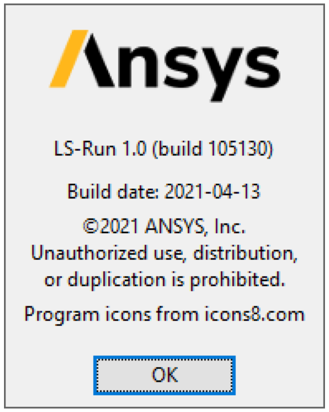
https://ftp.lstc.com/user Username: user Password: computer
SMP Version: ls-dyna
MPP Version: mpp-dyna
SMP/Windows: pc-dyna

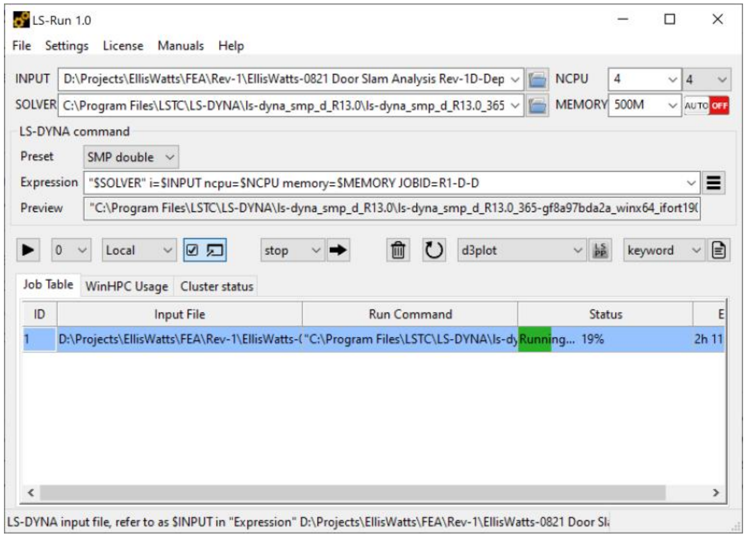
4.5 Submitting LS-DYNA Analysis Jobs with LS-Run

LS-Run is a LS-DYNA job tool that allows you to run jobs using multiple solvers and to queue jobs up for multiple runs. It can be downloaded the LST (see above) site.

Analysts' Notes: MPP (Massively Parallel-Processing or SMP (Symmetric Multi-Processing) depends on the number of CPU-Cores. SMP is faster using eight or fewer cores while MPP's sweet spot is eight and above CPU-Cores. In general, we recommend using the Double-Precision version of the code for initial analysis work although it can be anywhere from 10 to 20% slower; however, double-precision is required for the implicit solver.

See Class Reference Notes / MPP versus SMP / for additional discussion and scaling information.





Attention Windows Users: *LS-DYNA does not handle spaces in file folder names or regular keyword deck names. A simple workaround is to enable 8dot3name on your systems harddisks. This is done by opening up a CMD window (run as Administrator) and typing fsutil 8dot3name set 0 to enable the creation of 8dot3name'ing convention on all drives. With this setting, long file folder names and file names are truncated to 8 characters with a 3 character extension. Please note that after getting Windows setup for 8dot3name, one may have to copy file folders onto another drive and then copy them back to allow Windows to assign the 8dot3name "mask".*

4.5.1 INTERNAL LST FAQ - HTTPS://FTP.LSTC.COM/ANONYMOUS/OUTGOING/SUPPORT/FAQ/

This is the most outstanding information and is a recommended "desktop-reference" for those inquisitive simulation engineers. Please note that this link was valid as of 02/10/2021 and may be removed at the discretion of ANSYS LST – no guarantee is provided by Predictive Engineering, Inc.

Index of /anonymous/outgoing/support/FAQ

Name	Last modified	Size
Parent Directory		-
2d_general_condensed	2017-02-17 14:11	3.4K
ASCII_output_for_MPP_via_binout	2019-06-07 12:18	12K
Instructions_encryption	2018-10-24 12:08	8.8K
LSPP_curve_template	2014-05-22 11:16	1.6K
LSTC_LicenseManager-InstallationGuide.pdf	2019-12-05 10:26	213K
ReleaseNotes/	2020-12-22 08:34	-
Seatbelt_learning_aid.txt	2015-07-12 16:50	5.5K
adapt_general	2020-06-08 15:02	4.4K
airbag.recommendations	2018-12-04 13:25	4.4K
bc_in_cylindrical_coord_system	2019-08-06 17:02	4.3K
composite.models	2019-05-15 15:18	21K
concrete_models_faq	2019-02-22 12:28	18K
concrete_references	2016-03-05 09:16	5.2K
consistent_units	2019-12-20 11:50	2.2K
contact.1d	2015-07-13 17:06	2.1K
contact.13vs26	2014-05-22 11:16	4.2K
contact.beam-to-shell	2015-07-13 17:06	2.3K
contact.friction_public	2017-01-18 07:44	11K
contact.ignore1	2014-05-22 11:16	3.4K
contact.overview	2020-05-15 08:50	8.1K
contact.soft1	2020-01-24 08:21	3.8K
contact.thermal	2018-03-07 10:23	1.4K
contact_force_output	2017-04-28 08:37	1.2K
contact_stiffness_adjustment	2020-08-12 09:16	2.2K
contact_with_license_server_lost	2014-05-22 11:16	1.6K
create_solid_spotweld_from_one_node	2015-07-13 17:06	1.8K
damping	2018-03-05 15:21	9.8K
discrete_beams_for_faq	2020-09-21 10:31	12K
discretization_of_curves_faq	2019-10-28 12:06	2.0K
dynamic_relaxation_for_FAQ	2020-05-11 18:11	6.5K
effective_plastic_strain	2020-03-18 09:44	4.4K
encrypt.tar.gz	2018-03-29 14:04	345K
energy_balance	2020-03-05 12:42	11K
eos_general	2020-10-13 07:33	7.9K
finding_maxima	2019-10-28 12:06	2.8K
hex_forms	2020-05-08 09:17	7.1K
hourglass_condensed	2020-05-08 09:17	7.2K
implicit.dynamic_relaxation	2015-07-13 17:06	2.8K
implicit.dynamic_relaxation	2015-07-13 17:06	2.8K
implicit.materials_faq	2020-06-19 08:11	8.8K
implicit_guidelines	2020-11-04 09:52	5.4K
impulse_load	2017-11-14 15:30	648
instability.tips	2019-11-13 08:09	7.1K
integrated_beam_notes	2019-03-05 07:37	9.7K
interface_linking_lsda	2014-08-19 08:32	6.5K
job_queueing	2017-09-18 11:28	624
long_run_times	2014-08-08 10:48	4.2K
ls-dyna_news	2017-09-26 11:06	936
mass_scaling	2020-03-13 11:26	19K
mat77.stiffdamping_vs_freqindepdampGandSIGF.k	2018-08-13 15:11	4.2K
mpp.getting_started	2018-01-29 11:16	4.6K
mpp_bind_to_core	2018-01-25 13:11	2.3K
negative_volume_in_brick_element.tips	2020-03-13 11:23	4.2K
orthotropic_materials	2018-07-10 09:44	16K
preload.general	2018-04-16 16:29	5.5K
prescribe_body_rotation	2014-05-22 11:16	744
quasistatic	2020-06-13 09:03	3.4K
releasedates	2020-12-09 09:16	7.0K
restart	2015-07-13 17:06	2.5K
rigidwall_energy	2015-07-13 17:06	1.3K
seatbelt_pretensioner_slipring_faq	2015-07-13 17:06	13K
shell_output	2019-07-11 08:42	3.6K
shell_to_solid	2017-10-13 12:18	3.2K
shellforms	2019-02-15 12:56	5.9K
shellstrain	2019-02-15 08:13	8.2K
soil_public	2018-02-15 16:11	15K
solid_output	2018-10-04 16:46	21K
specific_heat	2018-11-07 14:37	7.5K
spin	2015-07-13 17:06	1.2K
springback	2020-03-18 15:52	8.1K
stress_vs_strain_for_plasticity_models	2014-05-22 11:16	5.0K
transform_units	2014-05-22 11:16	1.3K
user-material-notes	2019-10-30 14:02	14K
user_defined_materials.faq	2019-01-21 09:28	31K
visualizing_applied_pressure	2019-05-17 07:35	2.6K
welding_process_faq	2019-08-20 11:55	10K

4.6 LS-DNA Output Files (Results and Message Files) and Database Requests and Management

Introduction: LS-DYNA is built for speed and its file formats and file request are likewise designed for speed and the ability to efficiently handle gigantic file sizes (e.g., hundreds of GBytes). This section provides a brief overview and is not comprehensive or a replacement for the wealth of information provided in the Keyword Manual. A fundamental recommendation is to build small models and explore options with the Keyword Manual close at hand.

Results and Message Files

File Name	Type	Description
d3plot, d3plot0#	binary	Database for entire model (stress, displacements, strain and energy information (kinetic energy, internal energy, energy ratio). When LSPP reads in the d3plot file, it automatically reads in its daughter files (d3plot01, d3plot02, …..). This file can be augmented with additional results information by *DATABASE_EXTENT_BINARY
d3hsp	ascii	This file contains an echo of the submitted Keyword Deck and provides detailed analysis statistics from contact penetration, mass values of each part, warning messages and more. This file can be read into LSPP (Misc. / D3hsp View) and a summary overview is provided. Very useful for model verification (e.g., mass of model).
mes00#	ascii	Text file of on-screen messages during analysis. This file is often requested by technical support since it provides documentation on the LS-DYNA solver used, warning messages and solution statistics.
binout00#	binary	Non-contour'able results database (e.g., energies, spcforc (reaction forces at constraints), etc.). A key advantage is the ability is to request high-frequency output of specific items and not suffer from data overload within the d3plot file.
glstat, spcforc, bndout, sleout, etc.	ascii	If one is using the LS-DYNA SMP solver (in the class we default to the MPP solver), then one can also output non-contour'able results in ascii format. This will be apparent with multiple files with characteristic names.

Database Requests – The Minimum For Most Explicit Analyses (But Still One Must Read The Manual (RTM))

Keyword Command	Description
*DATABASE_BINARY_D3PLOT	One of the nine required Keywords for an LS-DYNA FEA model and of course, necessary if one wants to visualize the FEA results. This command creates the d3plot files.
*DATABASE_EXTENT_BINARY	The essential daughter Keyword command to _D3PLOT it controls what gets dumped into the binary file. Although optional it seems that most analyses require a few of the options contained within this Keyword to supplement the results dumped into the d3plot files. Too many to discuss and thus RTM.
*DATABASE_GLSTAT, _SPCFORC, & _MATSUM	No rule-of-thumb is provided but usually one wants to see the energies of the model (_GLSTAT), reaction forces (_SPCFORC) and individual energy, hourglass, mass scaling for each PART (_MATSUM).

4.7 Workshop: 1A - LS-DYNA Getting Started – Common Keyword Deck Format Errors

Objective: We will be working directly with the LS-DYNA Keyword deck and if the format is not "as required" – it'll bark at you with cryptic messages.

Introduction to LS-Run: These little workshops aims to get you a bit relaxed about working with Keyword Decks and using LS-Run. The runs listed on the right are prepared Decks where one runs'em, read the error messages and then correct the decks to run correctly to "Normal Termination".

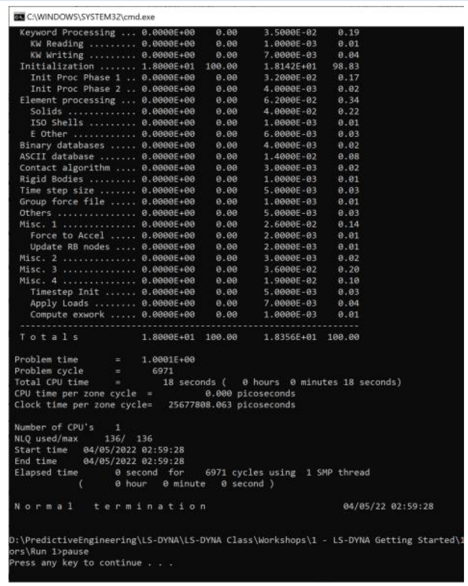
What To Be Aware of About LS-DYNA Keyword Deck Formating

- Run 1 - Start: Extra space before "*"{Keyword}
- Run 2 – Start: Data formatting error – Improperly Formatted Data
- Run 3 – Start: Data entry error – Two values within one Keyword field

Run 1 - Start

```
C:\WINDOWS\SYSTEM32\cmd.exe

*** Error 10450 (KEY+450)
    in keyword command:
*CONTROL_TERMINATION
        At line# 38 of file
D:\PredictiveEngineering\LS-DYNA\LS-DYN~1\WORKSH~1\1-LS-D~1\1A-COM~1\RUN1~1\RUN1-S~1.DYN

*** Warning 10435 (KEY+435)
    lines being skipped - See messag or d3hsp files

*** Error 10133 (KEY+133)
    input data failed with: 1 errors

E r r o r   t e r m i n a t i o n                        04/05/22 02:56:20

Memory required to complete solution   :     180K
Additional dynamically allocated memory:    2099K
                                 Total:     2279K

T i m i n g   i n f o r m a t i o n
                       CPU(seconds)   %CPU  Clock(seconds) %Clock
 ----------------------------------------------------------------
 Keyword Processing ... 0.0000E+00    0.00     6.0000E-02    0.33
   KW Reading ......... 0.0000E+00    0.00     1.0000E-02    0.05
   KW Writing ......... 0.0000E+00    0.00     6.0000E-03    0.03
 Initialization ....... 1.8000E+01  100.00     1.8144E+01   99.67
 ----------------------------------------------------------------
 T o t a l s            1.8000E+01  100.00     1.8204E+01  100.00

Problem time       =    0.0000E+00
Problem cycle      =        0
Total CPU time     =       18 seconds (   0 hours  0 minutes 18 seconds)
CPU time per zone cycle  =         0.000 picoseconds
Clock time per zone cycle=         0.000 picoseconds

Number of CPU's    1
NLQ used/max     136/  136
Start time   04/05/2022 02:56:20
End time     04/05/2022 02:56:20
Elapsed time       0 second  for       0 cycles using  1 SMP thread
            (      0 hour   0 minute   0 second )

E r r o r   t e r m i n a t i o n                        04/05/22 02:56:20
```

Run 1 - Finished



```
C:\WINDOWS\SYSTEM32\cmd.exe
 Keyword Processing ... 0.0000E+00    0.00     3.5000E-02    0.19
   KW Reading ......... 0.0000E+00    0.00     1.0000E-03    0.01
   KW Writing ......... 0.0000E+00    0.00     7.0000E-03    0.04
 Initialization ....... 1.8000E+01  100.00     1.8142E+01   98.83
   Init Proc Phase 1 .. 0.0000E+00    0.00     3.2000E-02    0.17
   Init Proc Phase 2 .. 0.0000E+00    0.00     4.0000E-03    0.02
 Element processing ... 0.0000E+00    0.00     6.2000E-02    0.34
   Solids ............. 0.0000E+00    0.00     4.0000E-02    0.22
   ISO Shells ......... 0.0000E+00    0.00     1.0000E-03    0.01
   E Other ............ 0.0000E+00    0.00     6.0000E-03    0.03
 Binary databases ..... 0.0000E+00    0.00     4.0000E-03    0.02
 ASCII database ....... 0.0000E+00    0.00     1.4000E-02    0.08
 Contact algorithm .... 0.0000E+00    0.00     3.0000E-03    0.02
 Rigid Bodies ......... 0.0000E+00    0.00     1.0000E-03    0.01
 Time step size ....... 0.0000E+00    0.00     5.0000E-03    0.03
 Group force file ..... 0.0000E+00    0.00     1.0000E-03    0.01
 Others ............... 0.0000E+00    0.00     5.0000E-03    0.03
 Misc. 1 .............. 0.0000E+00    0.00     2.6000E-02    0.14
   Force to Accel ..... 0.0000E+00    0.00     2.0000E-03    0.01
   Update RB nodes .... 0.0000E+00    0.00     2.0000E-03    0.01
 Misc. 2 .............. 0.0000E+00    0.00     3.0000E-03    0.02
 Misc. 3 .............. 0.0000E+00    0.00     3.6000E-02    0.20
 Misc. 4 .............. 0.0000E+00    0.00     1.9000E-02    0.10
   Timestep Init ...... 0.0000E+00    0.00     5.0000E-03    0.03
   Apply Loads ........ 0.0000E+00    0.00     7.0000E-03    0.04
   Compute exwork ..... 0.0000E+00    0.00     1.0000E-03    0.01
 ----------------------------------------------------------------
 T o t a l s            1.8000E+01  100.00     1.8356E+01  100.00

Problem time       =    1.0001E+00
Problem cycle      =     6971
Total CPU time     =       18 seconds (   0 hours  0 minutes 18 seconds)
CPU time per zone cycle  =         0.000 picoseconds
Clock time per zone cycle=   25677808.063 picoseconds

Number of CPU's    1
NLQ used/max     136/  136
Start time   04/05/2022 02:59:28
End time     04/05/2022 02:59:28
Elapsed time       0 second  for    6971 cycles using  1 SMP thread
            (      0 hour   0 minute   0 second )

N o r m a l   t e r m i n a t i o n                      04/05/22 02:59:28


D:\PredictiveEngineering\LS-DYNA\LS-DYNA Class\Workshops\1 - LS-DYNA Getting Started\1
ors\Run 1>pause
Press any key to continue . . .
```

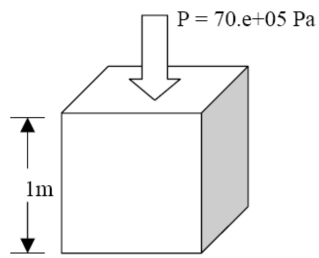
Run 2 - Start

```
C:\WINDOWS\SYSTEM32\cmd.exe
*** Error 10246 (KEY+246)
    line contains improperly formatted data
     reading *CONTROL_TERMINATION
        At line# 40 of file
D:\PredictiveEngineering\LS-DYNA\LS-DYN~1\WORKSH~1\1-LS-D~1\1A-COM~1\RUN2~1\RUN2-S~1.DYN
----------------------------------------------------------------------
      1.0       1.0
----------------------------------------------------------------------

*** Error 10133 (KEY+133)
    input data failed with: 2 errors

E r r o r   t e r m i n a t i o n                        04/05/22 03:17:37

Memory required to complete solution   :     180K
Additional dynamically allocated memory:    2099K
                                 Total:     2279K

T i m i n g   i n f o r m a t i o n
                       CPU(seconds)   %CPU  Clock(seconds) %Clock
 ----------------------------------------------------------------
 Keyword Processing ... 0.0000E+00    0.00     3.3000E-02    0.18
   KW Reading ......... 0.0000E+00    0.00     3.0000E-03    0.02
   KW Writing ......... 0.0000E+00    0.00     5.0000E-03    0.03
 Initialization ....... 1.7000E+01  100.00     1.7827E+01   99.82
 ----------------------------------------------------------------
 T o t a l s            1.7000E+01  100.00     1.7860E+01  100.00

Problem time       =    0.0000E+00
Problem cycle      =        0
Total CPU time     =       17 seconds (   0 hours  0 minutes 17 seconds)
CPU time per zone cycle  =         0.000 picoseconds
Clock time per zone cycle=         0.000 picoseconds

Number of CPU's    1
NLQ used/max     136/  136
Start time   04/05/2022 03:17:37
End time     04/05/2022 03:17:37
Elapsed time       0 second  for       0 cycles using  1 SMP thread
            (      0 hour   0 minute   0 second )

E r r o r   t e r m i n a t i o n                        04/05/22 03:17:37


D:\PredictiveEngineering\LS-DYNA\LS-DYNA Class\Workshops\1 - LS-DYNA Getting Started\1A - Common
ors\Run 2>pause
Press any key to continue . . .
```

Analyst's Note: *"Know what you Know" – With nonlinear analysis codes there are so many options, that one should only change those defaults that one knows (which often means reading the manual and maybe creating a pilot model to understand its effect).*

4.8 Workshop 1B – LS-DYNA Getting Started

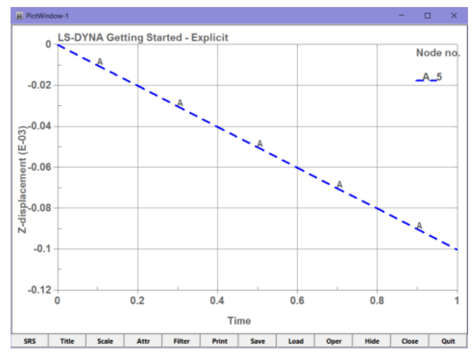
Objective: This workshop uses the LSTC Getting Started Example material and a LS-DYNA model has been prepared. This material can also be found in the Students' "Class Reference Notes" folder. A Workshop video is provided to walk you through the post-processing of the data but your job is to create the one element model that has an applied pressure load.



Aluminum 1100-O	
density	2700 kg/m^3
modulus of elasticity	70.e+09 Pa
Poisson Ratio	0.3
coefficient of expansion	23.6e-06 m/m K
heat capacity	900 J/kg K
thermal conductivity	220 W/m K

Tasks:

- Open your favorite text editor and build LS-DYNA Keyword deck using the existing deck: /Explicit Example 1 / ex01 – Start.dyn. The node positions and their constraints have been pre-entered to save you some of the more mundane work. The rest of the Keywords you'll have to figure out (***Workshop - GettingStarted.pdf and Keyword Manual Vol I. Please note that Material Keywords are located in Manual Vol II***).
- Analyze your model using LS-Run and post process the results within LSPP
- If time exists proceed to other examples.

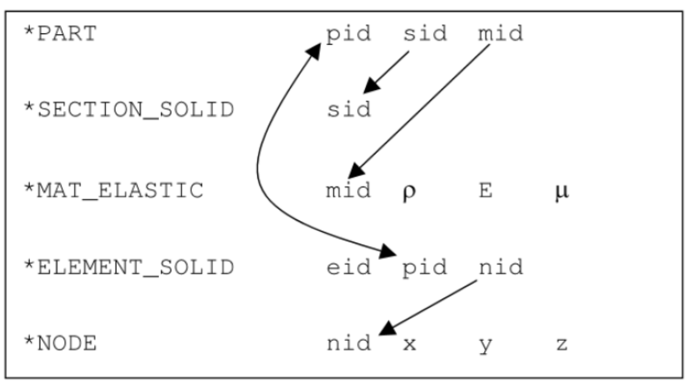


The vertical displacement due to a 70.0e+05 Pa pressure load can be calculated by

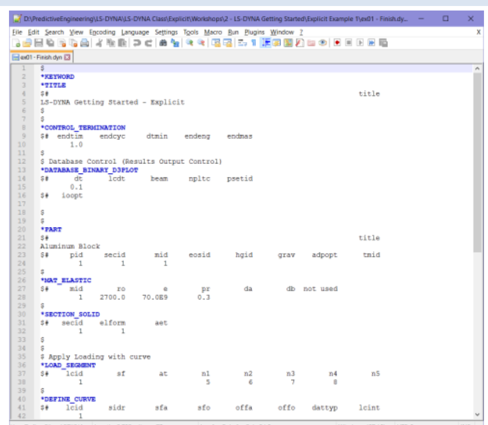
$$\Delta l = \frac{Pl}{E} = \frac{(70e+05)(1)}{(70e+09)} =$$

1.0e-04 m

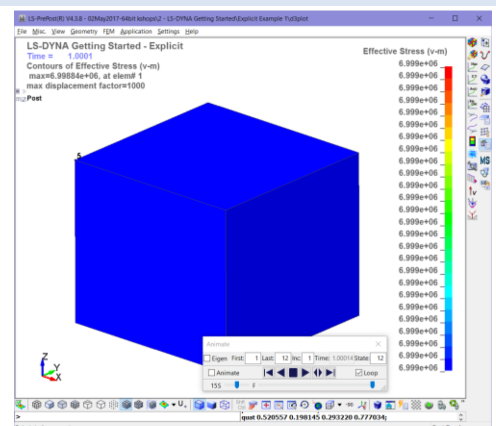
Pivotal Understanding: The "*PART"



Advantages of Using Notepad++



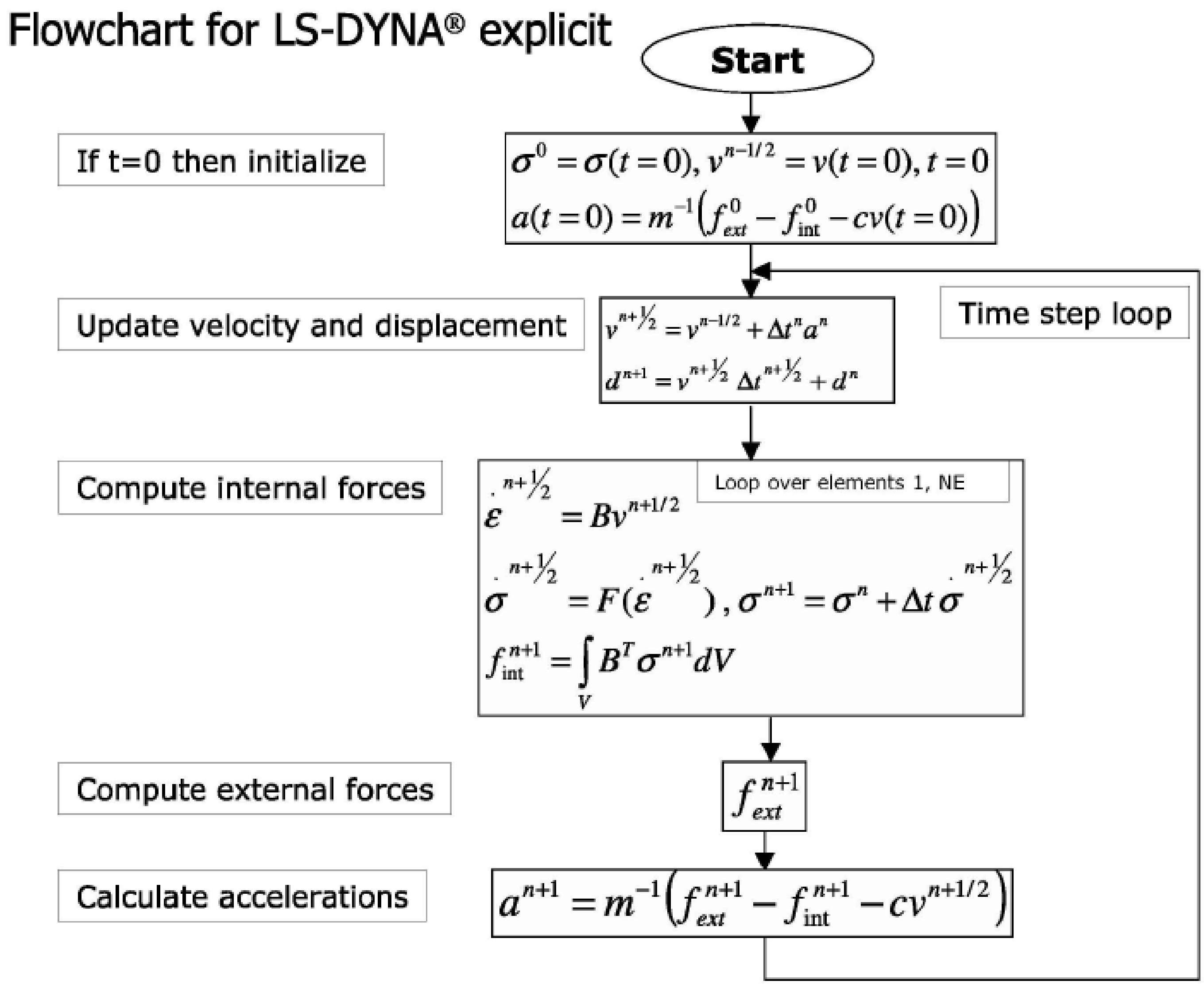
You Should See This



Take Away (Importance of Workshop): What is involved in building a LS-DYNA FEA model; that simple and direct.

5. FUNDAMENTAL MECHANICS OF EXPLICIT ANALYSIS

5.1 Explicit Numerical Flowchart



5.2 Time Step Significance (Courant-Friedrichs-Lewy (CFL) Characteristic Length)

- In the simplest case (small, deformation theory), the timestep is controlled by the acoustic wave propagation through the material.
- In the explicit integration, the numerical stress wave must always propagate less than one element width per timestep.
- The timestep of an explicit analysis is determined as the minimum stable timestep in any ***one (1) deformable finite element in the mesh***. (Note: As the mesh deforms, the timestep can similarly change)
- The above relationship is called the Courant-Friedrichs-Lewy (CFL) condition and determines the stable timestep in an element. The CFL condition requires that the explicit timestep be smaller than the time needed by the physical wave to cross the element. Hence, the numerical timestep is a fraction (0.9 or lower) of the actual theoretical timestep. Note: the CFL stability proof is only possible for linear problems.
- In LS-DYNA, one can control the time step scale factor (*tssfac*). The default setting is 0.9. It is typically only necessary to change this factor for shock loading or for increased contact stability with soft materials.
- As a note, the *tssfac* doesn't change the "wave speed" only the time step.

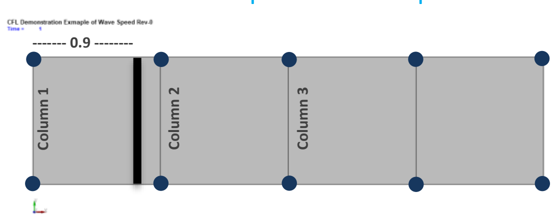
$$C_{AccousticWaveSpeed} = \sqrt{\frac{E_{Material}}{\rho_{Material}}}$$

$$\Delta ExplicitTimestep = \frac{Length_{Element}}{C_{Wavespeed}}$$

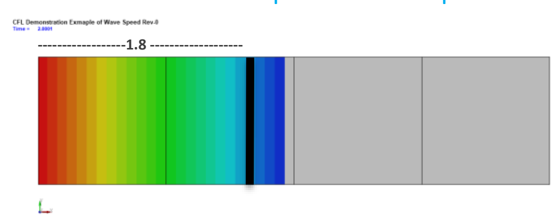
$$\Delta Timstep_{CFL} = (0.9)\Delta ExplicitTimestep$$

A bar is given a whack on its end. At time = 0.0, the state of the system is unknown except where loads and constraints are applied. At the first explicit time step, the stress wave (black bar) has advanced 90% or 0.9 into the element. The explicit calculation then can calculate the acceleration at the first column of nodes but everything else is still unknown. Hence, we still have zero stress in the bar. At the second explicit time step, we finally have all the information that we need to calculate stress in the first element. The stress state can now be contoured. Be aware that the stress fringe shown is extrapolated from the data set. We still do not know what the stresses are in the second element until the third time step has completed.

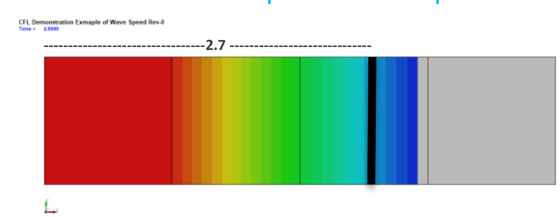
First Explicit Time Step



Second Explicit Time Step



Third Explicit Time Step



Analyst's Note: Based on these conditions, the time step can be increased to provide faster solution times by: (i) increasing the density of the material (e.g., mass scaling; (ii) lowering the modulus or; (iii) by increasing the element size of the mesh.

5.2.1 Is the CFL based on Elements or Nodes?

The CFL is based on neither but the product of stiffness and mass, i.e., in a purely mathematical sense. It is just common to describe it using wave speed and a characteristic element length whether for a rod (1D), shell (2D) or solid (3D) elements. But then this leaves out springs, which can be defined with coincident nodes, which can be denoted as 0D elements. Within LST's LS-DYNA Theory Manual is a section on time step calculations that covers the CFL explicit time step calculation for springs (0D). As shown, all one needs is stiffness and mass to calculate an explicit time step. At the end of this section, a little note is given: "*The springs used in the contact interface are not checked for stability.*" That is, when LS-DYNA starts an analysis, it will sweep through all elements from 0D to 3D and calculate the explicit time step but does not check contact interface explicit time step for ***stability*** for global control of the model's time step (although it does provide an estimate of the interface's time step). In LS-DYNA, the penalty formulation (virtual springs) *CONTACT has an explicit time step based on the stiffness and mass of the opposing surfaces. Although the explicit time step of the contact interface is not normally a problem, it is something that one should be aware of and if a warning message is displayed, it should be paid heed.

LS-DYNA Theory Manual

25.5 Time Step Calculations for Discrete Elements

For spring elements such as that in Figure 25.1 there is no wave propagation speed c to calculate the critical time step size.

The eigenvalue problem for the free vibration of spring with nodal masses m_1 and m_2, and stiffness, k, is given by:

$$\begin{bmatrix} k & -k \\ -k & k \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \omega^2 \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{25.20}$$

Since the determinant of the characteristic equation must equal zero, we can solve for the maximum eigenvalue:

$$\det\begin{bmatrix} k-\omega^2 m_1 & -k \\ -k & k-\omega^2 m_2 \end{bmatrix} = 0 \quad \rightarrow \quad \omega_{\max}^2 = \frac{k(m_1+m_2)}{m_1 \cdot m_2}, \tag{25.21}$$

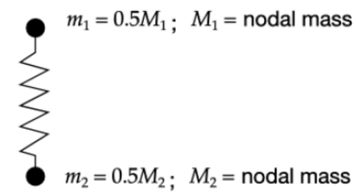


Figure 25.1. Lumped spring mass system.

Recalling the critical time step of a truss element:

$$\left.\begin{aligned} \Delta t &\le \frac{\ell}{c} \\ \omega_{\max} &= \frac{2c}{\ell} \end{aligned}\right\} \quad \Delta t \le \frac{2}{\omega_{\max}}, \tag{25.22}$$

and approximating the spring masses by using 1/2 the actual nodal mass, we obtain:

$$\Delta t = 2\sqrt{\frac{m_1 m_2}{m_1+m_2}\frac{1}{k}}. \tag{25.23}$$

Therefore, in terms of the nodal mass we can write the critical time step size as:

$$\Delta t_e = \sqrt{\frac{2M_1M_2}{k(M_1+M_2)}}. \tag{25.24}$$

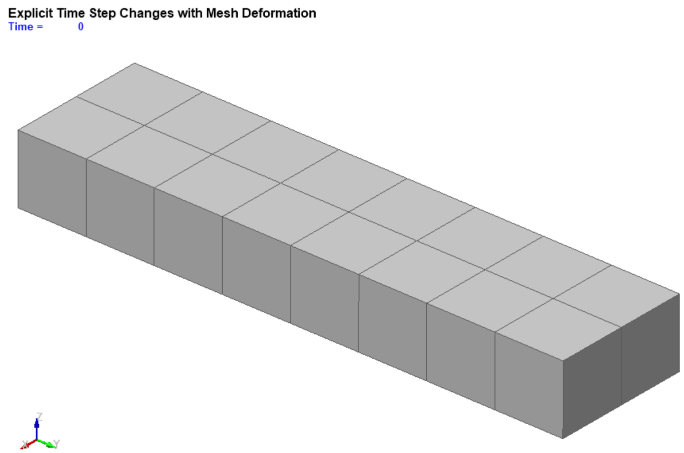
The springs used in the contact interface are not checked for stability.

Analyst's Note: In Workshop 6, within the file folder Zero Length, an example of a 0.0 length spring element model is given. As the model runs and the nodes separate from 0.0, one will notice that the time step does not change.

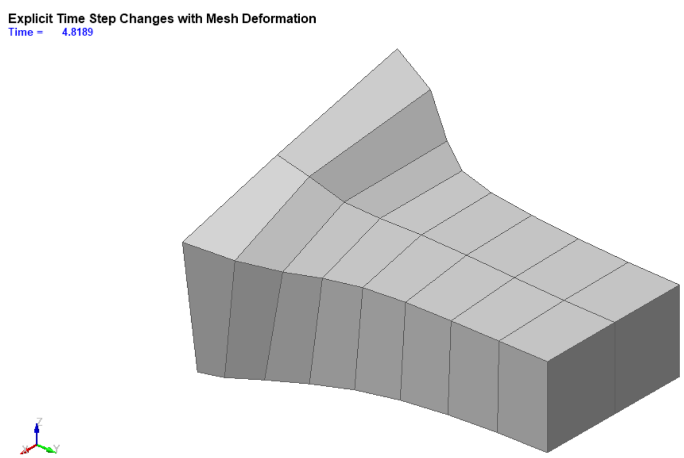
5.2.2 As the Mesh Size Changes, So Does the Explicit Time Step

Given the theoretical background of how the explicit time step is calculated, one must also consider that as the analysis progresses and the mesh deforms, the explicit time step will likewise change.

FEA Mesh	Explicit Analysis: Start dt = 3.92e-2 / Finish dt = 7.83e-3




```
initialization completed
      1 t 0.0000E+00 dt 3.92E-02 flush i/o buffers            05/08/19 04:56:25
      1 t 0.0000E+00 dt 3.92E-02 write d3plot file            05/08/19 04:56:25
      2 t 3.9205E-02 dt 3.80E-02 write d3plot file            05/08/19 04:56:25
      3 t 7.7181E-02 dt 3.70E-02 write d3plot file            05/08/19 04:56:25
      4 t 1.1417E-01 dt 3.62E-02 write d3plot file            05/08/19 04:56:25
      5 t 1.5036E-01 dt 3.53E-02 write d3plot file            05/08/19 04:56:25
      6 t 1.8569E-01 dt 3.45E-02 write d3plot file            05/08/19 04:56:25
      7 t 2.2021E-01 dt 3.38E-02 write d3plot file            05/08/19 04:56:25
      8 t 2.5406E-01 dt 3.33E-02 write d3plot file            05/08/19 04:56:25
      9 t 2.8735E-01 dt 3.28E-02 write d3plot file            05/08/19 04:56:25
     10 t 3.2014E-01 dt 3.23E-02 write d3plot file            05/08/19 04:56:25
```




```
    335 t 4.9323E+00 dt 7.97E-03 write d3plot file            05/08/19 04:56:25
    336 t 4.9403E+00 dt 7.95E-03 write d3plot file            05/08/19 04:56:25
    337 t 4.9482E+00 dt 7.94E-03 write d3plot file            05/08/19 04:56:25
    338 t 4.9562E+00 dt 7.92E-03 write d3plot file            05/08/19 04:56:25
    339 t 4.9641E+00 dt 7.90E-03 write d3plot file            05/08/19 04:56:25
    340 t 4.9720E+00 dt 7.88E-03 write d3plot file            05/08/19 04:56:25
    341 t 4.9799E+00 dt 7.87E-03 write d3plot file            05/08/19 04:56:25
    342 t 4.9877E+00 dt 7.85E-03 write d3plot file            05/08/19 04:56:25
    343 t 4.9956E+00 dt 7.83E-03 write d3plot file            05/08/19 04:56:25

*** termination time reached ***
    343 t 5.0034E+00 dt 7.83E-03 write d3dump01 file          05/08/19 04:56:25
    343 t 5.0034E+00 dt 7.83E-03 flush i/o buffers            05/08/19 04:56:25
    343 t 5.0034E+00 dt 7.83E-03 write d3plot file            05/08/19 04:56:25

N o r m a l    t e r m i n a t i o n                         05/08/19 04:56:25
```

*Analysts' Note: As one can imagine, with severe deformation, the explicit time step could approach zero. LS-DYNA allows one to stop the analysis or delete highly deformed elements based on their timestep (see *CONTROL_TERMINATION, dtmin and then *CONTROL_TIMESTEP, erode)*

5.3 Mass Scaling: (Everybody Does it But Nobody Really Likes It) – Changing the Wave Speed

Explicit Time Step Mass Scaling (*Control_Timestep)*

- Mass scaling is very useful and directly lowers the wavespeed and therefore increases the timestep given that the element sizing doesn't change. The concept is simple, **Larger Timestep = Lower Solution Time**
- One can also just simply increase the global density of the material for non-dynamic simulations (i.e., where inertia effects can be considered small).
- *CONTROL_TIMESTEP: Conventional mass scaling (CMS) (negative value of *dt2ms* *(Note in Keyword Manual "LT" means a negative value or "-" in front of the number))*: The mass of small or stiff elements is increased to prevent a very small timestep. Thus, artificial inertia forces are added which influence all eigenfrequencies including rigid body modes. This means, this additional mass must be used very carefully so that the resulting non-physical inertia effects do not dominate the global solution. This is the standard default method that is widely used.
- With CMS, a recommended target is not to exceed 5% of the mass of the system or 10% of the mass of any one part. Added mass can be tracked with *DATABASE options of GLSTAT for entire model and MATSUM for individual parts. But I prefer to visualize it within LSPP using *DATABASE_EXTENT_BINARY with *stssz* and/or *msscl* settings (RTM).

Analysts' Note: General recommendations and tips are given in Explicit Model Check-Out and Recommendations. A really good overview of mass scaling can be found here: https://www.dynasupport.com/howtos/general/mass-scaling

$$\Delta Timstep_{CFL} = tssfac \frac{Length_{Element}}{\sqrt{\dfrac{E}{\rho * Mass\ Scaling}}}$$

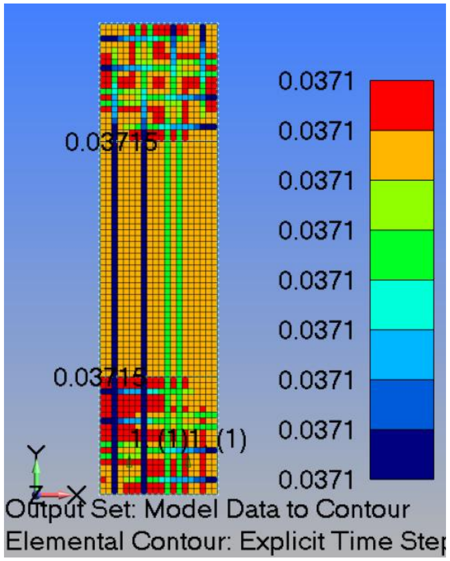
$$C_{Aluminum} = \sqrt{\frac{\frac{70}{(1-\upsilon^2)}}{2.71x10^{-6}}} = 5{,}384 \text{ mm/ms}$$

$$\Delta Timestep_{Al} = 0.9 \cdot \frac{200}{5{,}384} = 0.9 \cdot 0.0371 = 0.0334\, ms$$

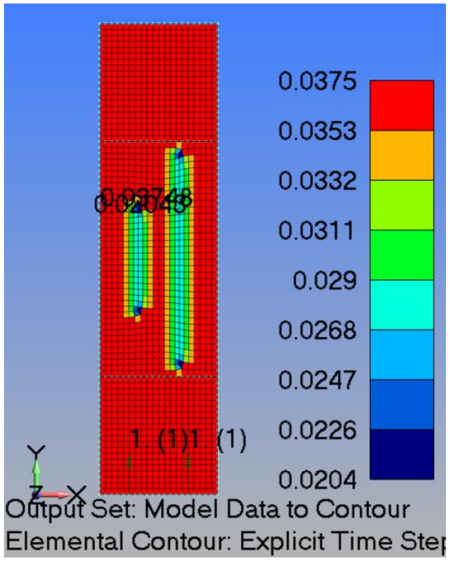
- LS-DYNA time step is different between LSPP and LS-DYNA due to *tssfac*=0.9 (default)
- Mesh quality affects Time Step – just tweak it

5.3.1 Instructor Led Workshop: 1 – Mass Scaling

Clean Mesh



Tweaked



5.3.2 Workshop: 2 - LS-DYNA Mass Scaling Basics

What You Will Learn:

Simple class exercise to reinforce the concept of mass scaling basics and how to view the explicit time step within LSPP.

The model is just a simple plate that is hit with a short force pulse along its bottom edge. This force pulse then propagates a stress wave through the bar. The physics are classic and your job is to manage the explicit time step.

Tasks:

- Open Keyword deck: /LS-DYNA Mass Scaling Basics / LSPP / Clean Mesh / LS-DYNA Mass Scaling Basics - Clean.dyn in LSPP. Verify elastic isotropic material (*MAT_ELASTIC) properties and then shell property (*SECTION_SHELL) with *elform*=2 and thickness = 1.0.
- Check explicit time step using LSPP's command Application / Model Checking / General Checking / Element Quality / Shell check item / *check* Time step.
- Change elastic modulus from 70 to 35 and re-contour time step.
- Submit the model for analysis using LS-Run

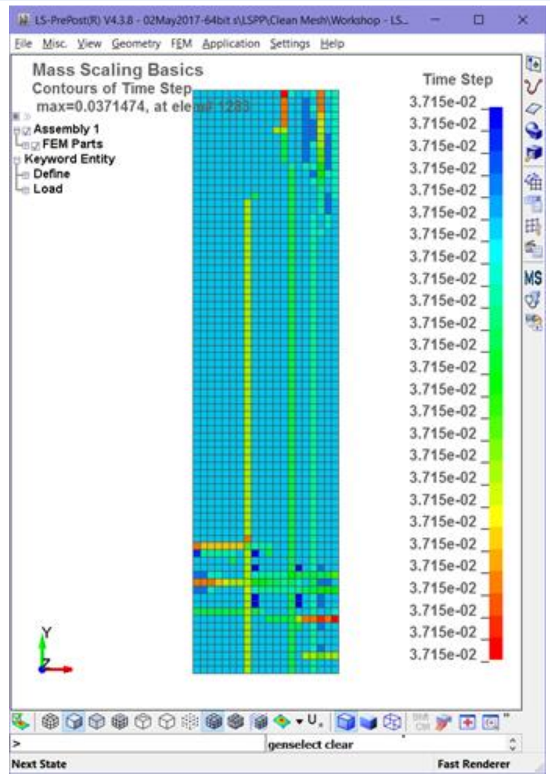
Analysts' Note: Although this model would run faster using SMP single-precision (~10 to 20%), we are aiming to keep it simple and use the most robust solver platform. As always, once a solution is in hand, one can seek efficiencies try running it with single-precision.

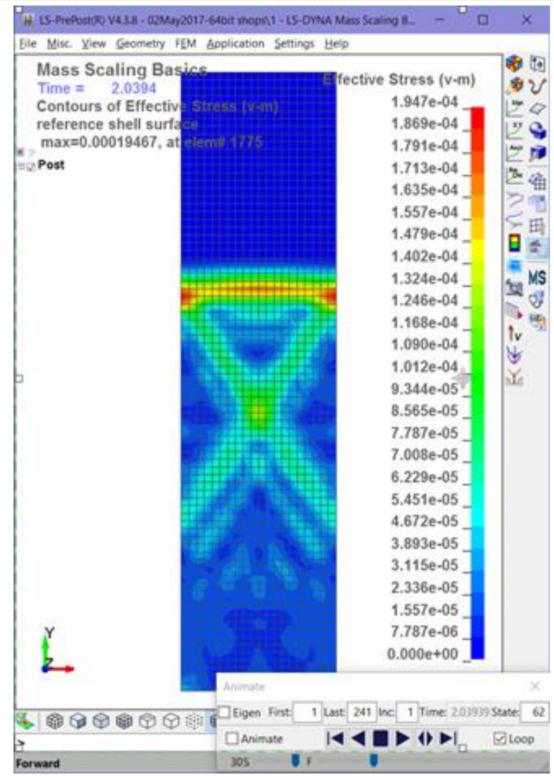
Units: kN-mm-ms-kg (stresses in GPa)

Linear, Elastic Material:

E	v	ρ
70	0.33	2.71e-6

Explicit Time Step	Stress Wave Propagation from Short Duration Force Pulse





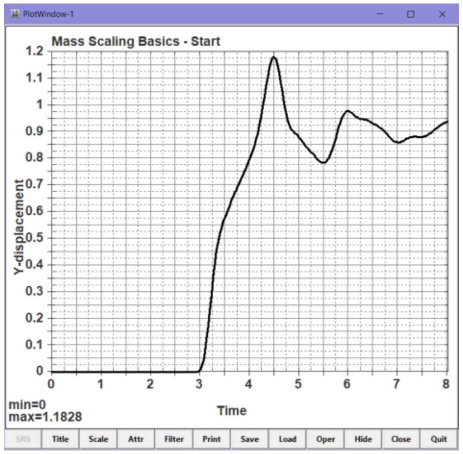
Workshop: 2 – LS-DYNA Mass Scaling Basics (continued)

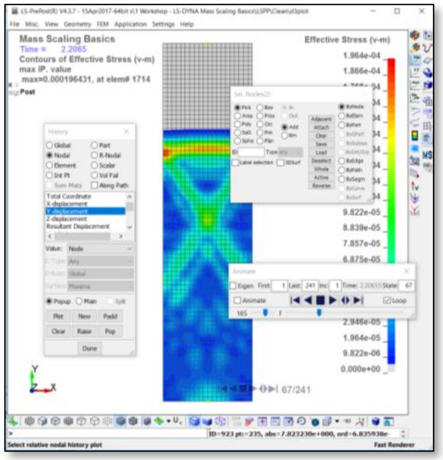
With the model working, let's harvest some data. We are going to make several runs of this model to investigate the relationship between mesh, explicit time step and mass scaling. As part of this process, you'll get comfortable working with LSPP and LS-Run. Our metric is going to be the maximum displacement from a node at the end of the bar (Node #1).

Tasks:

- Within existing LSPP model, open History, select Node, Y-Displacement and then pick the node at the very top of the bar (node #1) and hit Plot. (Note: The node is attached to a constrained nodal rigid body).
- Note that the maximum displacement at the top of the bar is 1.18 mm.
- Start filling out the Table at the bottom of this page.

{A filled-out Table is provided for you to check your work within a nested file folder labeled "Table".}





Documenting the Learning Objective:

Open the Keyword Deck LS-DYNA Mass Scaling Basics - Skewed Mesh - Start.dyn in your favorite text editor and apply conventional mass scaling (CMS) to the *CONTROL_TIMESTEP keyword card via the dt2ms option. The idea is to match the original time step in the clean mesh example and understand that mass scaling is invaluable but alas has drawbacks (i.e., one should carefully check your results).

Analysts' Note: Remember that the tssfac=0.9 and thus to get an explicit step of 0.0334, one must use a value of dt2ms=-0.0371.

Model	Time Step	% Mass Added by Mass Scaling	Max. Displacement
Starting Point	0.0334 ms	0.0%	1.18 mm
Skewed Mesh (-4x)	0.0184 ms	0.0%	mm
Skewed Mesh with Mass Scaling I	0.0334 ms	%	mm
Skewed Mesh with Mass Scaling II	0.0290 ms	%	mm

Got Extra Time? Open up: Abandon All Hope {Workshop - LS-DYNA Mass Scaling Basics - Skewed Mesh - Violation of CFL – FINISH.dyn} and see what happens when one forces LS-DYNA to ignore the CFL criterion. It'll bark at you but it'll run. For LS-DYNA non-newbies, take a look at the EXTRA file folder and contour the mass percentage added via CMS and create an XY plot of the added mass to the three PART's of the model.

5.3.3 Instructor Led Workshop: 2 - Mass Scaling Advanced

Explicit Time Step Mass Scaling (*CONTROL_TIMESTEP):*

- Mass scaling is no free lunch. For dynamic systems, added mass can affect the response of the system (i.e., like additional *un-wanted* KE).
- It is just something to monitor and make an engineering judgment about its effectiveness; time savings versus potential detrimental effects. Mass scaling is my universal modeling condiment, and the aim is typically no more than 5% additional mass.
- Conventional mass scaling (CMS) has morphed to using the negative (-)*dt2ms* option as the recommended default.
- Selective mass scaling (SMS): Using selective mass scaling, only the high frequencies are affected, whereas the low frequencies (rigid body bodies) are not influenced; thereby, a lot of artificial mass can be added to the system without adulterating the global solution.
- This method is very effective, if it is applied to limited regions with very small critical timesteps. SMS is invoked with the *imscl* command over a single part or multiple parts.

Example:

We are impacting a ball against a plate. The mesh is not uniform and mass scaling is used to speed up the analysis. The ball is first analyzed with no mass scaling, then with CMS and finally with SMS over the whole ball. To verify our analysis, we plot the kinetic energies from all three runs.

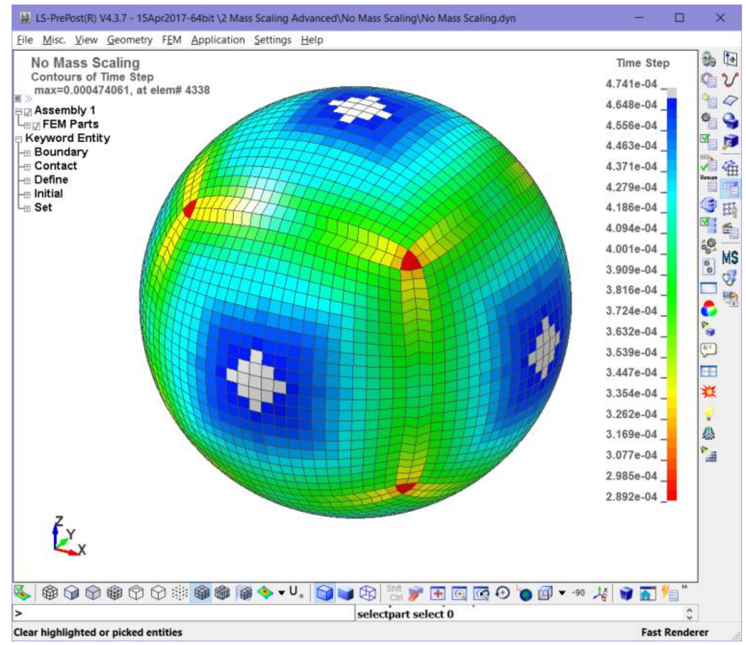
*Analysts' Note: Please understand that CMS is used **on all other parts** not called out within the IMSCL command (see Keyword Manual*

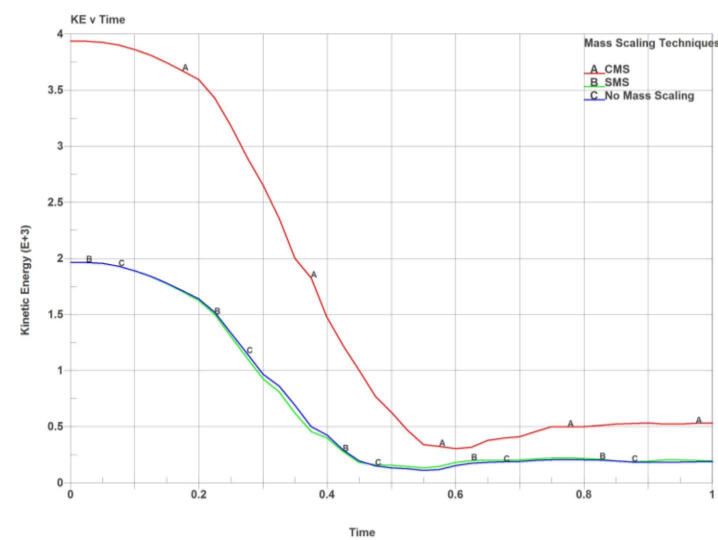
Some Finer Points:

- Solution time is 28 seconds for no mass scaling and 15 and 13 seconds for SMS and CMS respectively. SMS is more computationally expensive but has large benefits for some models.
- Question: Would mass scaling make your dynamic (F=ma) analysis more conservative or less?

Example Courtesy of www.DynaSupport.com

Time step ranges from 2.89 to 4.74*e*-4





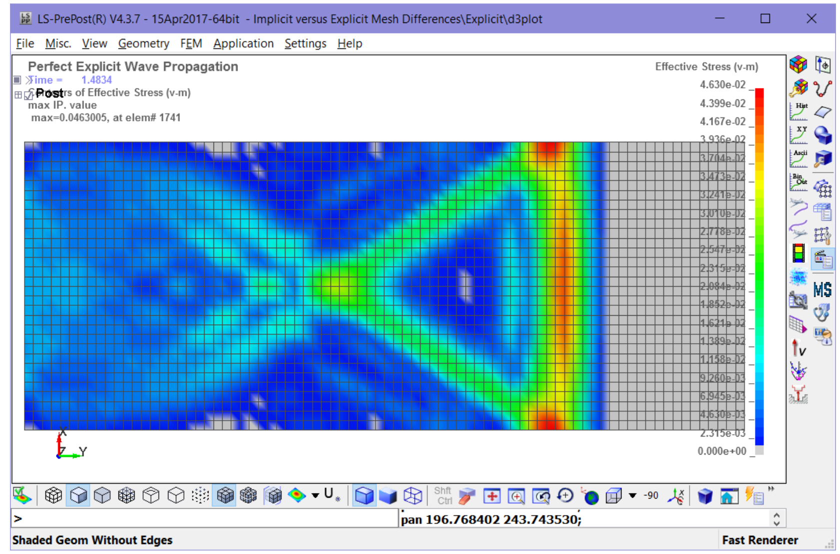
5.4 Implicit Mesh versus Explicit Mesh Characteristics

5.4.1 Instructor Led Workshop: 3 - Implicit versus Explicit Mesh Differences

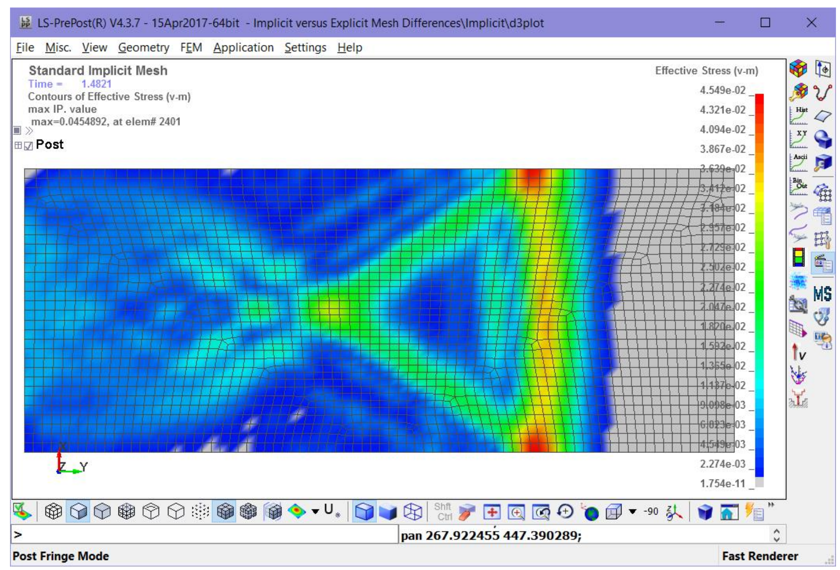
Meshing for Accuracy

- Solution time (number of nodes + time step) is often one of the most important considerations in setting up an explicit analysis; care should be exercised in setting up the mesh density.
- A good implicit mesh ***does not*** typically work well for an explicit analysis.
- In an explicit analysis, linear, elastic stresses are not often the most important analysis result. Typically, plastic strain, energy, crushing behavior, etc. are more important. These results are not as mesh sensitive as linear, elastic stresses and permit a much larger element size to be used.

Since the time step is controlled by wave propagation, the mesh should be graded gradually to likewise allow a smooth wave propagation through the structure whenever possible.



Analyst's Note: Mass scaling is great but it needs to be combined with a reasonable mesh gradient.

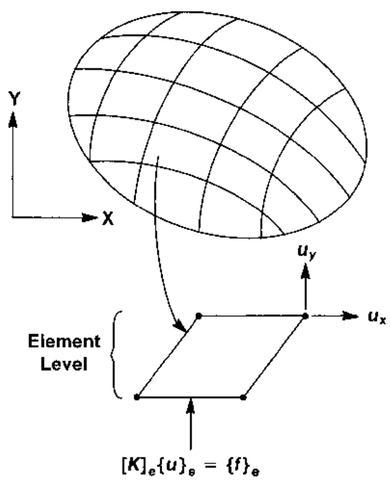


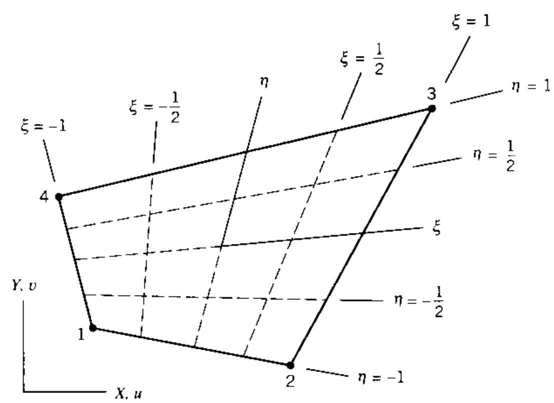
5.4.2 A Short Discussion on Element Quality (aka Jacobian)

Although this section covers material that might be understood by most students, it provides an introduction to the importance of element quality in performing explicit and implicit analyses. If one is not sure what are isoparametric elements, take a quick read from a chapter of Ed Wilson's book located in the Class Reference Notes / Elements / Isoparametric Element Theory (www_EdWilson_org - Book-Wilson - 05-iso.pdf).

Isoparametric (having the same parameters under different coordinate systems) are the bedrock of modern FEA. Simple functions are used to discretize oddly shaped surfaces or volumes. The basis of this method is given in the subsequent slides. Although the theory is given in 2-D it can be directly leveraged into the third dimension.

One starts with a random region that is normalized into a -1 to +1 coordinate system and two formulas that use a simple linear shape function to define interior coordinates and interior displacements:





$$x_{xp} = \sum_{i=1}^{4} N_i(\xi,\eta)x_{xi}$$

$$u_{xp} = \sum_{i=1}^{4} N_i(\xi,\eta)u_{xi}$$

N_i is known as the shape function, which does double duty as the interpolation function for both coordinates (x) and displacements (u). This is the "iso" in the isoparametric. With these formulas we can map displacements in the interior of our element and also map any coordinates. An example of a linear shape function for a four-node quadrilateral element (see FEA textbooks for quadratic shape functions use in parabolic eight-node quadrilateral elements):

5.4.2.1 An Example of the Assembly of Equations for Static Stress Analysis

We start with basic mechanics and apply the isoparametric method to these equations.

Step 1: Satisfy static equilibrium

$$\sum F = 0$$

Step 2: Relate strain to displacements (simple 2D example)

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \partial/\partial_x & 0 \\ 0 & \partial/\partial_y \\ \partial/\partial_y & \partial/\partial_x \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix}; \qquad \varepsilon = \partial u$$

Step 3: Incorporate the shape function

This is where it gets a little complicated. To get our generalized displacements (u, v), the shape functions discussed on the prior slide are used to take corner point displacements (nodes) u_i and v_i and generate displacements anywhere within the element.

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & \cdots \\ 0 & N_1 & 0 & N_2 & \cdots \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \end{Bmatrix}; \qquad u = Nd$$

Step 4: Relate strain to displacements (using the B matrix)

Matrix "B" is called the strain-displacement matrix and is common FEA matrix jargon. The concept is that you are using the shape function to determine the "strain characteristics" within the quadrilateral element.

$$\varepsilon = \partial N d; \qquad \varepsilon = Bd; \qquad B = \partial N$$

Step 5: Relate stress to strain

$$\sigma = E\varepsilon; \qquad \sigma = EBd$$

Step 6: Relate force to stress

$$F = E\varepsilon A; \qquad F = EBdA$$

Step 7: Relate force to displacement

$$F = Kd; \qquad K = EBA; \qquad u = d$$

The pivotal part is that "EBA" is the stiffness term of the element. It is this component that is calculated to form the stiffness matrix [K]. It doesn't seem that hard but just calculating the area of a quadrilateral by brute force (double integration) is a numerically very intensive task.

$$\{F\} = [K]\{u\}$$

The prior expression that formulated K=EBA leaves out a few numerical details. To actually calculate individual stiffness terms for the element, the formula EBA must be numerically integrated over the area or volume of the element. This is done by the following standard equation:

$$[K] = \iint [B]^T [E][B] dxdy$$

However, if a standard generalized numerical integration would be used, this operation would be very slow and model sizes would be limited to a few thousands of elements and not hundreds of thousands as in the norm today. To accelerate the numerical area or volume calculation, a process called Gaussian Integration is used. For this process to work, the first step is to transform the generalized X and Y coordinates into normalized -1 to +1 space. This is a linear transformation or a mapping process. The transformation matrix is called the Jacobian. Every element will have a unique transformation from its generalized coordinates into a normalized system.

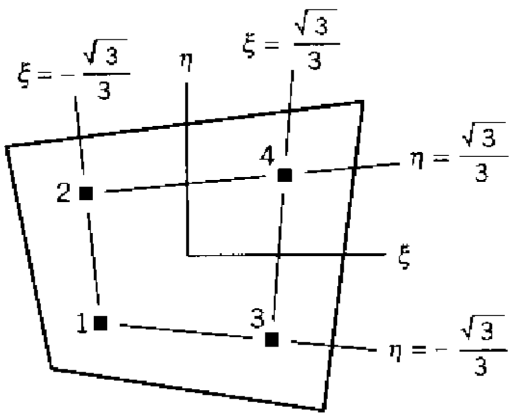
$$[K] = \iint_{-1}^{1} [B]^T [E][B][J]\, d\xi d\eta$$

The Jacobian is also a popular measure of the element's quality. If the element is distorted, one might say that the Jacobian has a lot of work to do in normalizing the element into a -1, +1 space. If the element is a square or a clean rectangle, the Jacobian practically does nothing. A value of 0.0 indicates a perfectly shaped element and a value of near 1.0 indicates something that might not be solvable.

5.4.2.2 Gaussian Integration for Isoparametric Elements

To numerically integrate the isoparametric element a technique known as Gauss Quadrature is employed. This technique is based on the element having a normalized coordinate system of -1, +1. Essentially, the inner terms of the stiffness equation given below are only solved at discrete points within the element and weighting functions based on Gauss Quadrature are then applied. The discrete points where this numerical integration is carried out are called Gaussian Integration Points (Gauss points). An example of the location of Gauss points in a quadrilateral element is given below.

Gaussian integration is at its best (i.e., most accurate) when the element is as near as possible to a perfect square. During the integration process, tabulated weighting values are used (terms Wi and Wj) to arrive at the final integrated value (I) for the elements area or volume:



$$\boldsymbol{I} = \sum_{i=1}^{n} \sum_{j=1}^{m} W_i W_j \phi(\xi_i \eta_j)$$

The location of these Gauss points is also used for strain recovery and with strain we have stress. That is, in isoparametric elements, stresses are calculated at the Gauss points and extrapolated out to the nodal points for contouring. Hence, a high-quality element (low Jacobian) will provide double benefits with a more accurate [K] and cleaner stress calculation.

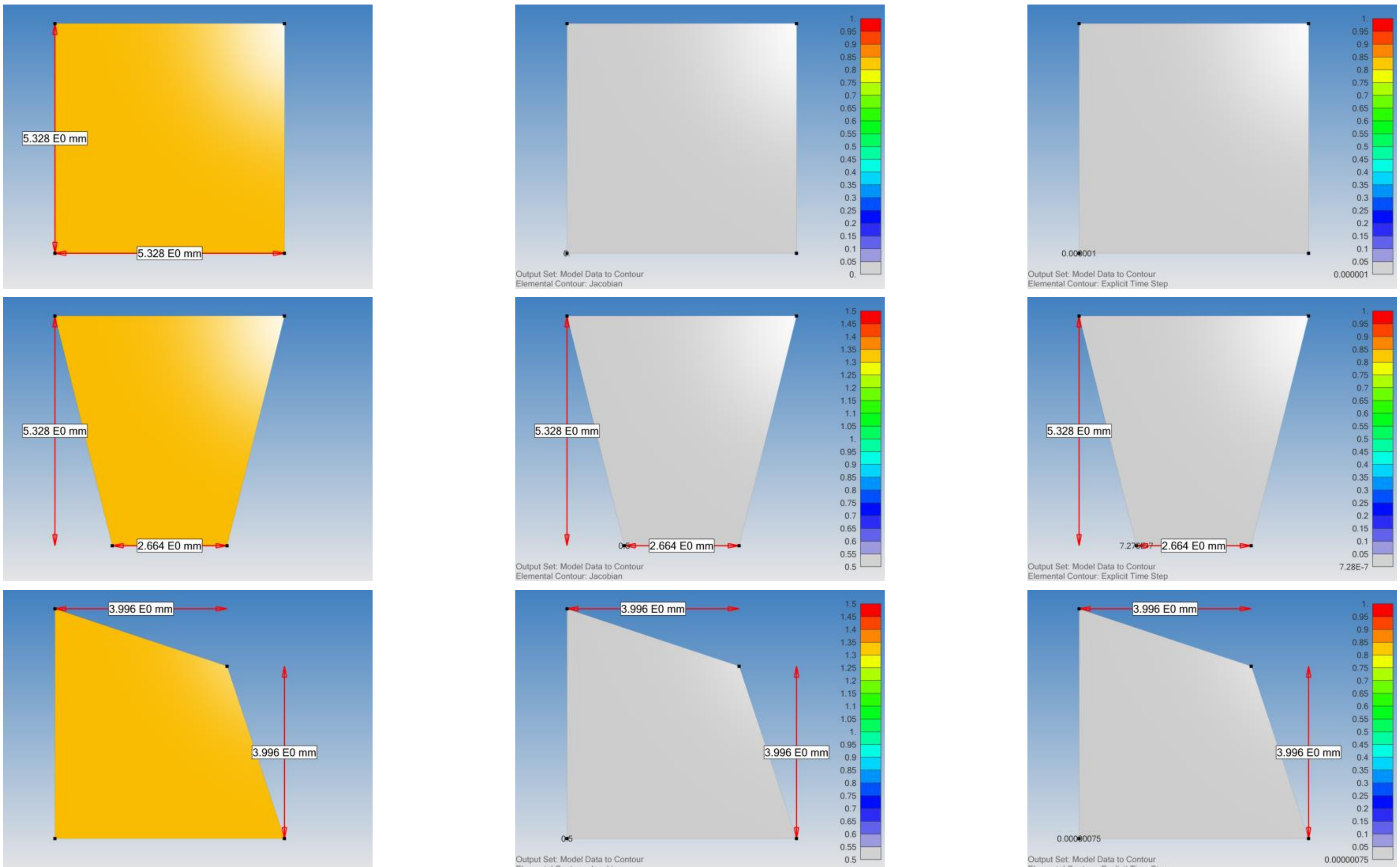
For additional reading on the subject, see the following references:

R. D. Cook, D. S. Malkus, M. E. Plesha, and R. J. Witt, "Concepts and Applications of Finite Element Analysis," 4th Edition, 2001.

K. J. Bathe, "Finite Element Procedures," 2007.

5.4.2.3 How Can One Leverage Element Quality to Create Higher Quality Analyses?

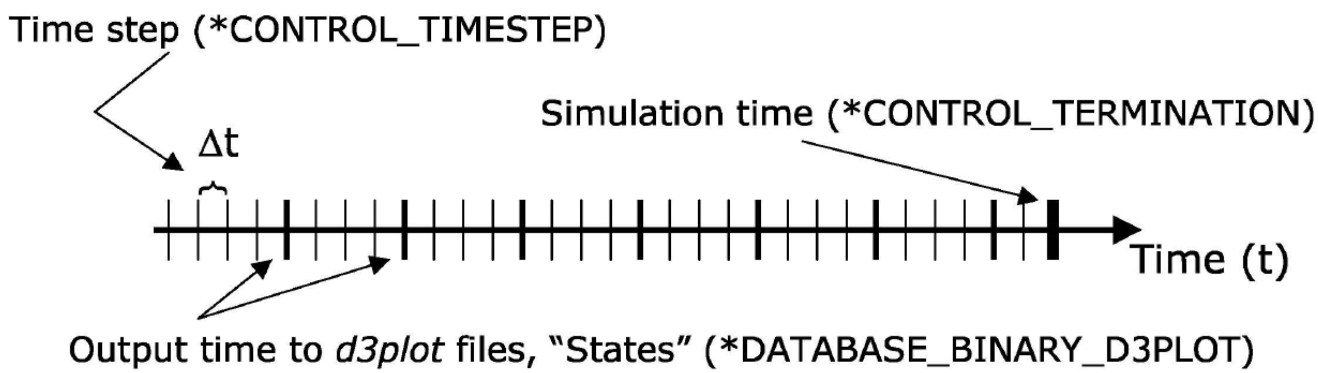
An efficient measure of a element's quality is it's Jacobian since it mathematically describes the transformation of the element's global coordinate system into a set of normalized coordinates whether in 2D or 3D. Contouring of the Jacobian can be readily done in most FEA pre-processors. Please note that the Jacobian quality check can't help you if the element is warped. Thus, a warping check should be included when working with shells elements that are meshed on curved surfaces. Below are some simple examples; however, there is not a 1 to 1 connection between a Jacobian value and the explicit time step (see LS-DYNA Theory Manual for explanation). At the end of the day, it is Jacobian (first) AND Explicit Time Step (second) AND Warping (last).



5.5 Summary of Explicit Time Integration

- Very efficient for large nonlinear problems (CPU time increases only linearly with DOF)
- No need to assemble stiffness matrix or solve system of equations (aka, implicit)
- Cost per time step is very low
- Stable time step size is limited by CFL criterion (i.e., time for stress wave to traverse an element)
- Problem duration typically ranges from microseconds to tenths of seconds
- Particularly well-suited to nonlinear, high-rate dynamic problems
- Nonlinear contact/impact
- Nonlinear materials
- Finite strains/large deformations

Some LS-DYNA explicit terminology that helps explain the relationship between these Keywords:



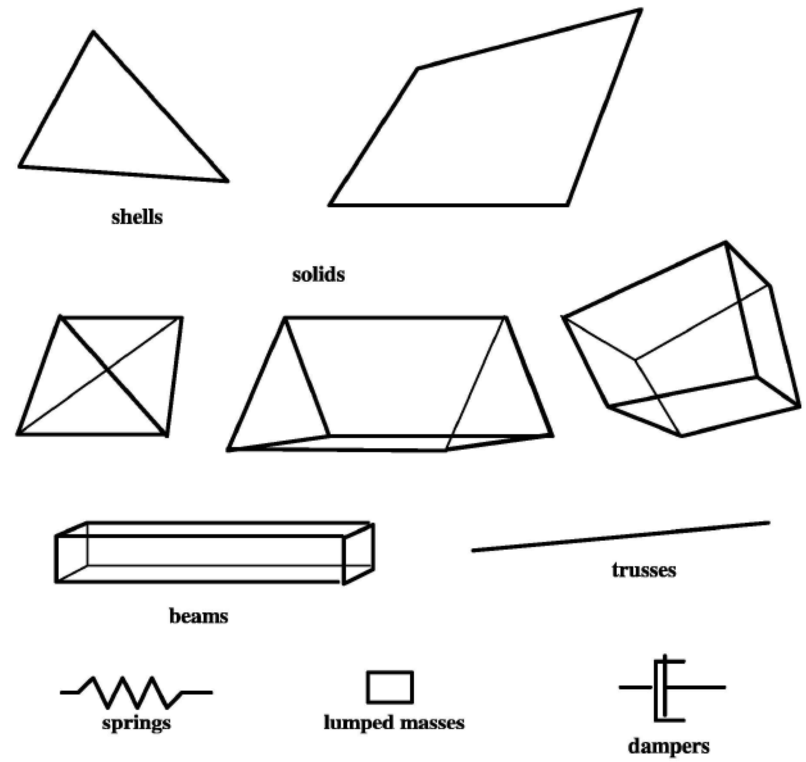
6. EXPLICIT ELEMENT TECHNOLOGY

6.1 Element Types in LS-DYNA

Element Toolbox:

If it numerically exists, then LS-DYNA most likely has it:

- Point elements (mass, inertia)
- Discrete elements (springs, dampers)
- Beams, cables, discrete-beams, etc.
- Solids (20 and 3D, Lagrangian, Eulerian, ALE)
- Shells
- Thick Shells (8 node)
- Cohesive elements
- Seatbelts (and related components)
- EFG and SPH (meshless methods)



Extremely Brief Recommendations:

- Hughes-Liu Integrated Beam, *elform*=1, is default. Stresses are calculated at the mid-span of the beam. Special requirements for stress output. Only your imagination limits the type of cross-sections available for beams using the *INTEGRATION_BEAM option. Since LS-DYNA is designed for nonlinear mechanics, beams require integration and care should be taken. More on beam element theory and modeling is provided in the Implicit Section of these notes.
- For solid elements, the default is *elform*=1 and uses one-point Gaussian Integration (constant) stress. This element is excellent for very large deformations. It is the standard recommend for explicit simulations.
- Shell elements are covered in detail.

Detailed Element Recommendations (see Class Reference Notes)

{Elements / Solid Elements} Review of Solid Element Formulations Erhart.pdf

{Aerospace Working Group} - AWG LS-DYNA Modeling Guidelines MGD_v22-1 (June 20, 2022).pdf

Analyst's Note: Beam elements represent the highest form of idealization and offer the most opportunities for optimization due to their ease of shape modification (i.e., cross-section modification). LS-DYNA is extremely powerful and can model all standard beams through the linear and nonlinear regime. However care must be taken to understand how stresses are calculated at integration points and that contact is with a cylindrical representation.

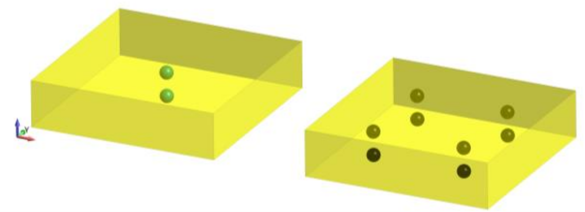
6.2 One Gaussian Point Isoparametric Shell Elements and Hourglassing

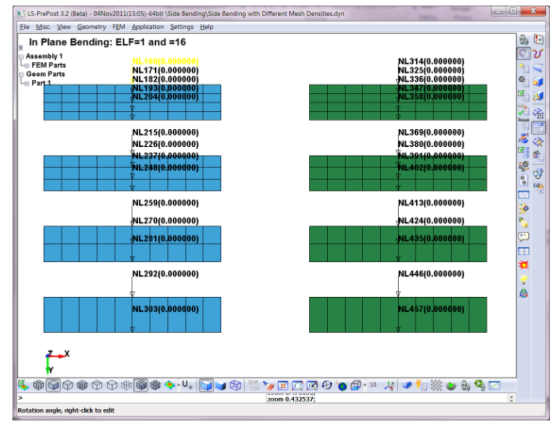
6.2.1 Instructor Led Workshop: 4 - Explicit Element Technology | A: Side Bending

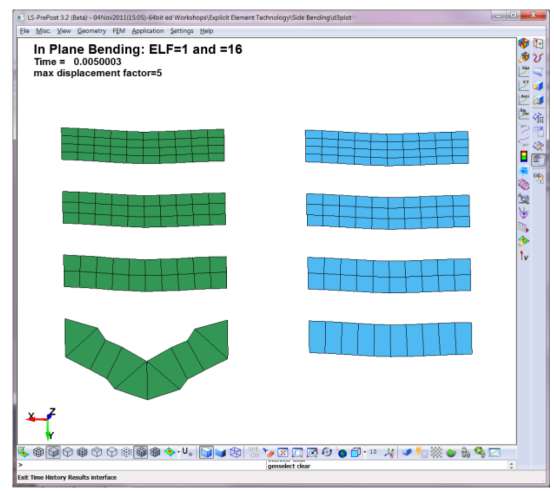
Isoparametric Shell Elements

Default element is **one** Gauss point in-plane (*elform*=2)

- This default formulation is efficient and generally the most robust formulation for large deformations.
- The example shows that under-integrated elements have severe problems in bending. The recommended number of through thickness elements is three (3). However, fully integrated (*elform*=-16) does an adequate job with one or two. Computationally 3x more expensive than the default formulation (*elform*=2).
- Importantly, it is not always possible to use only *elform*=-16 due to computational expense and care must be taken with using the default formulation in situations where only one element through thickness is possible.
- Increasing the number of elements can be problematic due the CFL timestep condition since three elements over a narrow width of strip will always cause a severe reduction in timestep.
- Recommended size is 5 mm for steel and aluminum and thus yields a time step of approximately 1 μs.







Acknowledgement: This section courtesy of LSTC and Paul Du Bois, Hermes Engineering NV

A Mechanics Observation on In-Plane and Out-of-Plane (Through Thickness Integration

For shell elements, the stiffness of the element is calculated based on the in-plane integration points. If the element is perfect (i.e., 1 x 1), then the stiffness calculation is exact for 1-point and for 4-point Gauss integration. Given that perfectly shaped shell elements are rare, the reality is that 4-point integration does a much better job given randomly shaped elements. The out-of-plane integration points or the through-thickness planes of Gauss integration only serve for the calculation of plastic strain. An example of how the through-thickness integration planes effect the calculation of a plate under bending given material plasticity is shown on the next page. The key observation is that in-plane gives you the in-plane stiffness calculation (K) while through-thickness captures the plastic strain behavior or one might say, the out-of-plane stiffness behavior if and only if plastic strain occurs.

6.2.2 Instructor Led Workshop: 4 - Explicit Element Technology | B: Out-of-Plane Bending with Plasticity

Isoparametric Shell Elements*

- Only one formulation is recommended: *elform*=-16.
- Number of through-thickness integration points (*nip*) controlled by user:

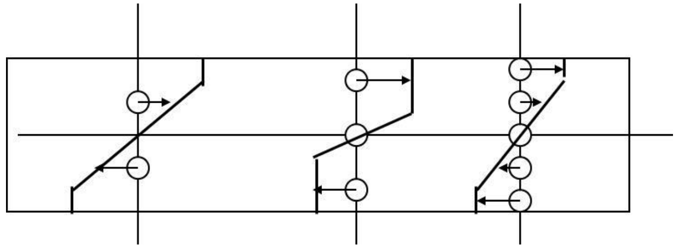
nip 1: Membrane Behavior

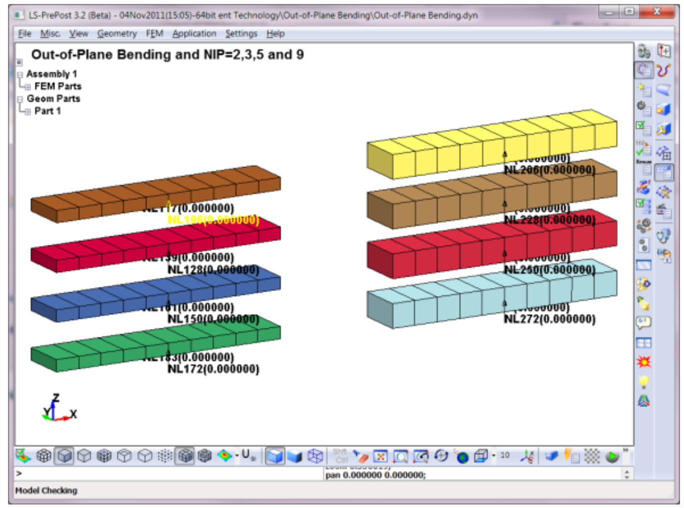
nip 2: *Barely Adequate* (default)

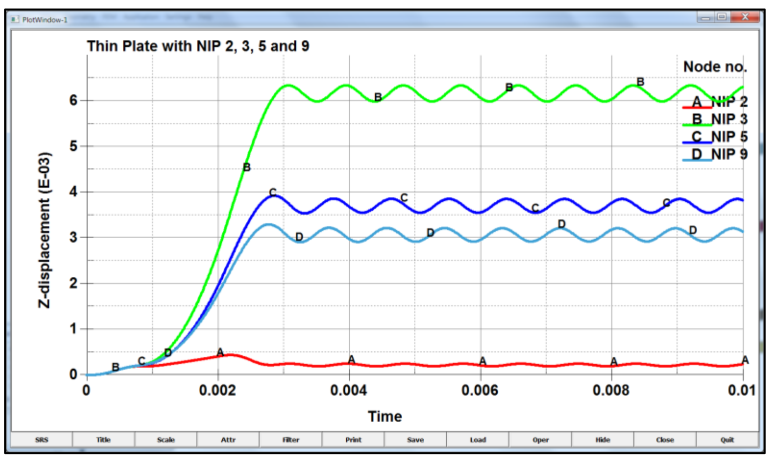
nip 5: Recommended for Nonlinear Materials

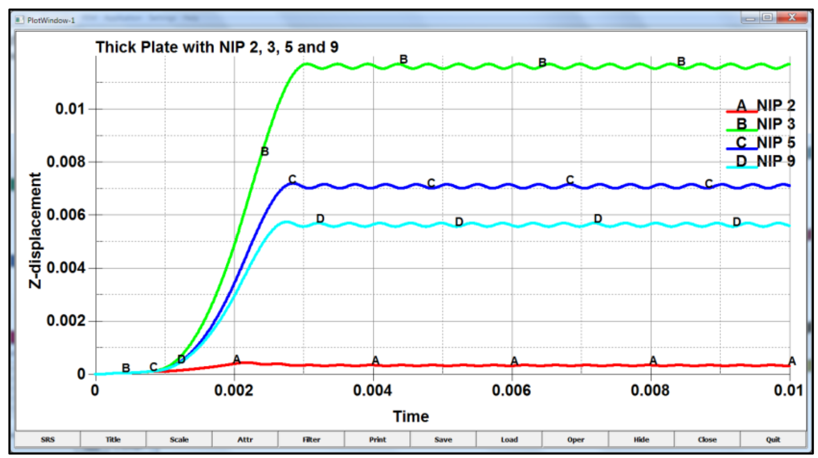
Recommend *elform* and *nip* for Nonlinear Plasticity = 5

- *elform*=-16 with *nip*=5

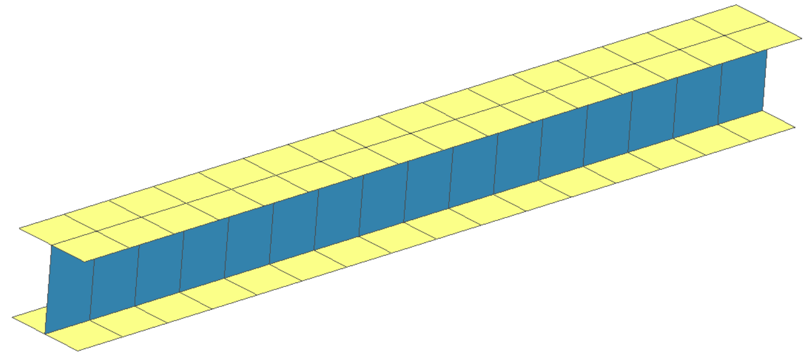








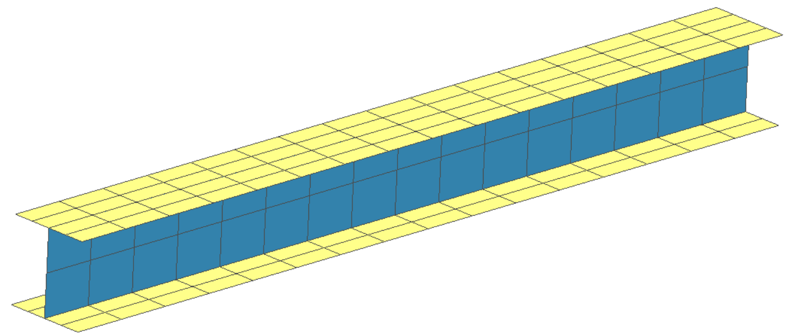
6.2.3 Workshop: 3 - Building the Better Beam



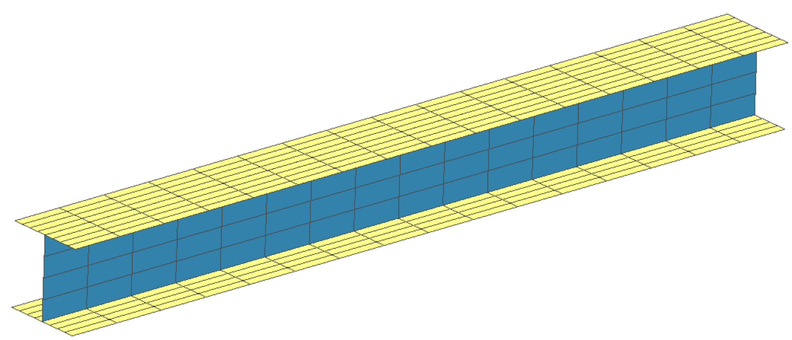
Objective: The importance of mesh density and plate out-of-plane integration will be demonstrated through the use of this simply-supported I-beam model (half-symmetry). The material model is steel with a yield stress of 100,000 psi and a tangent modulus of 200,000 psi. The workshop will start with a course model and then modify the mesh density and then finally change the element formulation (*elform*) from under-integrated (*elform* = 2) to fully-integrated (*elform* = -16). The results are surprising, and one has to think about the element shape function (linear) and how element integration calculates the stiffness of the element.

Tasks:

- Open / LSPP / Start / Building the better Beam – Start.dyn in LSPP. Inspect the model and note that it is using shell elements with *elform*=2 (see *SECTION_SHELL keyword command). Run the model and measure the maximum displacement at the end of the beam and record this value on the table below.
- Then, repeat this exercise for the three other models within the folder. Note that each time you run the model in the same folder it will overwrite the existing d3plot files. Record the end displacements for the 2x and 4x mesh refinement models.
- Now, edit the Keyword deck – Start and change its *elform*=-16. Rerun the model and note the end displacement. Do the same for the other two models (2x and 4x).



		Maximum Displacement	
Model	**Mesh Density**	***elform* = 2**	***elform* = -16**
Start	1x	-57.5	-2.7
Refine 2x	2x	-3.1	-2.9
Refine 4x	4x	-3.0	-2.8

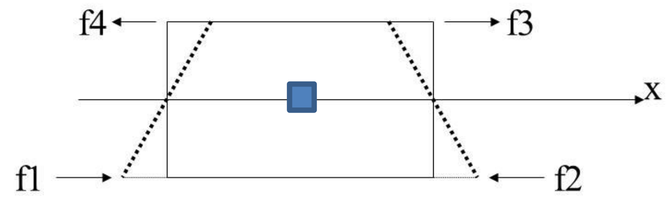


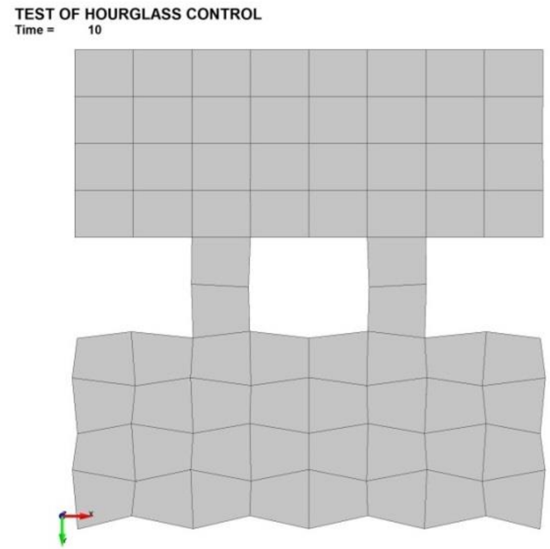
Analyst's Note: It can be surprising to see how the results move around but the mechanics are what they are given the mesh density and the problem physics.

6.2.4 Workshop: 4 - Hourglass Control/Hourglass

Isoparametric[1] Shell/Solid Elements and Hourglassing

- All under-integrated isoparametric elements (one Gauss point) have hourglassing present. It is a non-physical "zero-energy" mode of deformation.
- Fully Integrated formulations do not hourglass. Additionally, tetrahedron and triangular elements do not hourglass but are overly stiff in many applications.
- *CONTROL_HOURGLASS or *HOURGLASS to set hourglass control.
- Use default unless additional documentation is consulted (e.g., see Review of Solid Element Formulations Erhart.pdf (Class Reference Notes / Solid Elements).
- Hourglass energy should be less than 5% of the internal energy at any stage of the analysis (use *CONTROL_ENERGY (*hgen*=2) to calculate hourglass energy).
- In LSPP, check glstat for total hourglass energy and then matsum for individual part energy.
- For most applications, *hgen*=4 with *qh*=0.03 (see Class Reference Notes / Hourglass).





How to Limit Hourglassing

- Apply pressures instead of point loads.
- Refine mesh
- Selectively use *elform*=-16 (3x computational cost)

Workshop Tasks:

- Evaluate current model for hourglassing. Plot internal energy and hourglass energy.
- Read Hourglass Material.
- Attempt fix with different hourglass type.
- Switch *elform* to -16.

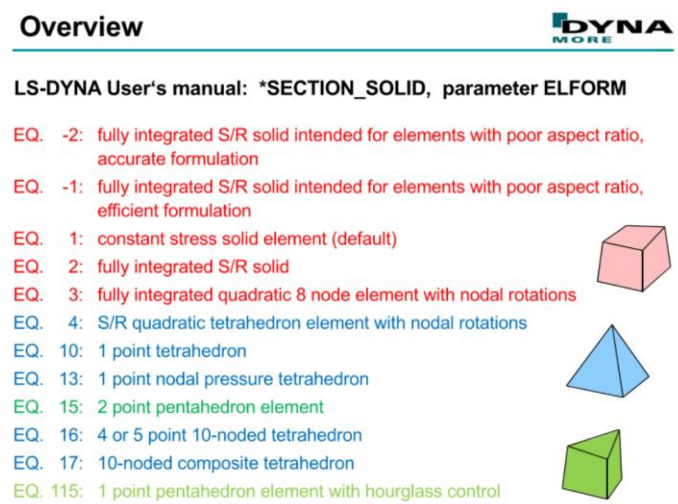
[1]*History Note: According to Ed Wilson (see www.edwilson.org) "The introduction of the isoparametric element formulation by Bruce Irons in 1968 was the single most significant contribution to the field of finite element analysis during the past 40 years."*

6.3 Workshop: 5 – Solid Element Technology – Hex and Tet Formulations

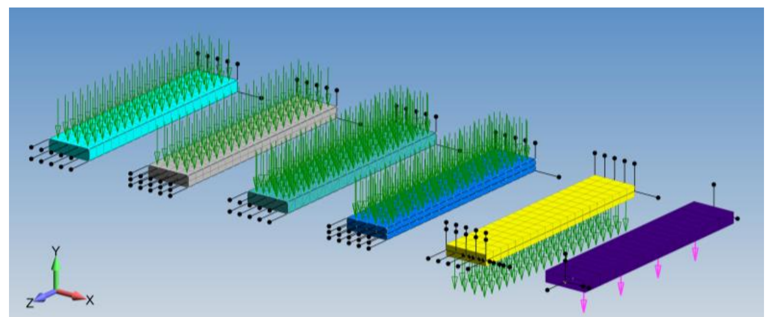
Objective: Be knowledgeable in your selection of solid element formulation (*elform*) whether brick or tetrahedron and its integration scheme.

Introduction: A simple way to become confident in your *elform* selection is to read and then build simple models of the behavior you would like to explore. The model simulates a simple supported beam under uniform loading. We have shell, hex and tet elements. Explore what can be done by simply changing the *elform*. The analytical solution for this model is:

$$\delta_{max} = \frac{5 \cdot \omega \cdot L^4}{384 \cdot E \cdot I} = \frac{5 \cdot 10 \cdot 10^4}{384 \cdot 1e7 \cdot 0.0013021} = 0.10\ inch$$



This paper is in Class Reference Notes / Elements / Solid Elements / Review of Solid Element Formulations Erhart.pdf



Tasks:

- Inspect the starting model in LSPP (Workshop - Solid Element Technology – Start.dyn). We have hex, tet, shell and beam elements);
- Look at the *elform* for each element type and look at the Manual under *SECTION to confirm your understanding of their behavior;
- Then record the end displacements for the models using:
- Run 1 – Hex: *elform* = 1; Tet: *elform* = 10
- Run 2 - Hex: *elform* = -18; Tet: *elform* = 13
- Run 3 – Hex: *elform* = 2; Shell: *nip* = 1

Please note that one should pick the lower, right-hand corner node of the solid and shell elements and, since the last one is a beam element, just the end node. If picked per the video, one will notice that the node numbers on the plot will be 1 to 6 going from left to right as picked on the model.

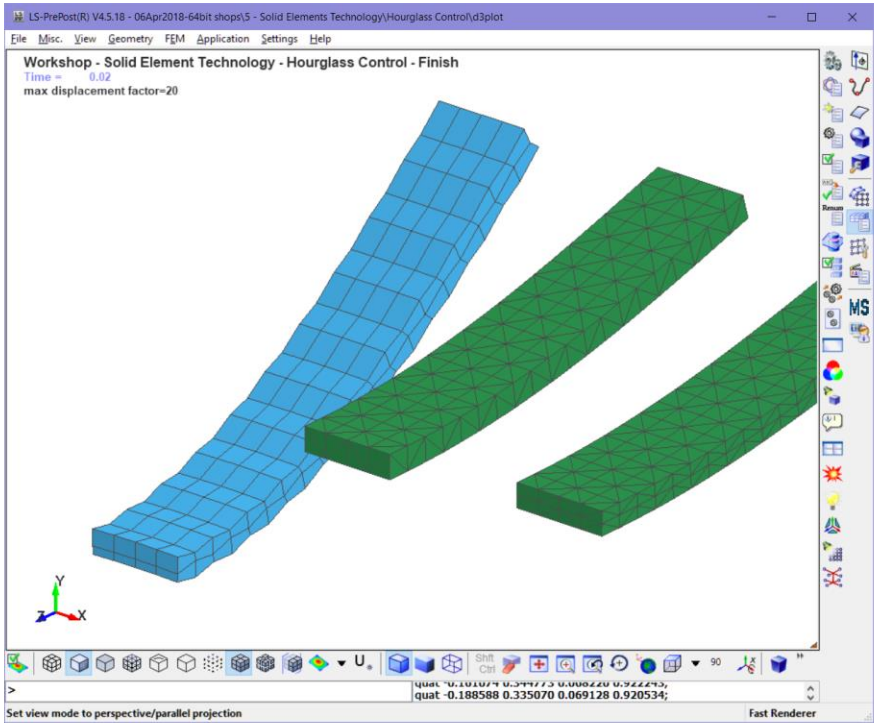
Run	Description	Hex – One	Hex - Two	Tet - One	Tet - Two	Shell	Beam
1	Hex: *elform* = 1 Tet: *elform* = 10						
2	Hex: *elform* = -18 Tet: *elform* = 13					***ditto***	***ditto***
3	Hex: elform = 2 Shell: *nip* = 1			***ditto***	***ditto***		***ditto***

6.3.1 Workshop 5 – Solid Element Technology – Hourglass Control

Objective: Leveraging the prior workshop on Hourglass Control, we gain a better understanding on how hourglassing works on standard explicit models

Introduction: When using under-integrated elements, hourglassing is real. A little side note is that hourglassing doesn't exist for tetrahedral elements (RTM). We didn't cover this at the beginning of the Workshop to keep the information flow manageable. The image on the left shows Run 1 with the displacement scaled by 20x (Please note that the one-layer hex model has flown off the screen at 20x!).

Your job is to add Hourglass Control and Re-Analyze.



Tasks:

- Run Start model in file folder Hourglass Control. Scale model by 20x.
- Edit deck and add Hourglass Control of *Ihq* =4 and *qh* = 0.1 and also enable the calculation of hourglass energy (*hgen* = 2). One will also note that _GLSTAT has been set to write out results.
- Run model and note results. Check Internal vs Hourglass Energy
- Re-run model with *qh* = 0.01

Run	Description	Hex – One	Hex - Two	Hourglass Energy Acceptable?
1	*Ihq* = 4 / *qh* = 0.1			
2	*Ihq* = 4 / *qh* = 0.01			

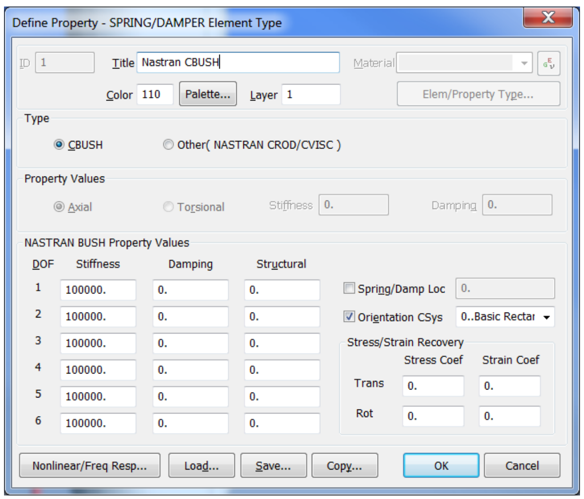
Analyst's Note: If you have time take a look at Pathology of elform 10 and also Implicit file folders. There are notes within each file folder as to what is what. If it doesn't make sense, ask me a few questions. If you have even more time, take a look at the 10-Node tetrahedral folder.

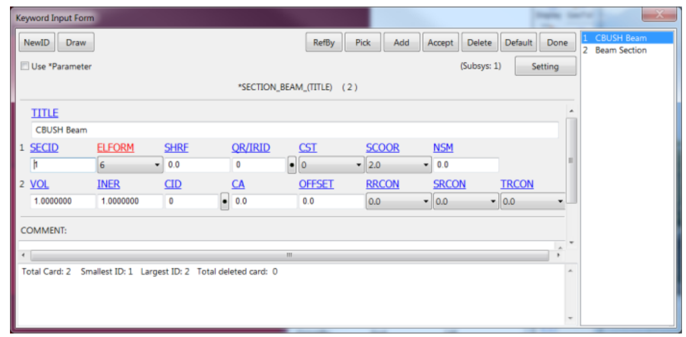
6.4 Scalar Elements (e.g., Nastran CBUSH) or LS-DYNA "Discrete Beam"

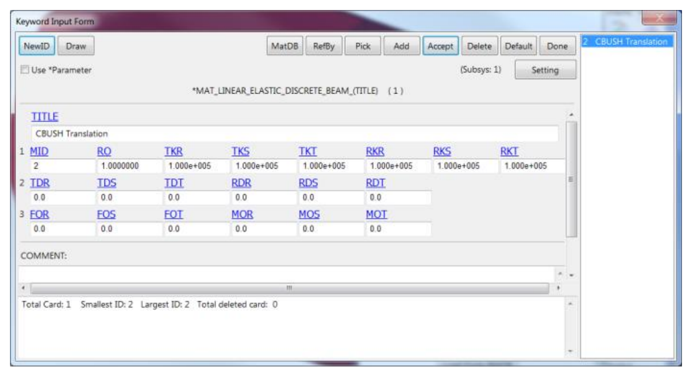
The Nastran CBUSH element is of such general utility that it merits its own special section on how to obtain equivalent behavior within LS-DYNA. Our first meritorious concept is that an explicit analysis always requires *mass* whereas a static implicit analysis only requires stiffness. Hence, in a static analysis, one can have elements with zero length (i.e., zero volume or zero mass) whereas in an explicit analysis *mass* must be present. In LS-DYNA, discrete beam mass is set by the material density (material card) and then within the *SECTION card under *vol* and *iner*. A very nice write-up can be found at DYNASupport.com (How To's / Discrete Beam).

Although LS-DYNA has several methodologies to arrive at simulating the behavior of Nastran CBUSH element (e.g., *ELEMENT_DISCRETE), we will present the most basic method and the one recommended by LST noted as "Discrete Beam". In Nastran, the CBUSH property card specifies orientation and stiffness. In LS-DYNA, these capabilities are handled by two cards: (i) *ELEMENT_SECTION, *elform*=6 (orientation) and (ii) *MAT_LINEAR_ELASTIC_BEAM or *MAT_066 (stiffness values).

Nastran CBUSH	LS-DYNA Equivalent *SECTION_BEAM & *MAT_66







If the connection has zero length, then the *SECTION_BEAM (*elform*=6) variable *scoor* must be set to a value corresponding to the desired direction of the beam. In other words, take a look at the manual and www.dynasupport.com under discrete beams since the choice of *scoor* is not obvious (or just click on this link: https://www.dynasupport.com/howtos/element/discrete-beam). Our general default is *scoor*=2 for beams of finite length and *scoor*=0 for zero length. The reality is that one should read the notes on *scoor* and do your research. **The short version is that *scoor*=2 aligns the beam's r-axis along its two nodes while *scoor*=0 sets it to the CID (e.g., when CID=0, the beams initial coordinate system is aligned to the global coordinate system, where r = X, s = Y and t = Z).**

All output results are in the beam's local (r, s, t) coordinate system.

The "mass of the element" (as we mentioned – in an explicit analysis one has to have "mass") is calculated by the product of the variables *vol* x *density* (ρ). If moments (i.e., torque) is to be captured, then an additional rotational mass must be specified using the variable *iner*. Importantly, the length of the element is not used to calculate the mass of the element but simply the *vol* and *density* variables of the material card. Whereas the *iner* variable is the mass moment of inertia about each of its three axes. A non-zero *iner* variable entry is ***required*** to activate any of the rotational DOF.

This element is extremely powerful since it is a fully 3-D, nonlinear spring that can simulate a lot of different actions. Thus, at first glance it may seem overwhelming complex but there are good reasons for its complexity.

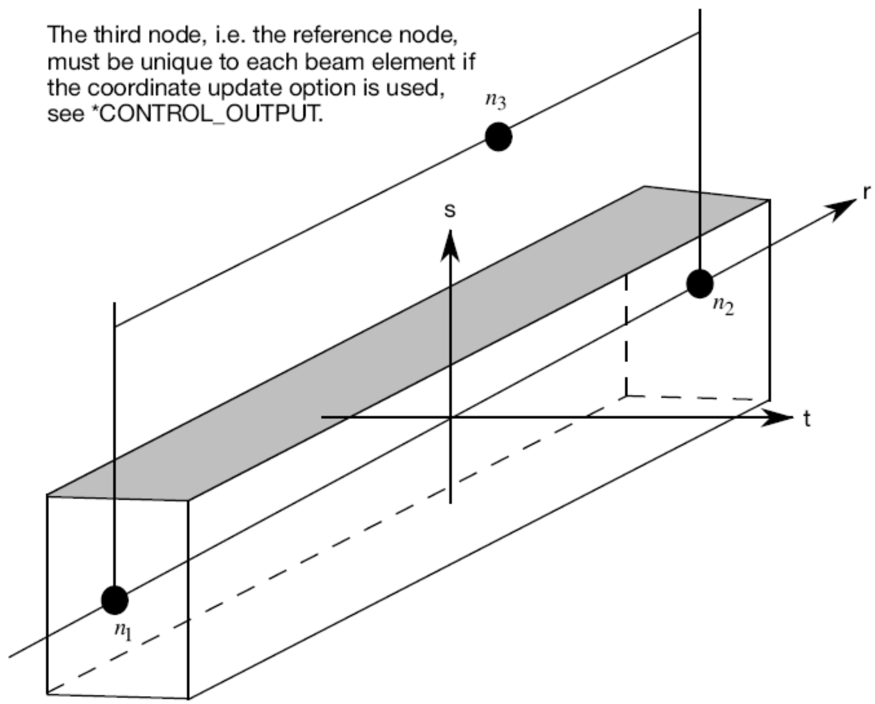
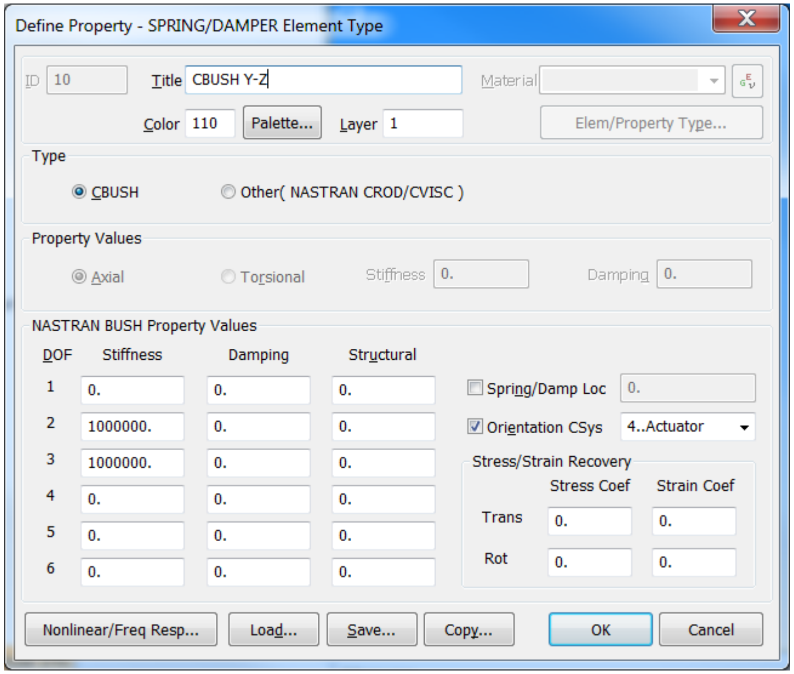


Figure 17-1. LS-DYNA beam elements. Node n_3 determines the initial orientation of the cross section.

A basic element is given below for a zero length "CBUSH" element coordinate with a user defined coordinate system:

Nastran CBUSH Property Card



Equivalent LS-DYNA Keywords

```
*SECTION_BEAM_TITLE
CBUSH Equivalent Element
$#  secid   elform    shrf  qr/irid     cst    scoor     nsm
        1       6  1.000000       2       2  3.000000   0.000
$#    vol    iner     cid      ca  offset   rrcon   srcon   trcon
 1.000000 1.000000      1   0.000   0.000   0.000   0.000   0.000
*ELEMENT_BEAM
$#  eid   pid    n1    n2    n3   rt1   rr1   rt2   rr2  local
      1     1   400   401     0     0     0     0     0      2
*DEFINE_COORDINATE_SYSTEM
4,  488.999,  -28.99994,  414.4192,  -105.6686,  -180.7164,  227.1846
  670.4222,  17.28623,  -199.297
*MAT_LINEAR_ELASTIC_DISCRETE_BEAM_TITLE
Spring Stiffnesses
$#    mid       ro      tkr      tks      tkt      rkr      rks      rkt
        1  0.001000   0.000 1.0000E+6 1.0000E+6   0.000   0.000   0.000
$#    tdr      tds      tdt      rdr      rds      rdt
    0.000    0.000    0.000    0.000    0.000    0.000
$#    for      fos      fot      mor      mos      mot
    0.000    0.000    0.000    0.000    0.000    0.000
```

Some things to note is that a third node is not defined within the *ELEMENT_BEAM card since the orientation of the element is handled by the CID definition. One will note that VOL and INER are both given values of 1 for simplicity since the mass of the element is then controlled by just the mass density on the material card. And, don't forget that a *PART card is also required to tie together the *SECTION and *MAT cards.

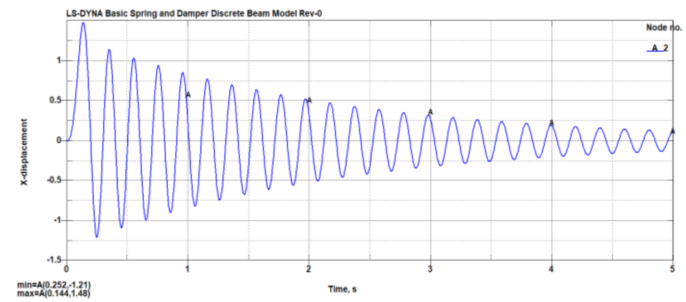
6.4.1 Workshop 6 - Discrete Beam (Spring Away)

Objective: Build a discrete beam (spring) and mass LS-DYNA model using the supplied file. Apply a transient load and then plot its damped response. The model represents a spring with a large mass (10 units) at its end (attached to node=2).

Why: There is nothing like building a model by "hand" to get the idea of how LS-DYNA works. There is no video since it would just be a crutch that you don't need.

Tasks:

- Open text file Discrete Beam (Spring Away) - Start.dyn and inspect the Keywords. They are left blank but provide the structure for you to fill in your own values to create a working model.
- Create a spring element having a unit length in the x-direction with node 1 at the origin (0,0,0) and node 2 at (1,0,0).
- Create the *ELEMENT_BEAM using nodes 1 and 2 and likewise the *ELEMENT_MASS at node 2. The *mass*=10.0.
- Use *MAT_LINEAR_ELASTIC_DISCRETE_BEAM to define a spring constant in the x-direction of 10,000 (*tkr*) and a damping constant of 10.0 (*tdr*) in the same direction. For density, use a value of 1.0 (*ro*).
- The element type is *SECTION_BEAM, with its element formulation as discrete beam/cable (*elform*=6) and *vol*=1.0. Given that we have a finite length beam (1 unit), we can leave *scoor* as default.
- To tie everything together, we need to define the *PART Keyword as *pid*=1, *secid*=1 and *mid*=1.
- Load is *LOAD_NODE_POINT (*nid*=2) with the direction (*dofx*=1) in the x-direction. The load curve is 1 (*lcid*=1). Then scale the load by 10,000 (*sf*=10000 or 1E4).
- The constraint (*BOUNDARY_SPC_NODE) is at the origin (*nid*=1) and only the x-direction needs to be constrained (*dofx*=1).
- Data output is every 0.001 time unit (*dt*=0.001) (*DATABASE_BINARY_D3PLOT).
- The model runs for 5 seconds (*endtime*=5) (*CONTROL_TERMINATION)
- Once the model runs, you should see a nice sinusoidally damped curve (i.e., History / Node (#2) / X-Displacement / Plot).
- What about the explicit time step? What will happen if you increase density by 10x? Can you predict the time step change?



Observations:

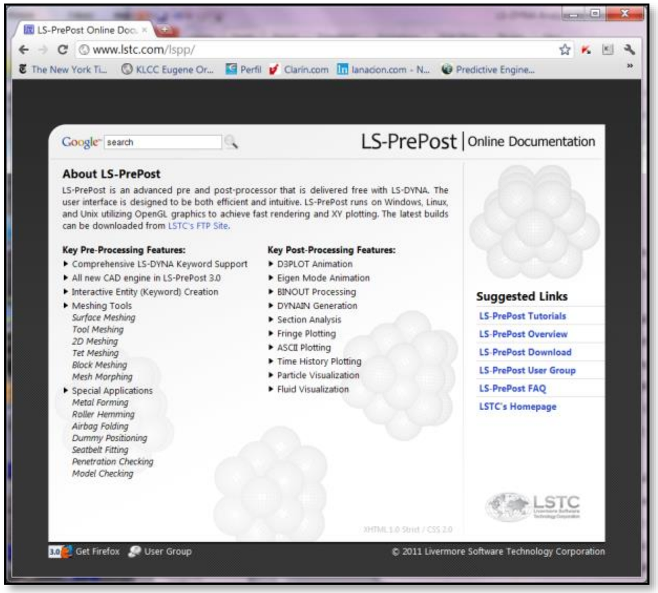
- This example was made purposively simple and since CID=0 (default), the springs material axis align themselves to the global coordinate system or in other words r, s, t = x, y, z. One could make the spring zero length and the same results would be forthcoming.
- Once you step away from this most basic setup, one needs to understand how to setup your own coordinate systems or if using finite length beams, employ a third node to orient the beam and likewise its material coordinate system (an example is given in the Workshop file folder under Third Node and note the use of *scoor*).
- If the beam is of zero length and one wants a non-global coordinate system, then you have to define a local coordinate system and apply the appropriate *scoor* value.

7. LS-PREPOST

7.1 Workshop: 7 - LS-PrePost | Workshop 6 (partial Execution)

Introduction to LS-PrePost (LSPP):

Reference and tutorial materials are provided at the LST site referenced in Section 4.4). Model manipulation is by Shift Key and the left, right and middle mouse button.

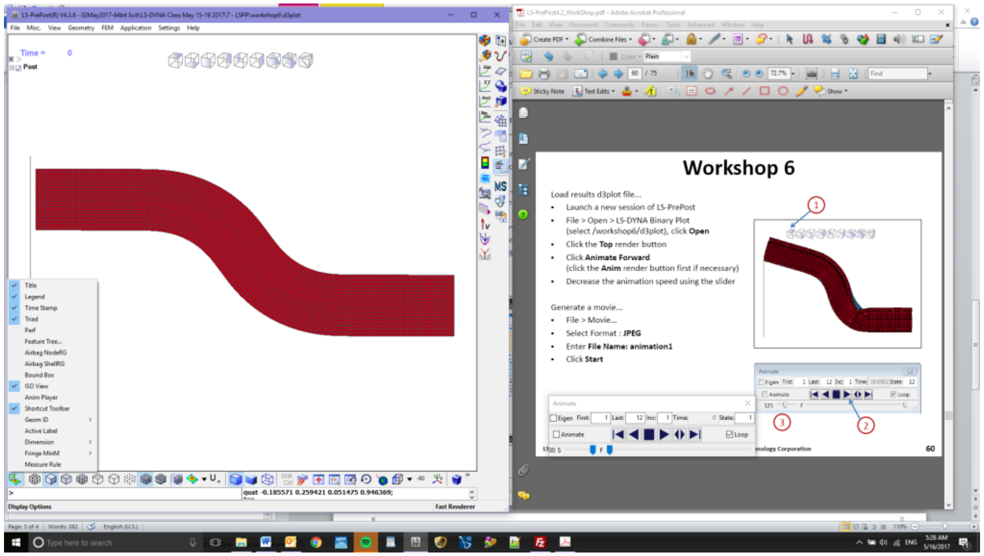


Class Referenced Note's Section:

The detailed usage of LS-PrePost in its own rights is a two-day class (see Class Reference Notes / LS-PrePost Documentation.

Workshop Goal: To Get Comfortable with LSPP Post-Processing:

- Open LS-PrePost_Intro_WorkShop - 6 - Post-Processing.pdf;
- Execute steps.
- Please note that on page 7, the dialog screens do not match the current LSPP version – {for now, the instructor will do this example (page 7) in class}.



8. MATERIAL MODELING

Material modeling is broken down into four parts: (I) Metals; (II) Plastics, Elastomers and Foams; (III) Composites and (IV) Equation-of-State (EOS). These are brief overviews but provide the basics to get you started. As would be expected, the Class Reference Notes provides our own internal references nots on material modeling.

8.1 Basic Review of Material Models Available in LS-DYNA

The broad array of material models in LS-DYNA can be over-whelming. In the material manual they are listed in numerical order based on their insertion into the code. Hence, the elastic material model (*MAT_001) was the first material model developed. These earlier models are well validated since they have been used extensively over the years. Later material models, e.g., *MAT_181 Simplified Rubber, was developed in early 2000, and although a somewhat recent development, it has seen wide-spread usage due to its advanced formulation and robustness. It is recommended for any hyperelastic material model since it provides an exact match to the experimental data.

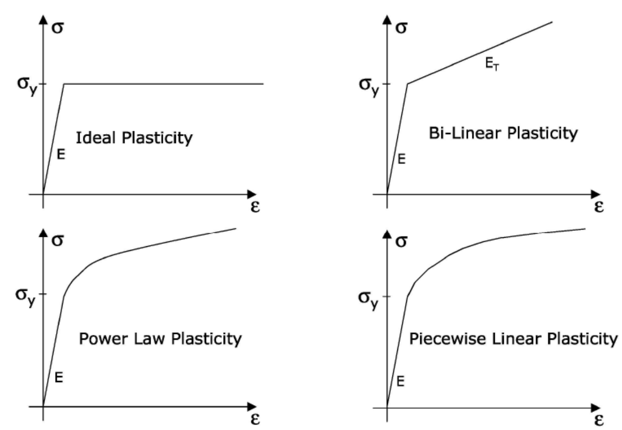
8.1.1 So Many Material Models and So Many Questions

In the LS-DYNA Keyword User's Manual – Volume II Material Models, at the start of the section *MAT, provides a listing of *what* material models can be used *with* what element types (beam, shell, solid, springs, etc.). After this section is MATERIAL MODEL REFERENCE TABLES provides a listing of *what* material models can capture *what* physics. Although most material models (90%) can be used for implicit there are a few surprises.

It is good practice to use as few as material models as possible such that one develops expertise in these models. For example, although many material models exist for metals, one of the most robust models is *MAT_024 or *MAT_PIECEWISE_LINEAR_PLASTICITY. This material model is the standard workhorse and is the recommended starting point for elastic-plastic simulation of metals and general plastics since it can also handle viscoelastic behavior (i.e., strain-rate dependency).

Major Categories

- Elastic (*MAT_001)
- Elastic-Plastic (*MAT_024)
- Rigid (*MAT_020)
- Orthotropic/Anisotropic
- Hyperelastic (*MAT_181)
- Foams (*MAT_083)
- Composites (*MAT_054)
- Viscoelastic
- Heart/Lung/Tissue
- Acoustic material
- Fabric
- Concrete/Soil
- High Explosives
- Laminated Glass
- User-defined



Analyst's Note: It is a recommended practice to build a pilot model (i.e., virtual test coupon using beam, shell or solid elements) using the material law of choice and replicate the material's experimental data. One may think that it is a waste of time but the art of "model debugging" is knowing what you know with 100% certainty.

8.2 LS-DYNA Keyword User's Manual: Volume II – Material Models

If you haven't looked at this manual, now is the time. These are my take-aways from Volume 2:

Start Here

*MAT

*MAT

LS-DYNA has historically referenced each material model by a number. As shown below, a three digit numerical designation can still be used, e.g., *MAT_001, and is equivalent to a corresponding descriptive designation, e.g., *MAT_ELASTIC. The two equivalent commands for each material model, one numerical and the other descriptive, are listed below. The numbers in square brackets (see key below) identify the element formulations for which the material model is implemented. The number in the curly brackets, {n}, indicates the default number of history variables per element integration point that are stored in addition to the 7 history variables which are stored by default. Just as an example, for the type 16 fully integrated shell elements with 2 integration points through the thickness, the total number of history variables is $8 \times (n + 7)$. For the Belytschko-Tsay type 2 element the number is $2 \times (n + 7)$.

What Materials Can Be Used with What Elements?

Key to numbers in square brackets

0	-	Solids (and 2D continuum elements, that is, shell formulations 13, 14, 15)
1H	-	Hughes-Liu beam
1B	-	Belytschko resultant beam
1I	-	Belytschko integrated solid and tubular beams
1T	-	Truss
1D	-	Discrete beam
1SW	-	Spotweld beam
2	-	Shells
3a	-	Thick shell formulations 1, 2, 6
3c	-	Thick shell formulations 3, 5, 7
4	-	Special airbag element
5	-	SPH element (particle)
6	-	Acoustic solid
7	-	Cohesive solid
8A	-	Multi-material ALE solid (validated)

And This is Also Really Useful

*MAT MATERIAL MODEL REFERENCE TABLES

MATERIAL MODEL REFERENCE TABLES

The tables provided on the following pages list the material models, some of their attributes, and the general classes of physical materials to which the numerical models might be applied.

If a material model, without consideration of *MAT_ADD_EROSION, *MAT_ADD_-THERMAL_EXPANSION, *MAT_ADD_SOC_EXPANSION, *MAT_ADD_DAMAGE, *MAT_ADD_GENERALIZED_DAMAGE or *MAT_ADD_INELASTICITY, includes any of the following attributes, a "Y" will appear in the respective column of the table:

SRATE	-	**Strain-rate effects**
FAIL	-	**Failure criteria**
EOS	-	**Equation-of-State required for 3D solids and 2D continuum elements**
THERMAL	-	**Thermal effects**
ANISO	-	**Anisotropic/orthotropic**
DAM	-	**Damage effects**
TENS	-	**Tension handled differently than compression in some manner**

And Material Models by the Numbers

*MAT_001:	*MAT_ELASTIC [0,1H,1B,1I,1T,2,3a,3c,5,8A] {0}
*MAT_001_FLUID:	*MAT_ELASTIC_FLUID [0,8A] {0}
*MAT_002:	*MAT_*OPTION*TROPIC_ELASTIC [0,2,3a,3c] {15}
*MAT_003:	*MAT_PLASTIC_KINEMATIC [0,1H,1I,1T,2,3a,3c,5,8A] {5}
*MAT_004:	*MAT_ELASTIC_PLASTIC_THERMAL [0,1H,1T,2,3a,3c,5,8B] {3}
*MAT_005:	*MAT_SOIL_AND_FOAM [0,5,3c,8A] {0}
*MAT_006:	*MAT_VISCOELASTIC [0,1H,2,3a,3c,5,8B] {19}
*MAT_007:	*MAT_BLATZ-KO_RUBBER [0,2,3ac,8B] {9}
*MAT_008:	*MAT_HIGH_EXPLOSIVE_BURN [0,5,3c,8A] {4}
*MAT_009:	*MAT_NULL [0,1,2,3c,5,8A] {3}
*MAT_010:	*MAT_ELASTIC_PLASTIC_HYDRO_{OPTION} [0,3c,5,8B] {4}
*MAT_011:	*MAT_STEINBERG [0,3c,5,8B] {5}
*MAT_011_LUND:	*MAT_STEINBERG_LUND [0,3c,5,8B] {5}
*MAT_012:	*MAT_ISOTROPIC_ELASTIC_PLASTIC [0,2,3a,3c,5,8B] {0}
*MAT_013:	*MAT_ISOTROPIC_ELASTIC_FAILURE [0,3c,5,8B] {1}
*MAT_014:	*MAT_SOIL_AND_FOAM_FAILURE [0,3c,5,8B] {1}
*MAT_015:	*MAT_JOHNSON_COOK [0,2,3a,3c,5,8A] {6}
*MAT_016:	*MAT_PSEUDO_TENSOR [0,3c,5,8B] {6}

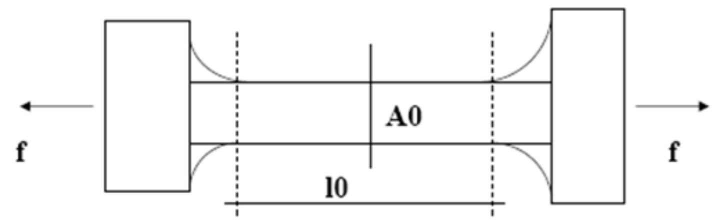
8.3 Part I: Metals

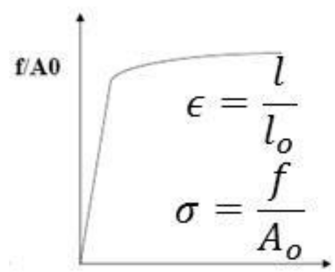
8.3.1 Engineering Stress-Strain vs True Stress-Strain

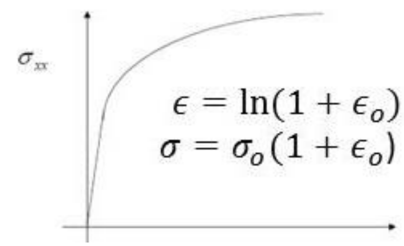
The constitutive large-strain modeling of all materials is based on the true stress-strain response of the material. Starting with a simple tensile test, the engineering stress strain is just the force over the original cross-sectional area of the coupon. The true stress-strain response accounts for the necking of the cross-section and can be elastically stated as shown on the graph on the right.

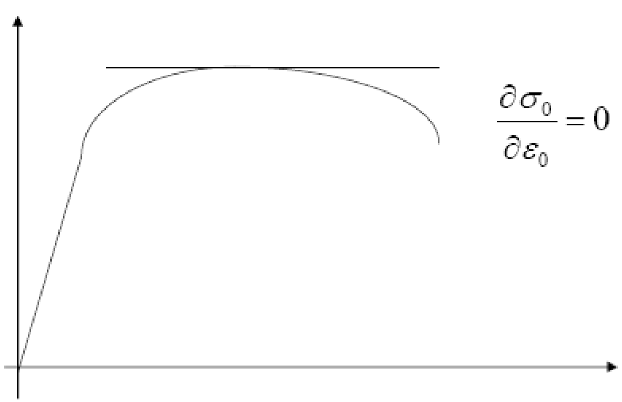
This method has its limitations and once the coupon starts to neck locally, this approach is no longer valid and an iterative approach must be used to calculate the true stress-strain response.

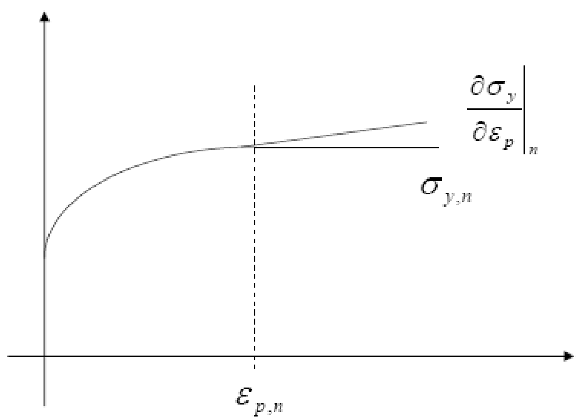
In many simulations, it is not exactly critical to have the necking response accurately characterized since once the material starts to neck we only have the end point data available and hence we draw a straight line from the initiation of necking to failure.











8.3.2 Material Failure and Experimental Correlation

Material Failure:

Simulation of material failure is a broad research avenue. Prediction of brittle material failure, as shown by the glass shatter patterns on the right, can be extremely mesh sensitive. For ductile materials, the failure prediction is more robust since ferrous and non-ferrous materials often have high-energy absorption characteristics prior to failure. That is to say, metallic materials generally tend to tear while brittle materials tend to snap.

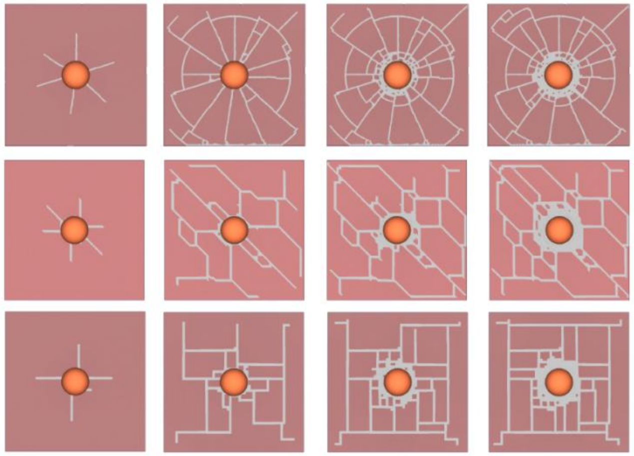
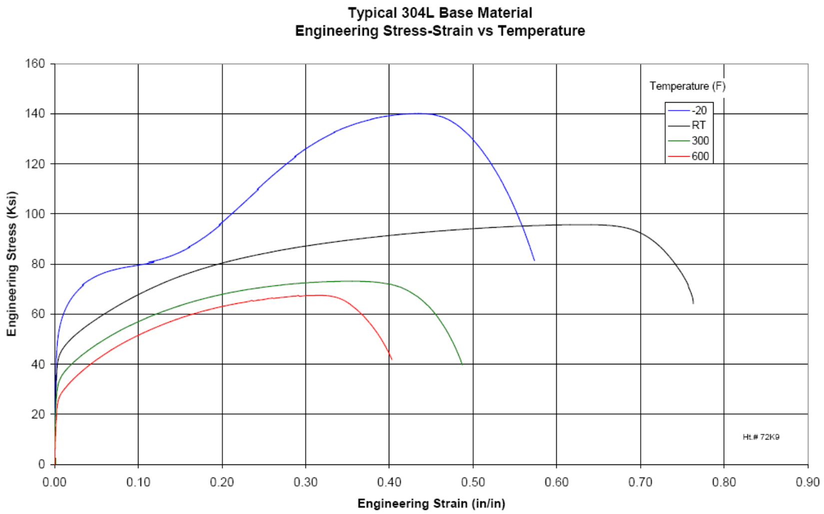


Image from article: A finite element model for impact simulation with laminated glass, Timmel, Kolling, Osterrieder, DuBois, IJIE, 2006.

Experimental Correlation:

The development of a material model often starts with stress-strain data from a standard mechanical test. This data can then be converted into true stress-strain and a very simplified approach used to extend this curve to failure.

An example that we will discuss is that for stainless steel which from an engineering stress-strain perspective appears to lose strength as it approaches its ultimate stress. What one should remember is that the mechanical test is pulling on the sample and then the force and elongation data are converted to stress and strain based on the original cross-section area and length of the sample, respectively.



8.4 Workshop: 8 - Elastic-Plastic Material Modeling (*MAT_024)

Objective: Convert engineering stress-strain to true stress-strain and then verify the material model using a small test coupon model.

Note: *We'll be using the research paper that can be found in the Workshop folder (see Tensile Stress-Strain…. By Blandford et al.). Starting with Figure 4 of the paper, we see the engineering stress-strain curves. From Table 2, the 304L material has a room temperature yield strength of 40 ksi and an ultimate strength of 95 ksi at a Total Strain of 75% and a Reduction in Area of 80%.*

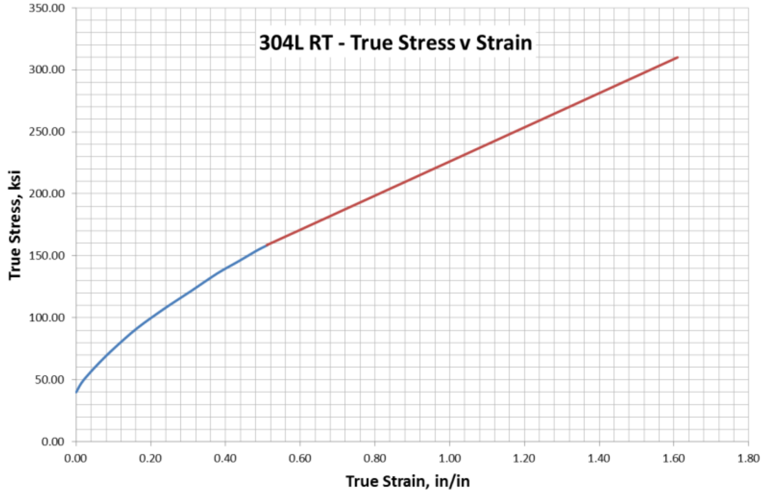
Tasks:

I.) The engineering stress-strain image has been discretized (courtesy of http://automeris.io/WebPlotDigitizer/ and the starting point for this workshop is: Elastic-Plastic Material….xlsx (a spreadsheet located in the Workshop file folder). With this data, create a true stress v strain curve. Please note that the final fracture strength is calculated by the engr. stress at fracture (62 ksi) and the final area of the necked specimen (*Note: since we have a reduction in area of 80%, we can calculate the final true stress as 62 ksi / (1-0.80) = 310 ksi*). Per LS-DYNA requirements, the true stress v strain curve starts at a plastic strain of 0.00 with the stress at the yield stress (40 ksi). If one gets stuck, a finalized spreadsheet is provided in the file folder Finish.

II.) With the true stress v strain data in hand, export in a CSV format suitable for LSPP (first column strain, second column stress).

III.) Don't forget to enter the failure strain (*fail*) from the spreadsheet.

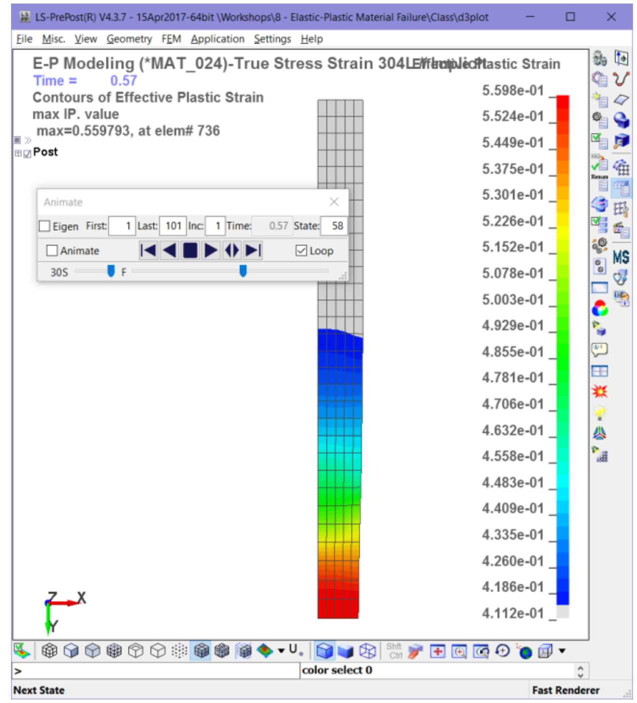
Figure 3. 'Cup and Cone' Type Failure



Workshop: 8 – Elastic-Plastic Material Modeling (*MAT_024) – (continued)

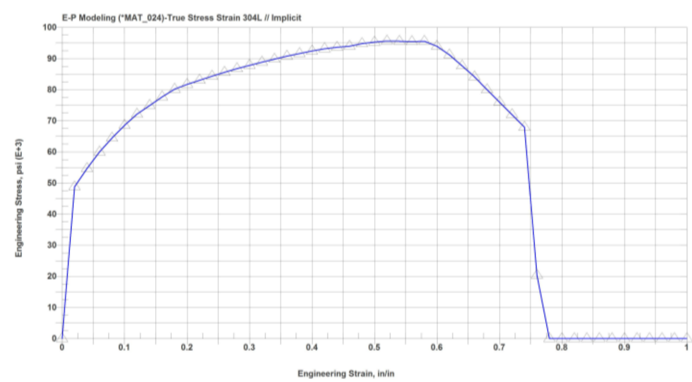
IV.) Open LSPP file in your favorite text editor: Elastic-Plastic Material Modeling (MAT_024) - Start - Implicit.dyn and paste in the true stress-strain data. This data will be inserted as *DEFINED_CURVE with *lcid*=2. Some formatting may be required.

V.) With the curve imported, fill out the material Keyword with the required values (only two are needed *fail* (1.6) and *lcss (2)*). The fail strain can be calculated per the procedure given in the research paper and if you have doubts, this value is also provided in the spreadsheet.



Note: The model is simple bar that is given a fixed displacement over time. At time=1.0, the bar has an engineering strain of 100%. The bar is slightly necked down in the middle to initiate necking.his is also experimental practice for the mechanical testing of pull coupons. The model is run using the implicit method. The file is commented to provide some background.

VI.) Once the analysis has finished, plot the resultant SPC force and compare with the engineering stress-strain graph from the paper. If all has been done correctly, it should correlate well. One can also do a ***cross-plot*** of stress v strain and this should compare nicely with the true stress-strain graph. However, keep in mind that your base units are force and displacement.

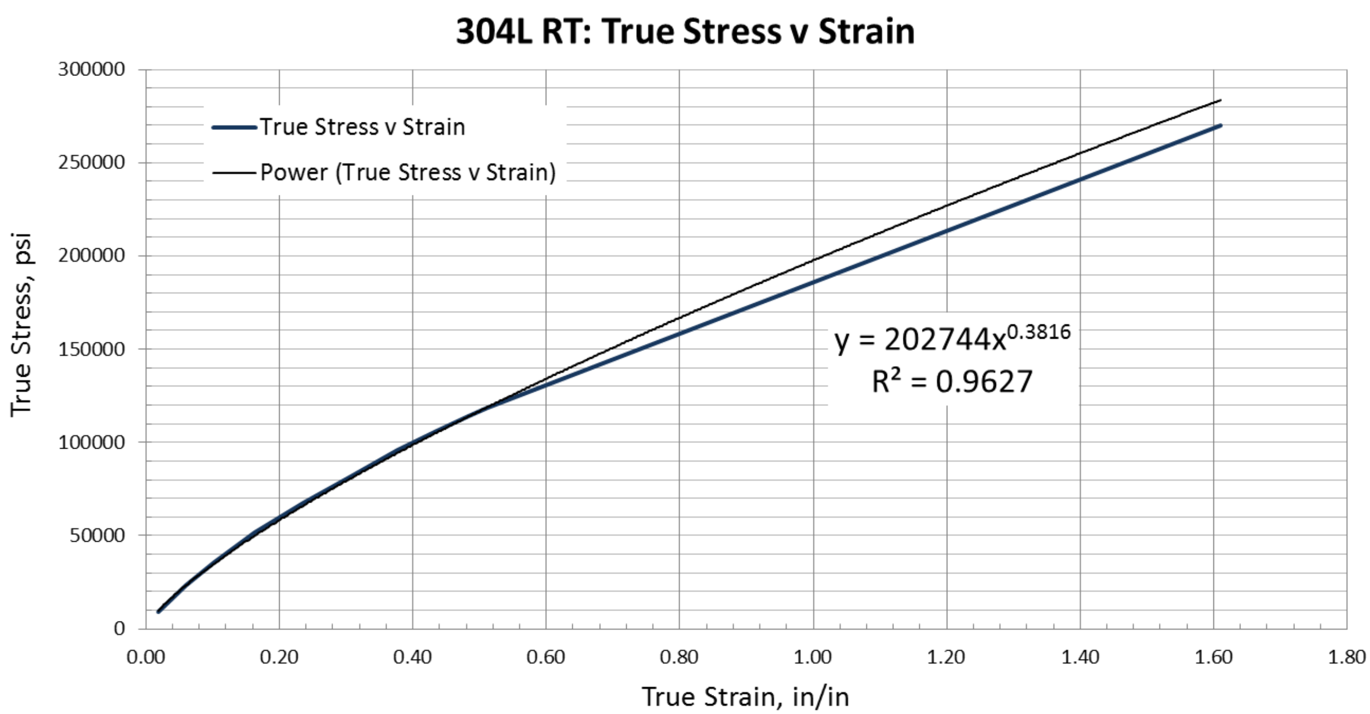


Analyst's Note: By taking our FEA results back to engineering stress vs strain, we can directly compare our FEA model to the raw experimental results. In this manner, one can have confidence that the engineered stress-strain data was converted successfully.

8.5 Material Modeling of Stainless Steel - *MAT_024 (curve) or *MAT_098 (equation)

Over the last several years of LS-DYNA work, I have become more attentive to the material modeling aspect of the idealization process. Although *MAT_024 is the workhorse of the engineered plastic and metallic material modeling world, it is using discrete steps (curve data) to approximate a material deformation process that is smooth and continuous. Toward this end, when appropriate, I like to model materials using a power law relationship via *MAT_098 - *MAT_SIMPLIFIED_JOHNSON_COOK.

The curve fitting can be done within Excel to arrive at the parameters.

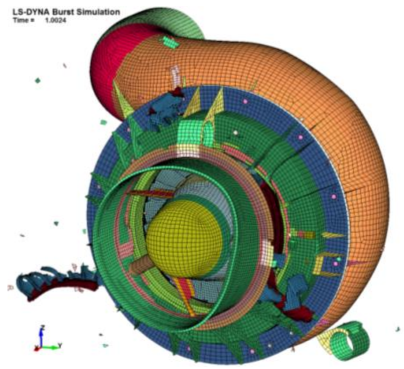


A little Note: If one would like to read a high-level overview, I wrote a small article in FEA Information Engineering Solutions - January, 2016. This article is within the Class Reference Notes / Material Database and is titled: LS-DYNA: Observations on Material Modeling.

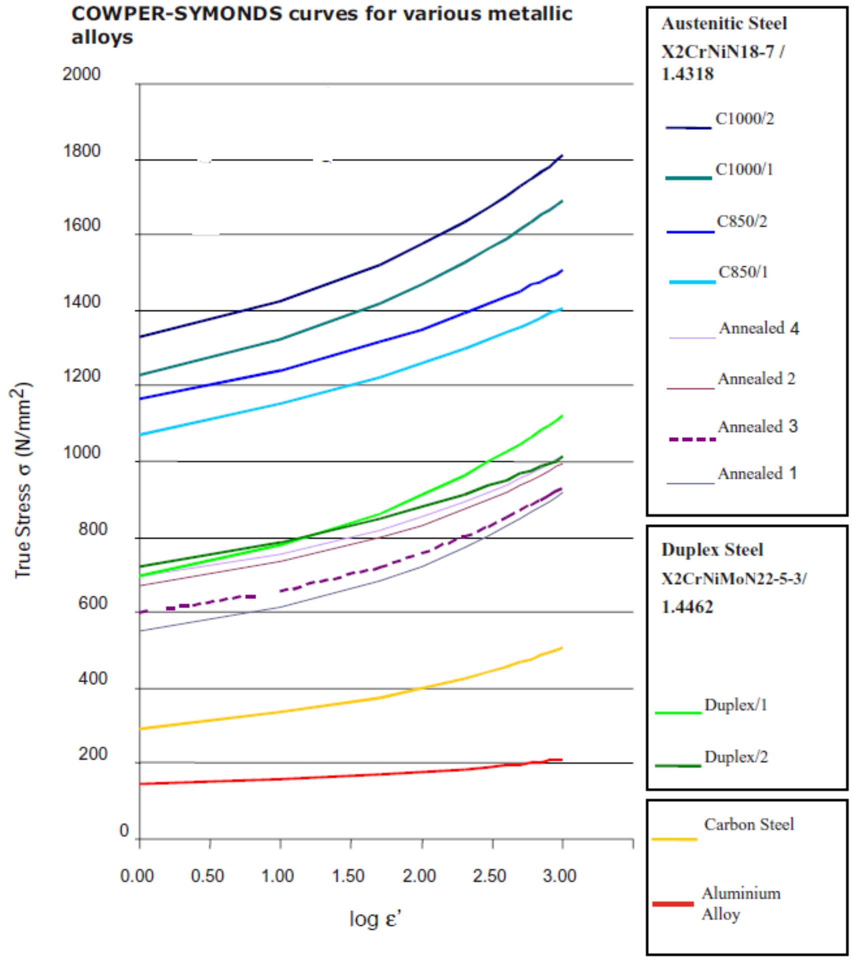
Extra Task: *Rework last model using *MAT_098 – hint A=40,000 also take a look at Varmint Al's data for formatting.*

8.6 Strain Rate Sensitivity of Metals

- This mechanical metallurgical behavior is due to the movement of dislocations within the crystalline structure. Dislocations move up to the speed of sound within the metal.
- The graph on the right provides a rough-order-of-magnitude idea of how strain rate affects the true stress in steels and aluminum. The strain rate is in seconds.
- An example of strain rate effects might be that for rotor burst. The rotor spins at 55,000 rpm. The tip velocity of the turbine blade at burst is 575 m/s. As the blade impacts the containment ring, one could expect to see significant strain rate effects.



- For example, given a 10 m tall, carbon steel cylinder that is compressed 10% at 575 m/s. This would give a strain rate of 575 s^{-1} or log ε' = 2.8. From the chart on the right, one could expect an increase in the yield stress of >50%.
- Strain-rate effects can be evaluated by numerical testing (i.e., exploratory work looking at maximums).



Analyst's Note: We were working on this blast project of a generator enclosure. We were concerned about strain rate effects since finding strain rate data for the client's material was problematic. As part of the project work, we also needed the pressure wave in CONWEP format, so we hired a blast expert, Len Schwer. Since Dr. Schwer was on-board, we asked him about material references. His reply was to pause and say something like "Don't you want the results to be conservative?" And then the light bulb went off and we realized, of course, the most conservative assumption is to assume no strain rate effects since the yield stress would then be the lowest.

8.7 Part II: Plastics, Elastomers and Foams

8.7.1 Modeling Plastics, Elastomers vs Foams (Viscoelasticity)

The Material section of the LS-DYNA keyword manual provides a wealth of practical information on the modeling of elastomers (e.g., silicones and rubbers) and foams. As for plastic materials, they have very unusual engineering stress-strain curve due to adjustment of the long chain molecules upon yielding and then classic necking at failure (just like ductile metals). However, once corrected for true stress-strain, the curve looks very reasonable.

The curve on the bottom right is for a uniaxial test performed on rubber. Experimental uniaxial data can be directly entered into the material card (e.g., *MAT_181). For the modeling of rubbers, this is a very common approach.

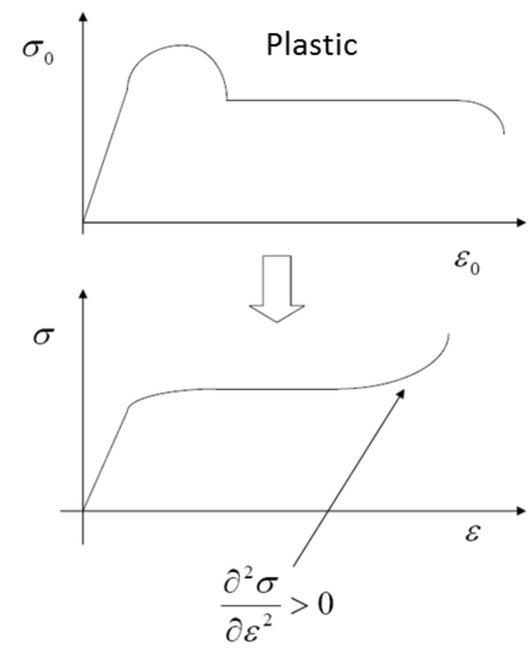
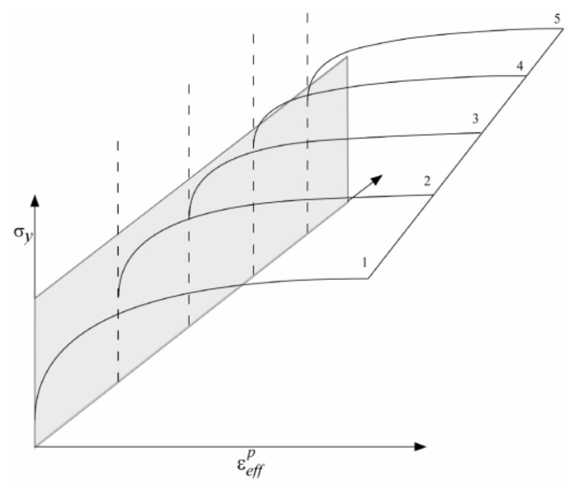
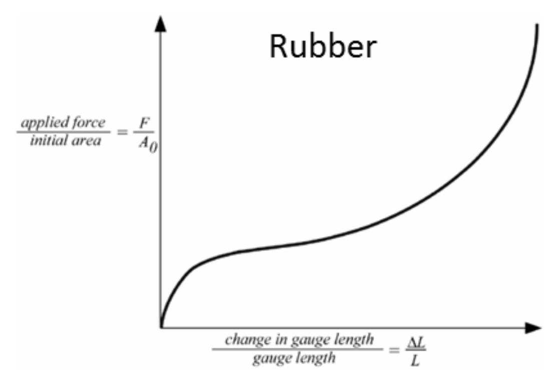


Figure 27.1. Uniaxial specimen for experimental data.

Viscoplasticity

This concept is fundamental in the modeling of plastic, rubber or foam materials. These materials deform via the stretching of their long-chain hydro-carbon network. As such, they are very sensitive to strain rate effects. As the strain rate increases, their complete stress/strain curve will shift upward.





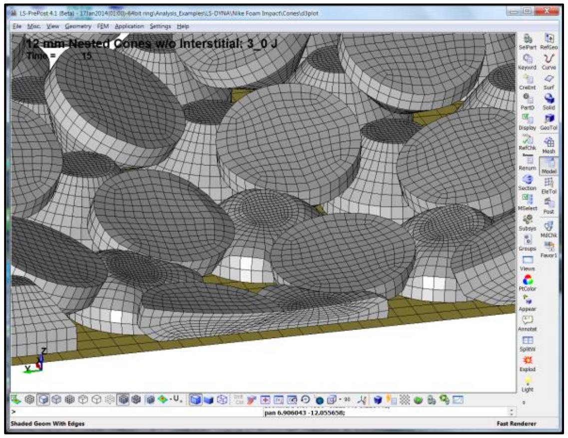
Analyst's Note: Whenever a new material model is simulated, a virtual test coupon analysis should be done and the results compared to the engineering stress vs strain test data. This is actually quite straightforward if one does a plot of force vs displacement (LSPP cross-plot). Since one can than scale these values to match the engineering stress (divide by original sample area) and engineering strain (divide by the sample length).

8.8 MATERIAL MODELS FOR MODELING FOAMS

The following table presents the currently available foam material models in use within LS-DYNA. One recommended foam model is that of *MAT_083 or *MAT_FU_CHANG_FOAM. Foams are perhaps the most challenging material to model due to their extreme nonlinear behavior upon loading and unloading plus their tendency to become crushed or damaged during loading, and then upon unloading, present a completely different stress/strain response.

Foam Material Models Available in LS-DYNA			
5	* MAT_SOIL_AND_FOAM	**142**	*MAT_TRANSVERSELY_ANISOTROPIC_CRUSHABLE_FOAM
14	* MAT_SOIL_AND_FOAM_FAILURE	**144**	*MAT_PITZER_CRUSHABLE_FOAM
26	*MAT_HONEYCOMB	**154**	*MAT_DESHPANDE_FLECK_FOAM
126	*MAT_MODIFIED_HONEYCOMB	**163**	*MAT_MODIFIED_CRUSHABLE_FOAM
53	*MAT_CLOSED_CELL_FOAM	**177**	*MAT_HILL_FOAM
57	*MAT_LOW_DENSITY_FOAM	**178**	*MAT_VISCOELASTIC_HILL_FOAM
62	* MAT_VISCOUS_FOAM	**179**	*MAT_LOW_DENSIlY_SYNTHETIC_FOAM
63	*MAT_CRUSHABLE_FOAM	**180**	*MAT_LOW_DENSIlY_SYNTHETIC_FOAM_ORTHO
73	*MAT_LOW_DENSIlY_VISCOUS_FOAM		
75	*MAT_BILKHU/DUBOIS_FOAM		
83	*MAT_FU_CHANG_FOAM		

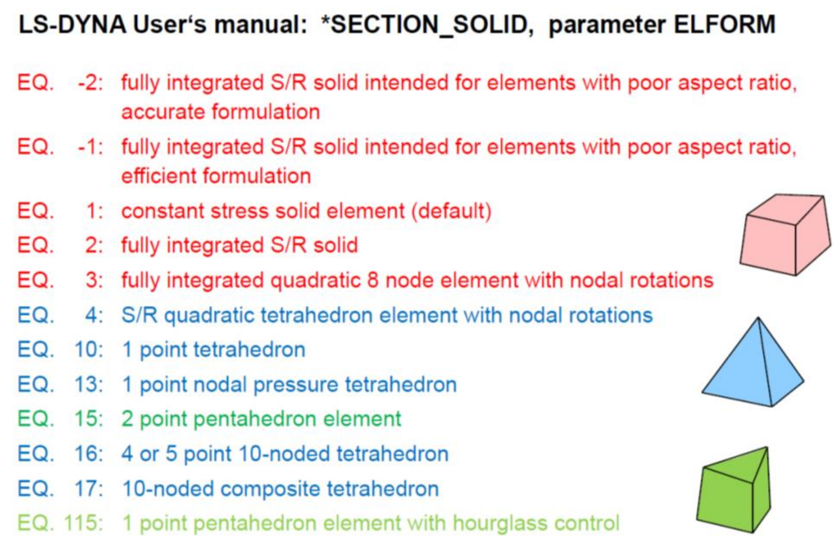
Analyst's Note: Since foams are modeled using solid elements, it is not uncommon to have numerical problems as the foam becomes highly compressed and crushed since the elements used to idealized this behavior, likewise become highly distorted or crushed. Typically, the workarounds are to use highly structured meshes with large element sizes. Another technique that is gaining utility is to use SPH (Smooth Particle Hydrodynamics) to model the foam material. This "mesh-free" technique will be covered at the end of these course notes.



8.9 Modeling Techniques for Elastomers and Foams

Solid Element Meshing for Soft Materials: Hexahedral versus Tetrahedral

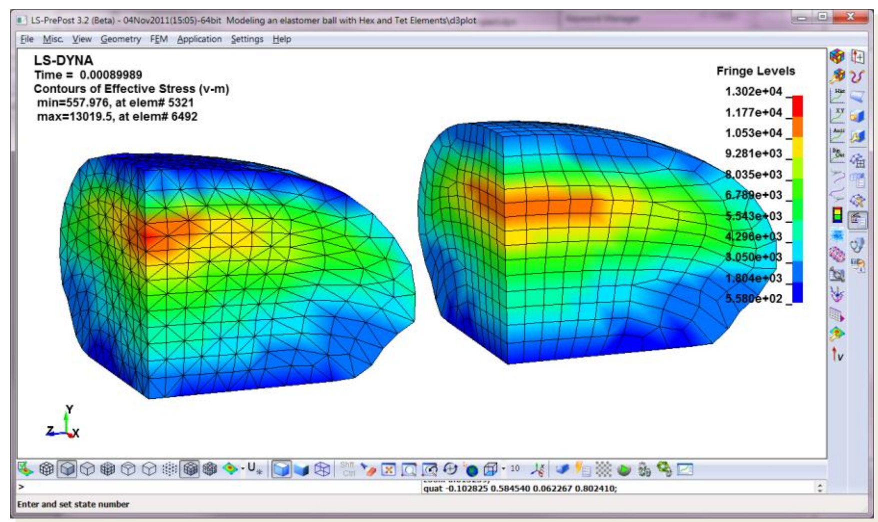
Whenever possible, a hex mesh should be used for the modeling of soft materials. The recommended element formulation for explicit is *elform*=1 (hex) or =13 (tet). Both are one-point Guassian integration formulations and can handle large-deformations without element aspect failure problems (not negative volume). Another recommendation is to use *elform*=-1 if additional accuracy is required and the computation cost to your model is not excessive.



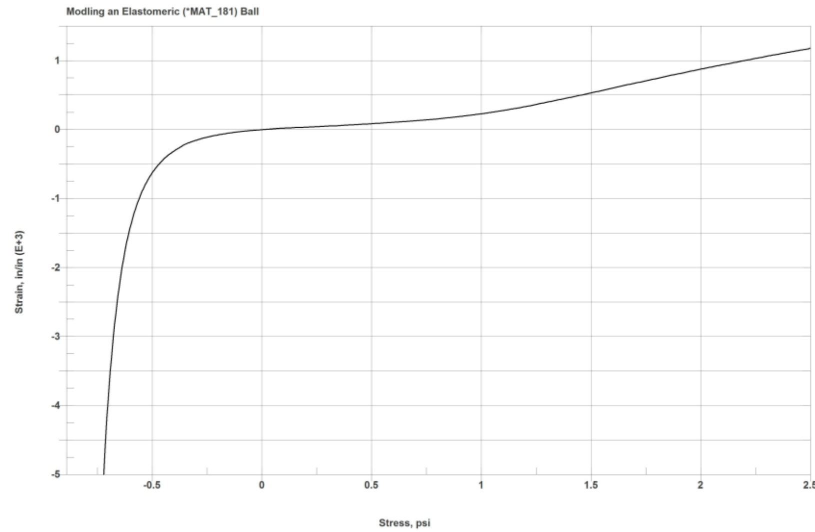
Analyst's Note: Please take a read within: Review of Solid Element Formulations Erhart.pdf (Class Reference Notes / Solid Elements) and it has an excellent section on Hourglass Control.

Element Quality and Negative Volumes:

Large-deformation behavior in soft materials is highly sensitive to mesh characteristics. In an ideal situation, brick elements are always preferred due to their regularity of formation (shape and distribution). Even with a high-quality mesh, under high compression loading, brick elements can generate negative volumes. The best solution is to refine the load application and/or improve the mesh quality. Prior to embracing any one path, it is recommended to consult references (i.e., www.DYNASupport.com and Keyword Manual).



8.9.1 Workshop: 9 - Modeling an Elastomer (*MAT_181) Ball with Hex and Tet Elements

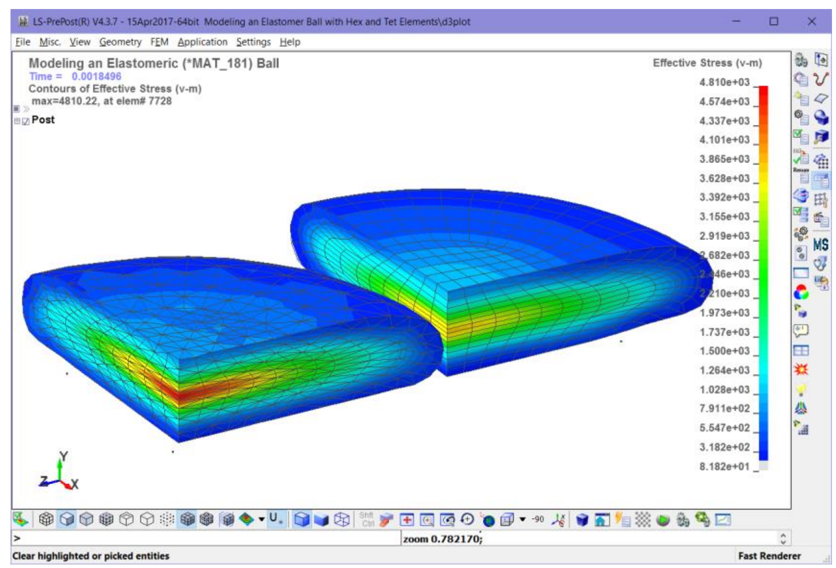


Objective: Understand how to model elastomeric materials using *MAT_181.

Background: Standard rubber material laws (e.g., *MAT_77) fit constants to the experimental data. One has to choose the order of the fit. This can be problematic. In the case of *MAT_181 there is no fitting since the material law is exact based on the curve.

Tasks:

- Inspect the deck (text editor) and then load it into LSPP and plot the material curve. Note that the ordinate has been scaled. This is a handy way to take one elastomeric curve and fit it to a variety of other durometers.
- Run model and display results.
- Let's explore some behaviors. Create a new file folder and drop the Keyword deck into this folder. Make the elastomer softer(sfo=10) and rerun it. One will see how the time step decreases and further decreases.
- Add conventional mass scaling (CMS) to the original deck via *CONTROL_TIMESTEP. Compare KE values before and with CMS. What would be your assessment of the value of mass scaling toward the accuracy of the simulation? To keep things moving along, let's fix the CMS at the starting time step (8.37e-07) of the simulation and see what happens.



*Analyst's Note: If one does a search on *MAT_181 within www.dynalook.com, it is apparent that this is popular material law for modeling rubbers and foams. Also, please note that if your test data is compressive, one must enter these values as negative force / stress and likewise, negative displacement / strain. The material law curve is based on engineering force (or stress) versus displacement (or strain). This means that your compressive strain must be less than 1.0.*

8.10 Part III: Composite or Laminate Material Modeling

LS-DYNA has extensive capabilities for the simulation of composite materials from basic linear to progressive failure and delamination. A simple example is presented on how to create a composite model, view the results and then progressively propagate a crack through the sheet.

There are lots of ways to setup a composite model, we are going to introduce our workhouse and what we feel presents the most robust and time-tested approach for the simulation of composites. We will start with the most basic and stop.

How it goes together (Keywords)

*MAT_054 | *MAT_ENHANCED_COMPOSITE_DAMAGE	Very capable material law that covers linear to progressive failure of the laminate composite. Extensive literature cover on its application and validation to test.
*ELEMENT_SHEL / beta	Beta sets the material orientation for the element. We prefer to set the beta angle in alignment of the 0-degree fiber of the laminate.
*PART_COMPOSITE	Assembly of plies into the final laminate. Each ply has one layer of integration points (e.g., *elform*=-16, four points per layer. If your laminate has 32 plies, then each element has 4x32=128 integration points. *This card combines the *SECTION_SHELL and the *PART card into one surfb card*.
*DATABASE_EXTENT_BINARY / neips and maxint	Failure indices are output as additional history variables. For *MAT_054 there are six extra variables, hence neips=6. For multiple ply composites, the data requests can be intensive, if one has an eight layer, fully-integrated element, then maxint=-8 to capture all the data.

Analyst's Note: In our LS-DYNA Conference Paper: 026_Jensen, A_PAPER_Broad-Spectrum Stress and Vibration Analysis of Large Composite Container.pdf (see Class Reference Notes / Composites), we present a fairly complete workflow for a composite analysis.

8.10.1 Workshop: 10 - Composite Materials - Basic Understanding Using *MAT_054

Objective: Analysis basic hole-in-plate composite shell model under linear and then progressive failure loadings. Contour linear results and then under the progressive failure contour failure indices.

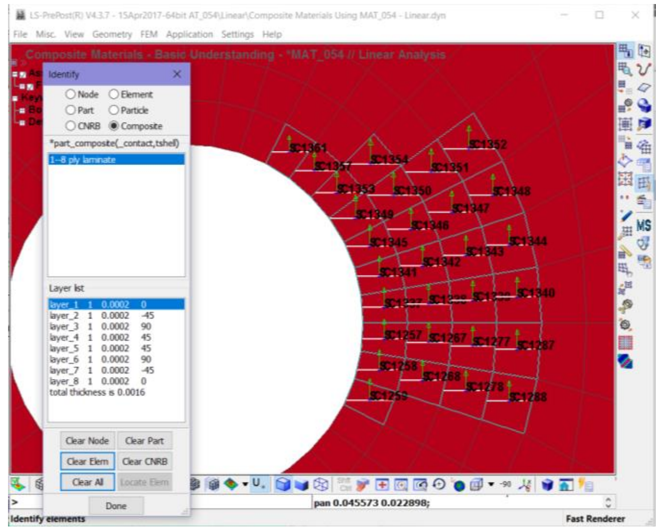
Tasks:

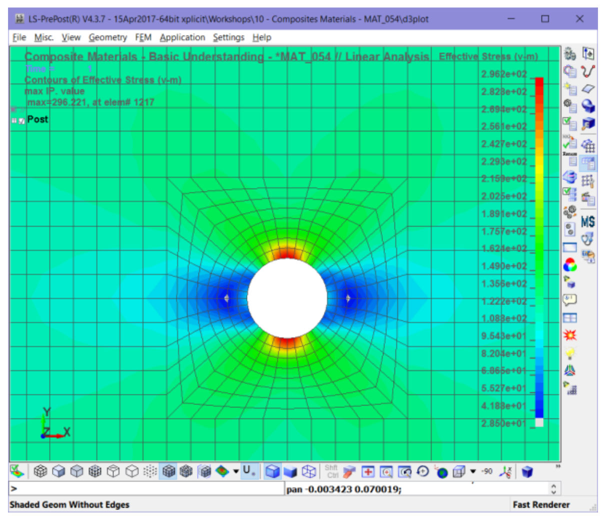
⇒ Load example model file in LSPP and display the (i) beta angle and then each (ii) ply's direction. This is done using (i) Element Tools / Element Editing / Direction (and then selecting the elements) and then (ii) Element Tools / Identify / Composite (and then selecting the elements and walking through the layers).

⇒ Now, look at the Keyword deck in your text editor. The setup is commented. *Be aware that LS-DYNA *MAT_54 uses the minor Poisson's ratio.*

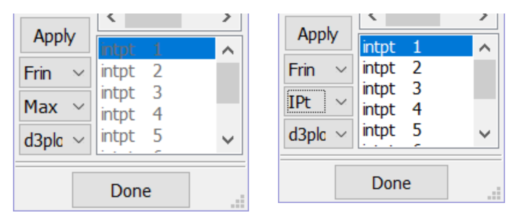
⇒ Element layers are with layer 1 at the bottom of the element as determined by the shell element's vector (+arrow top, -arrow bottom).

⇒ The analysis has been run and one can load the results and then contour the data. The default setting is to present the maximum stress that is found in all the plies. If individual ply stresses are desired, then one can toggle the tab to IPt and pick the ply layer (labeled as intpt 1, etc.).





Fringe Component



Workshop: 10 - Composite Materials - Basic Understanding Using *MAT_054 - (Continued)

Tasks: (continued)

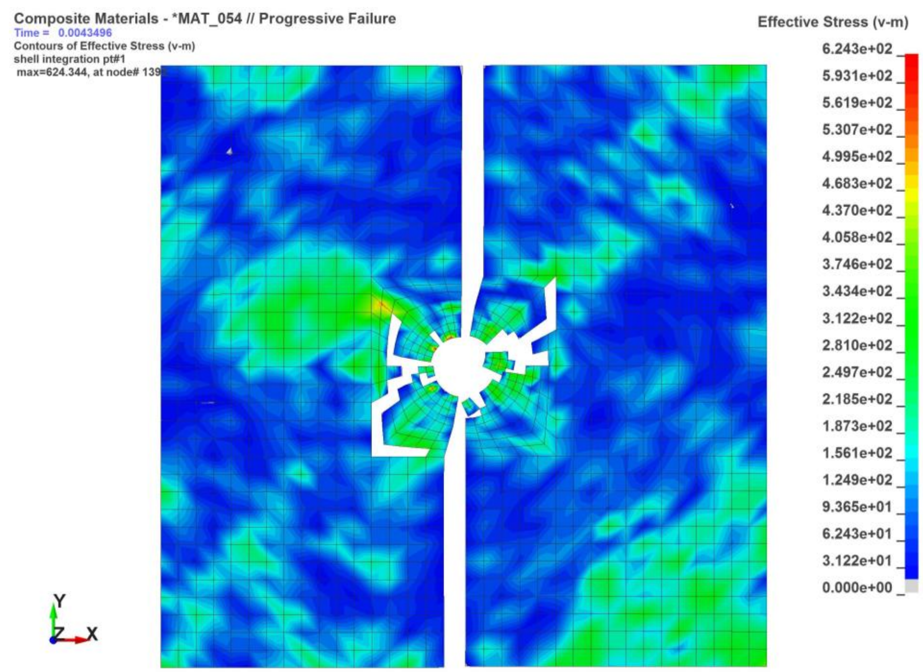
⇒ Open up in a text editor (Composite Materials Using MAT_054 - Progressive Failure – Explicit.dyn and review the settings.

⇒ Progressive failure is controlled by the failure stress and then failure strain. The component (fiber and matrix) yields plastically upon reaching its failure stress (i.e., xc, xt, yc, yt & sc). The element will not be deleted but yielding is occurs. Once the failure strain (dfailm, dfailt & dfailc) has been reached for ***all*** integration points, the ply is deleted.

⇒ Understanding how the failure indices work requires a bit of concentration but they are elegant.

History Variable	Description	Value	LS-PrePost History Variable
1 ef(i)	tensile fiber mode	1 – elastic 0 – failed	1
2 ec(i)	compressive fiber mode		2
3 em(i)	tensile matrix mode		3
4 ed(i)	compressive matrix mode		4
5 efail	max[ef(ip)]		5
6 dam	damage parameter	−1 – element intact 10^{-8} – element in crashfront +1 – element failed	6

```
$
*MAT_ENHANCED_COMPOSITE_DAMAGE_TITLE
Toray Carbon/Epoxy (T800S/3900-2)
$#     mid        ro        ea        eb      (ec)      prba    (prca)    (prcb)
         1    2.0e-6  147000.0    7580.0             0.01547
$#     gab       gbc       gca      (kf)      aopt      2way
    3960.0    3000.0    3960.0
$#      xp        yp        zp        a1        a2        a3    mangle

$#      v1        v2        v3        d1        d2        d3    dfailm    dfails
                                                                  1e-2
$#   tfail      alph      soft      fbrt     ycfac    dfailt    dfailc       efs
                                                        1e-2     -1e-2
$#      xc        xt        yc        yt        sc      crit      beta
      2000      2000       200       200       500        54
$#     pel      epsf      epsr      tsmd     soft2

$#  slimt1    slimc1    slimt2    slimc2     slims    ncyred     softg

$
```



8.10.1.1 Misc Important Notes on Composites

```
- If DFAIL values are NOT given, behavior is elasto-brittle in fiber
  tension whereby integration points fail when reaching the stress-based
  failure criterion in fiber tension.  For other modes (compression modes
  and matrix tension mode), the behavior is elastic-perfectly plastic.
  These criterion are listed in the Users Manual.
```

```
The 'plastic strain' stored in d3plot
and elout is not a strain value at all in the case of mat_54 but rather an indicator flag for failure.
If you're looking at plastic strain for a mat_054 integration point as written to
dynain or to elout, it represents "dam" where dam=-1 only means that the overall element is intact.
The history variable ("effective plastic strain") stored in d3plot is different by design in
3 of the integration points than in dynain and elout.  The 2nd table under *mat_54 gives the
meaning of "eff. plastic strain" for those 3 integration points.  What needs to be clarified
however is that the 3rd one is stored, not in the slot for IP #3 but rather in the slot for
the last integration point.***   So in the 7 IP example mentioned below, integration points 1, 2, and
7 hold the values shown in the 2nd table (d3plot only).  For IP's 3,4,5, and 6 in d3plot,
```

8.11 Part IV: Equation of State (EOS) Material Modeling

An equation of state is required for materials that undergo significant deformation (can be very large plastic deformation or a compressible fluid). The Cauchy stress tensor can be separated into a hydrostatic stress tensor ($p\delta_{ij}$) and a deviatoric stress tensor ($\sigma`_{ij}$):

$$\sigma_{ij} = \sigma`_{ij} + p\delta_{ij}$$

The deviatoric stress is calculated by the material model constitutive law. The pressure term, *p*, must then come from an equation of state (EOS). The EOS provides a relationship between pressure and the volume (can also be a relation of temperature and/or energy). Depending on the compressibility of the material, different types of EOS's are possible. A very popular EOS is the Gruneisen equation of state. The full version of this EOS (compression) is:

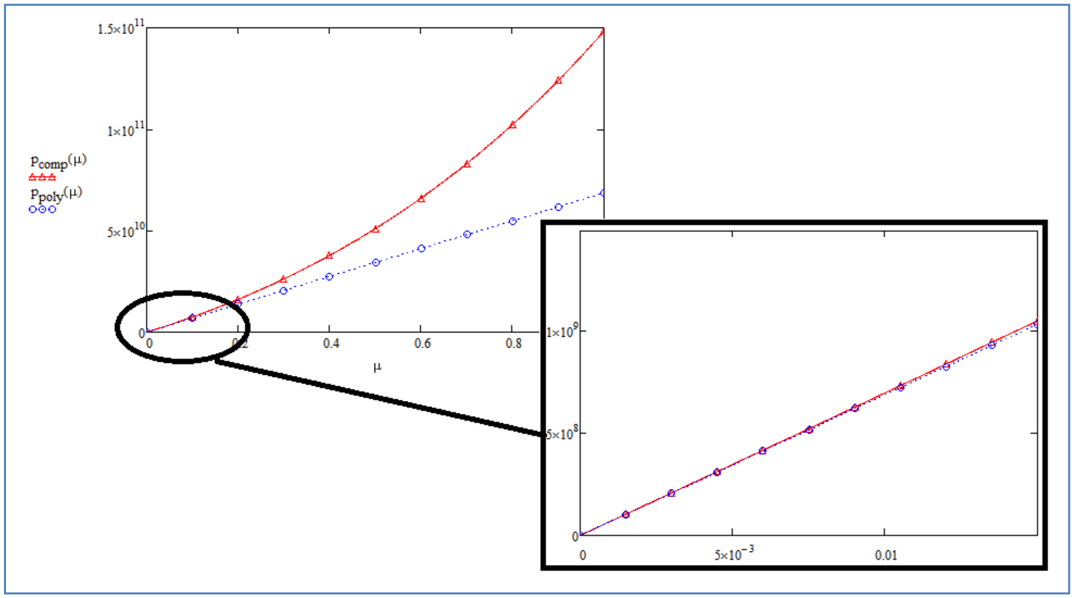
$$p = \frac{\rho_0\, C^2\, \mu \left[1 + \left(1 - \frac{\gamma_0}{2}\right)\mu - \frac{a}{2}\mu^2\right]}{\left[1 - (S_1 - 1)\mu - S_2\left(\frac{\mu^2}{1+\mu}\right) - S_3\left(\frac{\mu^3}{(1+\mu)^2}\right)\right]^2} + (\gamma_0 + a\,\mu)E$$

The equation appears rather complicated at first glance, if we focus on a weakly compressible fluid (many engineering fluids can be considered this way), the equation of state can be reduced to:

$$p = \rho_0\, C^2\, \mu$$

Where ρ_0 is the initial reference density for the material, C is the speed of sound in the material and $\mu = \frac{\rho}{\rho_0} - 1$. All the other parameters are curve found by curve fitting to a set of compression experiments. These parameters are typically only needed when the pressure loading is very high as in shock waves. In the figure shown on the next page, we can see a comparison between an EOS specified with only the speed of sound (linear dependency of pressure on volume) and an EOS to give a cubic dependency of pressure on volume.

In the subsequent workshops, the units are $p\ [Pa]$, $\rho_0\ [kg/m^3]$ (specified as RO on *MAT_NULL) ,and $C\ [m/s]$ (specified as C0 on *EOS_GRUNEISEN) and μ is a dimensionless parameter.



Comparison between weakly compressible and compressible

Fluids in LS-DYNA Explicit need to be described by a constitutive material law (such as *MAT_NULL for example) and an appropriate EOS. The reason is that solving the set of Euler equations (or full Navier-Stokes in the presence of viscosity) with a strictly explicit time integration scheme requires an equation of state to directly determine the pressure at each node point. A truly incompressible algorithm requires solving a Poisson equation (elliptical partial differential equation) to ensure that the flow is divergence free. The Poisson equation can only be solved iteratively or using Implicit time integration.

All this truly means is that in LS-DYNA SPH Explicit, a fluid that is commonly considered incompressible can be treated as weakly compressible with a simple EOS by defining only two parameters; initial density and the speed of sound in the material. Throughout the workshops, we use *MAT_NULL, but other material models such as *MAT_JOHNSON_COOK, *Elastic_Plastic_Hydrodynamic, etc. can be used with an EOS to describe various engineering materials.

8.11.1 Modeling Water with *EOS_GRUNEISEN and *MAT_NULL

As basic as this may sound, it is not that obvious. The technique for modeling water is well described in Class Reference Notes / Aerospace Working Group / AWG LS-DYNA Modeling Guidelines MGD_v19-2 (December 2019).pdf at page 62, Section 3.9.2 Water. The entry is as simple as the graphic shown on the right.

The only real difficulty is to ensure that one gets the units handled correctly.

*Analyst's Note: Make sure you remember to set the *PART card to use the *EOS law.*

3.9.2 Water

In LS-DYNA models, *MAT_NULL and *EOS_GRUNEISEN are commonly used to represent water and other liquids. Some commonly used input constants for water at 20 deg C follow.

*MAT_NULL:
Mass density, RO = 1.e-6 kg/mm^3
1.0 g/cm^3
1000. kg/m^3
1.e-9 tonnes/mm^3
9.37e-5 lbf-s^2/in^4
Dynamic viscosity, MU = 1.0e-3 N-s/m^2 (often taken as 0.0)
Pressure cutoff, PC = -100 Pa (often taken as zero)
All other parameters in *MAT_NULL should be set to zero or left blank.

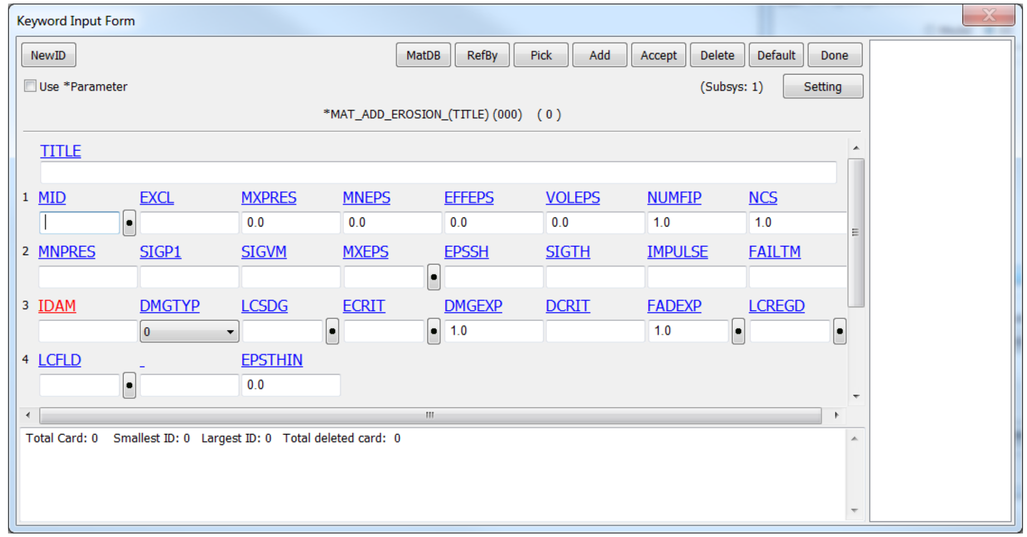
*EOS_GRUNEISEN:
Nominal sound speed C = 1500 mm/ms
0.15 cm/microsec
1500 m/s
1500e3 mm/s
59055 in/s
E0 = 0
V0 = 1.0 (unitless)

8.12 Material Failure Simulation

8.12.1 Basic Methods of Modeling Failure: Material versus Bond Failure

Standard failure in the modeling of materials is by specifying some sort of material based failure criterion. My favorite approach is to use the *MAT_ADD_EROSION card to specify the exact failure criteria that is needed. For metals, one approach is to set the EFFEPS (maximum effective strain at failure) to 3x MXEPS (maximum principal strain at failure). This ensures that the material does not prematurely fails under compressive plastic deformation but still remains true to the mechanical test data.

Exercise: Open LS-DYNA_manual_Vol_II_XX (*whatever is the latest*).pdf (see Class Reference Notes / Keyword Manuals) and read the *MAT_ADD_EROSION card section.



Another way of modeling failure is by *CONSTRAINED_TIED_NODES_FAILURE. With this formulation, bond failure can be modeled in a direct and simple manner by setting the plastic strain required to pull apart the nodes. Of course, this plastic strain is taken from that elements integration point. The setup for this failure mechanism is to take a clean mesh and let LSPP create the tied connections. This is done by breaking apart the elements and then tying together the adjacent nodes. Upon failure, the elements fly apart but are not deleted. An example of this concept is can be found at www.dynaexamples.com / Intro by J. Reid / Sphere Plate. Given all that, I prefer the simplicity of *MAT_ADD_EROSION.

*Analyst's Note: Element erosion (how general element failure is treated in LS-DYNA) is based on calculated values at the element's integration point. For standard shell (elform=2) and solid element (elform=1) formulations, only one integration point is used. Hence, erosion occurs when that integration point fails based on the provided criteria. For fully integration shell (e.g., elform=16) and solid (e.g., elform=-1) elements, erosion occurs upon failure of any of the integration points reach the failure criteria. This can be modified by the use of numfip within the *MAT_ADD_EROSION card such that all of the element's integration points must fail (e.g.., for solids numfit=8) or just a percentage (e.g., for shells numfip=-50 requires 50% of the integration points to fail for element erosion). Also, once an element erodes: 'Think of the elements as point masses being connected by springs, i.e. the stiffness/internal force of the element. The element being deleted is like the springs no longer holding the masses together, at which point they are free to fly about." **Courtesy of Ushnish, LSTC Technical Support**. Which means that athough the element is gone, the nodes stick around and have the same KE as prior to deletion. Once can even have these nodes still be active for contact, albeit not for *CONTACT that employs soft=2 (RTM).*

8.13 Workshop: 11 - Modeling General Material Failure

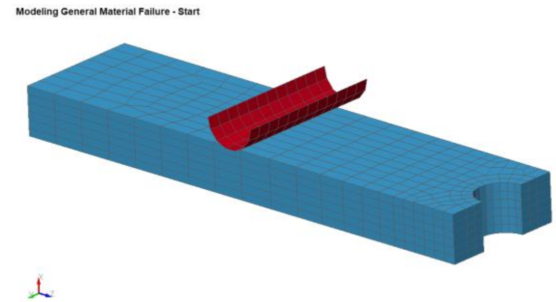
Objective: Modeling of material failure is not as complex as one might think if a reasonable expectation is taken from the outset. It should be noted that: "All models are wrong, but some models are useful" and hence when trying to replicate failure in a structure one should strive for simplicity prior to adding complexity. This concept segues into another saying: "elegant simplicity is deceptively difficult to achieve". In the following workshop, a basic failure mechanism within the *MAT_024 card is improved upon by using the *MAT_ADD_EROSION approach.

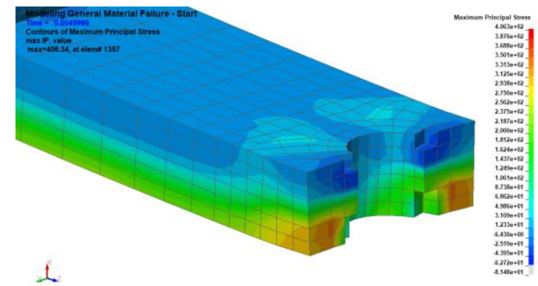
Model Introduction: A four-point bend test is conducted (1/2 symmetry). One will notice that elements under compression fail somewhat equally as elements under tension. This is wrong. Your task is to "fix" the model and make it fit reality using *MAT_ADD_EROSION.

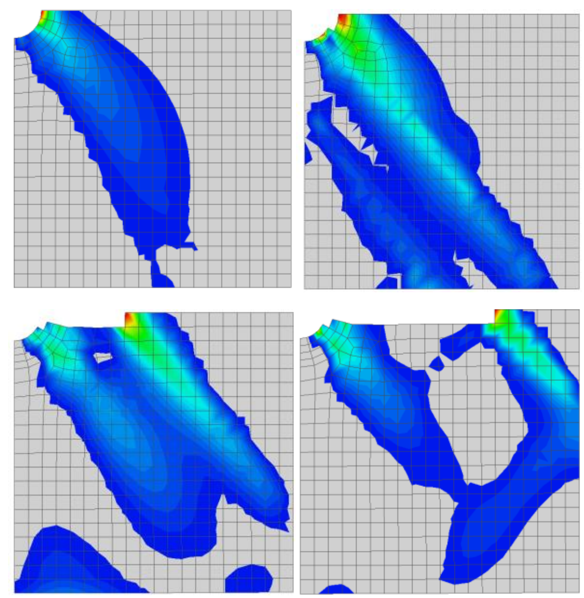
Workshop Tasks:

- Open General Material Modeling - Start.dyn and inspect material law used for the default simulation.
- Run Model and inspect failure mode of elements; i.e., deleting equally under compression and tension. This is just not physically possible. Why not? Read PDF's in Workshop 11.
- ***Delete*** failure criterion (fail) from MAT_PIECEWISE_LINEAR_PLASTICITY_TITLE card.
- Open LS-DYNA Material Manual and read-up on the *MAT_ADD_EROSION card. Then, within LSPP add a tensile strain failure criterion (*mxeps*) = 0.1 and also *effeps* = **-**0.3 to the existing material law. Note 1-3 ratio. Why? See News Article and scientific paper (pdf) in Workshop Folder. Please note "-" for *effeps* value; why? RTM.
- Rerun model and interrogate von Mises and 1^{st} principal stress.

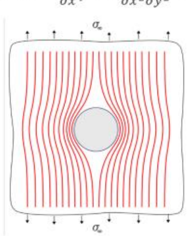
Extra Task: If a hole through a plate creates a stress concentration of 3x, if the load was compressive, what would be the tensile stress at the top of the hole (i.e., in the direction of the load? Inspect QS plate model, run it and interrogate the stresses (contour von Mises and then 1^{st}-principal stress. The material fails at 2,000 psi (tensile). Some background on stress ⮷

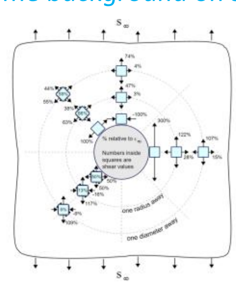






Stress Flows $\frac{\partial^4 \phi}{\partial x^4} + 2\frac{\partial^4 \phi}{\partial x^2 \partial y^2} + \frac{\partial^4 \phi}{\partial y^4} = 0$

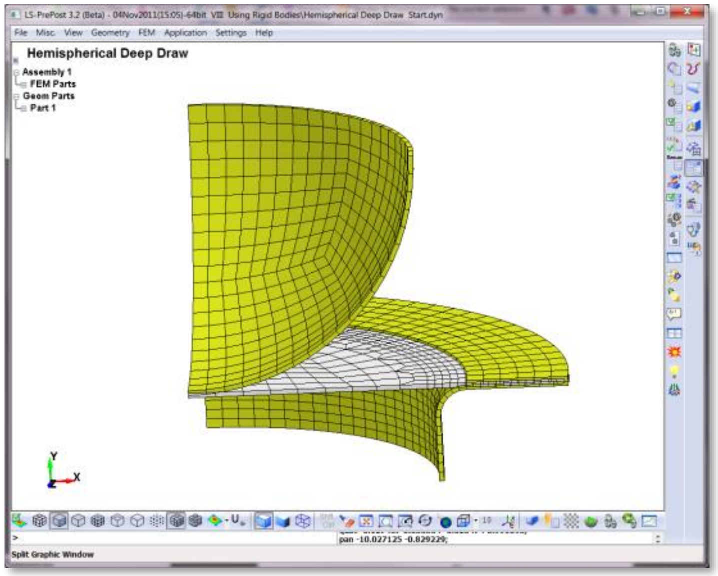




8.14 Modeling Rigid Bodies

8.14.1 Rigid Materials (*MAT_020 or *MAT_RIGID)

This is one of the most powerful modeling techniques within LS-DYNA. By setting bodies (i.e., parts) to use *MAT_RIGID, where deformation and stresses are not of interest, significant CPU savings can be realized. In the background, LS-DYNA retains the surface mesh of the part for contact behavior and calculates an inertia matrix to simulate the dynamic behavior of the body. What is useful with this approach is that the body still retains its inertia and physical characteristics as it interacts with other bodies within the simulation but at a fraction of the numerical cost of dragging around a fully deformable body. For example, the model on the right is of a deep drawing operation and only the plate is deformable.



*Note: Rigid bodies cannot have constraints applied to them. To constrain a rigid body, the CMO flag is set within the *MAT_RIGID card.*

For a very nicely done reference, please see Class Reference Notes / Rigid Bodies / LS-DYNA Intro Class Chapter 9 Rigid Bodies.pdf.

Other Rigid and Not-so-Rigid Idealization Connections

Nastran multi-point-constraints (MPC) equations, of which their two most common flavors are the RBE2 and RBE3 elements, are translated into two different LS-DYNA formulations. The RBE2 is translated into a rigid body where the nodes are placed into a group and then constrained rigidly per the number of dependent DOF's specified in the RBE2 element. The inertia or mass properties for this nodal rigid body are obtained from the elements attached to the nodes of the rigid body. Although this may sound a bit odd coming from the implicit world, in explicit mechanics everything needs a bit of mass to enable its calculation. Hence, for the rigid link to behave correctly, it borrows mass from its attached elements and two node CNRB's should be avoided.

The LS-DYNA card that is used is *CONSTRAINED_NODAL_RIGID_BODY (CNRB) with the CMO card specifying what DOF's are to be released.

8.14.2 Workshop: 12 - Using Rigid Bodies

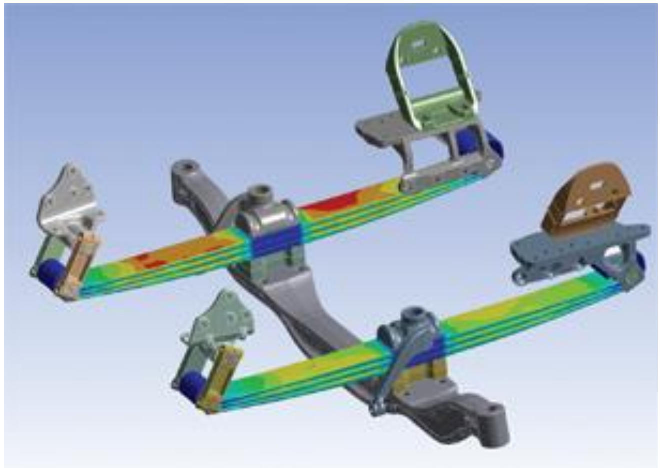
Introduction: For complex system level models, the use of rigid bodies allows one to capture the full kinematics of the system (inertia and contact) with very little numerical cost (rigid dynamics). The rigid body technology can also be used to create large multi-body interactions (e.g., four bar linkages, dynamic simulation of multi-body systems to extract joint forces along with part velocities and accelerations).

Objective: Get comfortable with the concept of rigid bodies as they Interact with deformable bodies (e.g., pressure loading, motion, and contact).

Tasks:

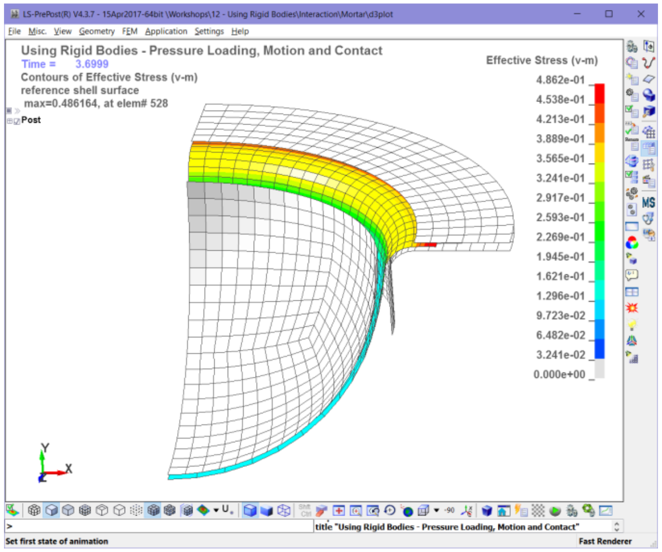
- Add *BOUNDARY_PRESCRIBED_MOTION_RIGID to move the punch (open Using Rigid Bodies - (i) Interaction (Loading and Contact) - Start.dyn). You'll need to move the punch down by -**3** (*sf*) units down using *lcid*=**2**. One should consult the Keyword Manual for definitions.
- Correct pressure load on Clamp Plate (Hint: Think negative pressure ***sf*** **= -10**). Although it is a rigid material, it still recognizes load application through its segment faces. Likewise, one can apply a load at a grid point of a rigid body. As one can imagine, the force is resolved thru the rigid body's center of mass.
- Inspect Contact definition. It is a single surface type, and everything is contacting everything else (all blank settings). Contact works on both deformable and non-deformable (i.e., rigid) bodies.
- Run model and contour von Mises stress and verify that it is logical.

Rigid Body Kinematics



Courtesy of Desktop Engineering, Pamela Waterman

Metal Forming



8.14.2.1 Instructor Led Workshop: 5 – Connections From RBE2 /CNRB and RBE3/ CI

Objective: Working with *CONSTRAINED_NODAL_RIGID_BODY and *CONSTRAINED_INTERPOLATION connections.

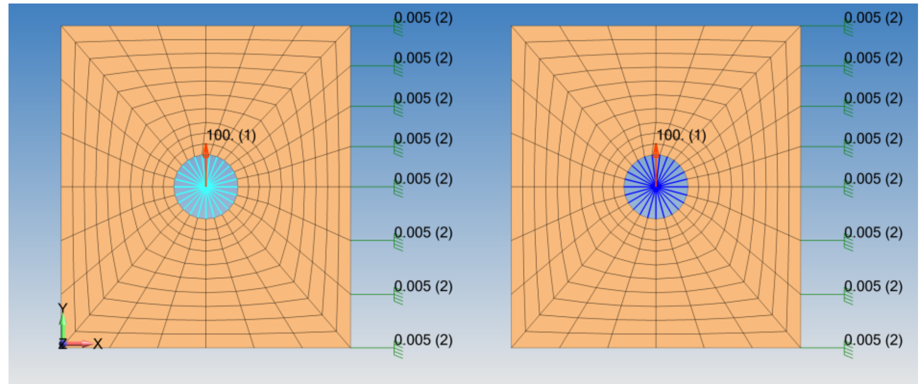
Why: In implicit modeling, the use of connector elements (e.g., Nastran MPC's (RBE2 & RBE3)) is common and likewise, the same technologies are useful in explicit modeling but require *mass*.

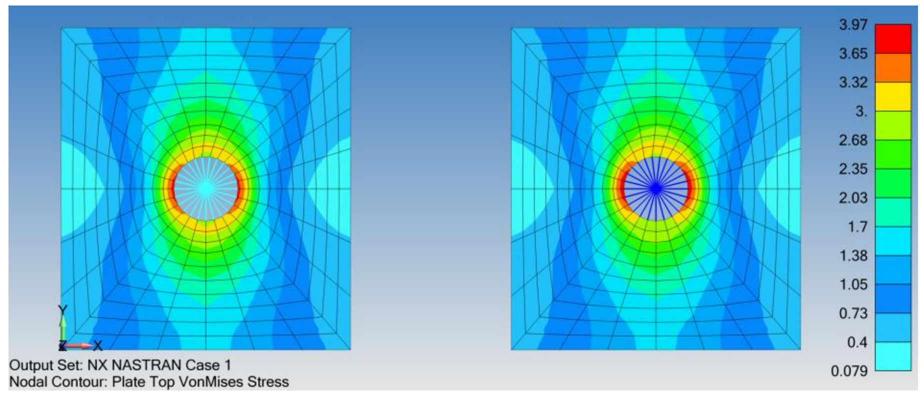
Task: Open the Instructor example model and run the model. Read the messages for clues on what is wrong with the model. If one wants to correct the model, the use of this element might be useful: *ELEMENT_MASS with a *imass* of some very small, insignificant value of say 1e-6.

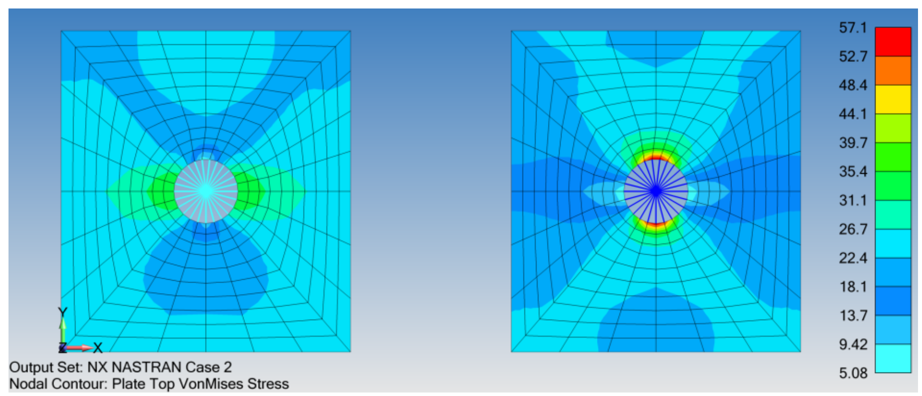
To start the model correction process, open the Keyword deck and add mass to the offending node and rerun. Of course, a final run-ready model is available if you get stuck.

Post-Processing Notes: The instructor model is Nastran with linear stresses extrapolated to the nodes. To extrapolate linear stresses in LSPP, enter extrapolate 1 in the lower-left hand corner box.

Analyst's Note: One can't apply constraints to a CNRB but one can fix'em via the _SPC option (which adds card 2 with the cmo, con1 and con2 options). Please note that the Nastran RBE2 or RBE3 element number is used as the LS-DYNA CNRB Part number or interpolation constraint ID.







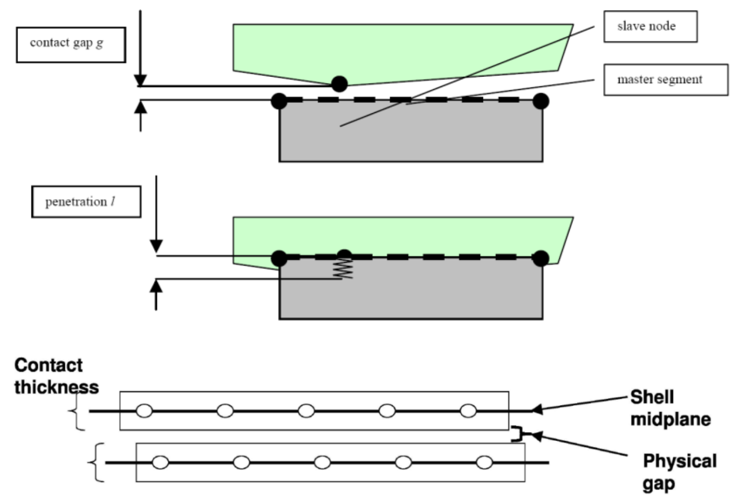
8.15 Verification of Material Model

"It is not that hard to have a simulation model 90% right and can be done quickly and inexpensively. It is very difficult to get it 99% right and a lot more expensive. The challenge is that most companies want to pay for 90% but require 99% right."

This came out in running a material test coupon in explicit mode. I asked Paul about how to ensure that we weren't picking up dynamic effects. The limitation to the model is that it required small elements and run times could get quite long given that many runs are necessary. I said: "Well, the kinetic energy (KE) was <1% of the internal energy (IE) and that should be good enough". He counted that using that assumption is right nine times out of ten. That the only way to be sure is to run it with longer run times and check for convergence. He said that many other simulation engineers fall into this trap of ratio'ing KE to IE and it works most of the time but then when it doesn't work you are really screwed.

9. CONTACT

9.1 Definition of Contact Types



LS-DYNA was developed specifically to solve contact problems (see Class Reference Notes / History of LS-DYNA). Contact behavior is enforced by two methods: (i) Penalty-Based using finite springs (see graphic) and (ii) Tied Contact (discussed within its own section).

Contact can be effortlessly implemented or it can be bewitching in complexity. A reasonable treatment of contact is a multi-day course in itself. To start the learning process see Class Reference Notes / Contact User's Guide / Contact User's Guide.pdf. This introduction will focus on the most effective and robust contact methods. They may not be the most numerically efficient, but they will allow the user to enforce clean contact behavior with a minimum of debugging effort.

9.1.1 What is Implicit with the _AUTOMATIC Option?

As one would suspect, *CONTACT has evolved over many years within LS-DYNA. Initially, given computational limitations, the analyst had to be careful with element normals and shell thicknesses. It is now more standard to just use _AUTOMATIC on the *CONTACT cards. This option does several things: (i) Turns on shell thickness for contact behavior; (ii) Looks for contact penetration in both shell directions (regardless of normal orientation); and (iii) Better search routines for when contact may occur.

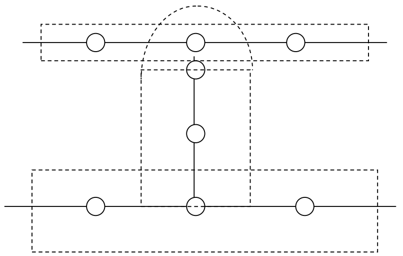
9.1.1.1 Efficient Contact Modeling

Whenever possible, interferences between parts should be avoided. It is standard contact practice that any initial interference is removed (nodes are shifted) and as such, sharp stress spikes can occur where parts/plates overlap.

Setting up contact surfaces appropriately that account for plate thickness can be time consuming. If necessary, contact thickness can be overridden within the *CONTACT Keyword card or tracked via ignore=1

*Analyst's Note: One can use *CONTROL_CONTACT and, as a default, for all defined *CONTACT_ cards to remove edge projection by setting shledg=1.*

Contact accounts for shell thickness and, for shells, there is "end projection" as shown. This can cause problems if your mesh doesn't account for such projections. What is shown below is not good since the end projection penetrates into the upper shell. This would create an initial interference that would have to be removed during the simulation.



9.2 General Contact Types

Although there are numerous contact types given in the Keyword Manual, these workhorse formulations are recommended:

1. *CONTACT_AUTOMATIC_GENERAL[1]
2. ***CONTACT_AUTOMATIC_SINGLE_SURFACE[1]**
3. *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE

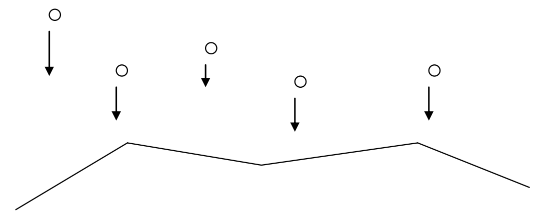
The first formulation is the "kitchen sink" and is computationally expensive but is very robust and will enforce contact between beam elements and other components. It is a *single-surface formulation* and only the *surfa* side is defined (it assumes that everything contacts everything else) and edge contact on shells, it adds null beams (within the solver) to enforce contact. Thus _GENERAL is the most capable since it can detect contact from solids to shells (segments and edges) and beams.

The second formulation is the workhorse and is highly efficient and with the "right settings) it handles everything except beam-to-beam contact. While the last formulation is general purpose and is advantageous when one wants to look in detail at specific contact pairs.

[1]Note: Take a look at our contact recommendation prior to using these formulations

Classical Contact

"Keeping Nodes on the Right Side"



9.2.1 In *CONTACT, What Does _SURFACE Mean?

When one thinks of "surface", it implies a smooth geometric skin whereas for LS-DYNA, it implies an element face. A _SURFACE can be one element face or a collection of faces. In LS-DYNA a face is termed a segment. That is, a "segment" is the base terminology to define contact faces using shells or solids. On a higher level, when one defines a _SURFACE using a *PART ID for a solid element, the free surface of the solid elements as defined by the *PART ID, is wrapped using segments. This is done internally within the code. One can also limit *CONTACT regions by specifying groups of segments using the *SET_SEGMENT Keyword command (within LSPP see Model and Part / Create Entity / Set Data / *SET_SEGM).

In general, our preference is to define contact regions using the *PART ID. This provides the flexibility to change the underlying nodes and elements without having to touch the *CONTACT definitions.

*Analyst's Note: Let's talk about *CONTACT in general terms. When the analysis start, the *CONTACT algorithm looks around its defined regions (i.e., surfb/surfa sets as defined by the analyst as *PARTs or segments or elements or nodes) and finds its nearest neighbors. This is called the "bucket sort" (i.e., bsort in Card A of the *CONTACT Keyword definition). It is not done every iteration but nominally once every 10-100 iterations depending upon the *CONTACT definition (i.e., see bsort in the *CONTACT Keyword). They are then tracked as the parts move around. When they get close (see depth), the algorithm starts checking for penetration (more numerically intense) and when they touch, the surfaces (i.e., nodes) forces are applied to the offending surfaces (nodes) to prevent interpenetration. The magnitude of the force is proportional to the interpenetration. Thus, the term "springs" is used to describe the *CONTACT interface. Keep in mind that *CONTACT interfaces have their own explicit time step since they have stiffness. They get their mass from borrowing a bit from the adjacent surfaces.*

9.2.2 Additional Options: Optional Card A - soft=2 and depth=5 "The Default"

The standard contact search routine is based on nodes looking for faces (i.e., segments). Sometimes, contact might be missed for a few steps, and when finally engaged a large restoring force is required to separate the interfaces or contact might just not occur. Additionally, the standard penalty approach calculates the spring stiffness based on global material stiffnesses. This can lead to contact instability in soft materials.

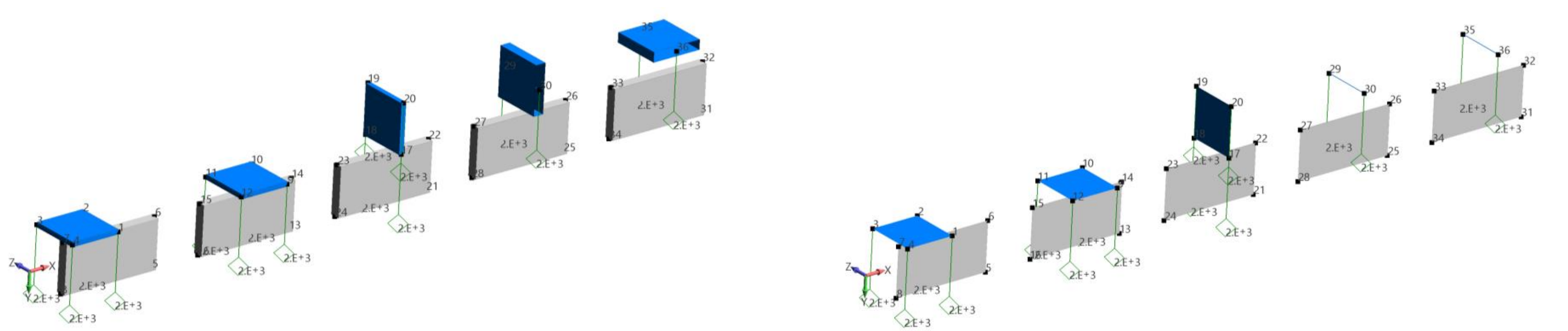
To counter these problems, the contact option *soft*=2 is recommended. This switches the contact search routine to segment-to-segment and locally calculates the stiffness for the penalty approach. Although more computationally expensive, it provides a more robust contact and it is used as our default. If one adds *depth*=5 to an existing *soft*=2, then edge-to-edge contact is enabled (but not beam-to-beam). Of course, once your analysis is running and one wants to tune it for speed, *soft*=2 (and *depth*=5) could be removed and the results checked to see if they are stable. The following Instructor Led Workshop helps elucidate how *soft*=2 and other options works.

9.2.2.1 Instructor Led Workshop: 6A – Basics of Contact

In the prior discussion, one saw how the standard contact (*CONTACT_AUTOMATIC_SINGLE_SURFACE with *soft*=2 with *depth* 5) can handle the basics of shell-to-shell contact and is preferred over the kitchen seat contact formulation *CONTACT_AUTOMATIC_GENERAL, ahh but only if LS-DYNA life could be so easy. That is, if we have shells covered, then likewise we have solids covered but what about beams? So, as has been emphasized in this class, let's look at a small pilot model and see how it can guide us to understanding beam-on-beam contact.

Shells-to-Shells and Beams-to-Shells

View with "thickness" | **View au Natural**



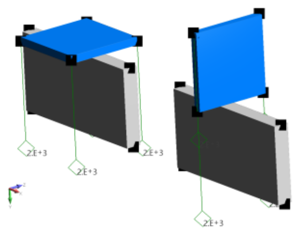
In this example, we walk thru the *CONTACT_AUTOMATIC sequence from: (i) _SINGLE_SURFACE; (ii) _SINGLE_SURFACE with *soft*=2; (iii) _SINGLE_SURFACE with *soft*=2 and *depth*=5; (iv) _GENERAL and (v) _SINGLE_SURFACE_MORTAR. As we add options from (i) to (v) we increase the numerical cost but gain in contact robustness.

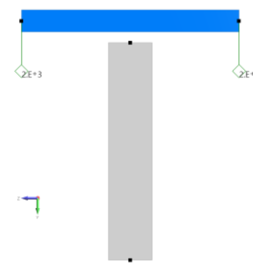
*Analysts' Note: One can always create contact sets and reserve the most expensive *CONTACT formulation for a subset of the model. For example, create one set for beams and have it contact the rest of the model or something similar. Of course, if the model is small enough, there is a temptation to just use _MORTAR contact and be done with it.*

9.2.2.2 Instructor Led Workshop 6A – Basics of Contact – A Little Detail That Could Whack You

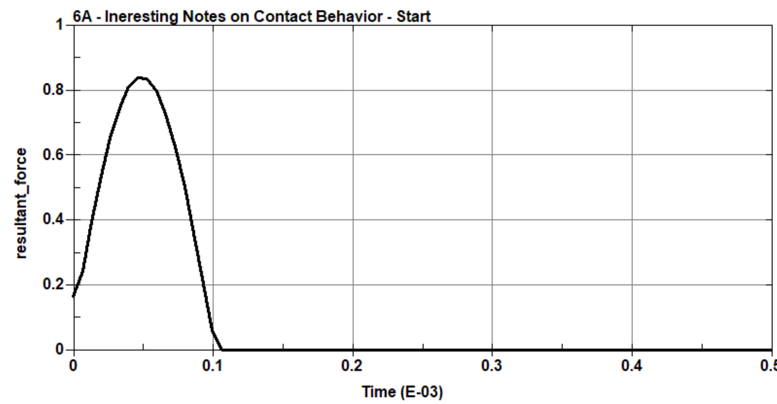
Just to cover all the bases, this thing has burned me a few times building models. It has to do with edge extension.

Shells Impacting Shells
Segment (face)-to-Edge and Edge-to-Edge





The last adjustment has to do with shell edge projection. If one looks at the model closely and check the spacing between the shells, one will note that if shell edge projection is enabled, the shells will contact. This can be seen numerically by plotting the *rcforc* component from the *DATABASE_RCFORC file (requested in the Keyword deck). It is our recommendation to use *CONTROL_CONTACT with *shledg*=1 and remove it as a default.

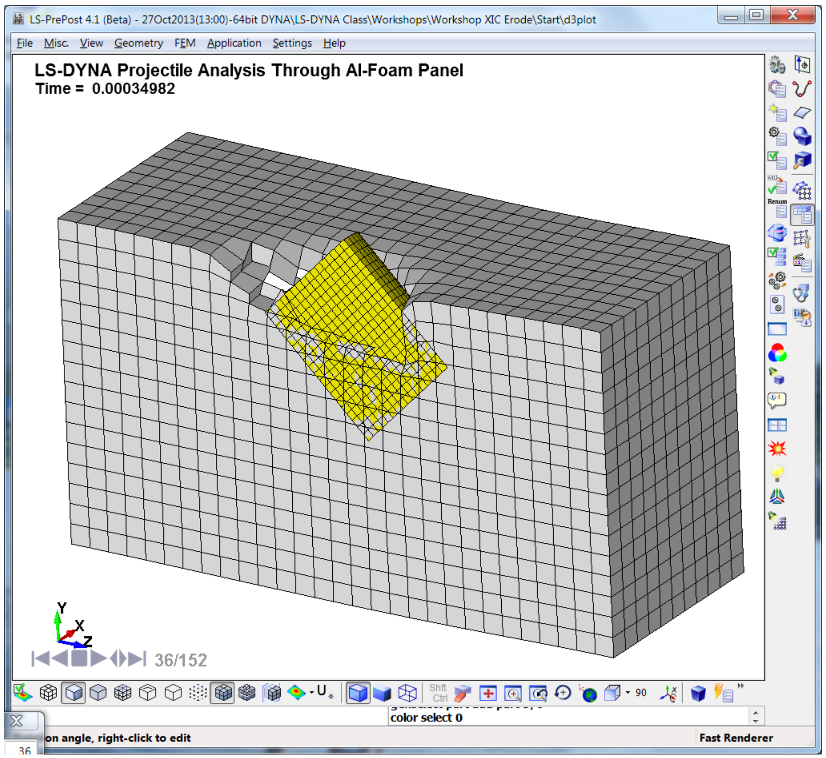


9.2.3 Contact when things ERODE

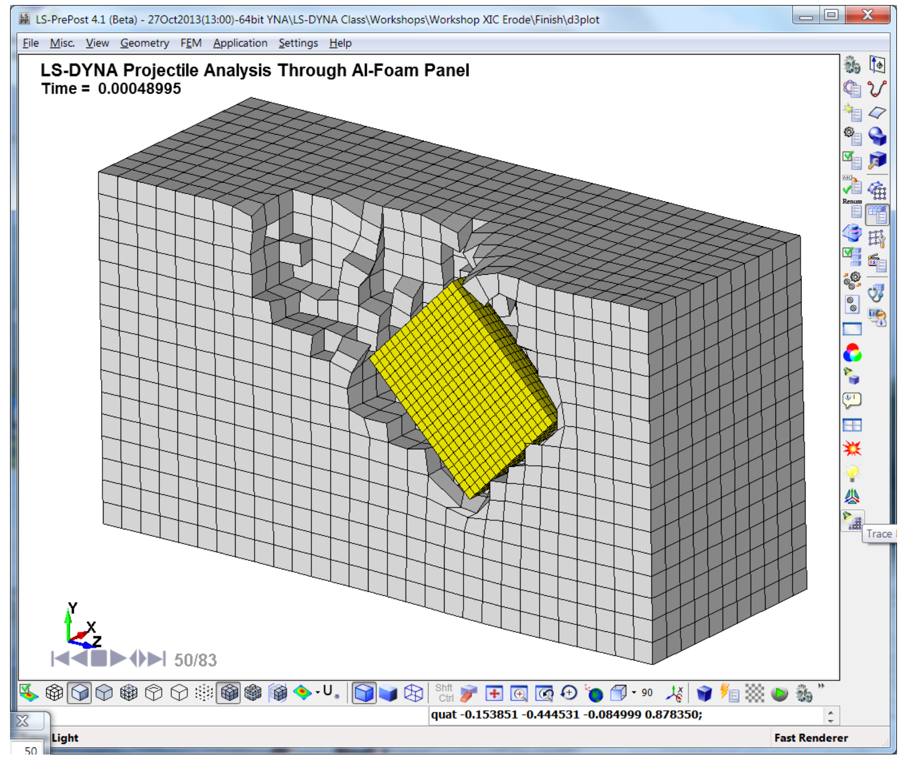
For numerical efficiency, the contact algorithm only looks at free edges and faces. If element erosion occurs (i.e., element failure), the standard contact algorithm is not prepared to look for contact on these newly generated faces. If one knows ahead of time, then the contact can be switched to _ERODING with erosop=1. The *erosop* variable is required to allocate memory storage for the newly created element surfaces. *(**Update:** As LS-DYNA evolves so does its Keyword defaults; as of 2019, the default is erosop=1.)*

This option should ***not*** be evoked for most element erosion situations since it is not very numerically efficient. For example, plate models are typically just fine without using element erosion, but as shown below, it is critical. Oh BTW, _MORTAR contact supports erosion as a default. This capability exists in the Development versions as of 2019 but expect to see it in R11.2.

_AUTOMATIC_SINGLE_SURFACE



_ERODING_SINGLE_SURFACE



Analyst's Note: An example model for eroding contact (as shown above) is provided for informational purposes within the Class References / Contact / Erosion file folder. It is provided for educational purposes.

9.2.4 MORTAR Contact

Just add… _MORTAR and all is good.

“Mortar contact is a penalty based segment-to-segment contact with finite element consistent coupling between the non-matching discretization of the two sliding surfaces”

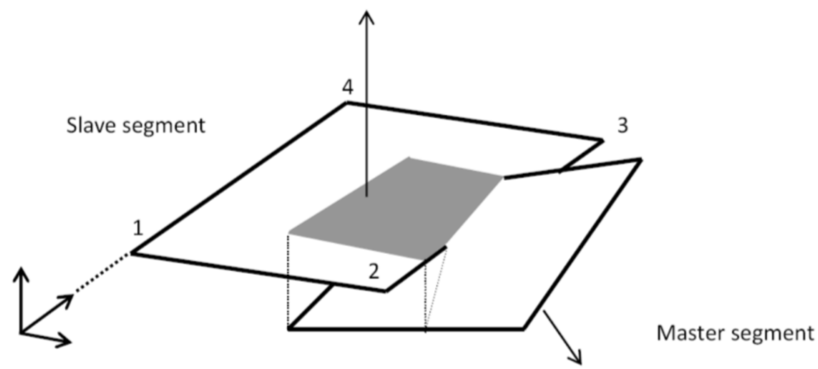
Thomas Borrvall, DYNAmore Nordic AB

Developer of _MORTAR contact and much more…..

In other words, no soft=2 setting is needed and pretty-much all other varied and sundry contact settings can be left un-touched.

For _MORTAR contact, the ignore option (card 9) has special significance and enables many useful capabilities.

Although _MORTAR was designed for implicit, it also provides more robust contact capabilities than the default settings for the standard _AUTOMATIC contacts typically used in explicit analysis. It is numerically costly and might be significant (e.g., 2x run time) but it can solve many contact challenges that with non-MORTAR formulations might require many hours of tweaking of various settings (which often requires a two-day course to go over their meanings).



3.4 Initial penetrations

Initial penetrations are always reported in the message files, including the maximum penetration and how initial penetrations are to be handled. The IGNORE flag governs the latter and the options are

IGNORE=0	Initial penetrations will give rise to initial contact stresses, i.e., the slave contact surface is not modified
IGNORE=1	Initial penetrations will be tracked, i.e., the slave contact surface is translated to the level of the initial penetrations and subsequently follow the master contact surface on separation until the unmodified level is reached
IGNORE=2	Initial penetrations will be ignored, i.e., the slave contact surface is translated to the level of the initial penetrations, optionally with an initial contact stress governed by MPAR1
IGNORE=3	Initial penetrations will be removed over time, i.e., the slave contact surface is translated to the level of the initial penetrations and pushed back to its unmodified level over a time determined by MPAR1
IGNORE=4	Same as IGNORE=3 but it allows for large penetrations by also setting MPAR2 to at least the maximum initial penetration

9.2.4.1 _MORTAR _ERODING {Built-In}

Just a little heads up, _MORTAR also handles element erosion.

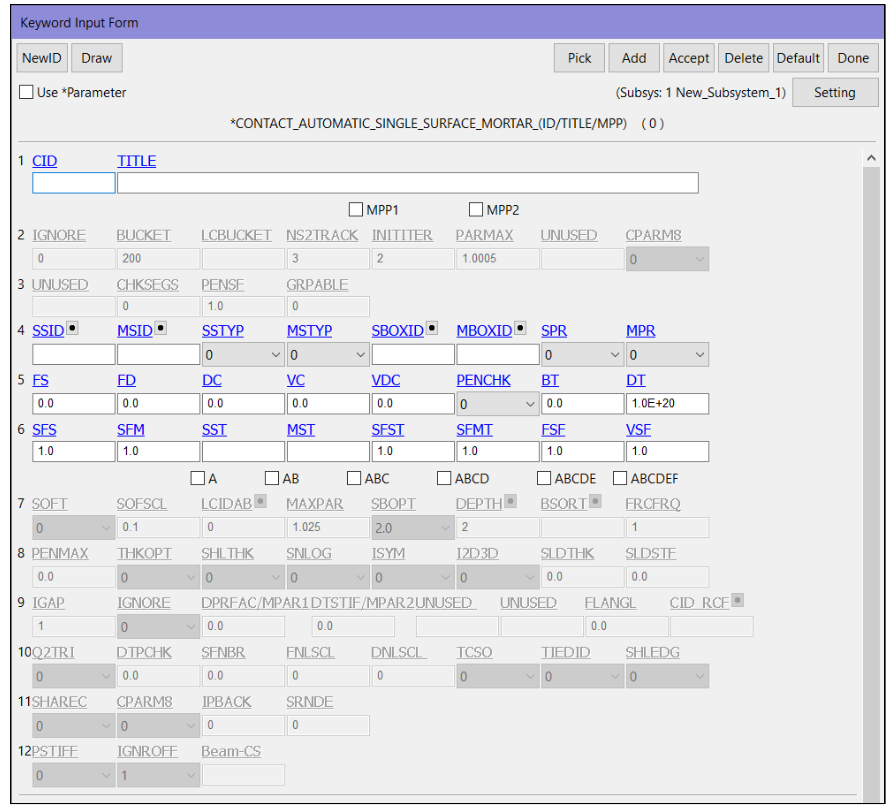
MORTAR Contact (continued)

Contact Card Discussion

This screen comes from LSPP and is what new users must face in trying to figure out the contact card. Although it has _MORTAR option, it is identical to the regular un_MORTAR card. In standard _AUTOMATIC[1] contacts. In general, the friction setting (FS) is set to a reasonable value with all others set to their defaults. Given that the modeling of complex systems is difficult, it is common to have some overlaps between adjacent components. Setting the ignore=1 is often very effective. This is done by checking ABC and then ignore=1.

If a non _MORTAR contact is used, then the recommended settings are to check ABC and set soft=2 with ignore=1.

Contact can be simple and it can be hard but there are general procedures to follow. This will be discussed in the following workshops.



[1]What does _AUTOMATIC mean? Many years ago, prior to _AUTOMATIC, users had to define which faces (i.e., normals) would contact against the opposing faces (*Surfb / Surfa*) and would have to be carefully defined or no contact would occur. With _AUTOMATIC, faces just contact faces or PARTs just contact PARTS or etc. This more robust formulation is more numerically costly but so much easier.

9.3 Contact Energy

The default LS-DYNA contact uses the penalty method. That means that the contact is enforced by the using springs. The use of springs implies that the opposing surfaces will have some net displacement interference to create the driving force that keeps them apart. This leads to a net positive contact energy since $\text{Energy}_{\text{Contact}}=\text{Force}_{\text{Contact}}*(\Delta\text{Interference})$. From a simulation viewpoint, this energy is not physical since contact results in an increase in the potential energy of the contacting parts and not in energy build-up within the interface. Although this energy is conserved within the simulation it is nevertheless somewhat detrimental since it represents energy that should have gone toward physical deformation of the part (i.e., Internal Energy) or toward Kinetic Energy of the structure.

A loose guideline is that this contact energy (see *DATABASE_GLSTAT / Sliding Energy) is that it should be no more than 10% of the maximum Internal Energy of the analysis. This guideline simply is saying that this non-physical energy is a bit dangerous and if you are doing critical work, one might want to have much less. This same guideline applies to Hourglass Energy However, it depends on lots of factors, such as the number of contacting surfaces in your model, the velocity of the impact, etc. If this energy is high, most likely it is due to interpenetrations or a large mesh size w.r.t. the physics of the problem. Within LSPP, there exists a tool for checking of contact penetration (Application / Model Checking / General Checking / Contact Check). Also, one should take a look at the Class Reference notes under Contact / Contact Energy.

If one is using friction within your contacts, then of course, the Sliding Energy will be much greater. It is not a bad idea to make one run with friction turned-off just to assure oneself that the contact setup is performing correctly.

Sliding Energy = Contact Interface Energy = (sliding (which only with friction) + interpenetration)

The whole concept of debugging contact is tricky and requires a bit of reading and experimentation. It is recommended to the student to read the sources provided in the Class Reference Notes when contact problems arise.

9.3.1 A Brief Comment on Energy Reports

Energy reports are quite useful to understand and verify the mechanics of the simulation. A bit more is said about energy plots in Section 18.5. The general gist is that everything revolves around the internal energy (IE). The IE is the stored energy in the mesh and if you have plasticity, most of your IE will be plastic damage. Since IE controls your damage or what you are often trying to investigate, other energy components are compared against the IE value; for example, hourglass energy (HE) or sliding interface energy (SIE). In the following workshop, we'll be plotting IE against HE and SIE with the idea that these values should be < 10% (at least) as compared to the IE. The reality is that HE and SIE are fictional energies of the numerical process and ideally we would like to see close to 0.0%.

9.4 Workshop: 13 - Understanding Basic Contact Mechanics

Introduction and Objective: Often, the challenge to modeling contact is not setting up the contact model but checking the results. In this workshop, the goal is to verify the contact behavior, plot the sliding interface energy and then also the contact force between the impacting bodies and the pipes hitting the rigid wall. The contact behavior should make good engineering sense. This workshop requires the student to do some independent homework by reading the *Keyword Manual* and about sliding interface energy at www.*DYNAsupport.com*.

Workshop:

Part I-A: Let's Get Started

1. Run model (Basic Contact – Pipe-on-Pipe Contact – Start.dyn) and look at contact behavior.
2. One will notice that the contact behavior is not logical. Let's debug! Create *DATABASE_ files rcforc, sleout and rwforc (read Keyword Manual) by removing the "$". Set *CONTROL_ENERGY card to the required setting (*Note setting about slnten*) to calculate the hourglass energy, rerun model. Then make an energy plot using data within the glstat file. Then one can extract the rcforc, sleout and rwforc data. Please enter this data into the Table at the end of this Workshop.
3. Slight problem – no rcforc output for _SINGLE_SURFACE? Activate _FORCE_TRANSDUCER and rerun model and repeat 2. above. For extracting maximum contact force, filter the data in LSPP using a SAE filter at 1,000 HZ. Please enter this max. RCFORC and RWFORC data into the Table at the end of this workshop.

Keywords Discussed

***CONTACT**

_AUTOMATIC_SINGLE_SURFACE

_AUTOMATIC_SINGLE_SURFACE_MORTAR

_FORCE_TRANSDUCER (see General Remarks under *CONTACT)

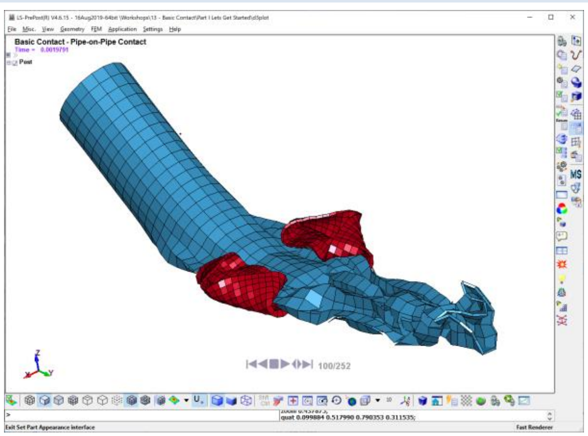
***DATABASE**

_GLSTAT, _RCFORC, _SLEOUT, _RWFORC

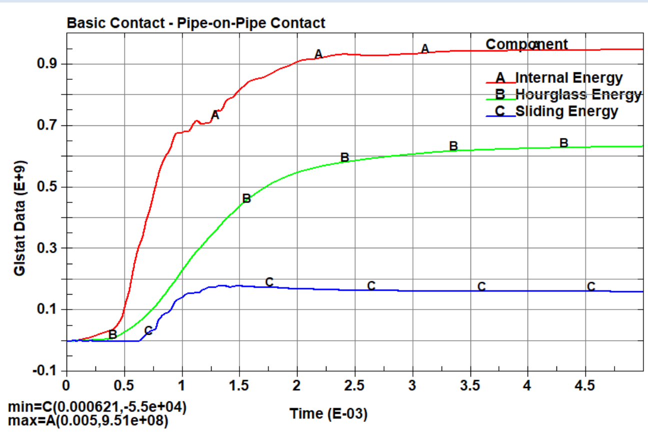
_BINARY_INTFOR

***RIGIDWALL**_GEOMETRIC_FLAT

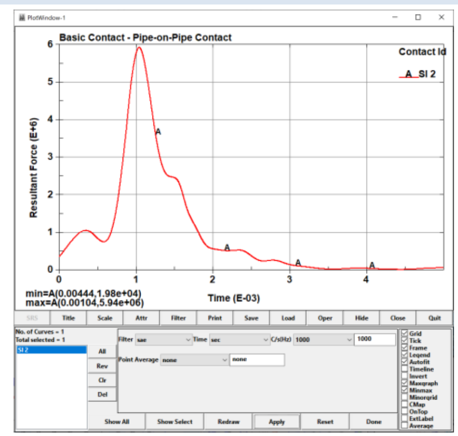
Part 1-A



Energy Plot



Contact Force (Filtered)



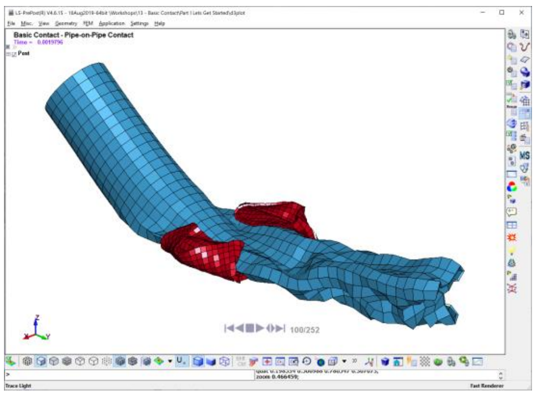
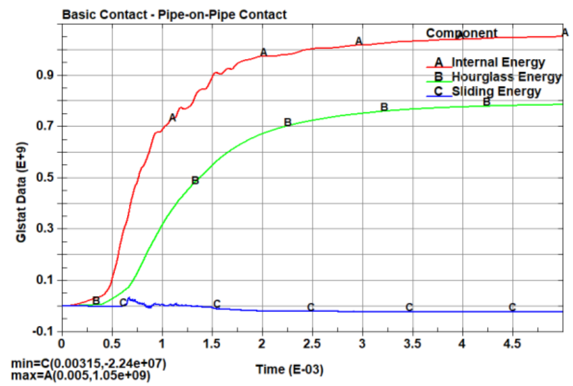
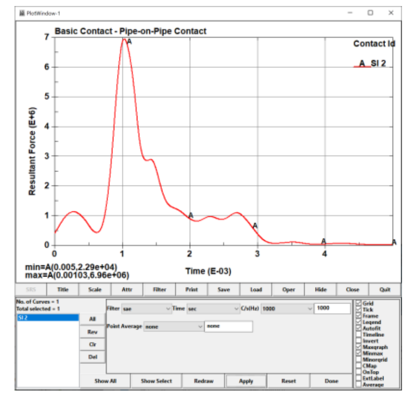
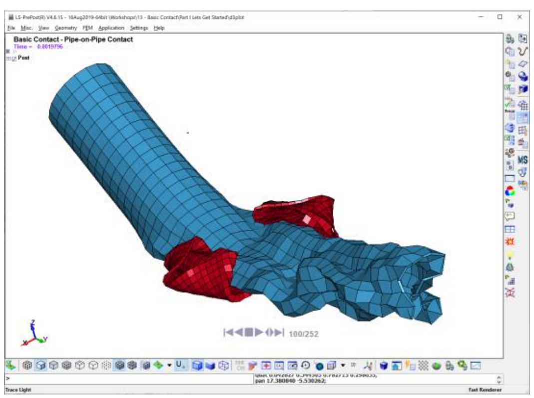
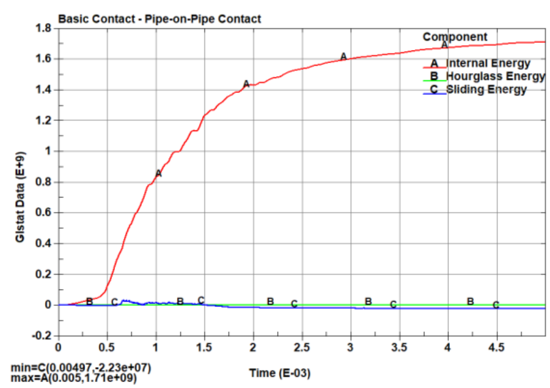
Workshop: 13 – Understanding Basic Contact Mechanics (continued)

Part I-B

- Add *soft*=2 to the *CONTACT_AUTOMATIC_SINGLE_SURFACE card 4 and generate new data for your observations (add to Table at the end of this Workshop).

Part I-C

- Change shell element formulation to *elform*=-16 and then add to Table.

Part 1- B	Energy Plot	Contact Force (Filtered)
		
Part 1-C	**Energy Plot**	**Contact Force (Filtered)**
		Well, you guys know what you will see..not that interesting….

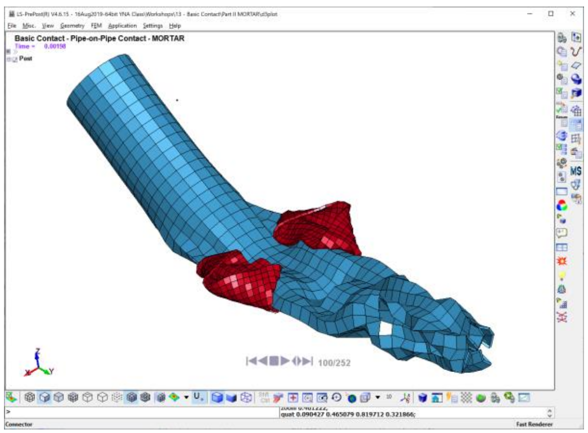
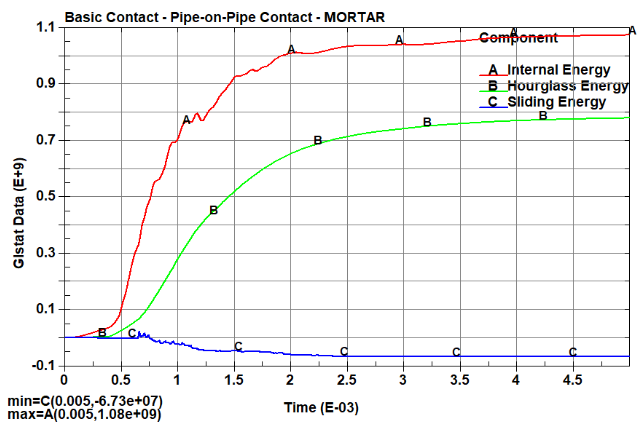
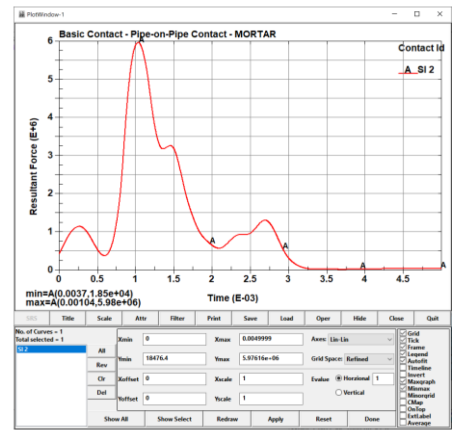
Workshop: 13 – Understanding Basic Contact Mechanics (continued)

Part II-A – MORTAR Contact

- Add _MORTAR to the contact. That is it. Plot data and fill out Table. You will notice that it runs slower since _MORTAR is a very numerically intensive contact algorithm.

Part III {Extra Credit}

- 1st you'll need to change the element formulations to *elform*=-16 and then change the contact formulation from _SINGLE to SURFACE_TO_SURFACE contact and remove _TRANSDUCER card. Be prepared to think about how a _SINGLE_SURFACE contact works and what is required to simulate this behavior with _SURFACE_TO_SURFACE if at all possible? A final runnable model is within the Workshop folder under Extra Credit.

Part II-A	Energy Plots	Contact Force (Filtered)
		

Analyst's Note: One will notice that the shape of the tube changes from simulation to simulation. It just indicates how random and complex the simulation is rather than any inherent numerical wrongness.

9.4.1.1 Student Notes for Workshop – Understanding Basic Contact Mechanics

Record your run data in this table to provide a quick comparison of the data. The sliding interface energy (SLEOUT), hourglass energy (HG Energy), rigid wall contact force (RWFORC) and contact force (tube-to-tube) are the maximum forces that are shown in your plots within LSPP. A final filled out table in provided in the Workshop folder as a PDF.

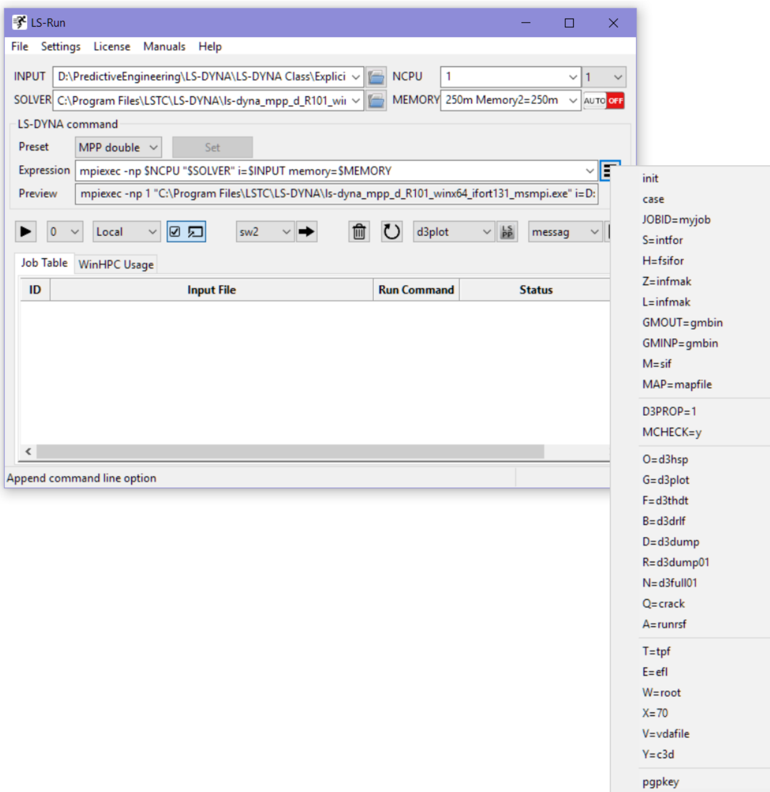
Run	Contact	Penetration?	SLEOUT	HG Energy	RCFORC	RWFORC
Part I-A	_AUTOMATIC					
Part I-B	_AUTOMATIC w/ soft=2					
Part I-C	*elform*=-16 w/ soft=2				*{It is just fine…a lot of picking…you guys know how to filter data…and nothing really changes.}*	
Part II-A	MORTAR (no soft)					
Part III[1]	{see below}					

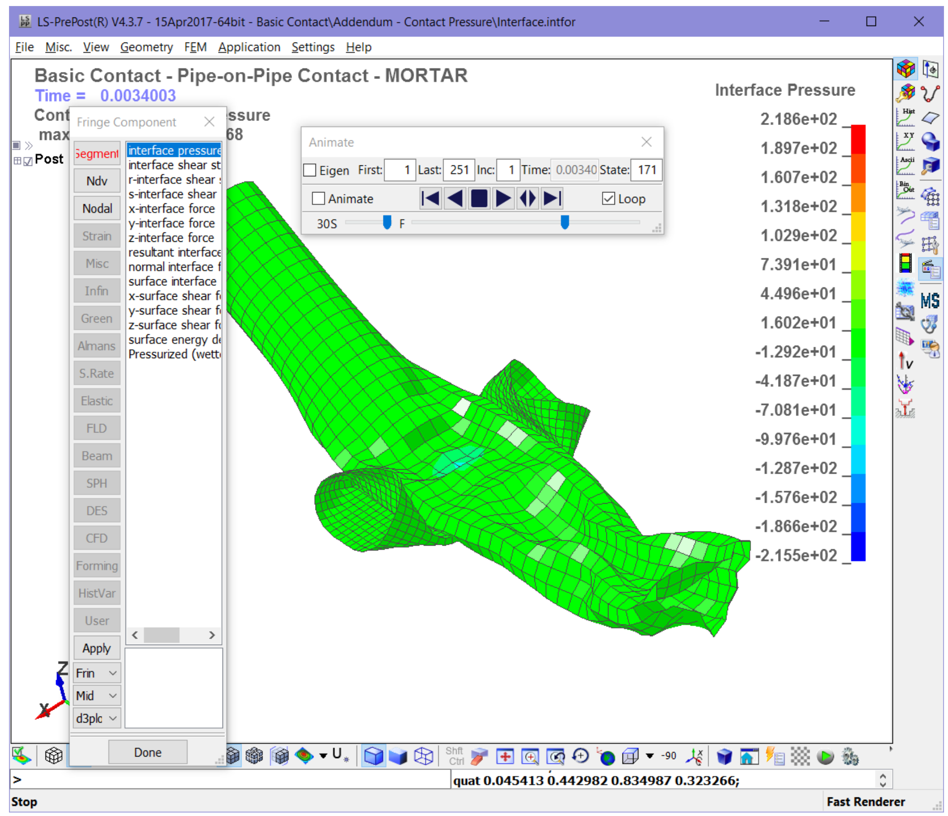
[1]Extra Credit: If one flies through this task, take the Keyword deck from Part II-A and change the contact from _SINGLE_SURFACE to SURFACE_TO_SURFACE and comment out the _TRANSDUCER card since one won't need it. However, one may realize that you still need a "SINGLE_SURFACE" contact to handle a distinct contact situation.

9.4.1.2 Addendum to Workshop: Contouring Contact Pressures

LS-DYNA doesn't automatically generate interface pressures developed during contact. To obtain this information, three items are required: (i) *DATABASE_BINARY_INTFOR DT={time interval} must be set; (ii) Print flag(s) on card 1 of *CONTACT_ must be set to SPR=1 and/or MPR=1; and (iii) and upon analysis (LS-Run), one must click on the drop down list and set s=(provide your own filename).intfor. This creates a *separate binary file* can then be read by LSPP as a separate post-processed file (LSPP – File / Open / Interface Force File).

This brings up one limitation of the _SINGLE_SURFACE contact is that it is then difficult to separate or view interface pressures between different components. The _FORCE_TRANSDUCER is only for the RCFORC and not interface forces (verified on January 2020). In the example shown below, we use the Extra Credit Model with separately defined contact interfaces as our starting point (Part II-C).





9.5 Workshop: 14 - Beam-to-Beam Contact

Problem Statement: A load of pipes is dropped onto a cargo rack. The model starts out using our standard default *CONTACT … (_SINGLE_SURFACE with soft=2).

Theory: The basic robust *CONTACT_ with *soft*=2 is a blend of robustness and numerical efficiency. Although for true speed, one might use just defaults, other small issues might occur with your contact behavior negating any solution speed advantage. It is always a bit of a tradeoff. To handle beam-to-beam contact, LS-DYNA has *CONTACT_AUTOMATIC_GENERAL. This is a general contact formulation (*it is not segment-to-segment (no soft=2)*) that has additional search logic to check for beam-to-beam contact and also adjusts the depth variable for increased bucket sorts and penetration checks. It is very numerically intensive and should be reserved to contact situations where edge-to-edge, beam-to-beam and other tricky contact situations arise. It should be noted that _MORTAR contact handles beam-to-beam contact; however it is several times more expensive than _GENERAL.

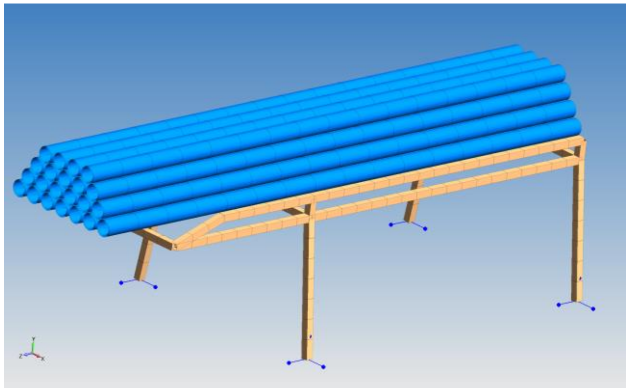
Script:

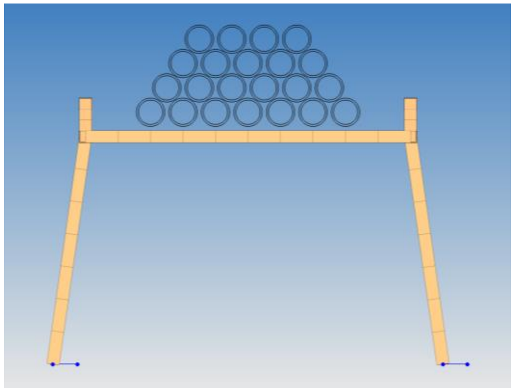
- Run model Beam-to-Beam Contact - Start.dyn and take a look at the results;
- Based on Theory, modify the existing – Start file to use _GENERAL contact. Rerun the model.
- That is it. Very simple, very direct.

Extra Tasks:

- Switch the *CONTACT to a _MORTAR formulation. How much slower is it?
- Ok, switch it to _GENERAL (heck, we don't have all day to run this model) and proceed with the following steps
- Extend the run time (*CONTROL_TERMINATION) by 5x and check out the physics of the problem. Are we on earth or in outer space? What happens when you add gravity (i.e., *LOAD_BODY_Y (read the manual (RTM) and also Remark 1 about how it works, i.e., opposite of your intuition) and then modify the existing *DEFINE_CURVE to create your "gravity".

Cargo Rack and Pipes





Visulization Tips and Tricks:

When you view the Keyword file in LSPP, the default is to show beam elements as lines. To show their shape, go to View / Beam Prism.

When viewing the d3plot file, one will notice that the beams will vibrate oddly. That is because the d3plot file does not contain beam element's shape. Thus, load the Keyword file into your d3plot session and then it will look "clean".

Analyst's Note: Yes, someday computers will be fast enough and all contact will just be done with _MORTAR. But for now, efficient use of contact formulations can make the difference between your model solving in one hour as versus two or three hours.

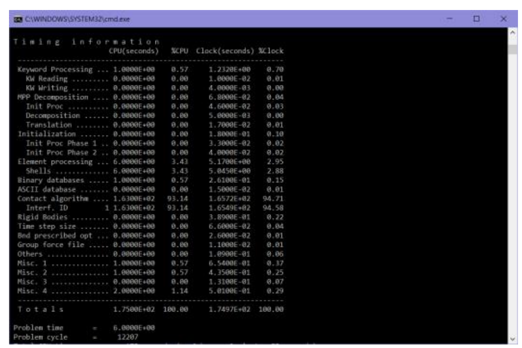
9.6 Miscellaneous Comments on Contact

9.6.1 Contact Numerical Efficiency or Why Not All _MORTAR All the Time?

Although _MORTAR contact solves many contact pathologies it does come at a price. Prior to _MORTAR contact, the order of increasing numerical cost for contact was _SINGLE, _AUTOMATIC_SINGLE and then _GENERAL. With perhaps _GENERAL being 2x more costly than _AUTOMATIC. Nevertheless this all pales in comparison with _MORTAR. The table below compares _SINGLE_SURFACE to _SINGLE_SURFACE_MORTAR for the small drawing model (Workshop – Using Rigid Bodies).

LS-DYNA Version	**Run Time – 1 CPU-Core**		**_MORTAR Speed x**
	_MORTAR Contact	**No _MORTAR Contact**	
MPP Double-Precision R8.1.0 (Released)	530	19	27x (slower)
MPP Double-Precision R9.1.0 (Released)	528	20	26x (slower)
MPP Double-Precision R10_116442	169	15	11x (slower)
MPP Double-Precision DEV_116523	186	15	12x (slower)
MPP Double-Precision R10.1 (Released)	91	11	9x (slower)
MPP Double-Precision Dev 126097	109	10	11x (slower)
MPP Double-Precision R11.0	140	14	10x (slower)
MPP Double-Precision Dev 137713	144	13	11x (slower)

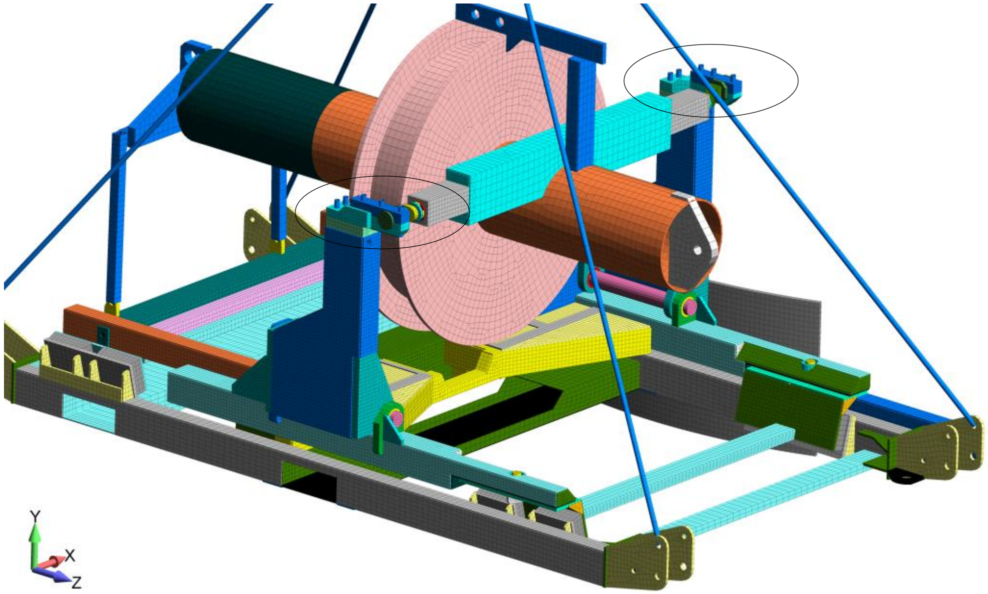
Analyst's Note: The example shown above was for a problem where 95% of the computational cycles was for contact. It is a very special case where the contact formulation dominated the Run Time. In many common models, contact may only require 10% or less and thus the effect of _MORTAR contact would be far less than shown in the Table.



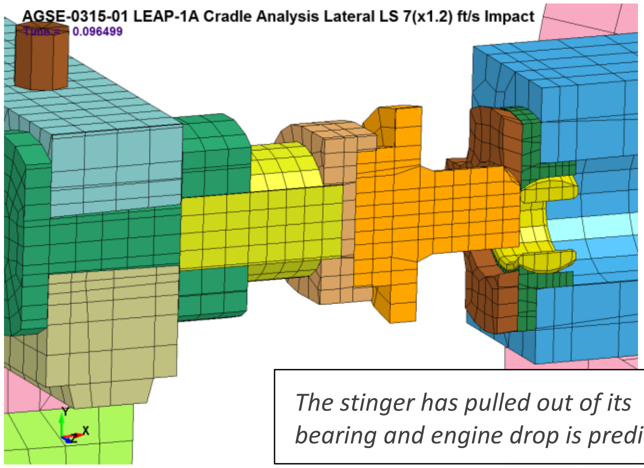
```
C:\WINDOWS\SYSTEM32\cmd.exe

 T i m i n g   i n f o r m a t i o n
                        CPU(seconds)   %CPU  Clock(seconds) %Clock
  ----------------------------------------------------------------
  Keyword Processing ... 1.0000E+00    0.57    1.2320E+00    0.70
    KW Reading ......... 0.0000E+00    0.00    1.0000E-02    0.01
    KW Writing ......... 0.0000E+00    0.00    4.0000E-03    0.00
  MPP Decomposition .... 0.0000E+00    0.00    6.8000E-02    0.04
    Init Proc .......... 0.0000E+00    0.00    4.6000E-02    0.03
    Decomposition ...... 0.0000E+00    0.00    5.0000E-03    0.00
    Translation ........ 0.0000E+00    0.00    1.7000E-02    0.01
  Initialization ....... 0.0000E+00    0.00    1.8000E-01    0.10
    Init Proc Phase 1 .. 0.0000E+00    0.00    3.3000E-02    0.02
    Init Proc Phase 2 .. 0.0000E+00    0.00    4.0000E-02    0.02
  Element processing ... 6.0000E+00    3.43    5.1700E+00    2.95
    Shells ............. 6.0000E+00    3.43    5.0450E+00    2.88
  Binary databases ..... 1.0000E+00    0.57    2.6100E-01    0.15
  ASCII database ....... 0.0000E+00    0.00    1.5000E-02    0.01
  Contact algorithm .... 1.6300E+02   93.14    1.6572E+02   94.71
    Interf. ID         1 1.6300E+02   93.14    1.6549E+02   94.58
  Rigid Bodies ......... 0.0000E+00    0.00    3.8900E-01    0.22
  Time step size ....... 0.0000E+00    0.00    6.6000E-02    0.04
  Bnd prescribed opt ... 0.0000E+00    0.00    2.6000E-02    0.01
  Group force file ..... 0.0000E+00    0.00    1.1000E-02    0.01
  Others ............... 0.0000E+00    0.00    1.0900E-01    0.06
  Misc. 1 .............. 1.0000E+00    0.57    6.5400E-01    0.37
  Misc. 2 .............. 1.0000E+00    0.57    4.3500E-01    0.25
  Misc. 3 .............. 0.0000E+00    0.00    1.3100E-01    0.07
  Misc. 4 .............. 2.0000E+00    1.14    5.0100E-01    0.29
  ----------------------------------------------------------------
  T o t a l s            1.7500E+02  100.00    1.7497E+02  100.00

 Problem time       =    6.0000E+00
 Problem cycle      =     12207
```

9.6.2 Why Paying Attention to the Contact time Step is Important

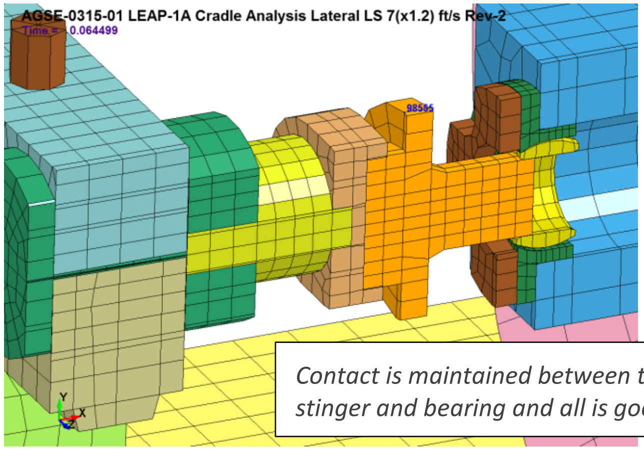
"The LS-DYNA time step size should not exceed X.XXXE-XX to avoid contact instabilities." If your solution time step is greater than this value, tread with caution. Why? The tale of a simulation where the dummy engine (the object in the middle) was predicted to separate from its stand and fall (a very bad event) using a timestep=-2.222E-06 and then predicted to stay attached to the stand using a timestep=1.111E-06. Of course, a warning message was provided saying that the timestep should not exceed 6.724E-07.



As the structure is swung against a barrier, the engine (the center object) must stay pinned between its two supports (circled). The engine is held in place by a stinger (top of support arm) that is inserted into a spherical bearing (engine attachment component). If during impact this stinger pulls out from the spherical bearing, the engine will fall and everyone becomes quite upset. The objective of this simulation is to demonstrate that the design can support the engine during impact without failure.

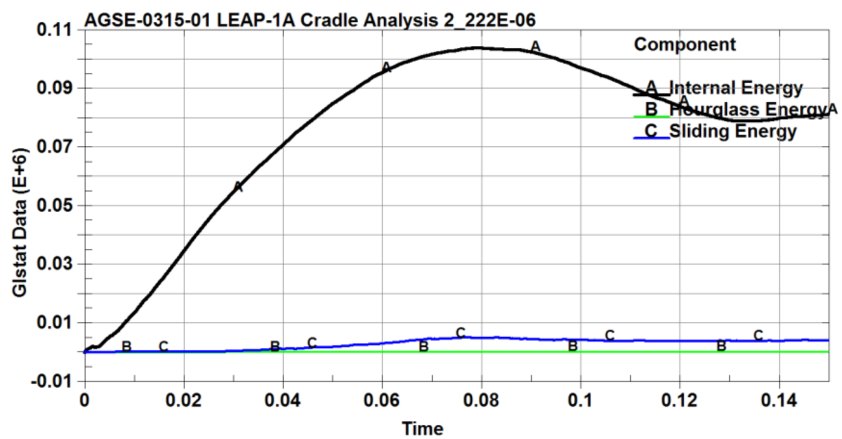


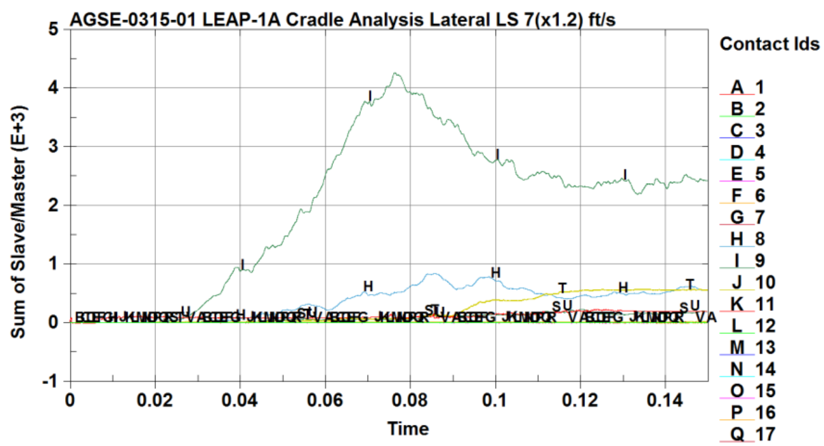
The stinger has pulled out of its bearing and engine drop is predicted

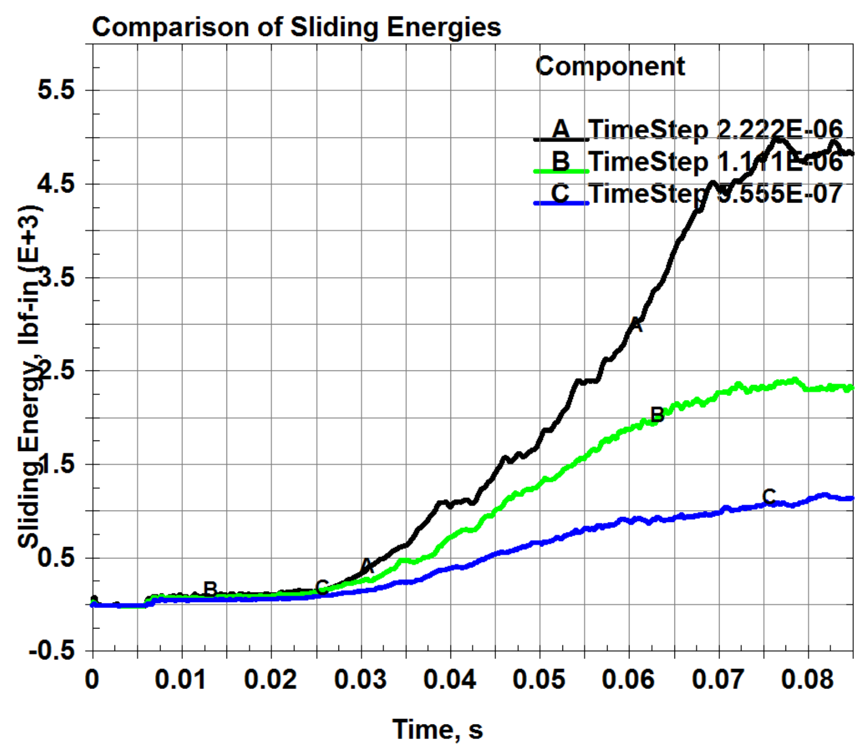


Contact is maintained between the stinger and bearing and all is good.

One technique that was learned in debugging this problem of a false prediction was to pay closer attention to the sliding interface energy (see discussions in this document on **Contact Energy**). The logic works this way – if contact is enforced by springs then each contact couple will have a bit of energy in maintaining the correct position of the interfaces. If the contact becomes unstable (surface-to-surface interpenetrations), then one can imagine that the contact energy will be greater than that which would occur under clean, stable contact behavior. For example, the prior project had decent energies during checkout with the global and local sliding energies at reasonable norms. However, as the timestep was decreased, the overall sliding energy decreased. Please note the large drop between the timestep at 2.222E-06 and 1.111E-06 and this drop in energy ties with the false prediction of "engine drop".



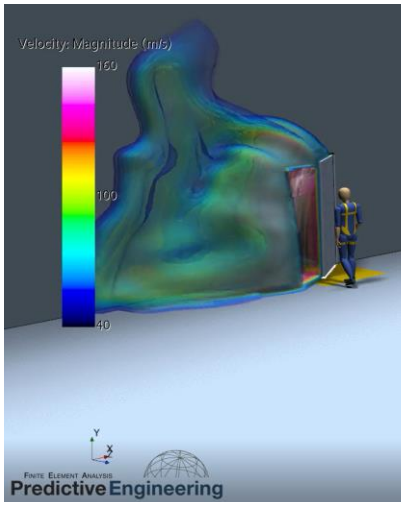


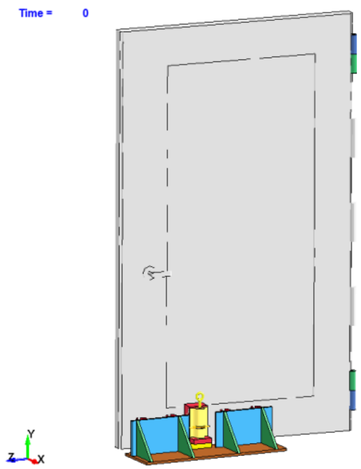


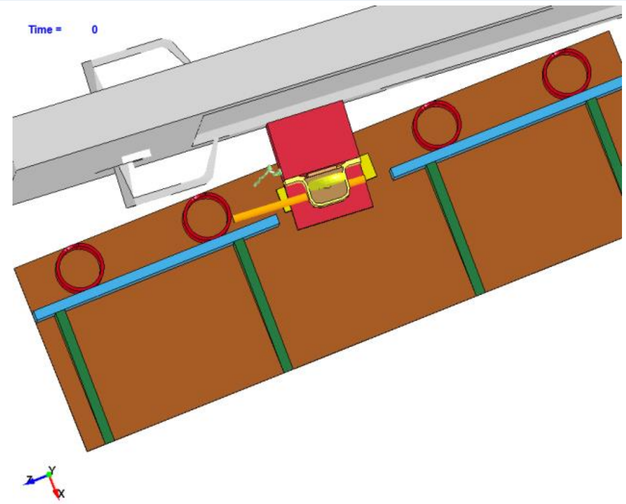
9.6.3 Instructor Led Workshop: 7A – Sliding Interface Energy – Not Always About "Rules-of-Thumb"

"Rules-of-Thumb" are guidelines and "good practices" but they don't replace "thinking". This analysis followed the guidelines but resulted in some suspicious behavior. After much time chasing "ghosts", I went back and looked at the sliding_interface_energy for each contact and determined and then set the contact Card A to depth=5. With this improved edge-checking, the model generated the expected results. Look at video for action and comments.

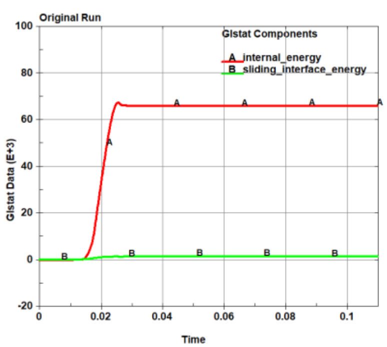
CFD Analysis (Door Velocity)	Door Model	Door Slam Retention Mechanism

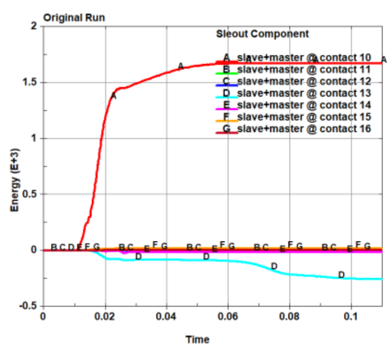


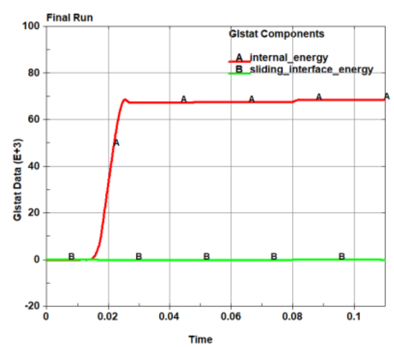


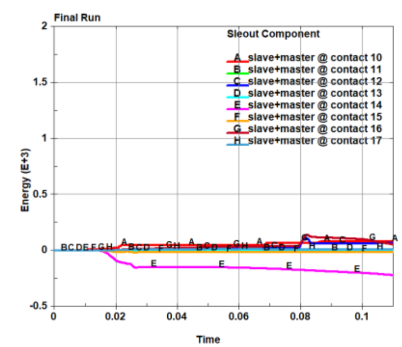


Original Run – Fly-Away Pin "Sticks"		Final Run – Correct Behavior – Fly-Away Pin "Flies"	
Internal Energy / SIE	SLEOUT (*Surfa+Surfb*) – ~1,700!!	Internal Energy / SIE	SLEOUT (*Surfa+Surfb*) – ~All Good









Analyst's Note: "Rules-of-Thumb" can be dangerous if one just follows them without thinking. We knew that the 3/8" diameter pin had to fly-away given the mechanics but the model was showing the pin sticking and plastically deforming. What was wrong? Added depth=5 to the contact card and all was good! However, the real trick was understanding the mechanics of the door retention mechanism and knowing that the original model was "cooked".

9.7 Contact Best Practices

Recommendation	Why
_AUTOMATIC_SINGLE or _SURFACE_TO_SURFACE w/*soft*=2 and *depth*=5	This is our recommended backup if greater numerical efficiency is required. The soft=2 option on the optional A card switches the contact formulation to a segment base and adjusts the contact stiffness and then depth=5 for improved edge contact behavior. Although _SINGLE_SURFACE is robust, it works best when all the parts are of one element type (shell or solid). With mixed element sets, one should carefully check the sliding_interface_energy. It is difficult to have "one recommendation" with many aspects of LS-DYNA.
*CONTROL_CONTACT, *shledg*=1	*shledg*=1 removes the shell edge projection.
Check your model for penetrations (LSPP – Application / Model Checking / General Checking / Contact Check). Keep in mind that _MORTAR, as a default, automatically accommodates small penetrations in explicit (*ignore*=1).	Be aware of their magnitude since they contribute to negative sliding interface energy, poor contact behavior and spurious contact forces.
For contact between deformable bodies, aim for a uniform mesh pattern.	Contact works by segments and transfers contact forces to the adjacent nodes. If the mesh is coarse, the contact forces will be high and dynamic. The smoother the mesh, the better the contact.
Verification	**Why**
Check "sliding interface energy"	Provides one small verification that contact is working well. Global (all contacts) are provided in GLSTAT while individual contact sliding energies are provided in a separate ASCII file under SLEOUT.

9.8 Mesh Transitions: Tied Contact for Efficient Idealization, Connections, Welding, Mesh Transitions and Etc.

9.8.1 _TIED Contact or Gluing

Given the idealization difficulty of system modeling, the ability to tie together different mesh densities (e.g., hex-to-hex or tet-to-hex), snap together parts along a weld-line or just glue sections together (e.g., plate edge to a solid mesh) is an amazingly useful ability and LS-DYNA provides a very complete _TIED Contact tool box to work with.

As a quick reference, one is always tying nodes to surfaces (i.e., to say segments) and _OFFSET indicates that the *surfa*-side and *surfb*-side do not have to be on the same plane. And with _TIED contacts we do have to pay attention to the *surfa* and *surfb* definition.

This is the short list for all _TIED relationships:

When Tying *Surfa*-Side 3-DOF (Translational)

- *CONTACT_TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET
- *CONTACT_TIED_NODES_TO_SURFACE_OFFSET

Usage: Surfa side nodes have 3 DOF (e.g., solid elements)

When Tying *Surfa*-Side 6-DOF (Translation and Rotation)

- *CONTACT_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET
- *CONTACT_TIED_SHELL_EDGE_TO_SURFACE_*BEAM*_OFFSET

Usage: Surfa and surfb side nodes have 6 DOF (e.g., beam and shell elements)

9.8.1.1 Summary and Recommendations of _TIED Usage

- *Surfa*-side nodes determine your _TIED formulation whether _TIED_NODES_ (3-DOF) or _TIED_SHELL_ (6-DOF).
- Whether to use _OFFSET or _*CONSTRAINED*_OFFSET is based on the *surfb*-side requirements. If the *surfb* is a rigid body, then it is _OFFSET. If rigid bodies or SPC's are not an issue, then the preference is to use _*CONSTRAINED*_OFFSET.
- As a recommendation, the finer mesh should be the *surfa*-side to avoid interpenetrations (see Workshop 16 – *Surfb* Class)

Theory-to-Practice

Tied contacts are not really "contacts" but are kinematic constraints *or* penalty relationships that uses the *CONTACT card entry format. What does this mean?

_OFFSET (and not _CONSTRAINED)

When the keyword _OFFSET is present (and not _CONSTRAINED), the _TIED formulation uses a penalty method (think "springs" just like a standard *CONTACT formulation between two surfaces) to enforce the _TIED connection. When you have "springs" you have the flexibility to tie multiple *surfa*-side nodes to the same *surfb*-side part or segments or tie to rigid bodies (read next section about *why*) or have the flexibility to apply SPC's to your *surfa*-side nodes. One just have to think about the connection as applying multiple "beam" elements, i.e., "springs" between your *surfa*-side nodes and the *surfb* interface. The downside to this _TIED connection is that it uses "springs" and in rare occasions one might see negative sliding interface energy arising from this _TIED connection and it is a bit more numerically expensive. Lastly, since it uses "springs", it means it adds to the stiffness matrix and for implicit analysis, it is just more overhead for convergence. All-in-all, if you can avoid using "springs", avoid using "springs"

_CONSTRAINED_OFFSET

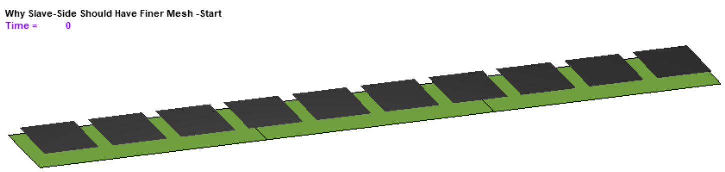
For an explicit analysis, the _CONSTRAINED_OFFSET creates mathematical "ties" between the *surfa*-side's nodes and the *surfb*-side's segments. Think of algebraic formulas that exactly link the *surfa*-side nodes accelerations (explicit) or displacements (implicit) to their opposing *surfb*-side segments (see **Theory Manual** for more detail). Since we numerically "trying", there are limitations such as: i.) Can't tie to rigid bodies; ii.) Can't apply SPC to the *surfa*-side's nodes and iii.) Can't tie multiple _TIED *surfa*-side nodes (multiple *CONTACT_TIED_) to the same *surfb* segment.

*Analysts Questions: How would one tie two rigid bodies together? (Answer – take a peek at *CONTRAINED_RIGID_BODIES*

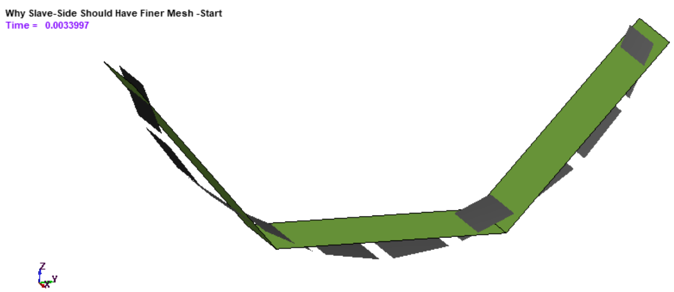
9.8.1.2 Some Important _TIED Concepts to Think About

In _TIED contact, the *surfa*-side node looks for a *surfb*-side segment to "tie" onto whereas the *surfb*-side is free. Below is an example where the top fine mesh is given a pressure load with the ends pinned.

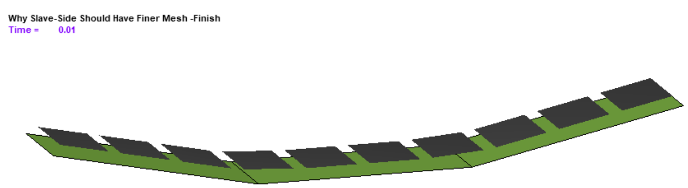
A fine shell mesh (10 elements) above a coarse shell mesh (3 elements)



Surfb-Side Segements (Fine) are Free to Move thru the _Surfa_-Side

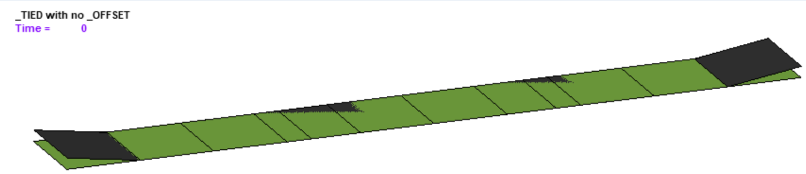


Surfa-Side Nodes (Fine) Tied to _Surfb_-Side Segements (Coarse)

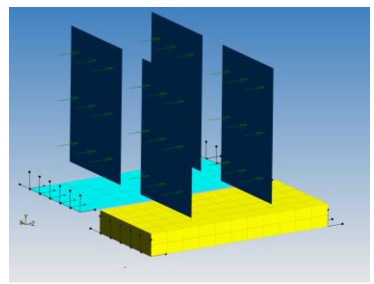


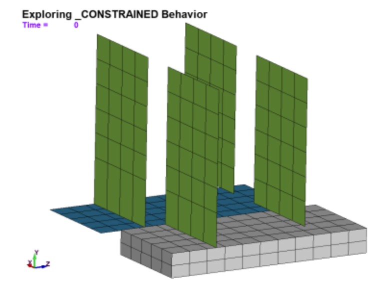
A _TIED connection without _OFFSET moves the *surfa* side nots onto the *surfb*-side segments. Whereas if _OFFSET is included, the *surfa*-side is not moved but it is required to be within a certain tolerance for _OFFSET to work (RTM).

An Example of _TIED without _OFFSET

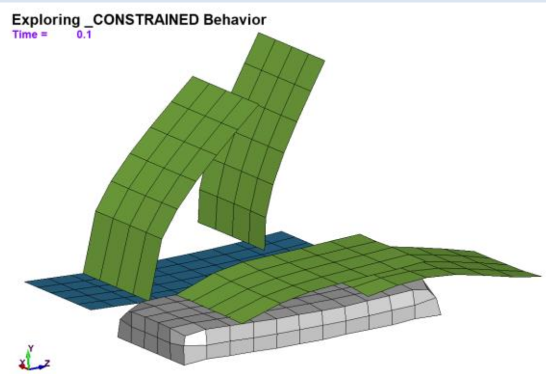


Another Concept About _TIED Connections 6-DOF to 6-DOF is Best





The 6-DOF to 6-DOF Works Best Regardless of _OFFSET or Not



9.8.1.3 What About All Those Other _TIED Formulations?

LS-DYNA has evolved over the years and to maintain compatibility, the code retains prior Keywords. For example, if one does not use _OFFSET, then the *surfa*-side nodes are moved at the start of the analysis onto the plane of the *surfb*-side segments or onto the *surfb*-side surface. Of course, if your *surfa*-side is planar with the *surfb*-side, then nothing is moved. This methodology was required for the constraint based formulation (kinematic constraints and no "springs"). As the code advanced, a new option was developed where constraints could be used with offsets, that is, _CONSTRAINED_OFFSET. As for the keywords the reference SURFACE_TO_SURFACE it is for flexibility since the code just grabs the nodes attached to the *surfa*-side segments and behind the scenes, it is just _SURFACE to _NODES.

9.8.1.4 For Those Believers in the KISS Method of _TIED Contact

Now, after years of wondering why it was necessary to have so many different _TIED contact formulations, it was reasoned that at the end-of-the-day, one could handle the full range of _TIED possibilities with two _TIED contact formulations. This is the new go-forward approach for both explicit and implicit analyses. Although the two Workshops are using the older formulations, the student can swap out these formulations after a successful completion and demonstration to themselves that the same results can be obtained.

There are some very sneaky situations where the _CONSTRAINED_OFFSET will not work well. It has to do with sharing a *surfb* surface (i.e., *surfb* segments). Since it is kinematic based it wants exclusivity to the *surfb* surface whereas the standard _OFFSET method (i.e., penalty) is happy to share *surfb*s.

Keyword	Card Option	Comment
_TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET	*ipback*=1	General purpose _TIED contact for tying solids (3 DOF nodes) to shells and solids. If the tie interface is coplanar or offset, the _OFFSET feature handles both situations. Plus being _CONSTRAINED it eliminates any problems with a spring formulation. The *ipback*=1 option is useful if one needs to tie to rigid bodies. If a rigid body is present in the tie definition, then the formulation is automatically switched to a penalty based algorithm.
_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET	*ipback*=1	With SHELL_EDGE option, one ties all 6 DOF of the *surfa* side nodes and if one needs to tie to rigid bodies, the *ipback*=1 option is available.

Analyst's Note: This recommendation comes after many years of _TIED introspection and troubleshooting. It is a work in progress but given the maturity of the LS-DYNA program, the go-forward quest is to only use only two _TIED formulation to handle all tied contacts. The advantage of the _CONSTRAINED formulation is that springs are not use and the constraint relationships are created between the adjacent surfaces. This eliminates the possibility of forming negative sliding interface energy and assures a clean tie that will work well for explicit and implicit analyses.

9.8.2 Workshop: 15A - Tied Contact for Solids (3 dof) _TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET

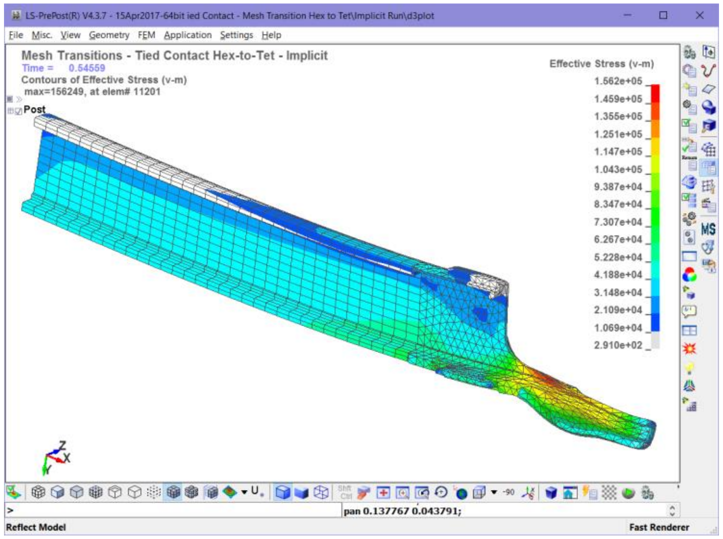
Background for TIED Contact Analysis:

The advantages of using a hex mesh for explicit work centers on better shape control during large deformation and the ability to maintain a larger time step. The last item is often pivotal in keeping your solution running fast without having to add excessive mass if automatic mass scaling is invoked (*CONTROL_TIMESTEP (DTMS = negative timestep value)). In the implicit world, the use of hex elements are desired for the ability to provide equivalent stress mapping using far less nodes (eight node brick versus the use of five 10-node tetrahedrals to fill the same space or 8/26 nodes) and often times, cleaner stress contours. This workshop shows how to setup the mesh transition and run the analysis using the implicit solution.

Simulation Objective: We are trying to simulation a static analysis using a dynamic approach. To obtain a quantitative estimate of how our analysis approaches static equilibrium, we will plot kinetic energy (KE) and internal energy (IE). A true static analysis will have zero KE.

Tasks:

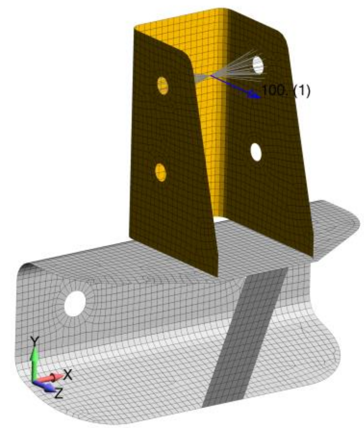
- Start with opening up the Keyword deck: Tied Contact - Mesh Transition Hex to Tet - Start.dyn and inspect the Keywords. It is setup for an explicit run. Input the settings required to enable _TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET. For quick and dirty, let's start with using *PART # for both *surfa* and *surfb* definitions.
- Run model and wait for it to finish. Better idea, apply CMS (conventional mass scaling). What value? Open model in LSPP and assess the time step Application / Model Checking / etc. Apply reasonable value of -2.222e-08 that will give you something under <10% mass scaling. Why 2.222? When multiplied by 0.9, one has a round number. Run model.
- Inspect model, check applied force (plot SPCFORC) and look at ratio of kinetic/internal energy. Is the model at static equilibrium or are dynamic effects present?
- Now, let's verify that the _TIED contact is working as intended. Open up model in LSPP then go to Application / Model Checking / General Checking / Contact Check and pick Tied and highlight the _TIED contact and click on the Check button. One should see that the _TIED interface is not as one might suspect with more nodes _TIED then desired. This can be fixed in two ways, reverse the interface and make the Hex elements the *surfa* or better yet, tell the _TIED contact not to search so deep during the tying process (MST=-1e-5 (*negative* 1e-5), i.e., RTM).
- Time to clean up the model. Inspect messag file and notice warnings. To fix this one has to create a node or segment set to restrict the *surfa* side. This final model is called: Tied Contact - Mesh Transition Hex to Tet - Finish - Clean.dyn and uses a segment set to define the _NODES on the *surfa* side. One will note that the results are the same as the prior run since the mechanics are the same.



Done Early? The prior model was run as implicit (see folder Implicit) and if one finishes the prior workshop quickly, one can inspect the Keyword deck and learn a bit more about implicit. Run it and see if it makes sense.

9.8.3 Workshop: 15B - Tied Contact for Shells (6 DOF) _TIED_SHELL_EDGE_SURFACE_CONSTRAINED_OFFSET

Objective: To understand how tied contact can easily glue structures together that are not coincident. There is much to learn and there are limitations. But if one understands the theory, it leads to confidence in how the tying or gluing is done.

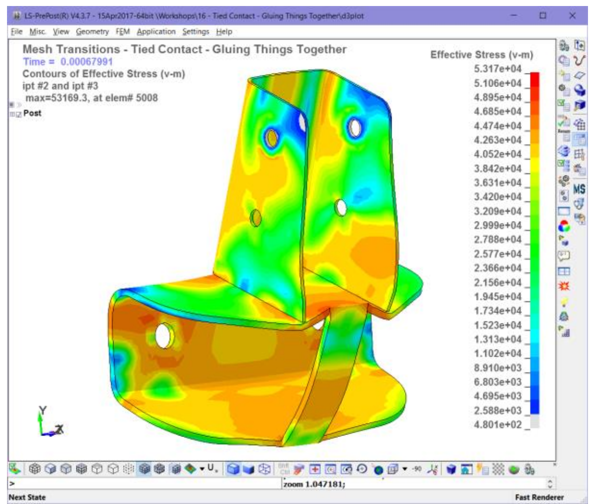


Tasks:

- Open up the start file: Tied Contact - Gluing Things Together - Start.dyn. It is set up to run minus the _TIED contact definition. That is your job. There is a predefined *surfa* node set and one can use the Part id for the *surfb* side definition. Since it is shells with six dof and the surface is offset, these hints should lead you to choose the correct _TIED setup. The real question is whether one uses the _OFFSET method or the plain vanilla non-offset method?
- For Curiosity, use a non-offset method and see what happens to the *surfa* side. Then reset the _TIED contact method and use an _OFFSET method. The – Finish _OFFSET and – Finish non-offset are both setup to provide guidance if your intuition and research fails you.

  Non-offset: *surfa* nodes are moved to *surfb* surface (if within reach, if not one can set sst and mst to a negative number that reaches)

  _OFFSET: does not move *surfa* nodes but they have to be in reach.
- With any luck, you'll see something like this when you are done.



Done Early? Rework the _TIED formulation to that of the recommended formulation and verify that it works as intended.

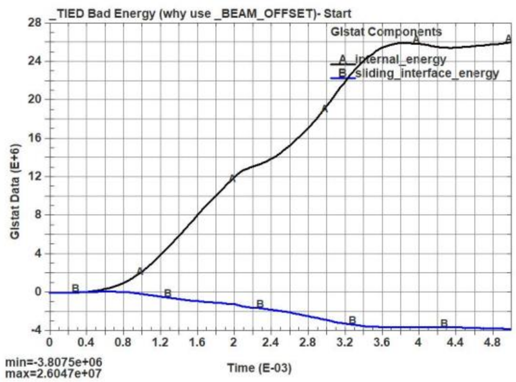
Analyst's Note: The Tied option considers a node "tied" if it is within 5% of the element's thickness. This applies to all _TIED formulations. As mentioned, the constraint option moves the surfa node to be adjacent to the surfb surface while the offset option accounts for the gap; but whether or not it is tied, depends upon the separation of the nodes. To override the default setting, one can set the SST to a negative number that reflects the absolute distance to search for a tie relationship between the surfa and surfb nodes. Of course, if one wants to know more, RTM.

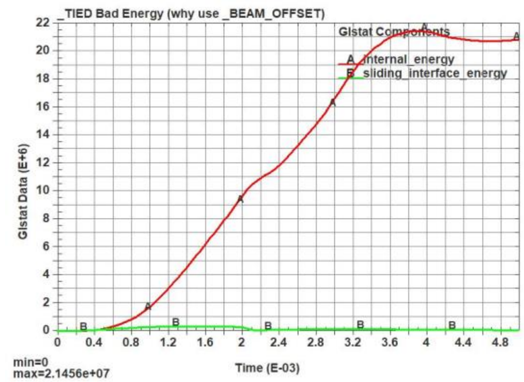
9.8.4 Instructor Led Workshop: 7B - _TIED Bad Energy (or why we use _BEAM_OFFSET)

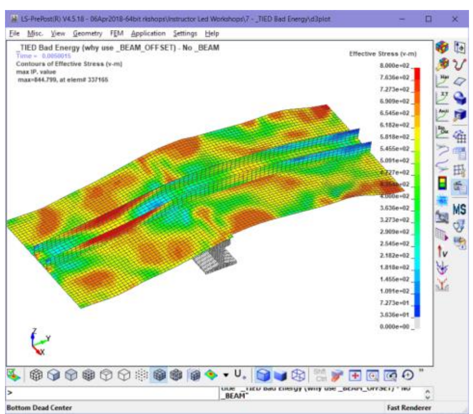
If anything this little dialog is to remind myself to be careful with Tied Contact's with "OFFSET". As mentioned, the Offset option indicates that the algorithm is using the penalty method to enforce the locked motion between parts. When there is "penalty" one has the opportunity to create negative sliding interface energy since springs are used to enforce the locked positions. This behavior killed a rather simple analysis. It was a bit amazing how it completely changed the behavior of the structure. The fix is just to change the contact to _BEAM_OFFSET.

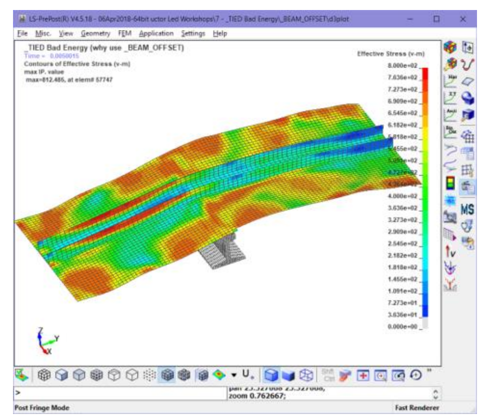
The two models are provided. Which model is right? {on the full model it was a complete FUBAR}.

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_OFFSET_TITLE	*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET_TITLE



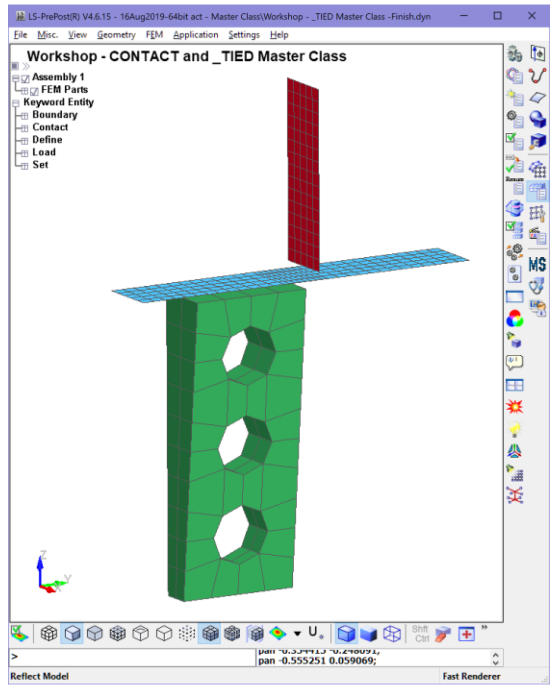
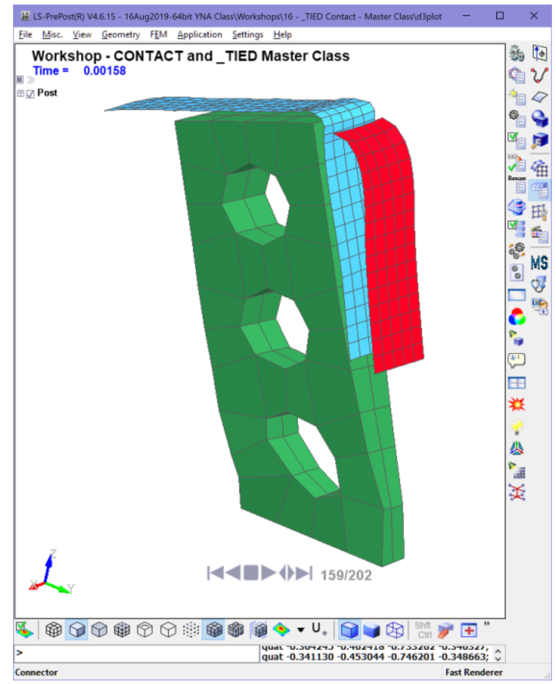






Analyst's Note: Why didn't we just use _BEAM_OFFSET as the default? It was an inherited model and I wasn't quite aware that _BEAM_OFFSET is the recommended go-forward formulation by LSTC and hey, how could a _TIED contact kill your model anyway?. It pays to understand your craft and stay abreast of new developments since this problem cost me several hours of wondering in the numerical wilderness.

9.8.5 Workshop 16: *Surfb* Class in _TIED Connections (Student Bonus)

Start	Finish
	

Objective: Just to work through how _TIED contact works and also not to forget that we need to enforce contact if we want parts of the structure to contact each other. At the end of your effort, you should see something like the image on the right. This implies that you have used the _TIED formulations correctly to capture 3 DOF and 6 DOF mechanics and also know how to use *CONTACT_AUTOMATIC_SINGLE_SURFACE.

Details of Model: The Center plate (*PART *pid*=2 Center Plate) is welded to the solid bar (*PART *pid*=3 Bar). The vertical plate (*PART *pid*=1 Vertical Plate) is welded to the Center Plate. A pressure load is applied to the Vertical Plate. The bottom of the Bar is fixed by constraints.

Tasks:

- A Start file is provided with all the necessary Keywords present, albeit not filled in. The idea is to edit the Keyword file with your inputs, make an analysis run and then repeat until you get a final model that appears as on the right.

Learning Goals:

- Comfortable with _TIED contact;
- Understand _AUTOMATIC contact;
- Clean-up contact behavior by decreasing the mass scaling by 10x (-7 to -8);
- Ability to take a model from its initial state, debug it and arrive at something good.

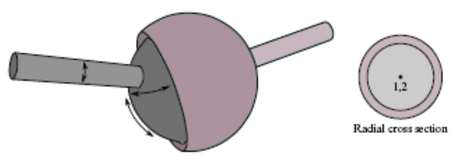
Analysts Note (Extra Credit): Run deck in file folder "Limitations of _CONSTRAINED_OFFSET". Take a look at the results. At the end of the day _CONSTRAINED_ means constrained and that constraint equations are used to create the _TIED connection. This means no sharing of the surfb segments with another _CONSTRAINED_ connection. For your extra credit try to fix the deck without looking at the solution (- Finish). If you are still looking for something to do, then fix the deck within Why Surfa Side Should Have Finer Mesh.

10. CONNECTIONS VIA JOINTS

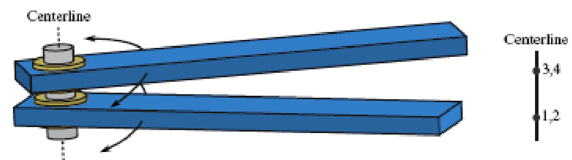
10.1 Joints or *CONSTRAINED_JOINT_

To model the motion and likewise, large movements in engineering systems, one needs joints. LS-DYNA has a very sophisticated set of commands that will allow one to model many types of common joints (e.g., hinges, spherical bearings, etc.). They are not that hard to setup if one just goes slow and perhaps build small pilot models of each joint that they are trying to simulate since debugging a large model can be laborious.

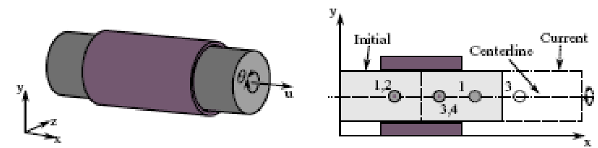
Spherical Joint



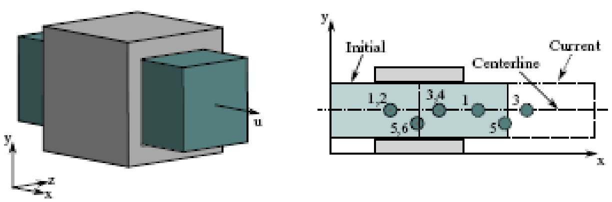
Revolute Joint



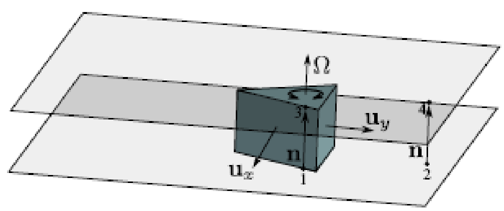
Cylindrical Joint



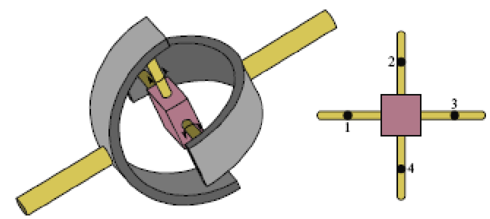
Translational Joint



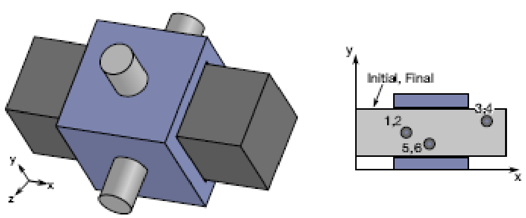
Planar Joint



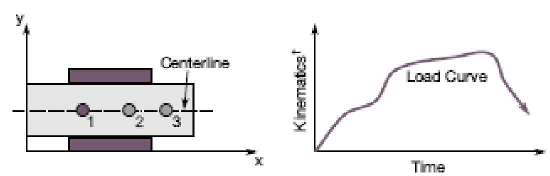
Universal Joint



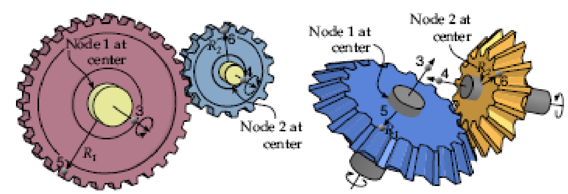
Locking Joint



Translation Motor Joint



Gear Joint, etc.



10.2 How Joints Work

The foundation of joints lie in the use of *CONSTRAINED_NODAL_RIGID_BODIES (CNRB) to provide the framework for the action of the joints. The joint mechanical behavior is implemented using the penality method (i.e., like contact). As such, a joint has a stiffness and energy. The stiffness of the joint is calculated based on more math that I want to describe in this brief note (*Please see LS-DYNA Theory Manual 2023 for the complete description*) what is directly of importance is that the joint stiffness is based on the mass of the joint's CNRB and their geometry, and that the reaction forces to enforce the joint's behavior, are applied at the center-of-mass of the CNRB.

*Analyst's Note: Given that joints use the penality method, one can have joint failure (i.e., unexpectively flying apart) by using CNRB's with too little mass and having the opposing CNRB's center-of-mass to close to resist displacements/moments. One may want to read the Keyword *CONTROL_RIGID for some interesting notes on CNRB w.r.t. joints and mass scaling.*

The simplest joint is the spherical joint which connects two coincident nodes between two CNRB's (locks translation DOF while rotation DOF are free). A more interesting joint is the *CONSTRAINED_JOINT_CYLINDRICAL which requires two colinear nodes within the opposing CNRB's. Please note that the node sets are separated by some reasonable distance to ensure that the joint doesn't "blow-up". What is reasonable? It depends on the mass of the CNRB's and the magnitude of the force that is applied to the joint. If this joint is expected to receive a high bending load (MZ), then it would be good to ensure a large separation between the node sets. There are no hard guidelines but the reality is that most joints function just fine.

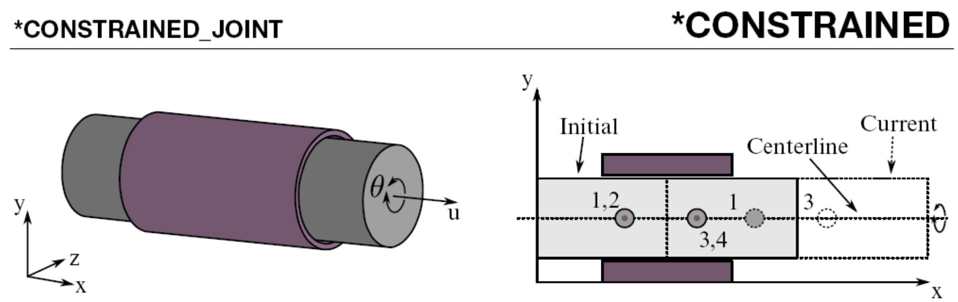
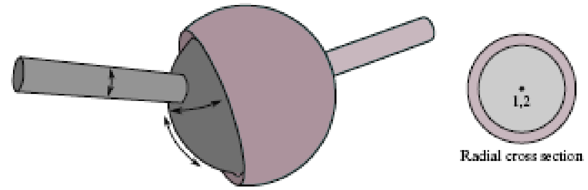


Figure 10-13. *Cylindrical Joint*. This joint is derived from the rotational joint by relaxing the constraints along the centerline. This joint admits relative rotation and translation along the centerline.

10.3 WORKSHOP: 17A – SPHERICAL JOINT BETWEEN A SHELL AND SOLID

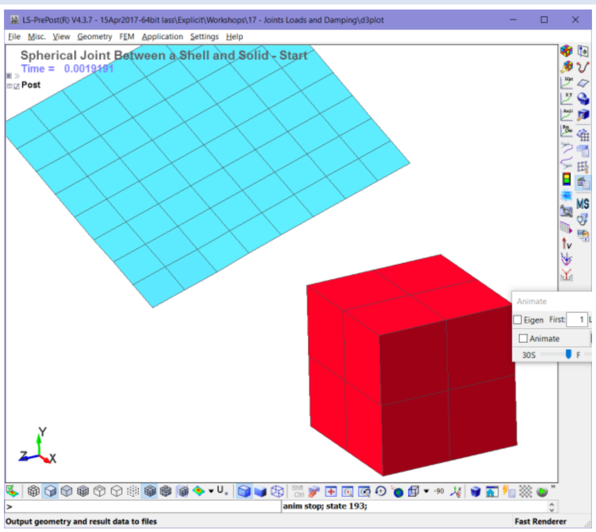
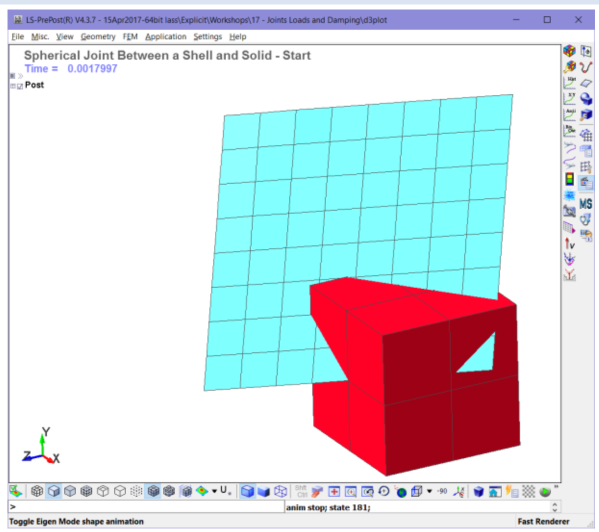
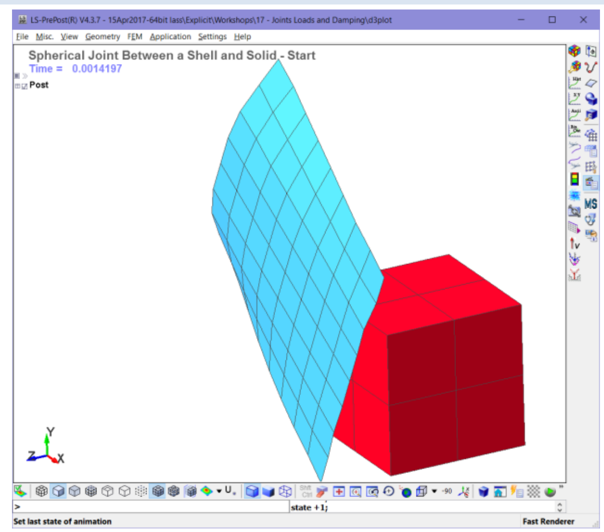
Objective: Understand how a spherical joint is setup and how it functions.

Background: The spherical joint is the easiest to setup of all the joints. All you need is two rigid bodies (Constrained Nodal Rigid Body (CNRB) having one node from each CNRB coincident with each other. As with all joints, try to ensure that the CNRB is well distributed (it has some spatial reach and mass).



Tasks

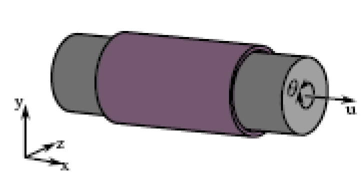
1. Open up Spherical Joint Between a Shell and Solid - Start.dyn and inspect the deck.
2. Read the Keyword Manual for *CONSTRAINED_JOINT_SPHERICAL and then setup your joint.
3. Run the model and see if it makes sense.
4. Add contact to the model and rerun (use our preferred defaults for *soft* and *depth* – not necessary for this model but it is a good habit to always go out-of-the-gate with these options – even though they do cost a few numerical cycles).

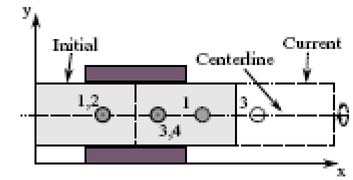
No Joint	Spherical Joint	Spherical Joint w/ Contact
		

10.4 Workshop: 17B - Cylindrical Joint Between Two Nested Cylinders

Objective: Understand how a spherical joint is setup and how it functions.

Background: The cylindrical joint opens the door to the standard more complex joint setup since now one has to define two sets of nodes. As in the spherical joint, all you need is two rigid bodies; however you need to have two nodes within each CNRB aligned as in the graphic on the right. Nodes 1 and 3 are within on CNRB while nodes 2 and 4 are in the other CNRB. As with all joints, ensure that the CNRB is well distributed (it has some spatial reach and mass). This means that the further apart the nodes sets, the more stable the joint will be.

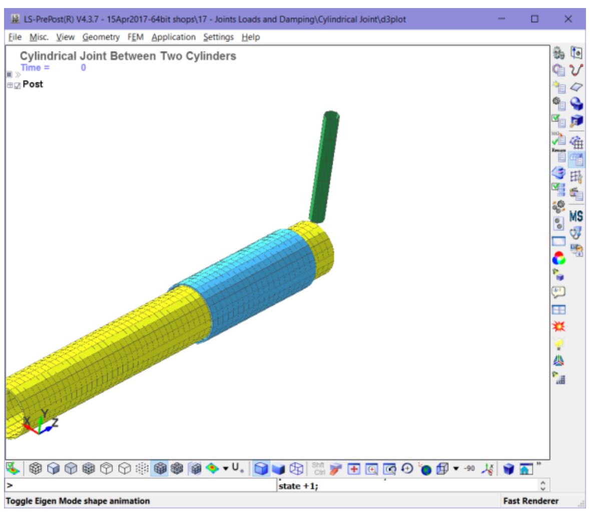


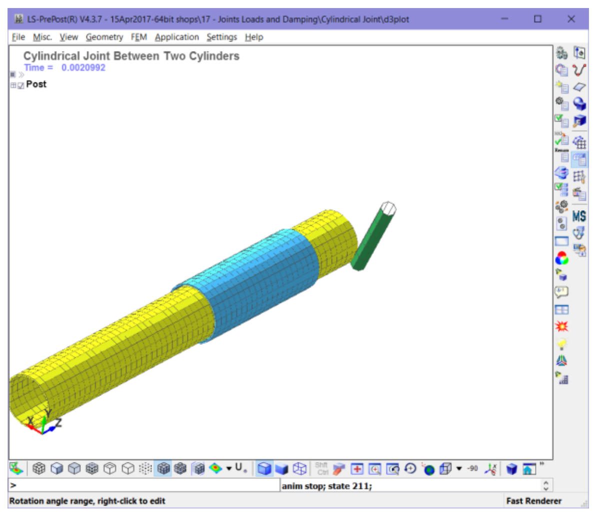


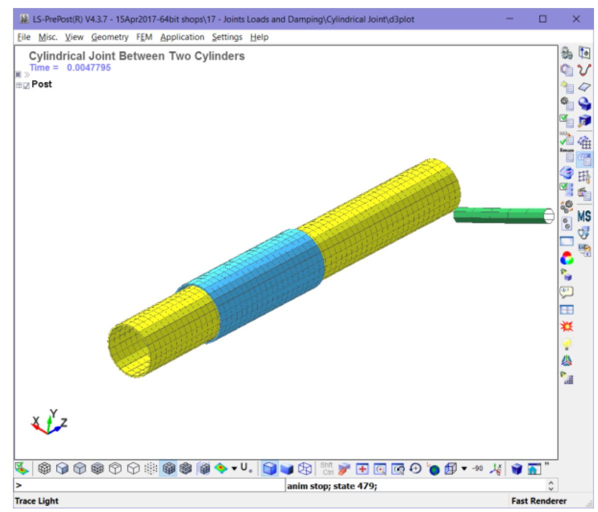
Tasks

5. Open up Cylindrical Joint Between Two Cylinders - Start.dyn and inspect the deck.
6. Read the Keyword Manual for *CONSTRAINED_JOINT_CYLINDRICAL and then setup your joint.
7. Run the model and see if it makes sense.
8. Add contact to the model and rerun.

Cylindrical Joint w/Whipping Beam Contact

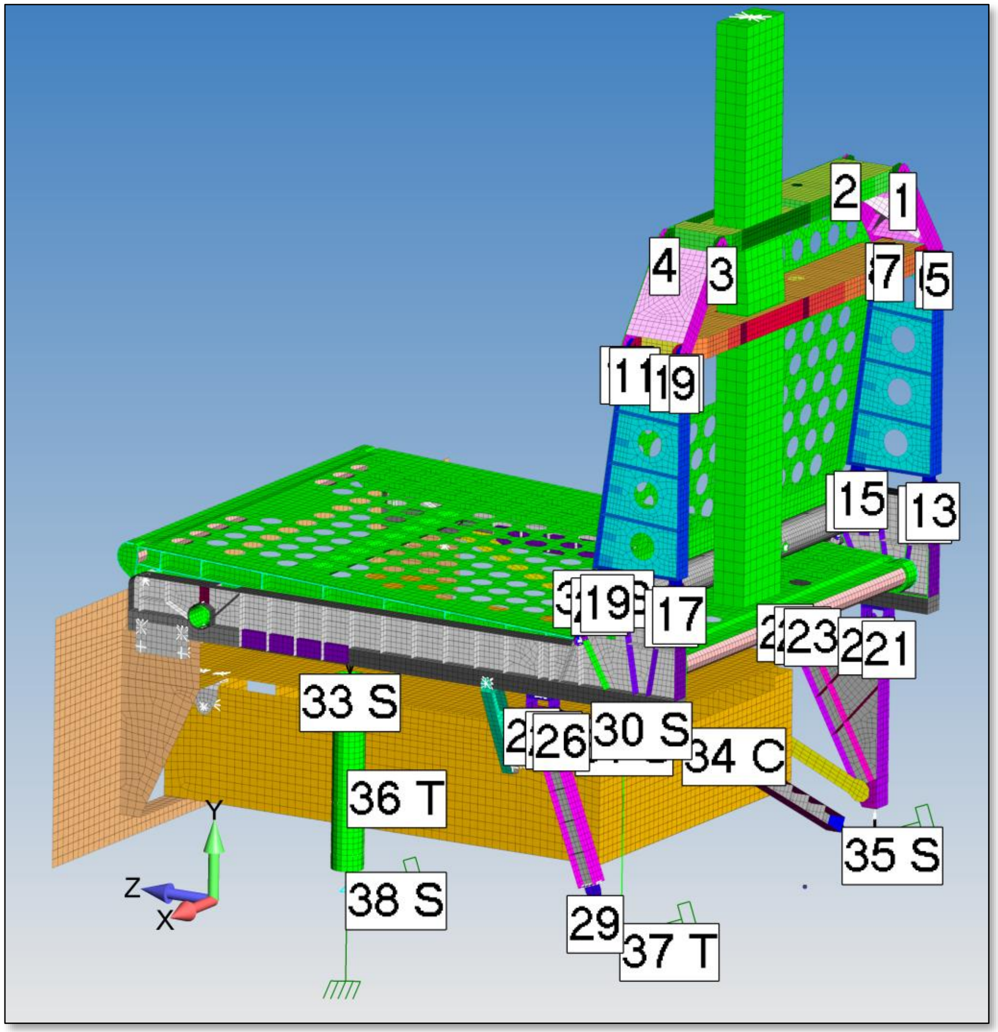






10.4.1 Who Uses Joints?

Background: Joints are very prevalent in seat analysis and automotive simulation. One could spend days working on joint setup and our only comment is that they do work if one is careful with the setup.



11. DAMPING

11.1 General, Mass and Stiffness Damping

In dynamics, there often can be some oscillations that the analyst would prefer to have damped out or to account for viscous behavior of some materials (see Material Damping). By default, LS-DYNA is undamped and why LSPP has XY data filter capabilities.

11.1.1 *DAMPING_*OPTION*

Introduces Rayleigh proportional damping based on: $[c] = \alpha[m] + \beta[k]$

- The mass damping constant α is specified by *DAMPING_GLOBAL, *DAMPING_PART_MASS and *DAMPING_RELATIVE.
- The stiffness damping constant β is activated by *DAMPING_PART_STIFFNESS.

Mass damping is for low frequency response (rigid body modes), while stiffness damping is more effective at higher frequencies. Since they are dissipative, their energy loss should be tracked. This can be done with the *CONTROL_ENERGY option of RYLEN=2. Energy loss is then reported in the glstat and matsum files.

11.1.2 *DAMPING_FREQUENCY_RANGE_DEFORM

This is a more elegant approach to damping and allows the user to specify the critical damping coefficient and the frequency range to damp. It is effective when used with low amounts of damping (e.g., 1 or 2%) and when the frequency range is no more than a factor of 5x (e.g., 100 to 500).

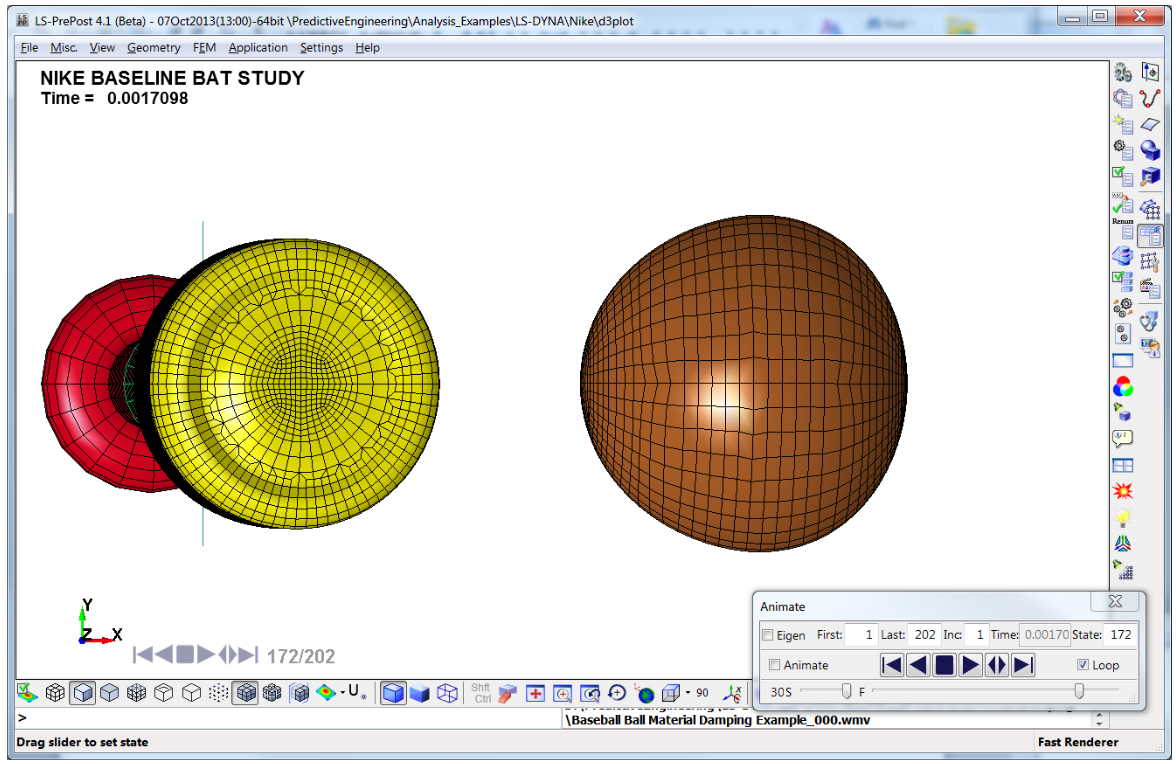
Analyst's Notes: I know of no shortcut to producing good agreement with the observed loss in a test I can only suggest good judgment and a trial-and-error approach in order to tune the numerical damping.

11.1.3 Material Damping (e.g., elastomers and foams)

Elastomers (e.g., *MAT_181 Simplified Rubber) and foams (e.g., *MAT_053 Fu Chang foam) have the ability to add damping directly within the material card. Recommended values are between 0.05 and 0.5. However, there is no true recommended value since each material application is a bit different and requires some observation by the analyst to determine the appropriate value. As a default *MAT_181 uses 0.10 damping.

11.1.4 General Example on Material Damping

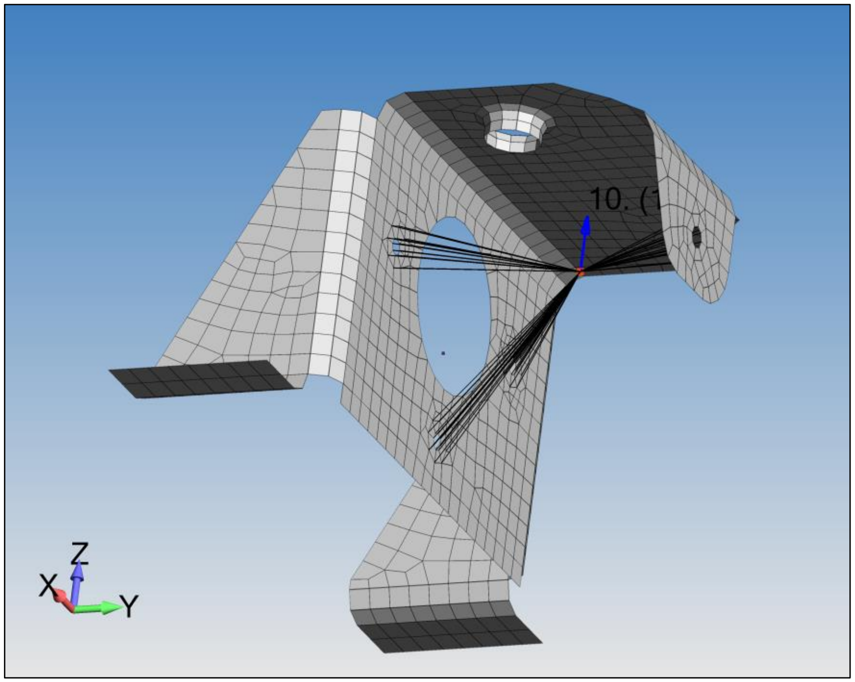
An example of material damping is provided in the Class Reference Notes / Damping titled: Baseball Ball Material Damping Example.dyn along with a movie file of what one can expect to witness.

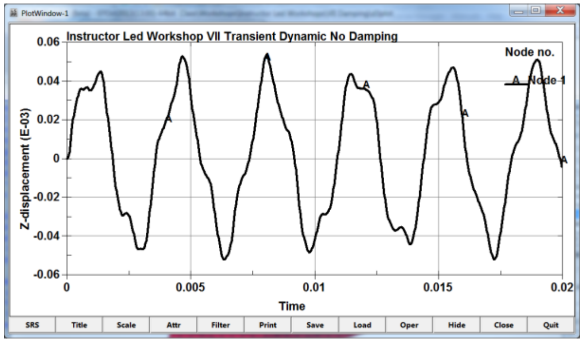


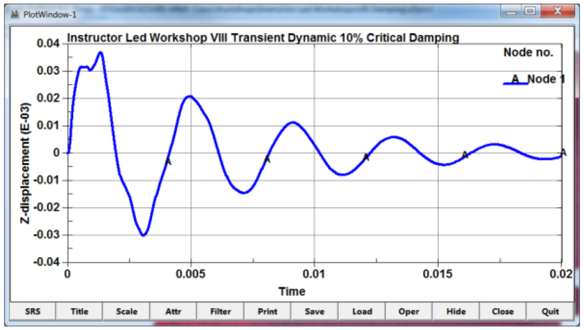
11.2 INSTRUCTOR LED WORKSHOP: 8 – DAMPING OF TRANSIENT VIBRATING STRUCTURES

Objective: Show An example is given of a vibration problem run by NX Nastran normal modes analysis and then by LS-DYNA as an undamped and damped transient analysis. If the student desires, the model is easy to switch to LS-DYNA implicit for an Eigenvalue run. **{Update Video}**

What to Learn: The model is inspected within the Keyword deck and LSPP *(The instructor will run a NX Nastran normal modes analysis for later comparison.)* The first natural frequency is noted. The LS-DYNA run is interrogated in LSPP. A History plot is made of node 1 in the **Z-direction**. Using LSPP X-Y plot tools, under Oper, a Fast Fourier Transform (FFT) is applied and the first hugely dominate frequency is noted. The model will then be given 10% critical damping between 250 and 350 Hz. We then modify the file by enabling *DAMPING_FREQUENCY_RANGE_DEFORM, rerun and plot results.







***Student Task:** One can switch it to an Eigenvalue analysis by adding these two *CONTROL cards: _IMPLICIT_EIGENVALUE (neig=10) and IMPLICIT_GENERAL (imflg=1) (One will note that LS-DYNA will ignore the *CONTROL_TERMINATION card). Once analyzed you'll want to view the d3eigv.*

12. LOADS, CONSTRAINTS AND RIGID WALLS

12.1 Loads

Within the Workshops of this course are a variety of load applications using initial velocity, point loads, pressure loads, etc. Other examples of load applications can be found at www.dynaexamples.com

12.1.1 Initialization Loads (*INITIAL_)

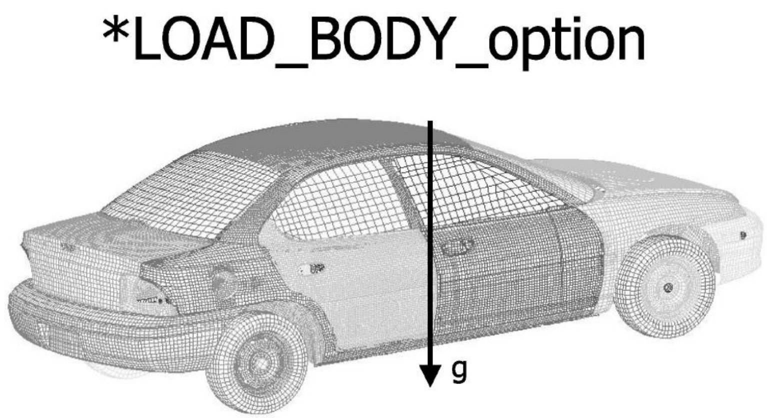
The most common initialization load is *INITIAL_VELOCITY_*option*. For example, for any type of drop test, the structure is given a uniform initial velocity and then allowed to instantaneously hit the target. The other common initialization command is *INITIAL_TEMPERATURE for thermal analysis work.

12.1.2 Point and Pressure Loads (*LOAD_NODE_ & _SEGMENT)

There is nothing complex to these loads. Point loads are those loads applied at nodes while pressure loads are applied over element faces. In LSPP, pressure loads are applied onto segments (i.e., faces).

12.1.3 Body Loads (*LOAD_BODY_)

Body loads are most commonly defined as constant acceleration to capture the effect of gravity. Keep in mind that LS-DYNA treats body acceleration loads differently and that to obtain the same direction of gravity in Nastran one must switch the sign of the acceleration load.



12.1.4 Rigid Walls (e.g., *RIGIDWALL_MOTION)

Rigid walls can move and provide a simple way to apply loading onto a structure.

12.1.5 Boundary (e.g., *BOUNDARY_PRESCRIBED_)

This command is for applying moving displacements, velocities and accelerations in every type of direction and on rigid bodies. A very powerful set of loading options.

12.1.5.1 Prescribed Nonlinear or Curvilinear Motion of Node, Node Sets or Rigid Bodies

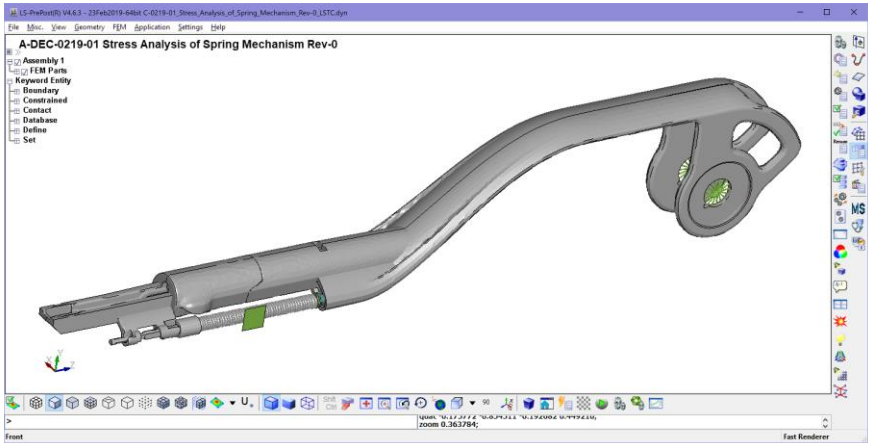
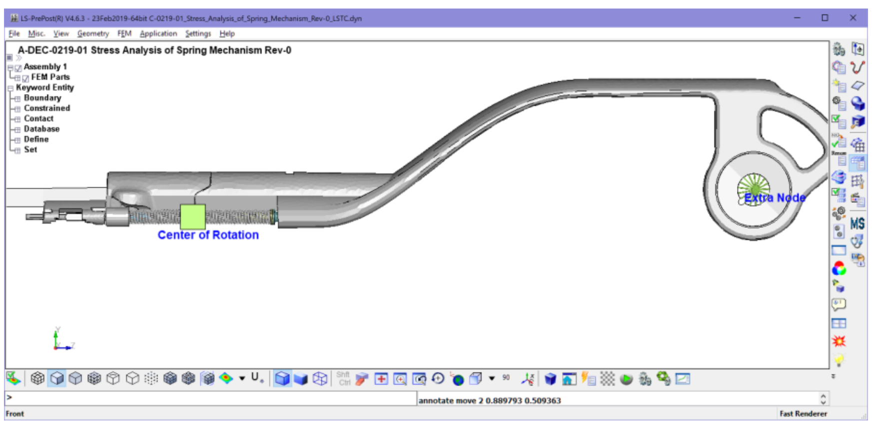
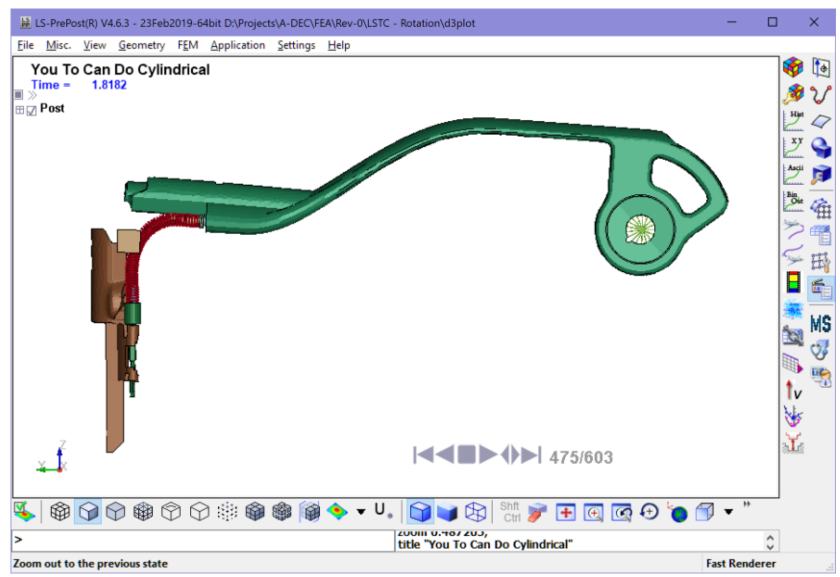
It should be mentioned that applying a prescribed displacement load to a node, node set or a rigid body is one of the key uses of *BOUNDARY_PRESCRIBED_MOTION_{option}. For example, one can make a node follow any type of curve (e.g., a circle) or a wandering path by simply prescribing a displacement in the global coordinates of x, y and z using three *BOUNDARY_PRESCRIBED_MOTION commands on the same entity. For example, if one uses cos and sin functions to define the x and y displacement, one can have the desired entity follow a perfect curve within the xy plane. This is identical to as if one used a cylindrical coordinate system in a linear package and prescribed a displacement in the theta direction.

12.1.5.1.1 Why LS-DYNA Doesn't Have Cylindrical or Spherical Coordinate System Options

It makes for a very powerful command once you realize you don't need a cylindrical or spherical coordinate system (e.g., as is common in linear codes) to define cylindrical or spherical motion since one can simply write the appropriate function and generate the x, y and z curves for the desired motion. This can be done of course for all the standard options of prescribed velocity, acceleration, etc.

12.1.5.2 Load Example: Fixed Cylindrical Displacement via Clever Use of Rigid Body and *CONSTRAINED_EXTRA_NODES_SET

Just an example of how one can enforce cylindrical motion using *PRESCRIBED_MOTION coupled with a "rigid body". This formulation was courtesy of LSTC Technical Support.

Spring Loaded Medical Device	Keywords
	*MAT_RIGID: CMO – CON1 = 7 with CON2 = 5 *CONSTRAINED_EXTRA_NODES_SET – Add node at location where cylindrical displacement is required *PRESCRIBED_MOTION_RIGID_ID – Set the PID to the rigid body at the center of rotation (the rectangular shell element at the center of the spring), set the DOF to 5 (rotation around the X-axis) and start the analysis. The rigid part and its extra nodes will rotate as one complete rigid body. The center of the rigid part is defined by the rectangular shell element. It is a very effective way to enforce cylindrical motion.
Rotation Around Center of Spring	**Rotated Cylindrically - Perfectly**
	

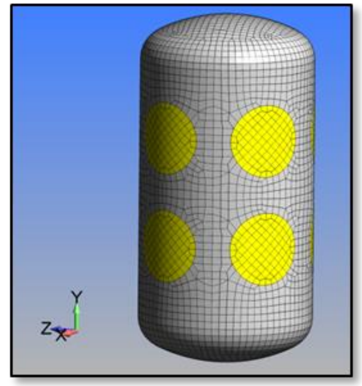
12.2 Workshop: 18 - Drop Test of Pressure Vessel

Objective: This exercise is geared toward increasing your confidence in using LS-DYNA. As our pre-processor, we'll be using LSPP. If one doesn't want to use LSPP, we have created a Keyword deck where one can hand enter the data.

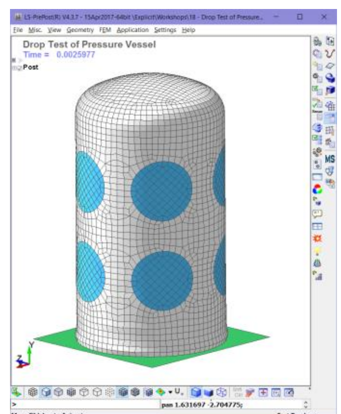
We are going to do a standard drop test where the body is given an initial velocity and a rigid wall is placed directly beneath the vessel. At the end of this exercise, we'll fill the vessel with a fluid. The workshop starts by inspecting the Keyword deck and then within LSPP. As in prior Workshops, a final completed Keyword deck is available if you get stuck.

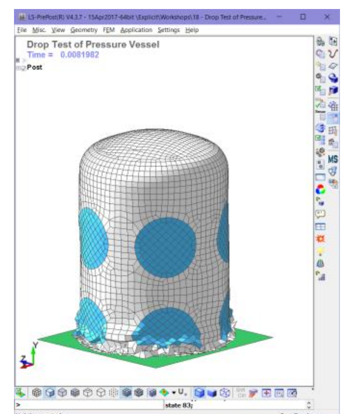
Please note that all the input values that you need for this analysis are provided within task list. What isn't provided is how to insert these values into the correct keyword, for that part of the Workshop, the student is advised to RTM is they are unsure of the format. Of course, the Instructor knows, but it will be more long-term useful for the student to learn how to use the Keyword Manual and self-learn. As always, if things don't make sense and you are going in mental circles, please ask the instructor to clarify. This is why this class is best taught "live" – to ask questions when you get stuck and keep you moving forward.

Tasks:

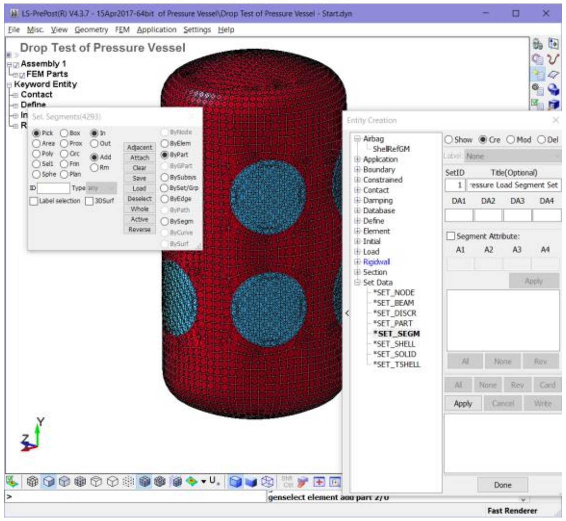


1. A thin-walled (0.05") aluminum vessel that is 24" in diameter and 36" tall is impacted at a speed of 100 MPH (1,760 in/s) against a rigid wall. The first task is to set the initial velocity for the vessel followed by creating the rigid wall.
2. Use *INITIAL_VELOCITY_GENERATION to get the vessel going. Set initial velocity to Vy = **-** 1,760 (negative). {*Note: LS-DYNA provides lots of _VELOCITY options, it is useful to standardize on one robust Keyword card and know it well.*}
3. Define rigid wall with *RIGIDWALL_PLANAR. We are placing this wall perpendicular to the bottom of the vessel. The bottom of the vessel is at 0, 0, 0. To avoid initial contact with the rigid wall, it is placed below the outer skin of the vessel at 0, **-**0.025, 0 and the head of the vector at 0, 1, 0 to define its position.
4. Run model and notice that the pressure vessel skin folds in upon itself.
5. Add *CONTACT_AUTOMATIC_SINGLE_SURFACE_ID. Please remember to add in soft=2 and depth=5 to switch the contact formulation to segment based with enhanced edge-to-edge detection. Then rename your model to Finish I. Run model and it should look better. (**Finish I Step Completed)**.



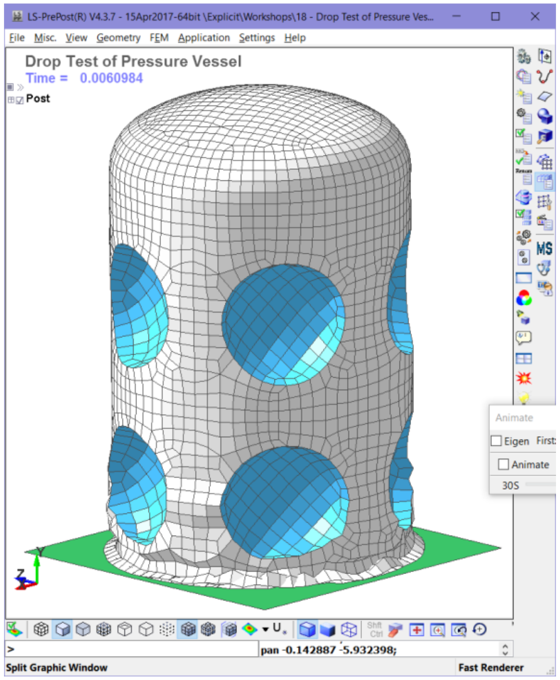


Workshop: 18 – Drop Test of Pressure Vessel (continued)



6. The next step is to apply a positive pressure load to the inside of the vessel. Pressure loads are applied to “segments” which is LS-DYNA terminology for element faces. One will need to create a segment set, define a load curve and then create the pressure load. The segment set is created in LSPP but everything else can be hand-entered if desired.
7. {This section has been changed – the Keyword Deck now contains the segment set (*sid*=1 to allow the student to focus more on analysis and less on pre-processing manipulation; however, if so desired, the student may learn how to create their own segment set.) To create your segment set, use the Create Entity option within LSPP. This option is located within the Model and Part toolbar below the Keyword Manager button. Within the Create Entity screen, expand the Set Data option and select *SET_SEGM option. Another pane will appear. Within this pane you’ll want to Create (**Cre**) the new segment set. I would recommend picking the segments using the ByPart option. After picking, hit Apply and you’re done.
8. Create your pressure load curve using the *DEFINE_CURVE command. Set the curve to ramp slowly and then hold (0,0; 0.005,1 and 0.015,1. When done, this will be *lcid*=2 since the rubber curve is *lcid*=1.
9. The pressure load is finally created using the *LOAD_SEGMENT_SET command. We are setting the pressure to 1.0 psi by setting SF=1.0 (positive). Run the model and interrogate (**Finish II**). Update file name to note that it is Finish II. One will notice that something is not quite right.

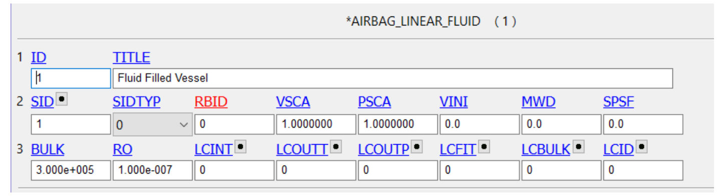
   We wanted positive pressure. So go back and change the *sf*=-1.0 (negative). Alternatively, one could flip the shell normals (see Workshop movies file).



Analysts Note: FEA pre- and post-processors don’t create an accurate analysis, they can facilitate but it is the simulation engineer’s judgement and understanding of the mechanics and the analysis code that makes the real difference. Thus, this course emphases understanding the mechanics and how it is implemented within LS-DYNA. If one knows the “mechanics” and the “code”, one can use any pre-/post-processor and be successful.

Workshop: 18 – Drop Test of Pressure Vessel (continued)

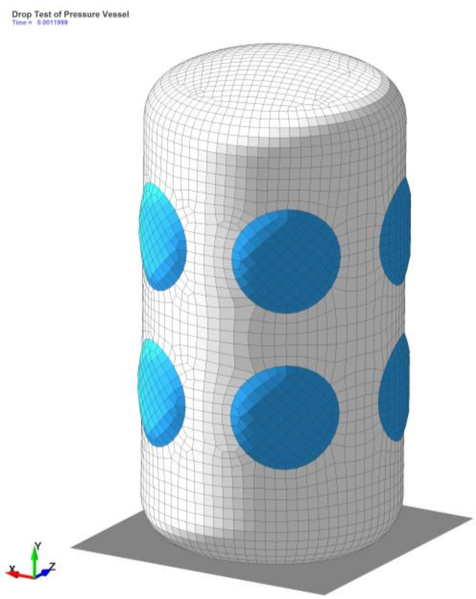
10. To simulate the incompressible fluid behavior within the vessel, add *AIRBAG_LINEAR_FLUID to the model with bulk=50 and ro=1e-7 with all other options default. The segment set created for the pressure load is re-used to define the enclosed fluid volume. The completed card is shown on the right.
11. Rerun model. You should see the rubber portals bulge (**FINISH III**).

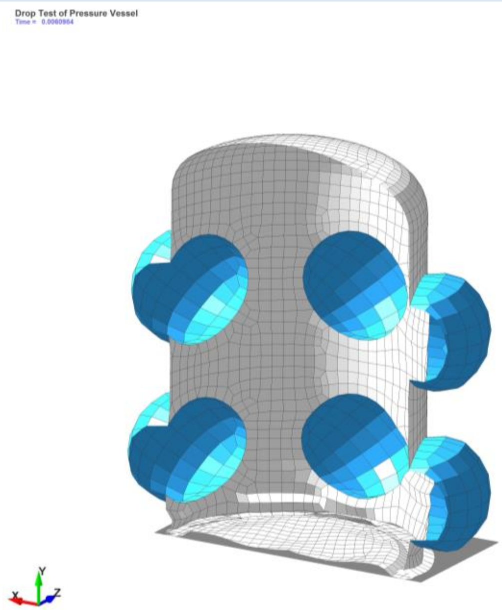


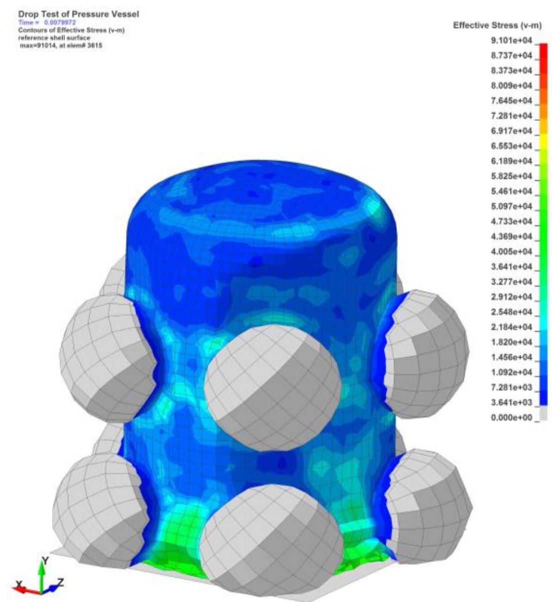
*AIRBAG_LINEAR_FLUID (1)

1	ID	TITLE						
	1	Fluid Filled Vessel						
2	SID	SIDTYP	RBID	VSCA	PSCA	VINI	MWD	SPSF
	1	0	0	1.0000000	1.0000000	0.0	0.0	0.0
3	BULK	RO	LCINT	LCOUTT	LCOUTP	LCFIT	LCBULK	LCID
	3.000e+005	1.000e-007	0	0	0	0	0	0

Color Change	Create Section Cut	Final w/ Stresses







Extra Time:

- *If You Can: Take the prior model and reorient the *RIGIDWALL_PLANAR to make it vertical or in the XY plane. Then change the *INTIAL_VELOCITY_GENERATION to have the vessel hit normal to the XY plane or in the Z-direction. We are simply re-orienting the impact from smashing down on the ZX plane to hitting “sideways” against a XY plane (so-to-speak).*
- *Switch contact to _MORTAR. How goes the run time? (How much longer?)*

13. DATA MANAGEMENT AND STRESS AVERAGING

When running large models, storing the complete dataset with the _BINARY_D3PLOT file can create unwieldy data sets. The idea of the d3plot file is to provide a full data picture (data and graphics) of the simulation. If the analyst needs higher resolution of just data, LS-DYNA provides lots of options to store just nodal (e.g., displacements, velocities, accelerations and forces) and just element data (e.g., stresses). This capability is accessed by specifying the nodal and elemental entities that one wishes to capture data at using the Keyword commands within the *DATABASE_HISTORY_*OPTION* section and then to specify the frequency of data output via the *DATABASE_ELOUT, _NODFOR and _NODOUT options.

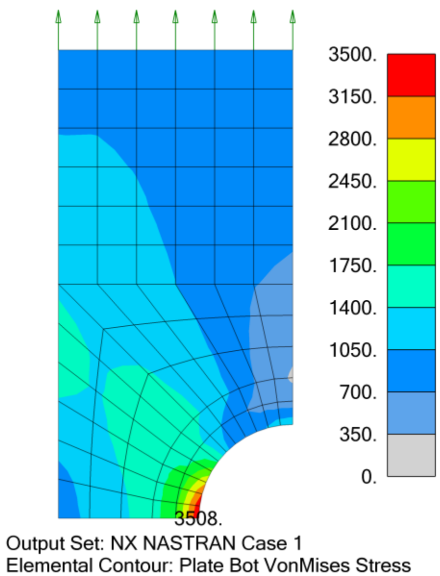
The flexibility of this system is quite useful, and the Keyword manual provides a listing of what data is available for post-processing and allows the output of such data in local coordinate systems.

13.1.1 Stress Reporting and Stress Averaging in LS-DYNA/LSPP

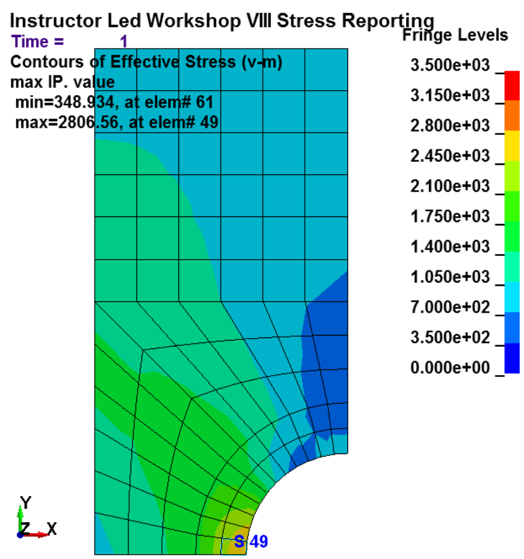
LS-DYNA default element formulations (*elform*=2 for shells and *elform*=1 for solids) are based on one-Gaussian point integration. Stresses are then reported at the centroid of each element since that is where the Gauss point lies. LSPP will then average these stresses at the nodal points (or not – depending on the setting). For highly nonlinear work, this approach works very well and keeps the solution time down and the database small. If one needs greater fidelity, LS-DYNA offers the complete menu of fully-integrated element formulations for explicit and implicit work. For example, *elform*=-16 for plates and for Hex elements *elform*=-18. However, keep in mind that with default settings, LS-DYNA will average the stresses from the integration points and report only the centroid stresses. For shell and solid elements, LS-DYNA can also extrapolate the integration point data via *CONTROL_OUTPUT, shlsig and solsig options (Note: This will be discussed in the implicit section).

This little example will show how to setup your implicit plate model to match which one sees in a standard implicit code (e.g., ANSYS or Nastran).

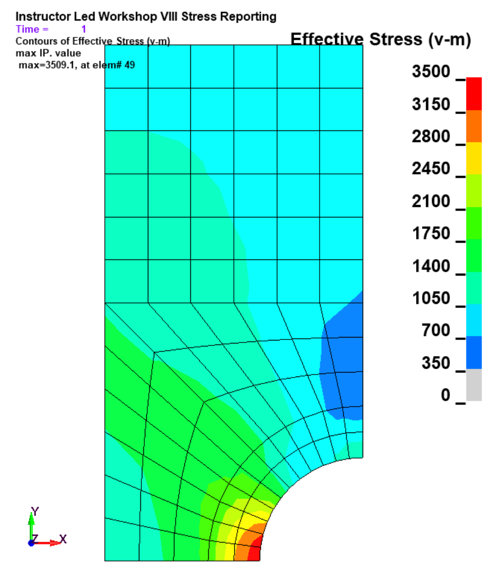
Instructor Led Workshop: 9 - Stress Reporting and Stress Averaging | Shells



Nastran 3,508 psi



LS-DYNA Default 2,800 psi



LS-DYNA *elform*=21 3,509 psi

*CONTROL_OUTPUT, *shlsig*=1 {*MAT_1}

*DATABASE_EXTENT_BINARY *maxint* = -2

14. LOAD INITIALIZATION BY DYNAMIC RELAXATION AND IMPLICIT ANALYSIS

14.1 Initialization of Gravity, Bolt Preload and Other Initial State Conditions

14.1.1 Stress Initialization

Given that explicit analysis work often involves timesteps in the range of microseconds, one can image the challenges of using an explicit approach to obtain quasi-static or static stress states in structures subjected to uniform constant loading. There are many applications where the ***start*** of the explicit analysis requires the initialization of steady-state loads within the structure. Here's a short list: rotating equipment (e.g., fans, turbine blades or rotating flywheels), pressurized vessels or tires, bolt preloads, shrink-fit parts, or spring mounted structures under constant gravity.

These static stress-state conditions can be simulated in LS-DYNA using two techniques: Explicit Dynamic Relaxation or Implicit Static Analysis.

14.1.2 Dynamic Relaxation (DR) *CONTROL_DYNAMIC_RELAXATION

DR is a heavily damped explicit analysis that is initiated prior to the main transient analysis. It has all the characteristics of a standard explicit run but it is assumed that stresses are relatively elastic and that displacements are small. The solution is heavily damped and unexpected results may be generated. Nevertheless, with some models, it does a great job with bolt preload, tire inflation or application of a shrink-fit. In the DR process, the load is applied (e.g., bolt preload) as a transient load with a sharp ramp up and then a steady-state application. The model dynamically responds to this load application with all the characteristics of a standard explicit transient analysis. As the model is solved, the nodal velocities are reduced at each timestep by the dynamic relaxation factor (default = 0.995). The kinetic energy (KE) is calculated at prescribed steps and when this energy has decreased sufficiently against the initial KE, the solution is considered converged and the DR process shuts down and the solution is handed over to the regular, normal explicit transient analysis sequence. There are lots of options to manage this process and the student is referred to the Class Reference Notes for more information.

Analyst's Note: I have often struggled to get DR to work correctly. It seems "finicky" since the process uses a heavily damped response and it just ain't natural. As such, whenever possible I strive to use the implicit approach.

14.1.3 Initializing Displacements and/or Stress with *INTERFACE_SPRINGBACK_LSDYNA

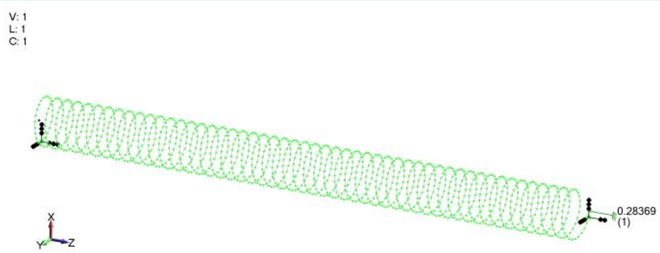
I have seen this Keyword around and wondered about its usage. As with a lot of LS-DYNA Keywords, the journey from reading the manual to actually using the Keyword can be a long, winding road with a few days of detours into the weeds. The solution was courtesy of LSTC Support who quickly provided a clear, simple example of how this Keyword works. It was so simple and obvious that I wanted to save it without my LS-DYNA Class Notes.

Challenge: How does one apply an initial tension to an extension spring?

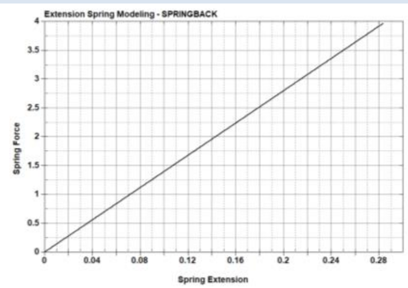
Background: A medical device company was developing a new dental tool using a preloaded extension spring. Extension springs are wound with initial tension such that internal forces hold the coils together. The initial tension of the spring is measured as the load required to separate the coils or to initiate extension in the spring. One then has a very useful mechanical element that can lock a structure together and but allow it to move once the extension load has been reached.

How To: As described by LSTC, it seems too simple. One just extends the spring to the desired "initial tension" with the Keyword *INTERFACE_SPRINGBACK_LSDYNA inserted into the deck and upon termination, a dynain file is written out. For our example, just delete all the results information except that for *INITIAL_STRESS_BEAM and save the file. This file is then *INCLUDE into your analysis model of interest (of course, not mucking up element numbers from your _SPRINGBACK model to your final analysis deck.

Beam Element Model



First Run to Extract Initial Tension Load (*INTERFACE_SPRINGBACK_LSDYNA)

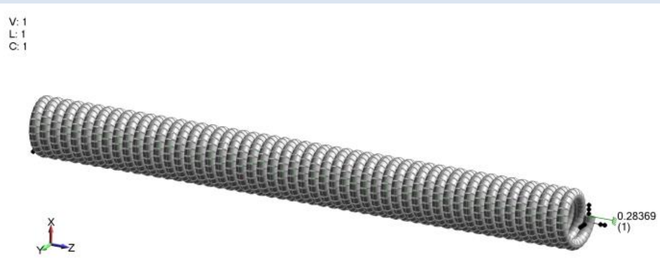


Keyword Guide

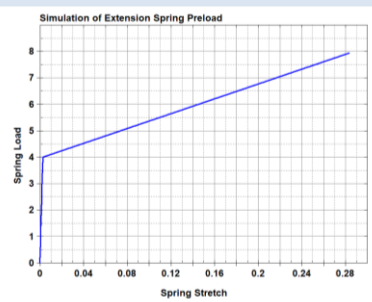
*INTERFACE_SPRINGBACK_LSDYNA is very clever and writes out the results in a format "ready-to-INCLUDE" into your analysis model.

*INITIAL_STRESS_BEAM writes out stress information per integration point. This example uses *INTEGRATION_BEAM with 144 integration points.

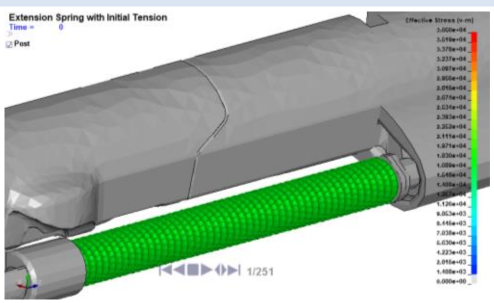
With Beam Thickness Shown (graphically)



Analysis Results with Initial Tension (*INCLUDE – dynain file)

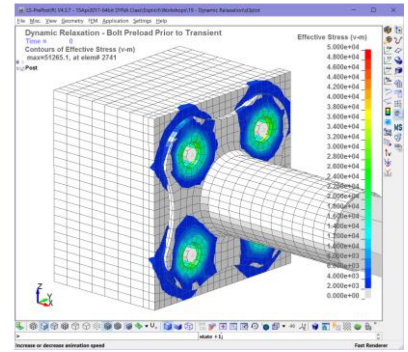


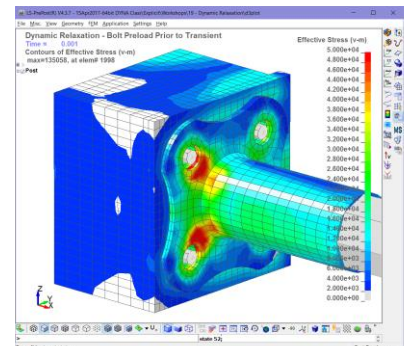
Inserted into Analysis Model Stresses are at Time = 0.0 (No Load)



14.2 Workshop: 19 - Dynamic Relaxation - Bolt Preload Prior to Transient

Background: As modal frequency response analysis is known as “the poor man’s transient dynamics”, Dynamic Relaxation might also be known as “the poor man’s implicit”. Nevertheless, in many cases it gets the job done albeit with some challenges and in some cases can be easier to setup.

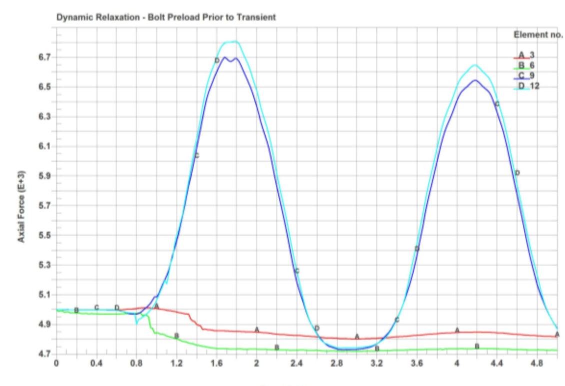




Tasks:

- Open up the Keyword deck: Dynamic Relaxation - Bolt Preload Prior to Transient - Start.dyn. The Keywords that you will be required to populate are commented out. Your task is to read the Keyword manual and correctly fill out the missing information.
- The bolts are preloaded to 5,000 lbf (axial load). This preload is simulated by using the *INITIAL_AXIAL_FORCE_BEAM command. The preload is applied via a curve (see *DEFINE_CURVE, *lcid*=1). It is missing some information to make it work under dynamic relaxation. Read about the settings for sdir and configure these two Keywords to run the model under dynamic relaxation with an applied bolt preload of 5,000 lbf (the model’s units are in English). Please note that a beam set is ready to use (*sid*=1). Please note that setting a beam element to simulate bolt preload requires the use of *MAT_SPOTWELD and that the beam *elform*=9 (spotweld). There is also another limitation is that only one element can be used between the connecting parts or one may say, only **one beam element for the bolt shank**. This is due to the beam’s element formulation that multiple connected spotweld beams are not stable.
- Run analysis and make a plot of the four bolts’ axial force.
- Check convergence of the bolt preload by activating the Keyword *CONTROL_DYNAMIC_RELAXATION that is present within the deck. Did the peak stress at time=0 change?

Analysis Run	DRTOL	Max Stress?
Std	1e-3	
Increase DRTOL	1e-5	



Analyst’s Note: We will come back to this exercise within the Implicit Section and rerun it using implicit preloading.

15. IMPLICIT-EXPLICIT SWITCHING FOR BURST CONTAINMENT

An application for implicit startup is the initialization of the steady-state stress field for rotating equipment. For systems with high-speed rotating components, the model can be initialized to its steady-state spinning condition using an implicit analysis. To model any downstream event that may be highly nonlinear and dynamic, the model can then be switched over to an explicit analysis. The trick is performing this type of analysis is to remember that LS-DYNA lets you apply boundary conditions and manage contact behavior in a “birth / death” manner. For example, one can apply constraints to all secondary structures that are not relevant, or in the case of a turbine analysis, not spinning to ensure that the implicit analysis runs smoothly and then remove them (i.e., death) once the explicit analysis starts up. Likewise, this can be done with contact behavior within the model to avoid numerical difficulties during the implicit solve. Both of these tricks can dramatically speed-up the implicit solution without affecting the accuracy of the simulation.

When initializing deformable and rigid bodies within the simulation, after dynamic relaxation, one has to repeat the *INITIAL_VELOCITY_GENERATION Keyword using *phase*=0 and then *phase*=1.

A Note from the Keyword Manual

***INITIAL_VELOCITY_GENERATION**

Purpose: Define initial velocities for rotating and translating bodies.

NOTE: Rigid body velocities cannot be reinitialized after dynamic relaxation by setting PHASE = 1 since rigid body velocities are always restored to the values that existed prior to dynamic relaxation. Reinitialization of velocities after dynamic relaxation is only available for nodal points of deformable bodies; therefore, if rigid bodies are present in the part set ID, this input should be defined twice, once for PHASE = 0 and again for PHASE = 1.

Analysts’ Note: The implicit-to-explicit switching is straightforward using a curve where any non-zero value implies a switch to implicit. For example, for analysis starting in implicit mode, the first value of the curve would be 0,1.00 and if a switch to explicit is desired at time=1.00, then one should be a point at 0.99, 0.99 and then a point at 1.00, 0.00 for the switch to explicit. The switch “drop” (1.00 to 0.00) should be less than your smallest implicit time step. For example, your implicit time step is 0.02, then your curve switch interval should be 0.01 or 0.99, 1.00 and then 1.00, 0.00.

15.1 High-Speed Rotating Equipment – *CONTROL_ACCURACY

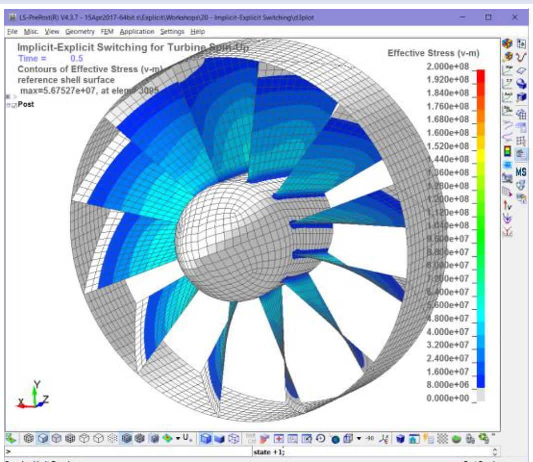
For structures that rotate, it is recommended that the *CONTROL_ACCURACY option *osu* = 1 and *inn* = 4 be set. The osu option adds additional terms to the stress update and improves the accuracy of the simulation while the INN option sets invariant node numbering to ensure the accurate calculation of element forces within elements that are become highly twisted and/or rotate through space. Interesting enough, the *inn* = 2 option is default for implicit calculations. And additionally, the osu=1 option is likewise default for implicit (when *iacc*=1). Both of these options will slow the simulation down by as much as 10%. The Keyword Manual under *CONTROL_ACCURACY has a list of activities activated by *iacc*=1.)

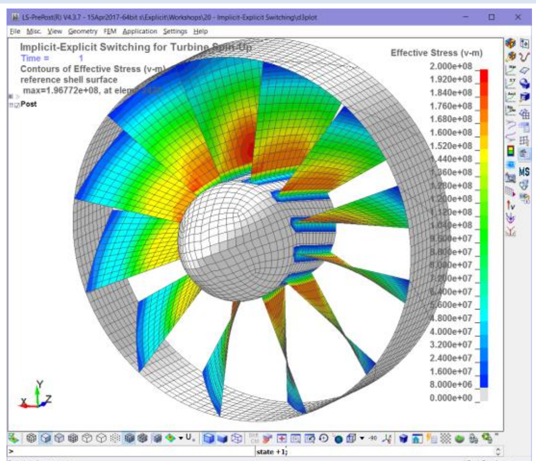
15.1.1 Workshop: 20 - Implicit-Explicit Switching for Turbine Spin Up

Introduction: This workshop will show the student how to setup a mock-turbine simulation from implicit ramp-up to steady-state rotating movement. The model has been prepared for the student to finish using the Keywords below. There is no video – just a finished file. The student is encouraged to read the manual and ask questions.

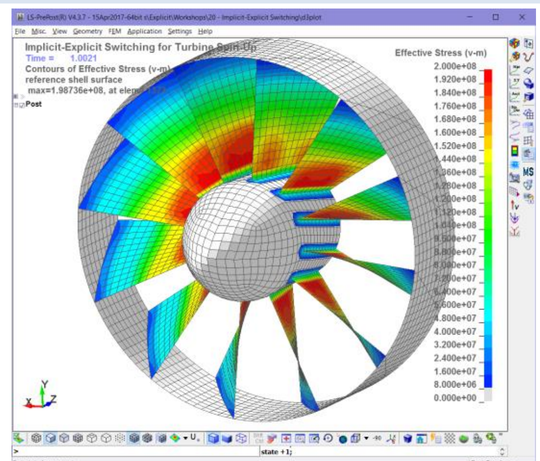
Keyword	What It Does
*CONTROL_ACCURACY (*osu*=1, *inn*=4, *iacc*=1)	*osu* & *inn* is for large rotation while *iacc*=1 is required for implicit analysis
*CONTROL_IMPLICIT_GENERAL (*imflag*=-1, *dto*= 0.25)	*Imflag*= (negative)# triggers implicit to explicit switching based on the *lcid*#
*INITIAL_VELOCITY_GENERATION (*nsid*=1, *styp*=1, *omega*=225, *nz*=1.0 and *phase*=1	Setup of initial velocity generation with a start time immediately after implicit start-up.
*INITIAL_VELOCITY_GENERATION_START_TIME (*stime*=1.00001)	And after the implicit solve, we kick off the "spin"
*LOAD_BODY_PARTS (*psid*=1), *LOAD_BODY_RZ (*lcid*=2, *sf*=225)	After initial spin, the rotation spin continues.
*BOUNDARY_SPC_SET_BIRTH_DEATH (*nsid*=1, *dofx*=1, *death*=1.0)	The structure is locked down during implicit spin-up and then released at the start of the explicit run.

Implicit Spin-Up





Explicit Spin



16. SMOOTHED PARTICLE HYDRODYNAMICS (SPH) {MESH FREE METHOD}

16.1 Introduction

A short introduction video for this section can be found in Workshops / SPH / Introduction to SPH Modeling

16.1.1 A Little Bit of Theory (skip this if you don't like math…)

Kirk Fraser, Sr. Staff Engineer, Predictive Engineering

Smoothed particle hydrodynamics (SPH) was developed in the 1970's by Monaghan, Gingold and Lucy for astrophysics problems. Monaghan has published an enormous number of papers on the SPH method. Libersky et al. [1] were the first to apply the method to solid mechanics problems. Lacome [2] was one of the first to implement SPH in LS-DYNA.

Mesh-based methods do a great job for all kinds of engineering calculations. When the deformation gets really large, mesh-based methods start to fail due to negative element volume, excessive mesh distortion and/or mesh tangling within contact region which then causes problems with the explicit time step and so on and so forth.

SPH is a Lagrangian based mesh-free method that can handle unlimited plastic deformation. The rate of change of the field variables for a given particle "i", with ***N*** "j" neighbors in the support domain is given by Lui and Lui [3]: $W_{i,j}$ is the smoothing function (interpolation kernel) and can take on many different forms depending on the type of problem being studied (e.g., the cubic spline function is popular), $\pi_{i,j}$ is an artificial viscosity term and H_i is an artificial heating term. The general idea is to use a finite number of neighbors within a radius of influence (also known as the smoothing length) on the central element. The graphic on the right depicts how the smoothing length can be visualized for a SPH mesh.

Continuity Equation

$$\frac{D\rho_i}{Dt} = \sum_{j=1}^{N} m_j ({v_i}^{\beta} - {v_j}^{\beta}) \frac{\partial W_{i,j}}{\partial x_i}$$

Conservation of Momentum

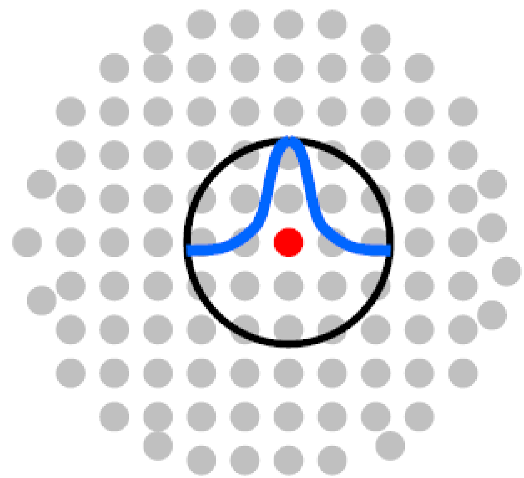
$$\frac{D{v_i}^{\alpha}}{Dt} = -\sum_{j=1}^{N} m_j \left(\frac{{\sigma_i}^{\alpha\beta}}{{\rho_i}^2} + \frac{{\sigma_j}^{\alpha\beta}}{{\rho_j}^2} + \pi_{i,j}\right) \frac{\partial W_{i,j}}{\partial x_i}$$

Conservation of Energy

$$\frac{De_i}{Dt} = \frac{1}{2}\sum_{j=1}^{N} m_j \left(\frac{p_i}{{\rho_i}^2} + \frac{p_j}{{\rho_j}^2} + \pi_{i,j}\right) ({v_i}^{\beta} - {v_j}^{\beta}) \frac{\partial W_{i,j}}{\partial x_i} + \frac{1}{\rho_i} {\tau_i}^{\alpha\beta} {\varepsilon_i}^{\alpha\beta} + H_i$$

The method converts a set of partial differential equations (PDE) into a set of ordinary differential equations (ODE). The ODE's can be integrated in time with many different schemes; in LS-DYNA a multi-step (fractional step) explicit method is used. This means that there is a stability condition on the time-step size (CFL):

$$\delta t_{min_i} = \xi \left(\frac{h_i}{c_i + v_i}\right) m_j$$



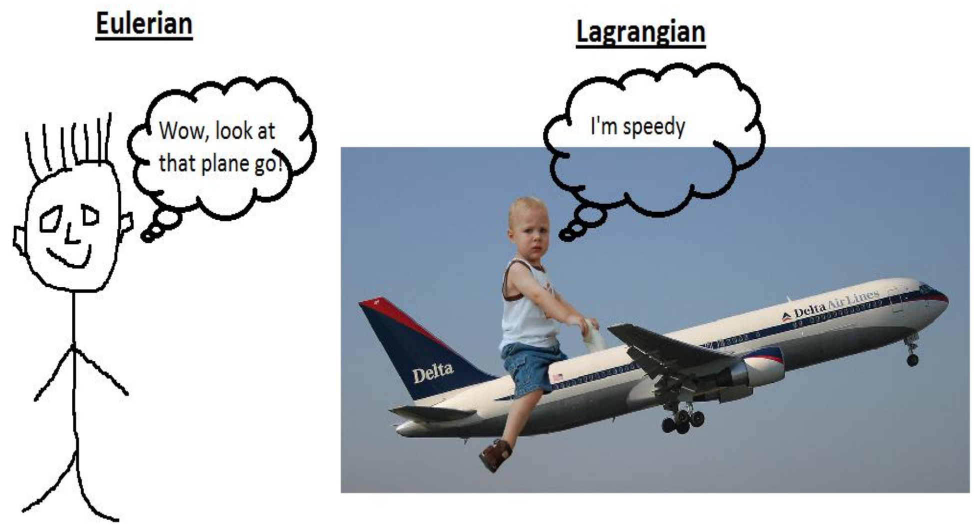
ξ is a constant and is typically 0.2 to 0.4. Implicit time integration a possibility in the future in LS-DYNA, but for now, only explicit is available. For the truly die-hard SPH gear-heads, I recommend the book by Damien Violeau [4] (you gotta really love math to enjoy this book) or for a less math intensive approach, see the book by William G. Hoover [5].

Lastly, for modeling constitutive relationships in SPH, one can use many of the same material cards (i.e., laws) as a regular Lagrangian analysis.

16.1.2 Lagrangian vs Eulerian

The two most common frames of reference for numerical simulations is Lagrangian or Eulerian. You can think of Lagrangian reference like you are sitting on a plane and Eulerian like you are on the ground (not moving) and watching the plane go by. Lagrangian makes following the history of each material point very easy compared with the Eulerian description. CFD codes (finite difference, finite volume and finite element) are written from an Eulerian formulation.

The Lagrangian nature of SPH makes it a very powerful numerical method, this opens many doors that Eulerian method closes (or makes it difficult to open the door).



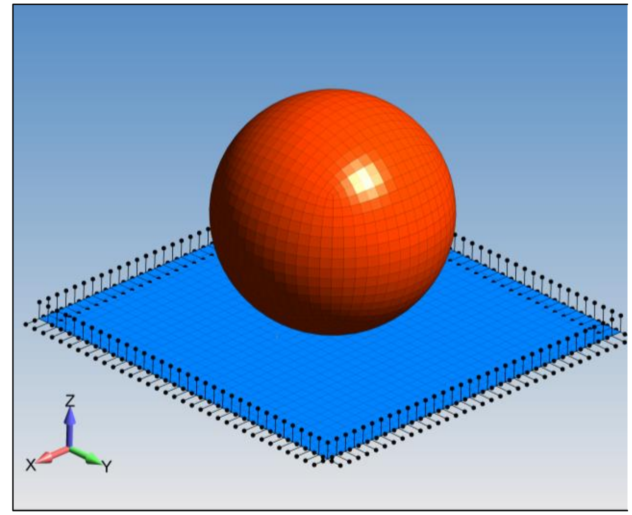
16.1.3 Types of Simulations with SPH

- Impact and ballistics (e.g., bird strike)
- Fracture and fragmentation
- Fluid structure interaction (e.g., sloshing)
- Linear and non-linear vibrations
- Microstructure evolution
- Heat transfer
- And many more…

16.1.4 Common Keywords for SPH

- *CONTROL_SPH
- *SECTION_SPH
- *CONTACT_AUTOMATIC_**NODES**_TO_SURFACE with SOFT=1 is recommended
- All *EOS_ and most *MAT_ (see Keywords manual for details, [6] and [7]) cards can be directly used)
  For example: *MAT_001 (ELASTIC), *MAT_024 (PIECEWISE_), *MAT_077 (OGDEN_RUBBER), *MAT_083 (FU_CHANG_FOAM)
- Most of the standard keywords work for SPH
- ***Node sets*** need to be used to define contact, boundary conditions, etc.

16.2 Workshop: 21A - SPH Getting Started – Ball Hitting Surface

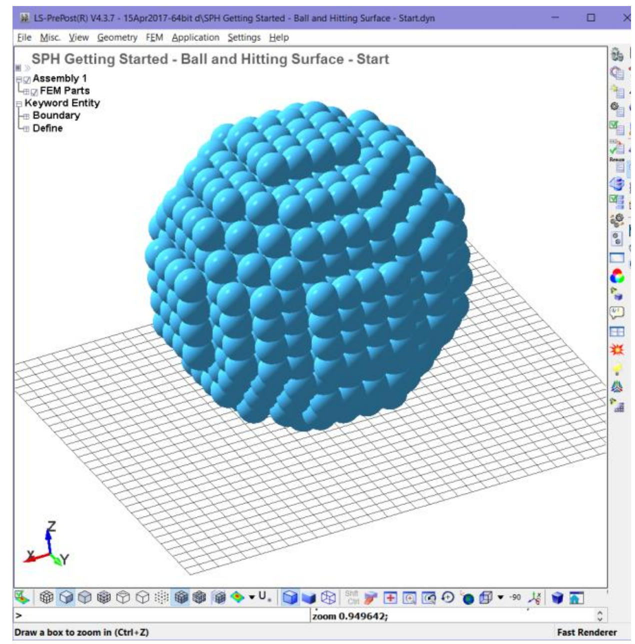


Objective: This example is geared toward demystify the process of creating and running a SPH model. The concept is that any type of closed shell mesh or standard geometric entity (box, sphere, cylinder, etc.) can be used to create a SPH mesh. The Workshop starts with the shell mesh setup shown on the right. The surface is converted into soft SPH material and then impacted against the plate.

Tasks

- Open SPH Getting Started - Ball and Hitting Surface - Start.dyn in LSPP and create the SPH nodes {Element and Mesh / SPH Generation /[Select Shell Surface (byPart), check Del Old Parts, Num Particles Definition PitX(Y)(Z)=0.05, Density=1,000 and then hit Set Params, Start PID: 10, Create and Accept].
- Visualize the spheres via Settings / General Settings / SPH / Particle.
- Setup SPH to run using the following Keyword commands (minus one):

*CONTROL_SPH, *idim*=3, *ithk*=1	For 3D problems and *ithk*=1 uses the SPH node radius for contact thickness.
*SECTION_SPH, secid=10	All defaults. The *cslh* merits attention.
*MAT_ELASTIC, mid=10	Elastic SPH, *ro*=1000, *e*=2e8 & *pr*=0.3
*PART, pid=10, etc.	{self-explanatory 10, 10 & 10}
*CONTACT_AUTOMATIC_NODES_TO_SURFACE _ID	One has to create a node set: Model and Part / Create Entity / Set Data then for contact set *soft*=1 and *bsort*=10. *Note: that a defined sst value will override the ithk value.*
*INITIAL_VELOCITY_GENERATION	Setup initial velocity card with *vz*=-10.0 with *styp*=3 and *id*=1
*CONTROL_TERMINATION	*endtim*=0.025



*Analyst's Explanation on *ELEMENT_SPH Mass (mass) and *MAT Density (ro): These items are used together in a couple of ways. Let's just explain what happens when *ELEMENT_SPH mass is positive (as in the example above). When LSPP creates the SPH nodes, it assigns a mass to each node based on how the SPH nodes were generated (spacing and density). This mass is used with the *MAT Density value (ro) to define the SPH nodes' volume. Thus both (mass and ro) must be defined to define the SPH "element".*

For a mathematical explanation of SPH theory, see Class Reference Notes / SPH and SPG/ LSTC SPH Short Course Notes.pdf

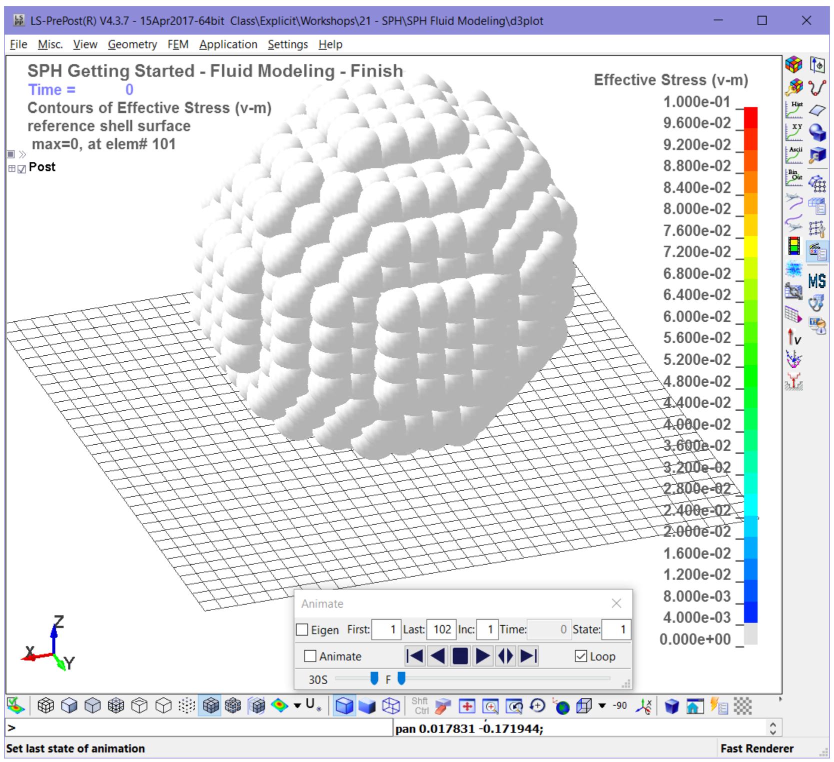
16.3 Workshop: 21B - SPH Getting Started - Fluid Modeling

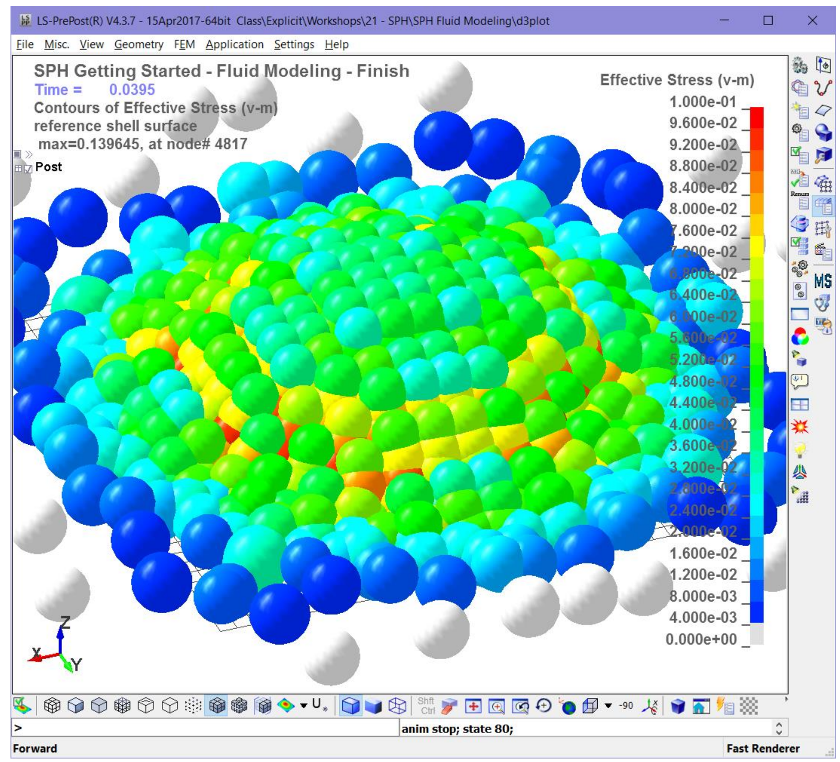
Objective: Learn how to model a fluid.

Introduction: The ball is turned into water by using two cards: *MAT_NULL and *EOS_GRUNEISEN. LS-DYNA uses *MAT_NULL as a placeholder to insert material card information for fluids. It is used in combination with an *EOS type of material law. Please note that this operation is done within LSPP using the prior SPH Getting Started - Ball and Hitting Surface - Finish.dyn as your starting point.

*MAT_NULL	mid=10, ro=1,000 & mu=0.001
*EOS_GRUNEISEN	eosid=10, c=342 (speed of sound in water)
*PART, pid=10	Update *PART card with eosid=10
_TERMINATION	Update endtime=0.05 to let the fluid flow

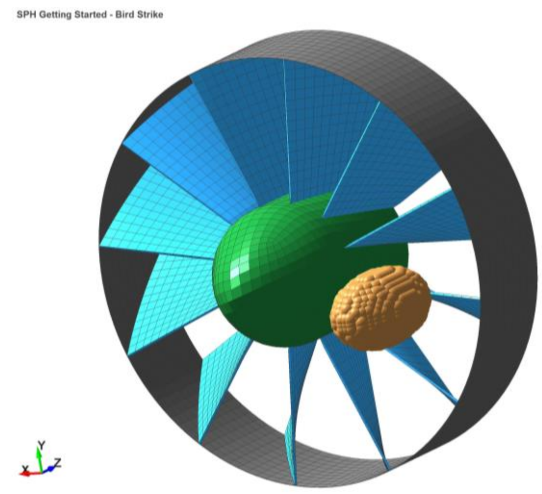
Post-Processing Note: The movie file covers post-processing techniques for color contouring etc.





16.4 Workshop: 21C – SPH Getting Started – Bird Strike

Introduction: The Bird Strike workshop essentially leverages all the prior workshops and explores in a bit greater detail the *CONTROL_SPH control card options ISHOW, IEROD and ICONT to make the simulation more efficient. Lastly, Blade Failure is activated by enabling material failure using *MAT_ADD_EROSION, mxeps=0.05 and contact between the fan blades and everything else (_SINGLE_SURFACE).



Tasks:

⇒ Open SPH Bird Strike Default - Start.dyn in a text editor and inspect the file.

⇒ What is the bird's material model? _______________

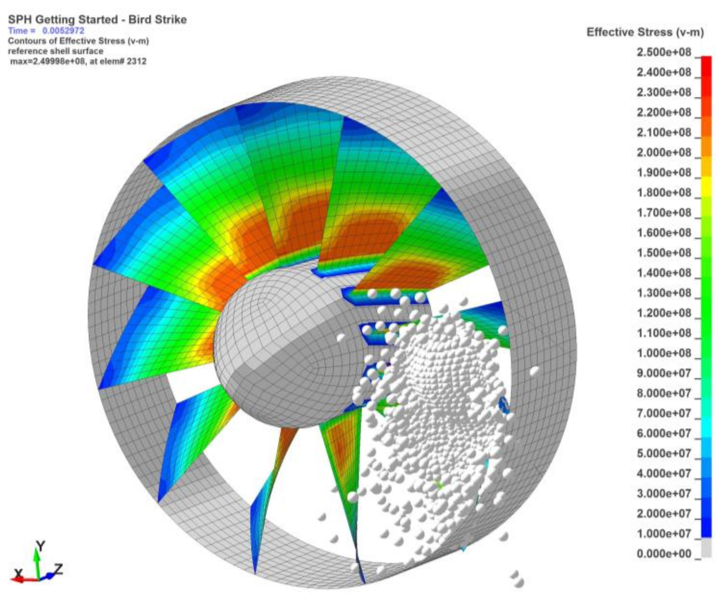
⇒ Is the turbine initialized prior to bird strike? Could you make it happen?

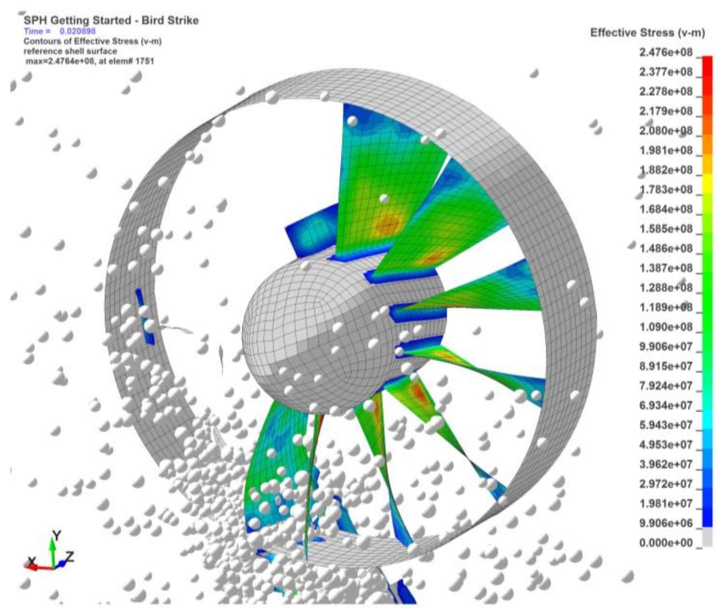
⇒ Run the simulation. Does it appear reasonable? Check energies (hourglass, slnten, etc.)

⇒ Activate Blade Failure {see Keyword Deck}

⇒ If this goes smoothly keep reading the deck and look at the commands for *CONTROL_SPH. Also, could your SPH material be *MAT_024?

Defaults	BLADE FAILURE





16.4.1 Bird Strike Models

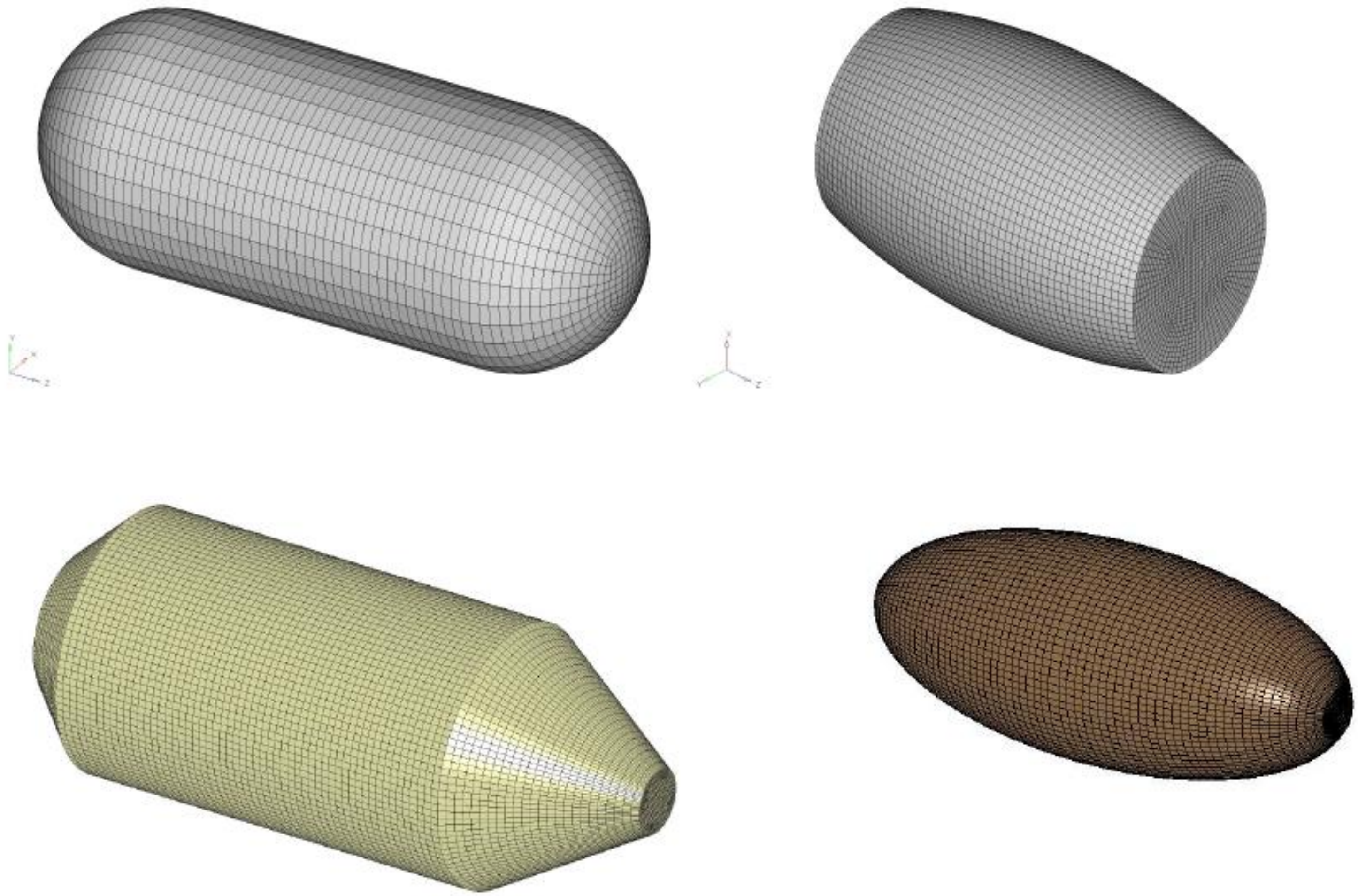


Figure 5.4 Various bird models used in rotating bird strike analyses.

If one ever wonders what a spherical chicken looks like. Image courtesy of Aerospace_MGD (see Class Reference Notes / Aerospace Working Group)

16.5 References

[1] L. D. Libersky, A. G. Petschek, T. C. Carney *et al.*, "High Strain Lagrangian Hydrodynamics: A Three-Dimensional SPH Code for Dynamic Material Response," *Journal of Computational Physics,* vol. 109, no. 1, pp. 67-75, 11//, 1993.

[2] J. L. Lacome, "Smooth Particle Hydrodynamics (SPH): A New Feature in LS-DYNA."

[3] G.-R. Liu, and M. B. Liu, *Smoothed particle hydrodynamics : a meshfree particle method*, Hackensack, New Jersey: World Scientific, 2003.

[4] D. Voileau, *Fluid Mechanics and the SPH Method: Theory and Applications*, Oxford, UK: Oxford University Press, 2012.

[5] W. G. Hoover, *Smooth Particle Applied Mechanics: The State of the Art (Advanced Series in Nonlinear Dynamics)*, Singapore: World Scientific Publishing, 2006.

[6] LSTC, "LS-DYNA Keywords User Manual Volume 1," no. Version 971, July 12, 2012, 2012.

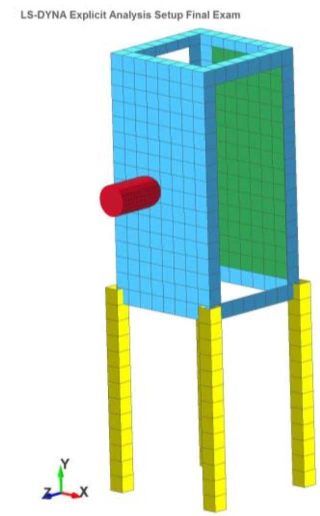
[7] LSTC, "LS-DYNA Materials User Manual Volume 2," no. Version 971, July 12, 2012, 2012.

[8] S. Marrone, A. Colagrossi, D. Le Touzé *et al.*, "Fast free-surface detection and level-set function definition in SPH solvers," *Journal of Computational Physics,* vol. 229, no. 10, pp. 3652-3663, 2010.

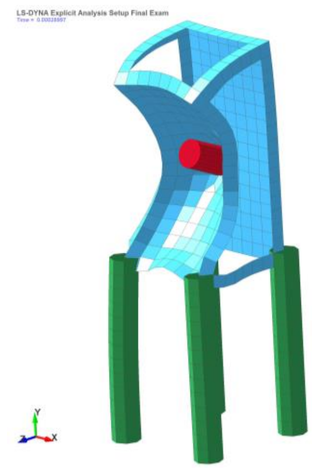
17. EXPLICIT EXAMINATION

Objective: This is the chance for the student to test their knowledge and build an explicit analysis from the ground up. The mesh is provided (nodes and elements) but that is it.

Introduction: A steel bullet (PART 1) is given an initial velocity and then impacts the structure. The structure is steel (PART 2). The structure is anchored at the base. The legs are beam elements (PART 3) that must be _TIED onto the steel structure. Setup the model for impact and report contact force between the bullet and the structure (Force Transducer). Also tie the impact force to the reaction forces (SPC) at the base. One should also provide a quick verification graph of the simulation energies. A final model with a report is provided in the folder to see if you pass or not. If you have time, please add conventional mass scaling using *CONTROL_TIMESTEP with a –dt2ms value of your logical choosing to speed up the simulation. Also, knowing what you know about explicit elements (even fully integrated), would you trust this mesh? LSPP has remeshing tools and one could split the mesh quite easily and check your results. If you are ahead, try this option.

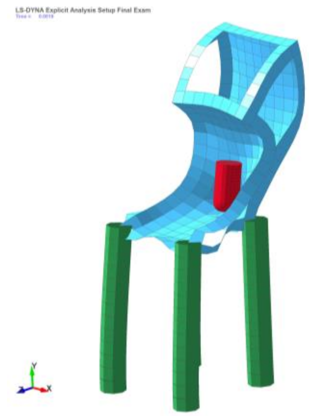


Task	Keyword to Add:	Workshop Used:
Constrain all 6 DOF at base of legs (nodes 1, 2, 3 & 4)	*BOUNDARY_SPC_NODE	Building The Better Beam - Start.dyn
Assign initial velocity (Z-Axis -15,000 in/s) on PART 1	*INITIAL_VELOCITY_ GENERATION	Basic Contact - Pipe-on-Pipe Contact - Start.dyn
Stainless steel material for all PARTS	*MAT_098	Elastic-Plastic Material Modeling (MAT_098) - FINISH - Explicit.dyn
Enforce Contact between all components	*CONTACT_AUTOMATIC_SING LE_SURFACE _ID	Basic Contact - Pipe-on-Pipe Contact - Part II-A.dyn with *soft*=2 and *depth*=5
Create a force transducer for Part 2	*CONTACT_FORCE_ TRANSDUCER_ID	(see file above). We want the force on the bullet, *ssid*=1, *sstyp*=3
Define element *SECTION properties for solid, shell and beam. The shell thickness is 0.050 and the legs are solid cylinders with an OD=0.50.	*SECTION_SOLID_TITLE *SECTION_SHELL_TITLE *SECTION_BEAM_TITLE	Elastic-Plastic Material Modeling (MAT_098) - FINISH - Explicit.dyn and Dynamic Relaxation - Bolt Preload Prior to Transient - Finish.dyn



EXPLICIT EXAMINATION (CONTINUED)

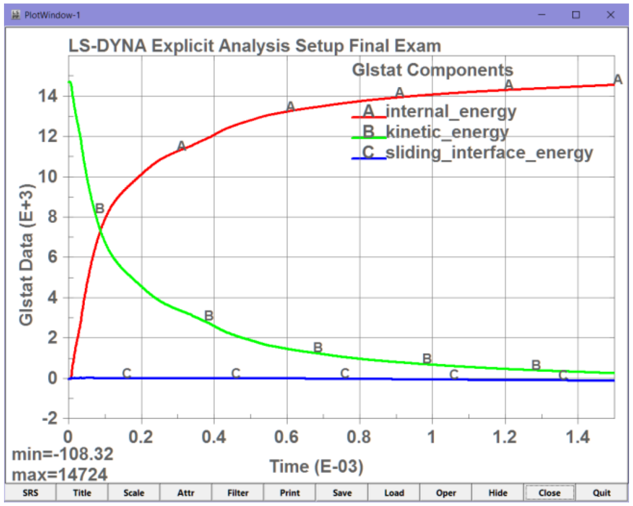
Set up PART definitions 1 thru 3.	*PART	Any workshop since it is a mandatory card
Weld the legs onto the frame. This can be done simply by a PART to PART definition in the Keyword card.	*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET_ID	Tied Contact - Gluing Things Together - Finish non-OFFSET and _OFFSET.dyn. *Please* note that the beam part should be the *surfa* (think nodes to surface).
Create database requests for d3plot and asci files (GLSTAT, SPCFORC and RCFORC).	*DATABASE_BINARY_D3PLOT, *DATABASE_GLSTAT, _SPCFORC and _RCFORC	Basic Contact - Pipe-on-Pipe Contact - Part II-A.dyn
And of course, we need to end the simulation at some time, say 0.0015	*CONTROL_TERMINATION	Any workshop since it is a mandatory card



Extra: If you have time, speed up the simulation with mass scaling and also refine the mesh.

Post-Processing: Graphing data is essential toward the verification of model. Although it can seem tedious and sometimes confusing wandering around the LSPP interface, time spent during this process can often be the best time one can spend as a simulation engineer.

The two tasks that the student should be comfortable with is to plot energies (as shown on the right) and then to compare the SPC summed reaction forces with the contact forces. This requires one to save curve data and then replot it along with the rcforce data.



18. EXPLICIT MODEL CHECK-OUT AND RECOMMENDATIONS

If you think you might have a simulation that is working, here's a short list of things to check for and review.

Here's an order of checking: Units | Mesh | d3hsp File | History Plots | Material Modeling | Contact Behavior | Etc.

18.1 Units

It is recommended to settle on one unit system for as much of your LS-DYNA work as possible to avoid unit problems when one is unfamiliar with a specific system. A commonly recommend system for dynamic events is the kN, mm, ms, kg system. Stresses are then in GPa. We have covered this before but it is hard to overstate the importance of getting your units straight. In a dynamic analysis, the mass of the system should always be checked.

18.2 Mesh

When looking at your mesh, it should look good and if it looks good, it will generate a smooth stress contour. This is never more so important than for an explicit analysis. If this sounds odd, please see Class Reference Notes / Stress Visualization / Desktop Engineering Stress Visualization Article March 2011.pdf.

Besides this Zen of meshing statement, here are some bulleted items to consider:

- Is the mesh density sufficient to capture the mechanical response?
- A contact interface has equal mesh densities across opposing faces. Remember, contact is by segments and contact forces are resolved at nodes; thus a uniform opposing mesh will provide smooth contact behavior;
- Lastly, check the explicit time step. Seriously, a couple bad elements can completely explode the analysis (personal experience that cost me a weekend). This can be done easily via LSPP and viewing the D3hsp file under 100 smallest timesteps.

18.2.1 Using Surface Elements to Improve Stress Reporting Accuracy

An old school technique is to create a skin of membrane elements over an existing solid mesh. This skin then allows a clear representation of surface stresses. However, since the analyst usually sets the thickness of these membrane elements to a very small number, numerical issues can arise due to very low rotational stiffness. An elegant solution is just to use the membrane element formulation (*elform*=5) and this also is recommended for an implicit solution. This technique was provided by Roger Grimes, LSTC in his paper "A Tutorial on How to Use Implicit LS-DYNA" (see www.DYNAlook.com).

18.3 Mass Scaling

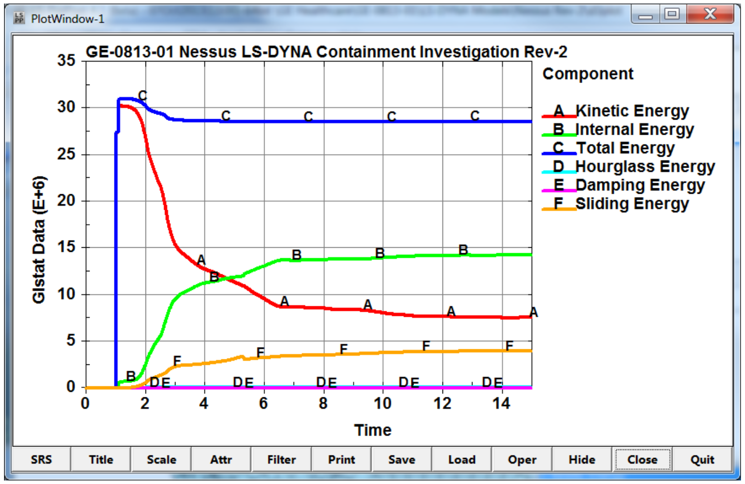
If mass scaling is used, one should look at the *GLSTAT / added_mass plot to assess how much mass was added to the simulation. In concert, one should contour the added mass at the beginning and the end of the simulation. To request this fringe plot, see *DATABASE_EXTENT_BINARY / *msscl.*

18.4 D3hsp File (LS-DYNA Equivalent to the Nastran F06 File)

- The *d3hsp* file summarizes the input in descriptive terms and can be viewed via a text editor or from within LSPP via Misc. and then D3hsp View.

  Within the d3hsp, one should review:
  - Verify mass of system (find "summary")
  - 100 elements listed in ascending order of time step (find "smallest")
  - Review rigid bodies for any that might be deleted or with super-small mass
  - Warning and error messages (find "Warning" or "Error")

18.5 Energy Plots



The GLSTAT file is your first stop for checking the analysis. A fundamental check is that your Energy Ratio should be 1.0 (+/- 0.01). This can be quickly checked within the ASCII, GLSTT file under Energy Ratio. An example of a more complex energy plot is shown on the right. The high Sliding Energy is because the model is simulating a burst containment of a fragmenting X-ray target with friction. For more information on Energy Data, see www.dynasupport.com under LS-DYNA User's Guides and then Energy Data also see Class Reference Notes / Energy Balance / Total energy LS DYNA Support.pdf

18.5.1 Sliding Interface Energy (Contacts)

Sliding energy (GLSTAT – global – complete model) and sliding interface energy (ASCII (SMP) or Binary Output (MPP)– individual contacts) say a lot about the numerical validity of your contacts. If friction is zero, than the sliding energy value should be 5% or less of the internal energy. Interrogate local sliding energy by plotting all of the SLEOUT values for the individual contacts. If individual (contact) sliding energy values are negative greater than 5% of the peak internal energy, be worried and start digging.

18.5.1.1.1 Reporting Frictional Sliding Interface Energy

If one needs friction within the model for various physical realities, then one can also just request that the frictional energy be reported as a separate item. This is done by setting *frceng*=1 within the *CONTROL_CONTACT card. Values can then be plotted as just another item within the sliding interface energy file (*DATABASE_SLEOUT). It does get complicated but any sliding interface energy that is not related to friction is energy that is non-physical and is moving your simulation away from physical reality.

18.6 Material Modeling Errors

Although much-ado is made about strain rate sensitivity, for most engineering applications, the only really strain rate sensitive materials are carbon based materials (rubbers, elastomers, plastics and foams) and to obtain such data is not that difficult from any good material testing laboratory. What is more common is just plain screwing up the material model. Thus, it is almost mandatory to demonstrate with a virtual coupon test model that one can match test data with the LS-DYNA model. Such correlation should be within every engineering report. Never underestimate the power of "KISS" and always attend to the basics before making your life more difficult.

18.7 Contact Options with Recommendations and *CONTROL_CONTACT Options

Our recommendation is to start with _MORTAR and get your simulation to run. Once everything is working well, one can then increase the numerical efficiency (i.e., make it run faster), otherwise here are some recommendations:

soft=2 / *depth*=5	*soft*=2 is really a quite good standard contact option. To pick up edge contact, one might want to employ *depth*=5.
vdc=20	Contact is often noisy and adding 20% damping can be a nice option to smooth things out. It is one of those tweaks that is often times worth investigation once you have the model working.
isym=1	If you have symmetry faces via SPC's, please be aware of this option.

*CONTROL_CONTACT

This Keyword allows the override of all contact options set on individually defined contacts. It has some utility to ensure consistent treatment of contact within a complicated model with lots of contact types.

frceng=1	This option turns on the calculation of friction. It is reported under SLEOUT (ASCII or BINOUT file) as friction_energy.

Analyst's Note: One should always check the Sliding Interface Energy and if it is high (<10% of the Internal Energy) with friction enabled, one should make an additional run with all friction set to 0.0 to confirm that the contact setup is not generating inappropriately high Sliding Interface Energy.

Likewise, one should always heed the message about "The LS-DYNA time step size should not exceed X.XXXE-XX to avoid contact instabilities. It is suggested that one run analyses not exceeding this time step to evaluate its effect and of course, check the positive sliding energy as a function of the time step. Lower sliding positive energy (assuming little friction) implies more numerically efficient contacts. Lastly, if you are reading this far…go thru all the warning messages and verify that you are okay. You will be happier in the long run to spend some quality time understanding what the program is telling you and yes, most likely one can safely ignore many of the warning messages but one should nevertheless understand prior to dismissing.

18.7.1 *CONTROL_TIED GLOBAL RECOMMENDATION

Keyword	DOF	Card Option	Comment
_TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET	3	*ipback*=1	General purpose _TIED contact for tying solids (3 DOF nodes) to shells and solids. If the tie interface is coplanar or offset, the _OFFSET feature handles both situations. Plus being _CONSTRAINED it eliminates any problems with a spring formulation. The ipback=1 option is useful if one needs to tie to rigid bodies. If a rigid body is present in the tie definition, then the formulation is automatically switched to a penalty based algorithm.
_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET	6	*ipback*=1	With SHELL_EDGE option, one ties all 6 DOF of the *surfa* side nodes and if one needs to tie to rigid bodies, the ipback=1 option is available.

Analyst's Note: This recommendation comes after many years of _TIED introspection and troubleshooting. It is a work in progress but given the maturity of the LS-DYNA program, the go-forward quest is to only use one _TIED formulation to handle all tied contacts. The advantage of the _CONSTRAINED formulation is that springs are not use and the constraint relationships are created between the adjacent surfaces. This eliminates the possibility of forming negative sliding interface energy and assures a clean tie that will work well for explicit and implicit analyses.

18.8 Control Cards with Recommendations

Commonly used options are shown in the table below:

*CONTROL_CONTACT	*ssthk*=1, *ignore*=1 & *shledg*=1	With *soft*=2 and _MORTAR, *ssthk, ignore* and *shledg* are automatically set to 1.
*CONTROL_ENERGY	*hgen*=2 & {*slnten*=2}	Calculates hourglass energy and sliding interface energy. Although it adds computational expense it should be added and checked and then one can delete it later. If fully integrated elements are used (e.g., *elform*=-16 (shells), no hourglass energy calculation is needed since there is no "hourglass". Please note that slnten is automatically set to 2 with *CONTACT.
*CONTROL_SHELL	*esort*=2	Switches shell element formulation (elform) for triangular elements to elform=17. This is recommended for shell meshes (e.g., one has a mixed mesh of quad's and tri's where the elform=-16. Setting *esort*=2 will switch the tri's elform from -16 to 17 at the start of the analysis.)
	nfail1=1 & *nfail4*=1 **Let's Talk about istupd**	Deletes highly distorted elements prior to them causing harm to your simulation. There are also other element checks that can be useful where one can set the distortion level (e.g., stretch).
*CONTROL_SOLID	*esort*=1	Automatic sorting of tetrahedron and pentahedron elements to treat degenerate tetrahedron and pentahedron elements as tetrahedron (formulation 10). However, most LS-DYNA models don't have tetrahedrals but it is included for completeness.
*CONTROL_TIMESTEP	*dt2ms* (negative)	Just admit it, you'll going to use some mass scaling. If so, don't forget to contour the added mass to verify it didn't get to crazy (see below for recommendation on contouring added mass). Assume that mass scaling is bad and understand where the "bad" is in your model via the use of msscl (see *DATABASE_EXTENT_BINARY).
	erode=1	Just good practice when using solid elements. If the element becomes highly distorted to the point of a negative volume, it'll be deleted without killing the simulation.
*CONTROL_RIGID	*plotel*=1 or 2	If one is using CNRB's, this command will add plot-only elements to the d3plot file and allow visualization.

18.9 Database Cards with Recommendations

*DATABASE_{option}	GLSTAT, MATSUM, SLEOUT & SPCFORC	This is the minimum recommended set. Please note if you have rigidwalls in your simulation, you should also have RWFORC enabled.
*DATABASE_EXTENT_BINARY	*beamip* (1) and if you are using mass scaling, you might want to set *msscl*=2	Beam elements are isoparametric elements and have integration points, setting beamip=1 writes beam stress out to the d3plot file. To dump out added mass information and then also check *msscl*=1 or =2 to indicate incremental or percentage increase of added mass (my choice is *msscl*=2). This is very useful to check your model for mass scaling effects. To contour the added mass, see LSPP, Fcomp / Misc / Mass Scaling (whereas in this case, it is the added mass). *Please note, it is really mandatory if you are going to be aggressive with your mass scaling to contour this item and be aware of where you are adding mass to your structure.*

18.10 Explicit Element Recommendations

*SECTION_BEAM	*elform*=1	Default. Remember that beams output stresses at the middle of the beam per the integration rule.
*SECTION_SHELL	*elform*=-16, *shrf*=0.833 and *nip*=5	Default *elform*=2, one point integration – perfectly fine for most explicit analyses. If hourglassing is high and/or your mesh is course, *elform*=-16 can help since it is fully integrated. Set shear factor to 5/6 (recommended).
*SECTION_SOLD	Hex: *elform*=-1 | Tet: *elform*=13	Standard recommendations

18.11 Etc

I'm a big fan of building stupid, simple, itty-bitty test models to evaluate a proposed behavior. A standard downfall of many simulations is an attempt to model all the physics out-of-the-gate without prior evaluation of the effects of individual items, in brief, the more complex the model, the more heinous is the debugging.

19. IMPLICIT ANALYSIS

19.1 Introduction

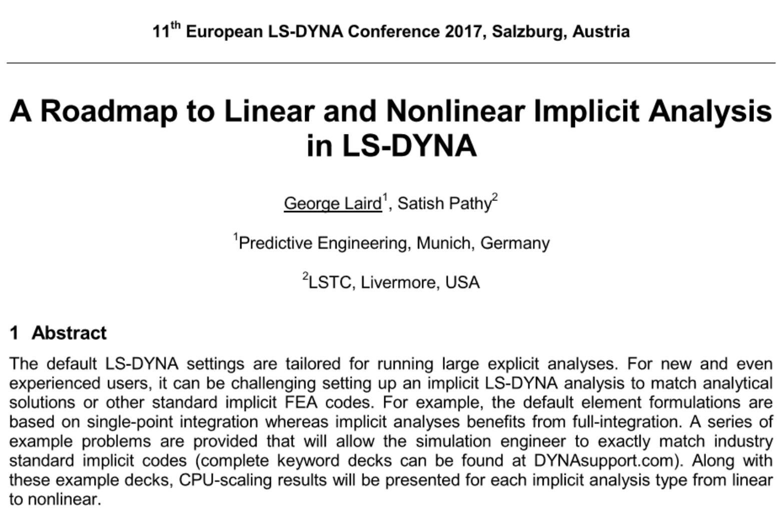
19.1.1 Why Implicit?

It has been our experience that one can learn a lot about the mechanical behavior of a structure by a quick linear elastic implicit analysis and somedays, that is all you need to stop what you are doing and call out for a redesign. And, of course, there is a large class of FEA simulations where dynamics effects do not exist or add complexity that is counter-productive. Our objective with these notes is to show how LS-DYNA can easily be used for classic linear elastic analysis work (implicit) and also for highly nonlinear implicit work that would typically require an explicit approach. We advocate modeling techniques where one model can serve multiple analysis requirements and truly exploit the capabilities of LS-DYNA.

19.1.2 What we cover

- Implicit FEA Mechanics – Always use a LS-DYNA Double-Precision Solver
- The technology of creating accurate nonlinear, implicit FEA models
- How to do your own research to create more advanced simulations
- Our condensed experience and that of our colleague's to help you *not* repeat our mistakes

See Class Reference Notes / Implicit


11th European LS-DYNA Conference 2017, Salzburg, Austria

A Roadmap to Linear and Nonlinear Implicit Analysis in LS-DYNA

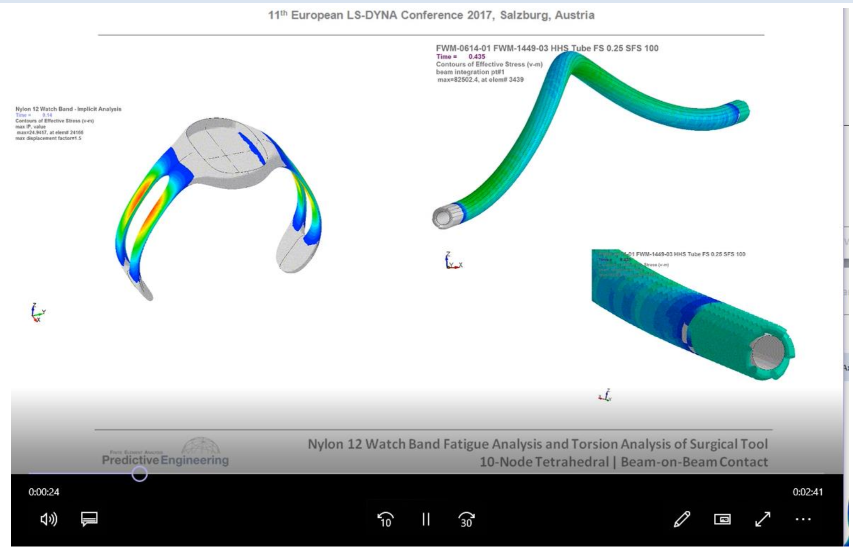
George Laird[1], Satish Pathy[2]

[1]Predictive Engineering, Munich, Germany

[2]LSTC, Livermore, USA

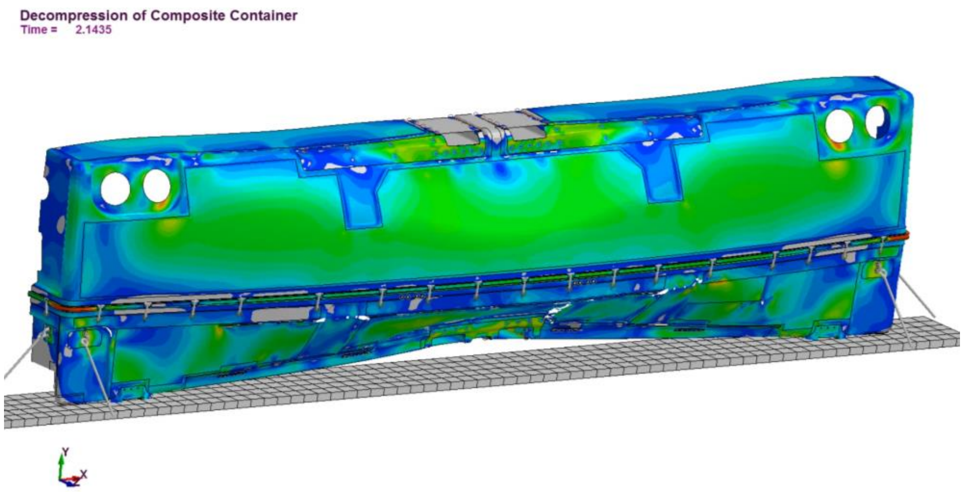
1 Abstract

The default LS-DYNA settings are tailored for running large explicit analyses. For new and even experienced users, it can be challenging setting up an implicit LS-DYNA analysis to match analytical solutions or other standard implicit FEA codes. For example, the default element formulations are based on single-point integration whereas implicit analyses benefits from full-integration. A series of example problems are provided that will allow the simulation engineer to exactly match industry standard implicit codes (complete keyword decks can be found at DYNAsupport.com). Along with these example decks, CPU-scaling results will be presented for each implicit analysis type from linear to nonlinear.

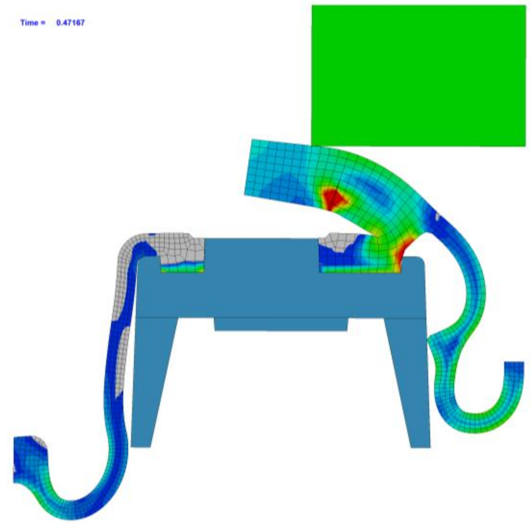


19.1.3 What Sort of Problems Can We Solve in Implicit?

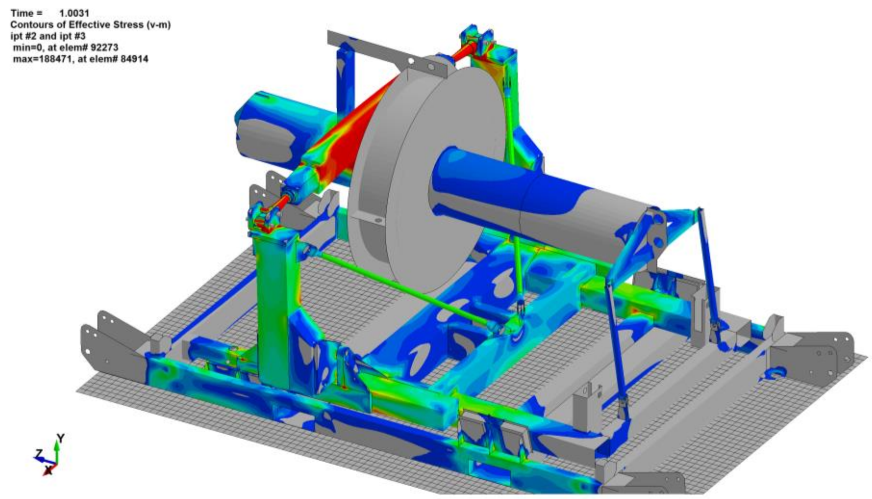
Drop, Rail Impact and PSD Analysis of Composite Container (MAT_54 Failure)



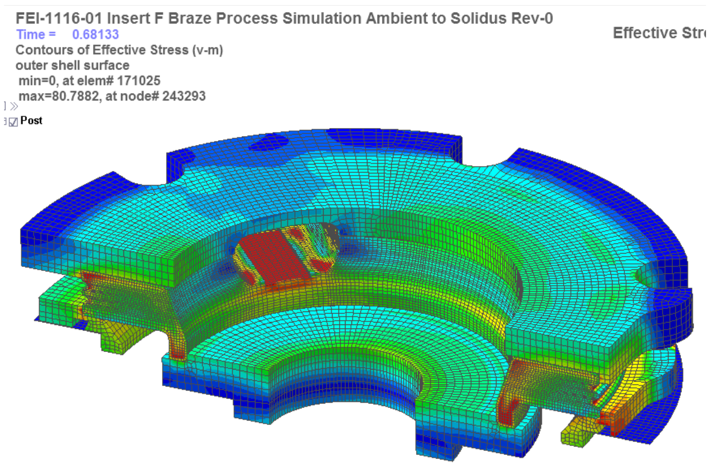
Axisymmetric Rubber Seal Analysis



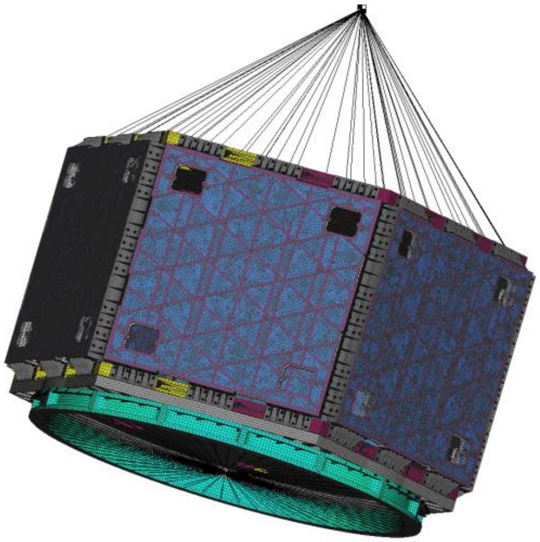
9g Crash Analysis of Jet Engine Stand



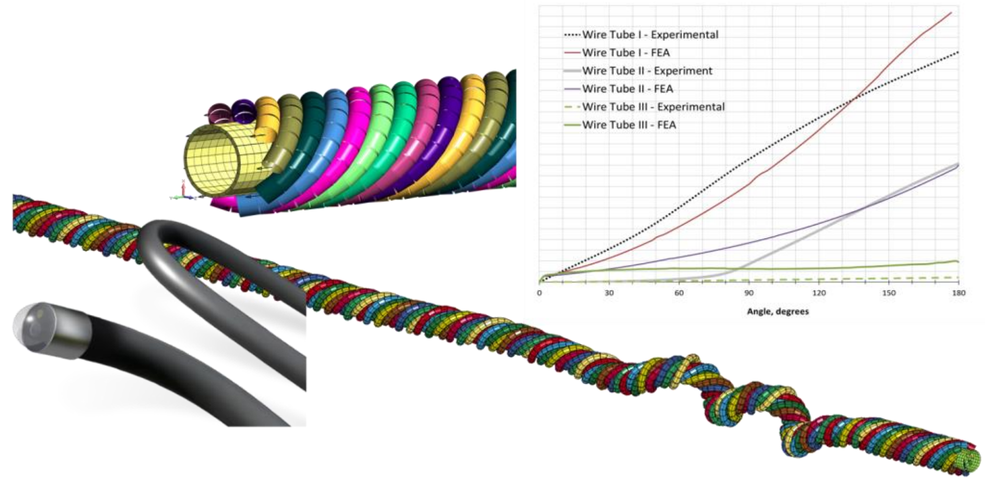
Braze Process Simulation (MAT_188)



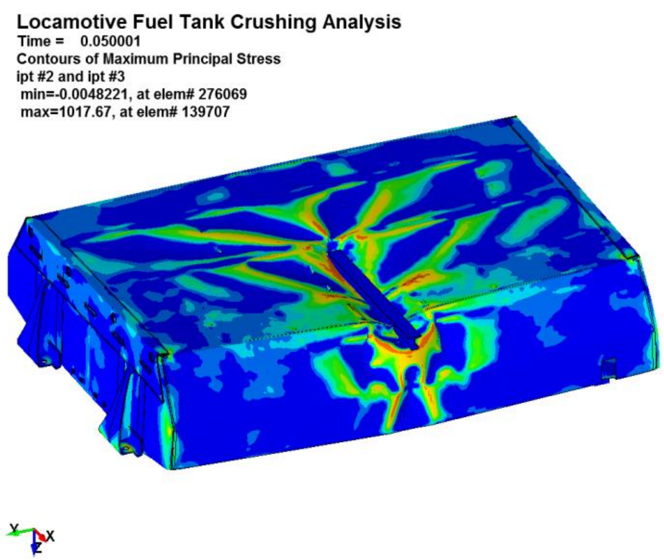
Stress and PSD Analysis



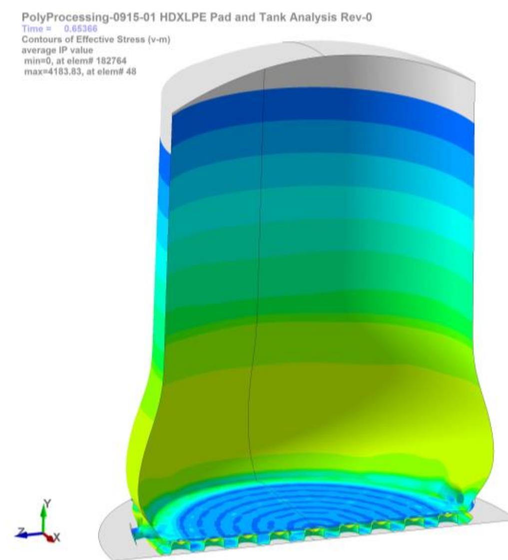
Torque Analysis of Endoscopic Medical Device (Beam-on-Beam Contact)



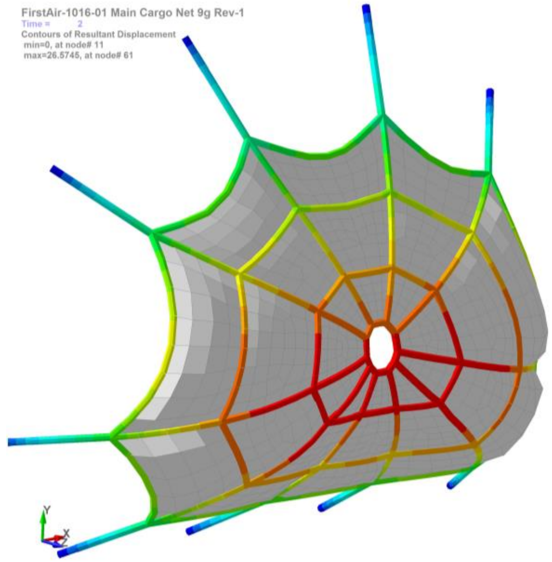
Fuel Tank Static Impact Analysis



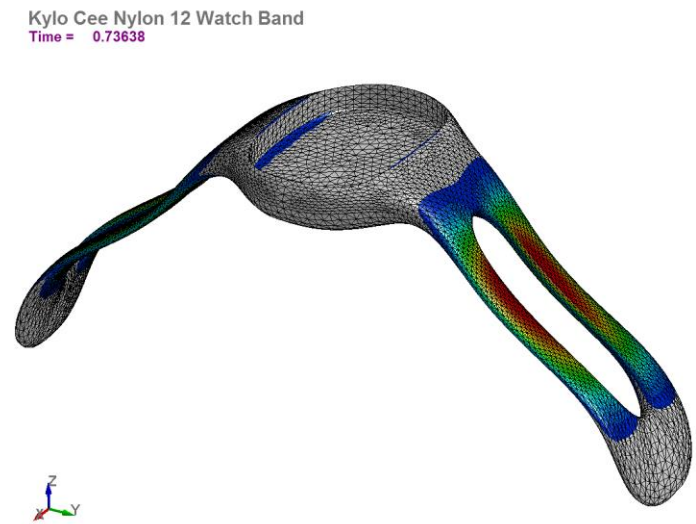
Plastic HDPE Acid Storage Tank



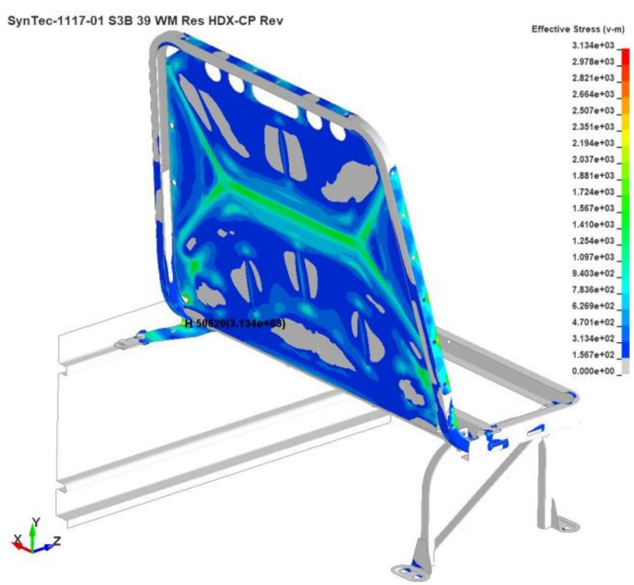
Cargo Net Analysis – 9g Crash Landing Load



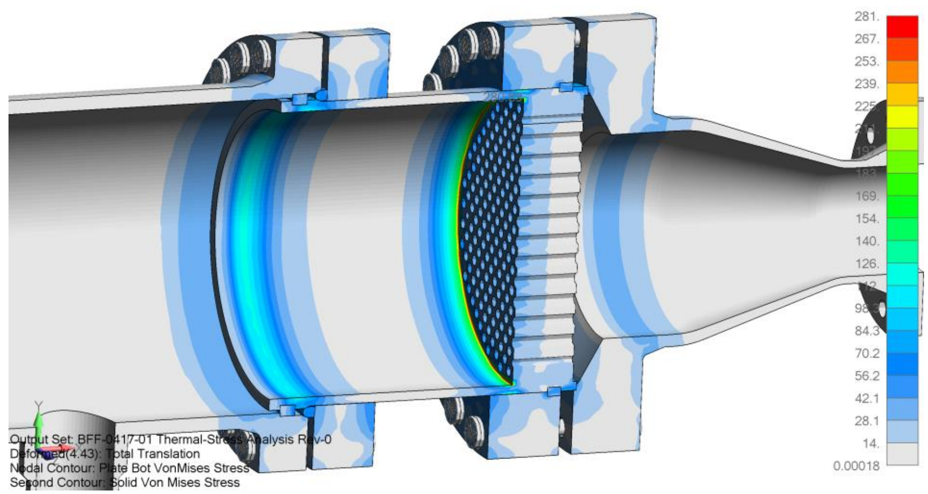
Nylon 12 Watch Band



PSD Analysis of Bus Seat under Road Vibration



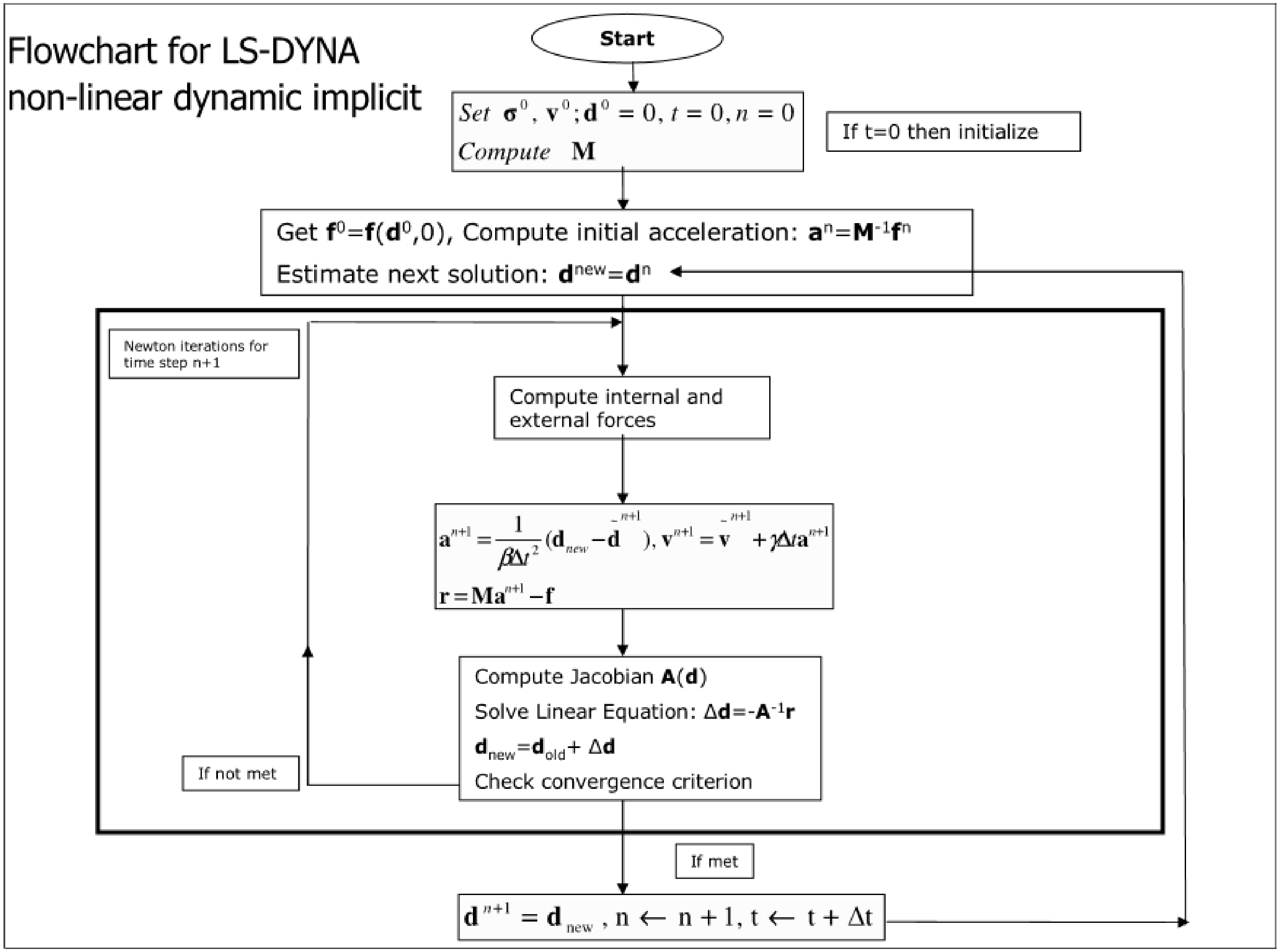
Nonlinear Contact – Fatigue Analysis of Pressure Vessel



19.2 Implicit versus Explicit Analysis

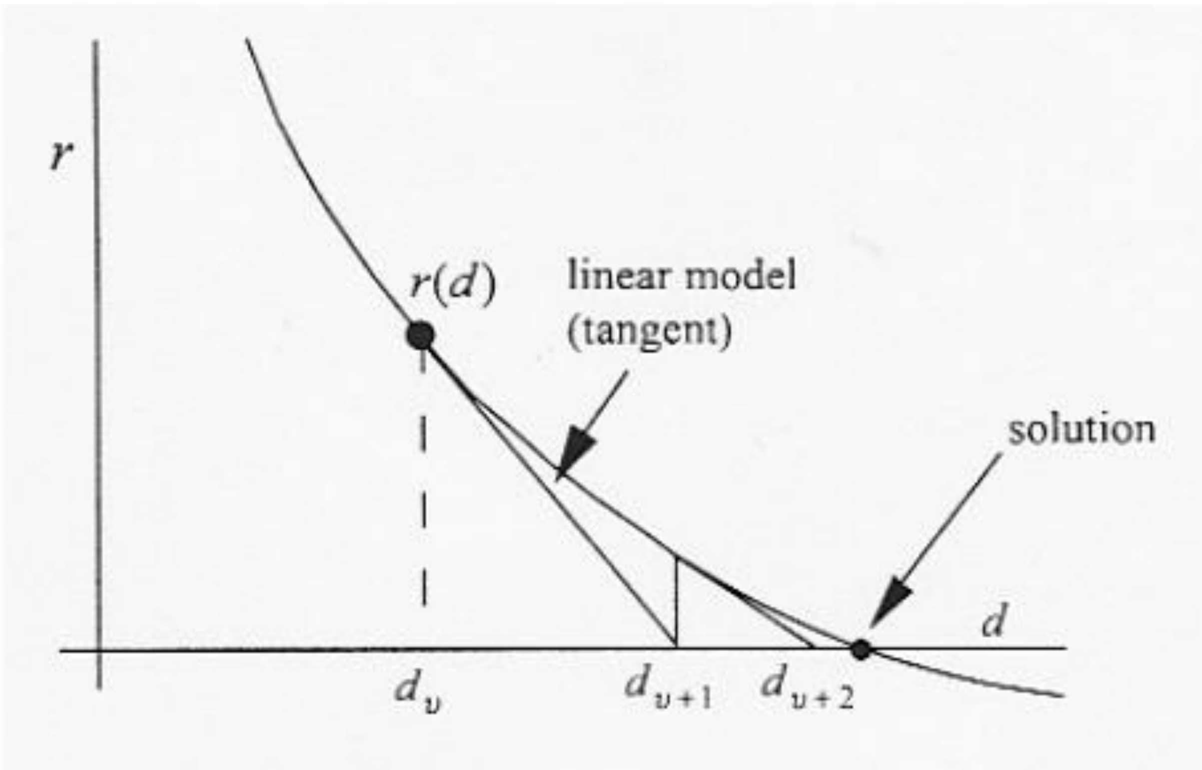
LS-DYNA is a non-linear transient dynamic finite element code with both explicit and implicit solvers and single and double-precision. For implicit work, one is required to use the Double-Precision solver.

19.2.1 What We Are Solving



19.2.2 Review of Mathematical Foundation of Nonlinear Dynamic Implicit Analysis

Review Implicit Analysis in LS-DYNA Theory Manual.



NR Convergence illustration is from LSTC Implicit Class Notes by Morten Jensen 2012 (courtesy of LST, an ANSYS Company)

19.3 Linear Elastic Implicit Analysis (LS-DYNA Double-Precision Solver)

LS-DYNA assumes that the default is an explicit transient, nonlinear analysis. This means that the element formulations and control settings aim for numerical efficiency and to minimize storage requirements since most analyses require thousands if not hundreds of thousands of solves. Whereas, for a linear, elastic implicit analysis we are typically performing one solve for a given constraint and load case and the standard LS-DYNA defaults are not applicable. This section provides all the Keyword settings necessary to perform a linear, elastic analysis with results that are equivalent if not identical to that of other commercially available implicit codes, e.g., Nastran, Abaqus or Ansys.

Please note that LS-DYNA requires a curve to apply most loads (e.g., forces or pressures). Thus, for a linear analysis, we need to give the load a curve that scales the load of interest. For convenience, most people just use a curve with two points: 0,0 and 1,1; with your *CONTROL_TERMINATION, *endtim*=1.

19.3.1 Keywords Used in this Section for Isoparametric Shell and Solid Elements

The practice going forward is only to mention Keywords that are unique to the implicit analysis procedure and it is assumed that the reader is familiar with the standard Keywords to run an explicit analysis.

Keyword Setup for Linear Analysis

Keyword	Card Variable	Description
*CONTROL_ACCURACY	*iacc*=1	The *iacc*=1 is specific for implicit to account for the solution potentially taking large steps through the solution. This card setup is a recommended standard for all implicit analysis work.
*CONTROL_IMPLICIT_GENERAL	*imflag*=1 & *dto*=1.0[1]	This is where we start by telling LS-DYNA that an implicit analysis is being requested. The *dto*[1] is arbitrary. It just ties to your load curve. That is, your load curve must be two points: 0,0 and then 1, 1 (where your load magnitude is scaled by 1.0) – but this is arbitrary.
*CONTROL_IMPLICIT_SOLUTION	*nsolvr*=1	Linear (*nsolvr*=1) The default (no *Keyword) assumes a nonlinear solution. Although not necessary, one might be surprised by geometric nonlinearity upon executing a "linear analysis" by setting *nsolvr*=1.
*CONTROL_OUTPUT	*shlsig*=1 (shells) & *solsig*=1 (solids)	These variables enable stress extrapolation from integration points in fully-integrated elements. Please note that stress extrapolation shlsig and solsig have the same conventions to handle linear and nonlinear materials. It pays to read the manual prior to using these options.
*DATABASE_EXTENT_BINARY	*maxint*=**-2** (shells) & *nintsld*=8 (solids)	Maxint is for shell elements (correctly set to negative 2 (-2) for linear, elastic analysis only (RTM) and *nintsld* is for fully-integrated solid elements.
*SECTION_SHELL	Shells: *elform*=21	Fully integrated linear assumed strain C0 shell (5 DOF) (Wilson's plate element combined with a membrane element). This is equivalent to Nastran's Quad4 element.
*SECTION_SOLID	Solids: *elform*=**-**18 (8-node bricks) and *elform*=16 (10-node tets)	For solids, *elform*=18 is an 8 point enhanced strain solid element and *elform*=16 is for 10-node tetrahedrals. These solid elements can also be used for nonlinear work; however the shell *elform*=21 is only for linear analysis.
*CONTROL_TERMINATION	*endtim*={end of curve}	For linear, elastic, we just need one result set, at the end-of-the-curve. Thus, if are using a curve with an end point of time 1.0 (abscissa), then this is where the analysis will end.

19.4 Shell Element Technology for Linear Elastic Implicit Analysis

19.4.1 In-Plane and out-of-Plane (*nips*) Shell Element Integration

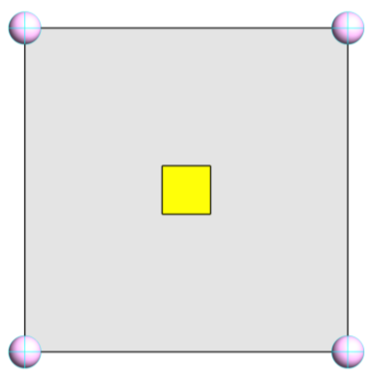
This is the first numerical hurdle to get over is that the default for LS-DYNA explicit is one integration point per shell and likewise one integration point for eight-node solid or four-node tetrahedral. But the reality is that fully integrated elements (e.g., shell elements with four integration points) are becoming common for explicit and are the default for implicit. But remember it is the in-plane layer and then one defines the out-of-plane layers using the *SECTION_SHELL, *nip*s variable where the recommendation is *nips*=5.

19.4.1.1 Why Is This Important to a Simulation Engineer?

In the linear, elastic stress analysis world, simulation engineers are typically focused on the stress state of their structure since many engineering codes require that the stress state be below a certain percentage of the yield stress of the material. For example, the ASME Section VIII, Division 2 code requires pressure vessels to have operating stresses that are nominal 2/3rds of yield. Simply put, understanding how stresses are calculated in finite elements gives greater confidence in the accuracy of your finite element stress results.

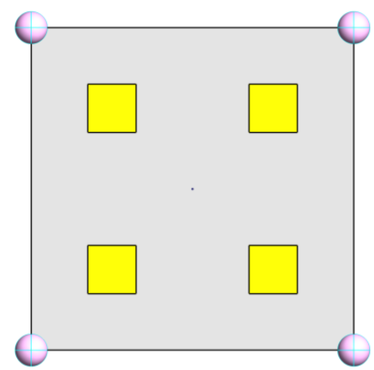
As one might remember from your FEA theory, displacements are calculated at the nodes and then strains are calculated at the integration points using the element's shape function. For elastic materials, stress is calculated by the elastic modulus times the strain (i.e., Hook's Law). The default element whether shell or solid is an element that has one integration point. Thus, stress results are only calculated at the center of the element. With a fully-integrated element, one has four integration points and each element has four unique stress points. For linear, elastic work, it is common to extrapolate integration point stresses out to the nodes. This provides greater accuracy but is only valid for linear, elastic materials.

Explicit (Default Element Formulation)



- One Integration point (*elform* = 2)
- One set of stress items (σ_x, σ_y, etc.) x *nips*
- Fast and efficient for large strain
- Requires Hourglass control for stability

Implicit (Non-Default Element Formulation)



- Four Integration Point (*elform* = 21 (linear) / -16 (nonlinear)
- Four sets of stress items x *nips*
- Numerically intense and can be unstable when distorted
- Hourglass control is not necessary but can improve accuracy

19.4.1.2 Workshop: 22A - Linear Elastic Analysis – Shells - Stress Concentrations

Objective: Setup deck to run a linear elastic analysis and work through the concept of using fully-integrated elements.

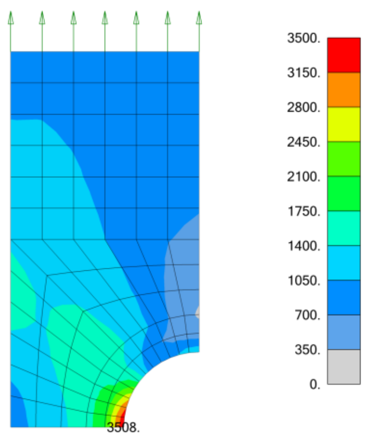
Problem Description: Quarter-symmetry plate under uniform tension loading. A hole in the plate creates a stress concentration of 3x (analytical solution for a plate of infinite width, since our plate is of finite width, the maximum stress value will be higher than 3x).

Theory: To achieve stresses that match a standard implicit code, we need to extrapolate stresses from the integration points. This is done by setting *shlsig* =1 (see *CONTROL_OUTPUT). Then, to write out the stresses to the d3plot file, we set maxnit=-2 in *DATABASE_EXTENT_BINARY (please note that it is negative two (-2).

Tasks:

- Open: / Linear Elastic Analysis – Shells / A – In-Plane / Linear Elastic Analysis with Shells – In-Plane – Start.dyn and setup model to run using the default settings discussed in this section but leave out maxnit and shlsig. Will call this the “default” since maxint and shlsig are “special commands”. Then follow the table and add-in shlsig command to enable extrapolation from the integration points out to the nodes.
- Answers are provided in the Answers file folder.
- Change *elform* from 21 to -16 and then change the *nip*s from 2 to 5: How do the Results Change? Think mechanics – it is linear, elastic…

Nastran Stress Results – Max Stress 3508

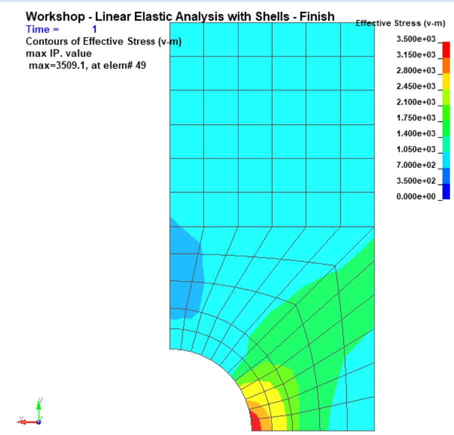


LS-DYNA (Default)

Stress Maximum?

(Please note that LS-DYNA Default refers to dumping out the stresses without extrapolation or with shlsig *blank or default*)

LS-DYNA – Recommended – Max Stress 3509



Model	Max Stress	*elform*	maxint	shlsig	*nip*s
Nastran	3,508	Not Applicable	Not Applicable	Not Applicable	Not Applicable
LS-DYNA - Default		21	*blank*	*blank*	*2*
LS-DYNA - Classic		21	-2	*blank*	*2*
LS-DYNA - Recommended		21	-2	1	*2*
LS-DYNA *elform*=-16 and *nips*=5		-16	-2	1	*5*

19.4.1.3 Workshop: 22B - Linear Elastic Analysis – Shells – Out-of-Plane Integration

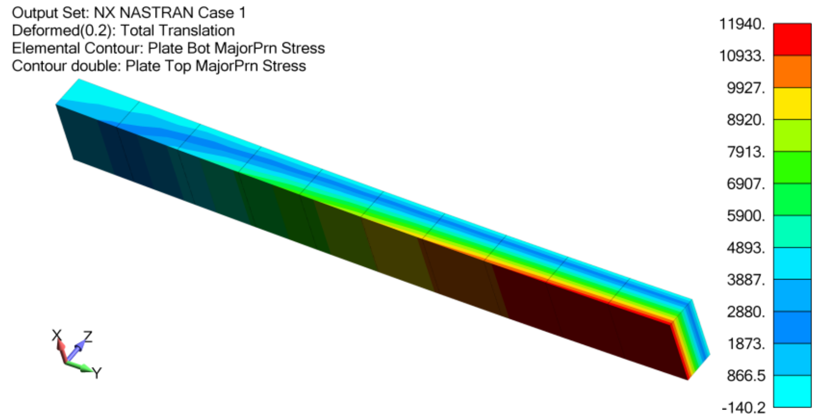
Objective: Linear Elastic Analysis of shells under bending – understanding the uniqueness of linear elastic analysis in LS-DYNA

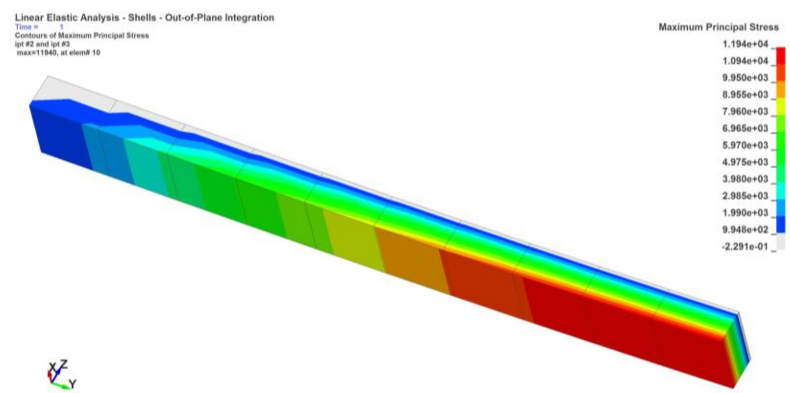
Problem Description: A simply-support-beam (SSB) is analyzed for its linear elastic stress state. This is about as simple as it can get. We have a half-symmetric beam under a uniform pressure load. One can run the model with either LS-DYNA SMP or MPP Double-Precision.

Tasks

- Open / B – Out-of-Plane / Linear Elastic Shell Analysis – Out-of-Plane – Start.dyn in a text editor and fill in the Keywords that would be required to make this analysis correct. One may note that they are similar to those in the prior analysis.
- Follow the table below and fill out the max stress values as you change the *Keyword variables.
- At the end, Answers are provided in the Answer file folder.

Analyst's Note: Keep in mind that linear, elastic stress analysis techniques are designed to be accurate for small strains (e.g., ~< 0.2%).





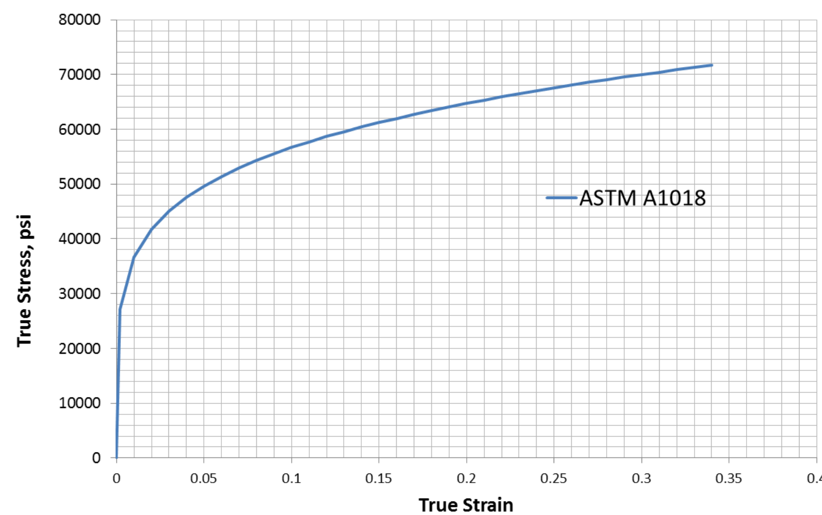
Model	Max Stress	*elform*	maxint	shlsig	*nips*
Nastran	11,940	Not Applicable	Not Applicable	Not Applicable	Not Applicable
LS-DYNA - Default		21	*blank*	*blank*	*2*
LS-DYNA - Classic		21	-2	*blank*	*2*
LS-DYNA -Recommended		21	-2	1	*2*
LS-DYNA *elform*=-16, *maxint*=-3 and *nips*=5		-16	-3	*blank*	*5*

19.5 Solid Element Technology for Linear Elastic Stress Analysis

Solid elements almost seem simple after dealing with bi-directional integration points since they only have volume integration schemes to think about. For brick elements it is either one-point or eight-points while tetrahedrals provide options for four-point and five-point integration. The same general concepts will be covered in this section; however, with greater brevity since the reader is getting acquainted with the nomenclature of LS-DYNA implicit mechanics.

The only basic concept is that the reader should carefully understand the significance of the various element formulations that LS-DYNA offers since they all pretty much work for implicit as for explicit. The reason this is done is that LS-DYNA allows the user to flip back and forth between explicit and implicit and vice-a-versa. Thus, the code must support the various element formulations between both solvers. However, and a big however is that linear elastic element formulations are very sensitive to slight strain variations while highly nonlinear elements are designed to handle large strain.

As illustration, let's look at the stress-strain graph for an 1018 steel. The elastic region has a stress/strain slope of the elastic modulus (30e6) while the plastic region has slope of something like 100,000. Another way to think about it is that a linear elastic element has to be 300x more sensitive to strain than a highly nonlinear element.



For our linear elastic work, we will be using *elform*=-18 for 8-node brick elements and *elform*=16 for 10-node tetrahedral elements.

19.5.1.1 Keywords Used in this Section for Solid Elements

The practice going forward is only to mention Keywords that are unique to the implicit analysis procedure and it is assumed that the reader is familiar with the standard Keywords to run an explicit analysis. For example, the Keyword *CONTROL_TERMINATION is not discussed although it is required to run the analysis.

Keyword	Card Variable	Description
*CONTROL_ACCURACY	*iacc*=1	The iacc=1 is new and is something specific for implicit to account for the solution potentially taking large steps through the solution. This card setup is a recommended standard for all implicit analysis work. While OSU=1 is standard for rotating equipment, it is not necessary for standard implicit work unless one has large rotations.
*CONTROL_IMPLICIT_GENERAL	*imflag*=1 & *dto*=1.0	This is where we start by telling LS-DYNA that an implicit analysis is being requested.
*CONTROL_IMPLICIT_SOLUTION	*nsolv*=1	The ***linear elastic hex element*** requires this trigger.
New Keyword Commands Just for Solid Elements		
*CONTROL_OUTPUT	*solsig*=1	For stress extrapolation of fully-integrated solid elements.
*DATABASE_EXTENT_BINARY	*nintsld*=8	Writes out all integration points for fully-integrated solid elements.
*SECTION_SOLID	*elform*=-18 (Hexs)	-18: 8 point enhanced strain solid element for ***linear*** and ***nonlinear*** statics.
	elform=16 (Tets)	16: 10-node tetrahedral.

19.5.1.2 Workshop: 23 – Linear Elastic Analysis – Solids - Hex & Tets

Objective: Verify that Bricks and Tets can provide high-quality linear stress results.

Problem Description: A quarter-symmetry plate of unit thickness is given a uniform pressure load of 1,000 psi. Nastran results are presented for the baseline comparison. (*Note: Stress interpolation is done by *CONTROL_OUTPUT, solsig=1 so make sure that LSPP extrapolate is turned off.*). A movie file is not provided since this Workshop follows prior practice of filling out the – Start keyword deck and if the student runs into trouble, there is a – Finish deck available.

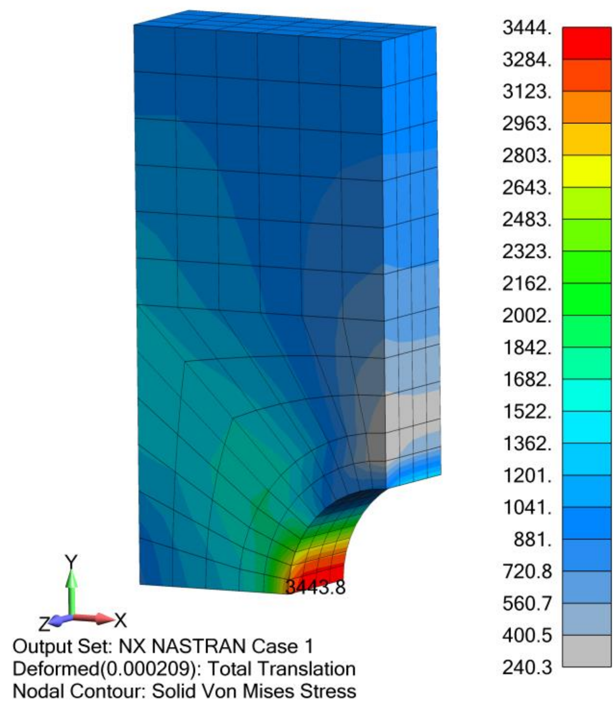
Tasks:

Hexahedral Analysis

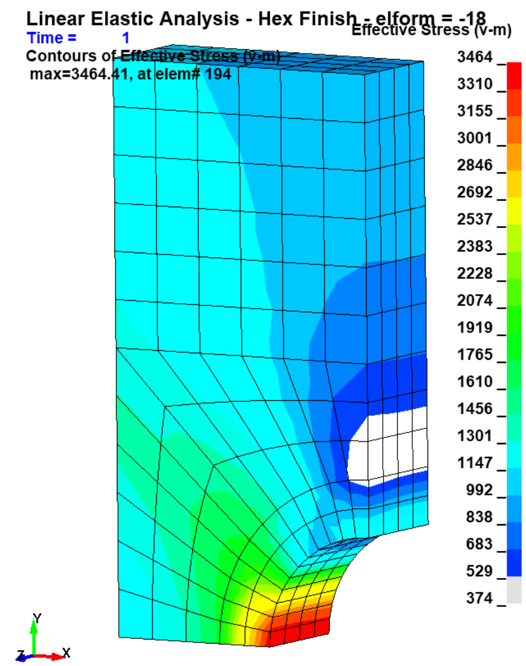
- Load Keyword file: / Linear Elastic Analysis – Solids / Hex / Linear Elastic Analysis - Hex - Start.dyn into a text editor and start filling in Keywords.
- Run and post-process.
- Change *elform*=-2 and note any differences.

Elform	Max Stress
Nastran	3,440
-18	
-2	

Nastran Max VM=3,440



LS-DYNA Max VM=3,464

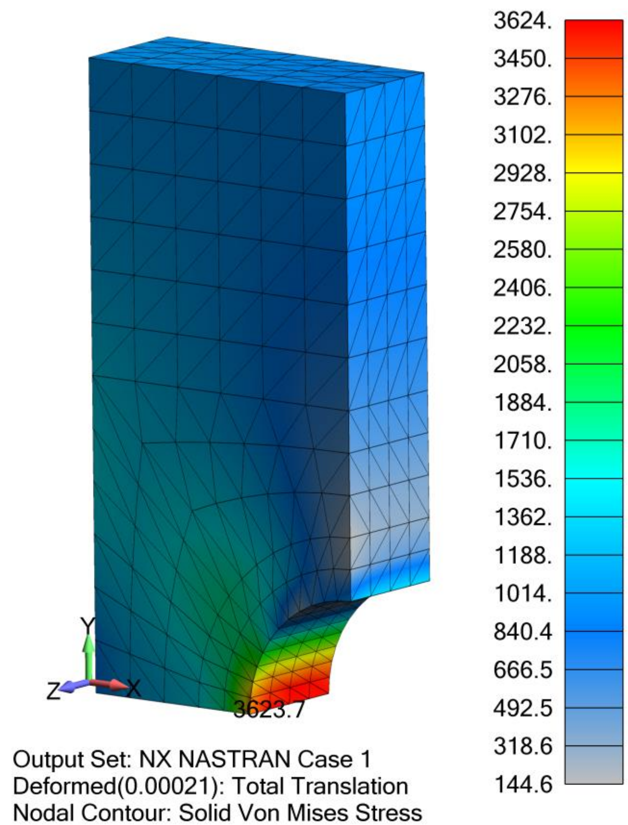


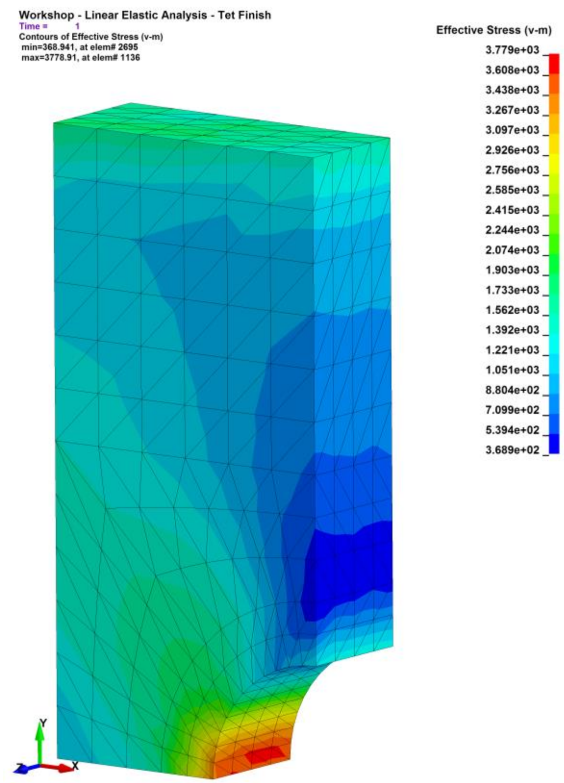
Tasks:

Tetrahedral Analysis

- Not much to discuss. The model is only presented for completeness.

Nastran Max VM=3,620	LS-DYNA Max VM=3,780



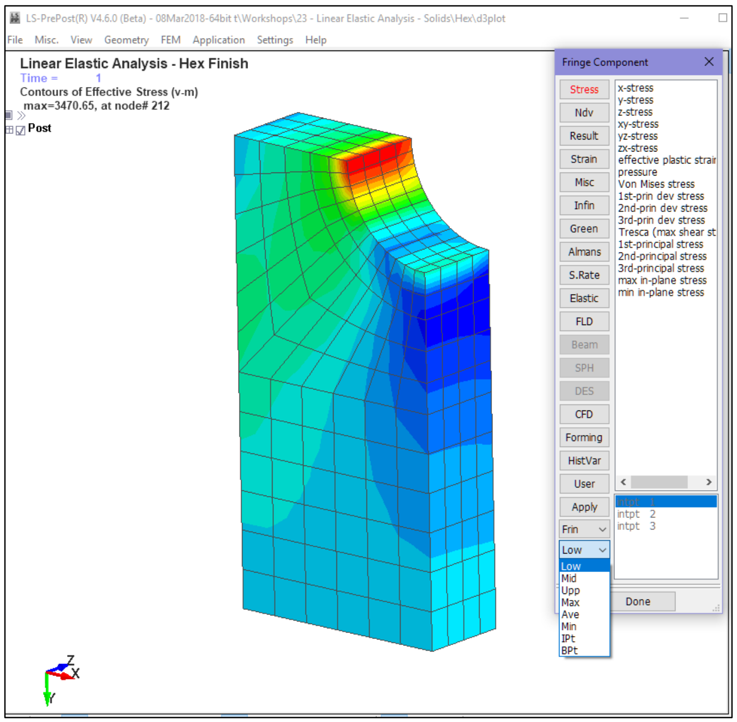


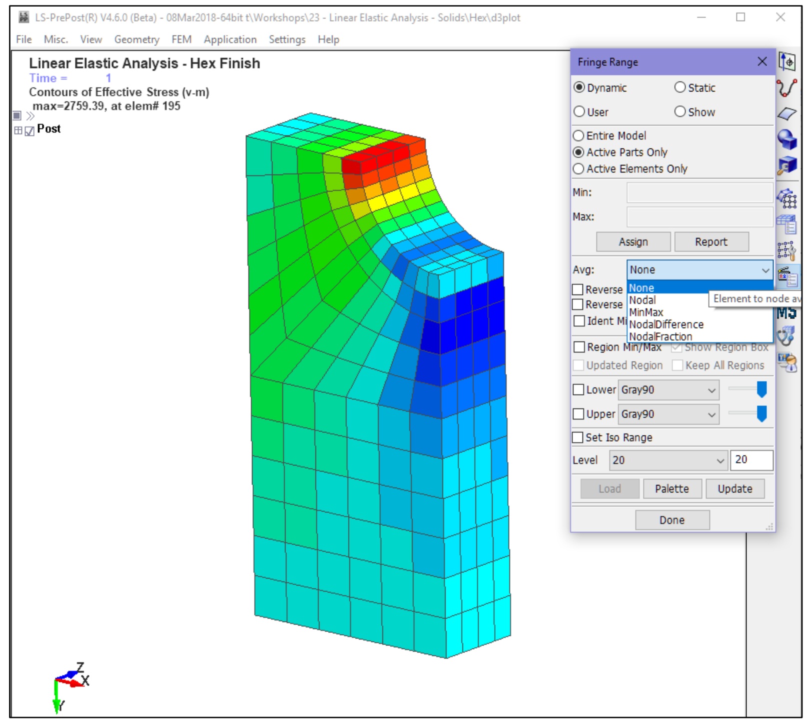
*Analyst's Note: As discussed, LS-DYNA was developed to be a super-fast explicit code and 10-node tets are really something only used extensively in the implicit linear world. So, one has to turn on a few extra Keyword variables to get the standard implicit linear output data. In default, LS-DYNA does not write out the mid-side nodes so one uses the *CONTROL_OUTPUT variable tet10s8.*

19.5.1.2.1 Solid Elements: Stress Post-Processing in LSPP

How Does LSPP Display Stresses in LSPP for Solid Elements?

Inside the Element	Outside the Element
In Fringe Component, the Low, Mid and Upp all do the same thing for solid elements, they average the stress of integration points adjacent to connected nodes. This is the classic, default stress averaging method. Then we have Max, Ave and Min which take the Max, Ave or Min of the eight integration points and uses that as the “single-value” stress item for that element.	Under Fringe Range is where one determines how to average (Avg:) the stresses between connected elements. The options allow no averaging (none) and then Nodal and MinMax (averaging the Min and then the Max value at the connected nodes). The last two options can be used to assess the mesh convergence (NodalDifference and NodalFraction).





Analyst’s Note: See file folder Understanding Stress Results for an example model that will write out the non-extrapolated and extrapolated integration point stresses. If one post-processes the results from this model, it will be obvious how (low/mid/upper) = average stresses and what Max, Min and Ave are doing.

19.6 Beam Element Technology for Linear Elastic Stress Analysis

LS-DYNA beam element formulations are quite diverse and provide broad modeling flexibility. When the term “resultant” is used for the *elform* description, it signifies that only displacements and forces are calculated for the element. For implicit and explicit work, *elform*=1 is the default. Although *elform*=13 matches exactly with standard Nastran beam element formulation, it is a resultant formulation and stresses are not calculated and it is only applicable to linear behavior.

For all the standard LS-DYNA beam element formulations where stress is calculated, one needs to think about the type of quadrature rule (QR) is desired to recover stresses. It is identical to that for shell and solid elements but with a twist. For example, LS-DYNA calculates stress at the mid-point of the beam element while ANSYS / Nastran codes perform the calculation at both ends of the beam. This is because LS-DYNA is using an integration scheme to calculate the beam’s stiffness and recover its stresses. For example, think shells but much more complicated (see below). Nevertheless, the same identical results can be obtained for linear elastic analysis as that generated by hand-calculations or standard implicit codes. And, if nonlinear, elastic-plastic, large strain behavior is required, LS-DYNA provides quite a bit of flexibility to obtain high-accuracy results.

19.6.1 Beam Integration (QR) Setting: Rectangular

Beam integration is defined by the QR setting and can be one point or 2x2 Gauss, 3x3 Gauss, 3x3 Lobatto or 4x4 Gauss as shown in the figure. Additionally, if necessary, one can also define specialized integration rules via *INTEGRATION_BEAM.

For beam elements, the integration is performed at one-point along the axis (mid-point of the beam) and then multiple points in-plane per QR=2, 3, 4 or 5.

The integration rule can be considered similar to that for shell elements based on the *nip* setting to capture through thickness plasticity where for accuracy reasons *nip*=5 is preferred.

For linear analysis, QR=4 (Lobatto) is recommended and for plasticity analysis, QR=5 is a better choice.

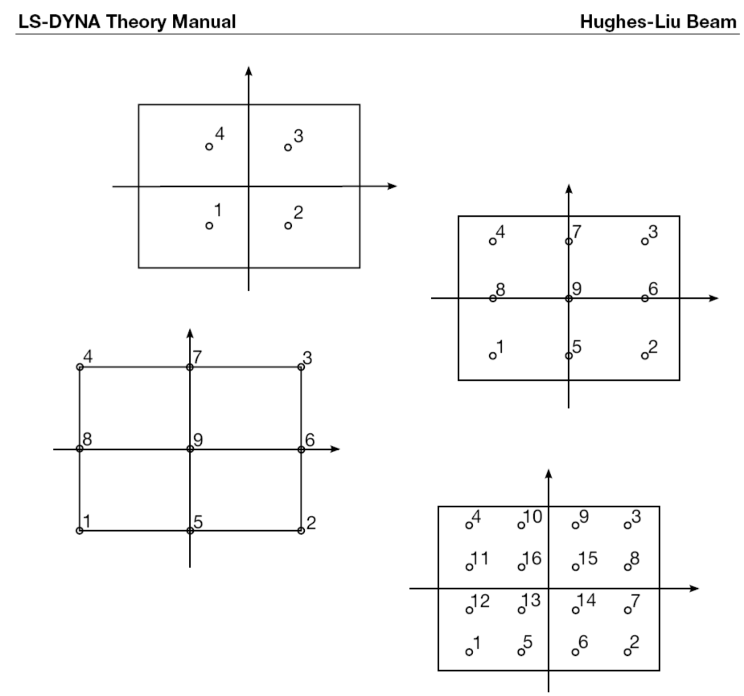


Figure 6.3. Integration possibilities for rectangular cross sections in the Hughes-Liu beam element.

19.6.2 Contact with Beams

For explicit, it is _GENERAL while for implicit, it is _MORTAR. For both contacts, beams are treated as cylinders regardless of their shape. This can cause problems if *INTEGRATION_BEAM is used since the contact algorithm can’t determine a contact cylinder. A workaround is to overlay the beams with null beams with the desired diameter and make these beams the active contact set.

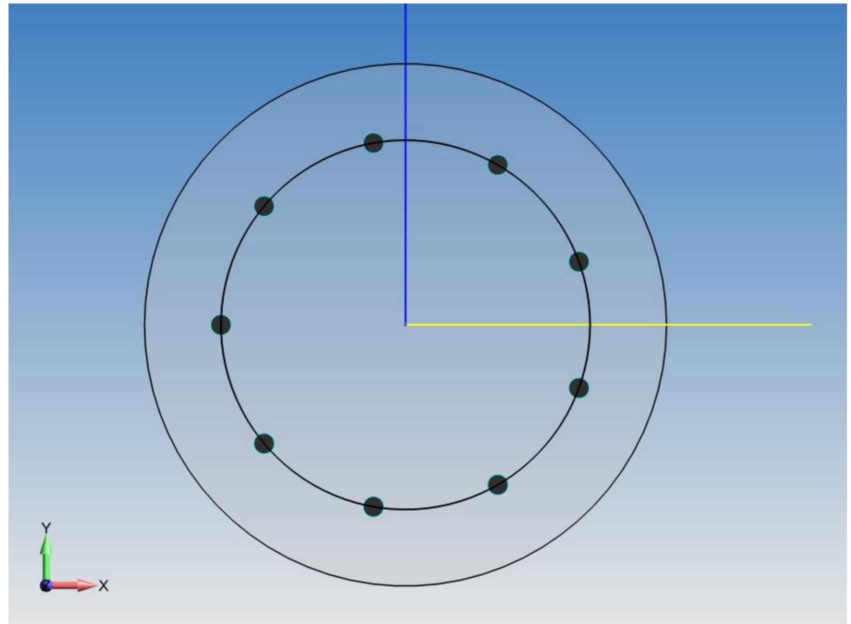
19.6.3 Beam Integration (Cylindrical Solid and Tube)

For circular solid (cylinders) or tubes, the default beam integration formulation is based on a circumference of integration points at a radius defined by:

$$Integration_{Radius} = \sqrt{(R_{Inner}^2 + R_{Outer}^2)/2}$$

Unfortunately, there is not a Lobatto type integration scheme for circular cross-sections. Please note that the standard integration schemes only allow 4 (QR=2) or more points around one fixed circumference (e.g., the figure on the right shows nine integration points (QR=3 or 4).

This scheme has advantages and disadvantages. For explicit analysis with large strain plasticity, it works fairly well but for linear stress analysis, where stresses on the surface of the beam are of more importance, it presents some limitations.



***INTEGRATION_BEAM**

Although it may appear computationally costly to request 100 beam integration points (K=2), most FE models don't have hundreds of thousands of beam elements. The only input requirements for a solid cylinder (ICST=8) is to specify D_1 and then K.

To output this data into the binary file, the *DATABASE_EXTENT_BINARY, *beamip*=1 (just greater than zero – RTM) must be set. An example is presented on the next page.

Overall, the simulation engineer should know that the results are exact to theory with respect to the stress results being extracted at the mid-point of the beam (midway between the nodes) and at the specified integration point.

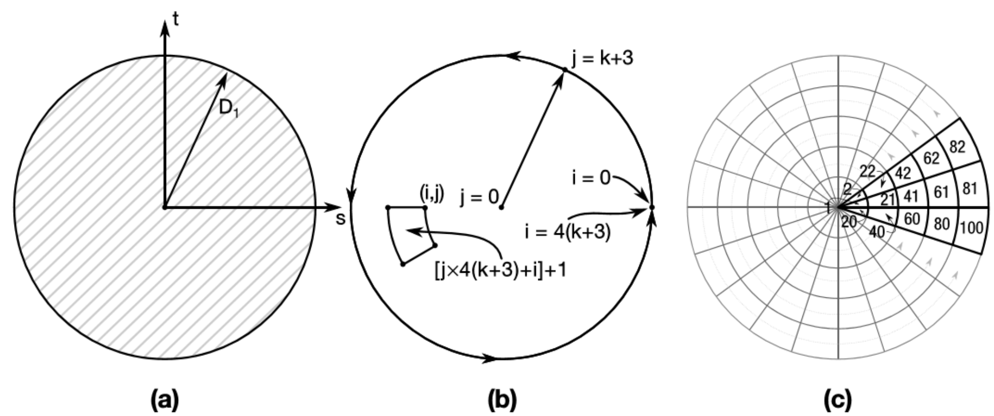


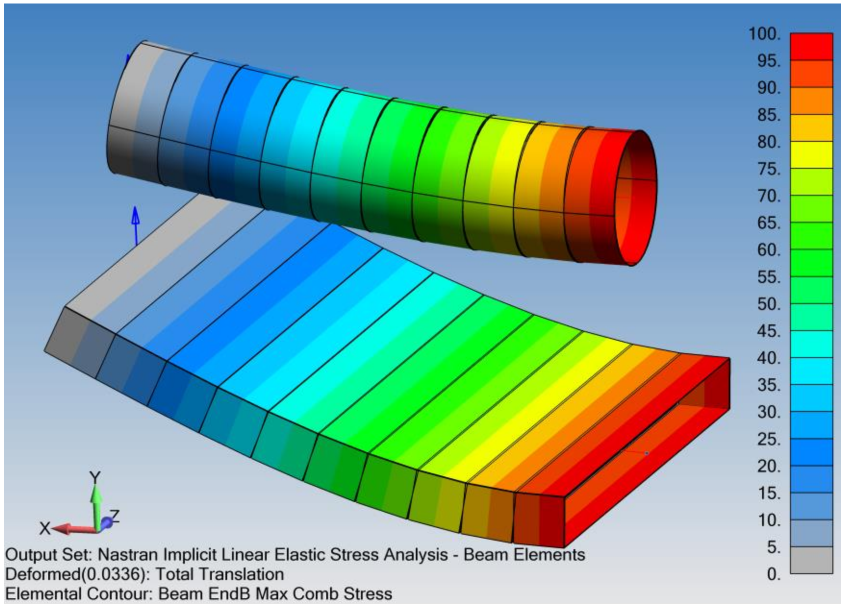
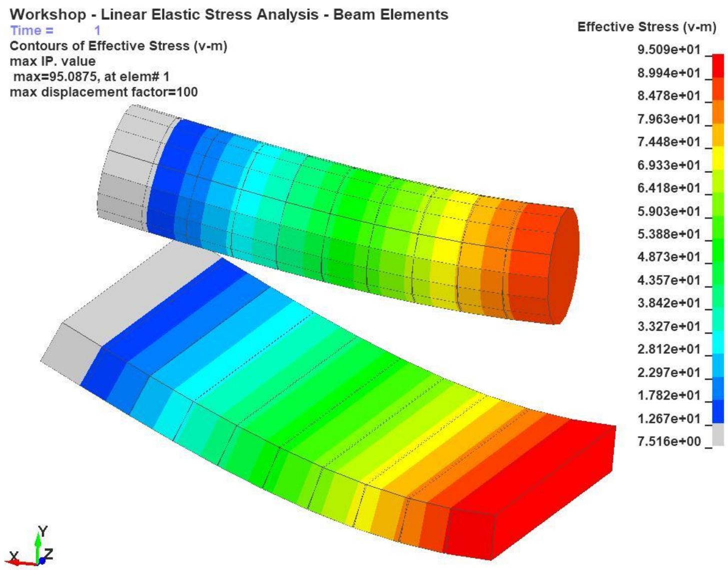
Figure 24-10. Type 8: Circular. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

19.6.4 Workshop: 24 - Linear Elastic Analysis – Beam Analysis

Objective: Setup a beam model to generate linear elastic stress analysis results that match a Nastran run.

Problem Description: Cantilevered beam with a load of 16.67 units at its end. The student needs to setup the file to run and to be able to contour the beam results in LSPP.

Tasks: Using the – Start file, the student will need to set the standard linear analysis commands (e.g., *nsolvr*=?) and then some new options. The first is to setup the circular section to use *INTEGRATION_BEAM. Review the manual and then it should make sense to set icst=8, k=2 and d1=1.28505. The *irid*=10 and thus under *SECTION_BEAM, *qr/irid*=-10. Then, we will need to request beam stress information to be dumped into the D3Plot files. This is done within the *DATABASE_EXTENT_BINARY, *beamip* > 0. For post-processing, if all is setup correctly one can contour beam results under FriComp / Beam / Von Mises stress. To get the displacement scaled up, see Settings (up on the main bar) / Post Settings / Displacement Scale Factor. The image shown below is using a Displacement Scale Factor = 100.

Nastran Results	LS-DYNA Results
	

Analysis Notes: Nastran beam stress results are provided at the end of the beams. Each beam has a length of 1 unit. Total length of the cantilevered beam is 10.

Analysis Notes: LS-DYNA calculates beam stress results at the middle of the beam. To compare with Nastran, multiply by (10/9.5)
Importantly, the stress results for the rectangle beam analysis are identical once adjusted for stress recovery location. The rod still lacks – why?

Extra Credit: Increase the *INTEGRATION_BEAM variable k from 2 to 3 (remember to change the beamip to > 0) and reanalyze.

19.7 Checklist for Implicit Static, Linear Elastic Analysis in LS-DYNA

- ☑ *CONTROL_ACCURACY with *iacc*=1
- ☑ *CONTROL_IMPLICIT_GENERAL with *imflag*=1
- ☑ Check element formulation: (i) Shells *elform*=21 (*nip*>2); (ii) Bricks *elform*=-18; (iii) 10-node Tets *elform*=16 and Beams *elform*=4 with *qr*=3 (Lobatto). If your beams are circular, then one might want to use *INTEGRATION_BEAM.
- ☑ To ensure linear elastic solution (no geometric nonlinearity), set *nsolvr*=1 (*CONTROL_IMPLICIT_SOLUTION)
- ☑ Use *MAT_1 since it is a linear elastic analysis and sets you up for *shlsig* (*CONTROL_OUTPUT)
- ☑ For consistency set load curve to end at Time=1.0 and Termination=1.0
- ☑ Be aware of your integration scheme (Gaussian or Lobatto) and your setting on *DATABASE_EXTENT_BINARY to recover the correct number of integration point data for shells (i.e., *maxint*=-2 based on *shlsig*=1), solids (i.e., *nintsld*=8) and beams (i.e., *beamip*>0).
- ☑ Think about extrapolation in *CONTROL_OUTPUT, whether for shells or for solids via *shlsig*=1 and *solsig*=1. Please keep in mind that if the element goes out of the elastic range (plasticity), then *shlsig* and *solsig* must be set to 2 (i.e., RTM). Also, if one is using 10-node tetrahedrals, then set the variable *tet10s8*=1 to write out the mid-side nodes to capture displacements.
- ☑ Within LSPP, be careful with how you pick stresses. In general, your selection choices will depend upon your element type.

19.8 Geometric and Material Nonlinearity

In this section we show how LS-DYNA implicit handles geometry and material nonlinearity. For the explicit solver, the solution of severe nonlinearities is typically not an issue unless we are talking about massive element distortions or compaction of foams. We will treat this section in the classic sense and cover geometric nonlinearity as buckling and material nonlinearity as plasticity.

A nice engineering review of nonlinearity and how LS-DYNA can solve these problems using the implicit technique is found in the Class Reference Notes / Implicit Analysis / DYNAmore Implicit Users Guide / LS-DYNA Implicit Users Guide.pdf. If you have not read this article, one should review it while the Workshop exercise on buckling is solving (since it'll take a few minutes).

To summarize what it means to do nonlinear analysis, the DYNAmore article provides a succinct list:

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

19.8.1 Material Nonlinearity in Shells: Out-of-Plane Gaussian Integration is Important

In the prior linear elastic discussion, material behavior was elastic. This allowed the clever use of shlsig and solsig to extrapolate stresses from a minimal set of integration points. With material plasticity, multiple layers of through-thickness integration planes are required to accurate capture the plastic deformation response of the shell element. This was discussed in the Explicit Section and holds equally true for implicit analysis.

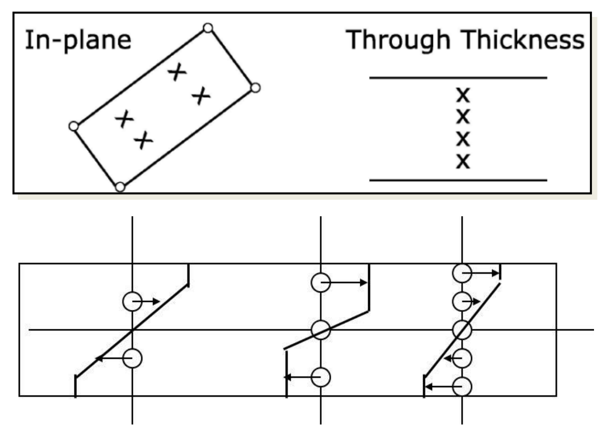
Why Multiple In-Plane Integration Points are Necessary for Nonlinear (Plasticity) Material Behavior

nip 1: Membrane Behavior

nip 2: Linearly Elastic Behavior (default)

nip > 3: Starting point for Nonlinear Materials

Optimum *nip* for Nonlinear Plasticity = 5



19.8.2 For Limited Material Plasticity (<20%) Go Lobatto (see *CONTROL_SHELL, *INTGRD*=1) – Finessing Implicit Stress Results

Analyst's Note: Only bother with this section if you are interested in a fatigue analysis, that is to say, you really want to finesse the calculation of stresses and plastic strains. From a general simulation viewpoint, your simulation stress and plastic strain variability will most likely be far greater due to loads calculation or material variability than the few percent difference that can be obtained by fooling around with Gauss vs Lobatto integration.

LS-DYNA offers three ways to place integration points: Gaussian, Lobatto (see *CONTROL_SHELL, *intgrd*) and a custom scheme of your own devising (see *INTEGRATION_SHELL). When high levels of plasticity can be expected, say greater than 20%, Gaussian Quadrature provides maximum efficiency in capturing the plastic deformation of the shell element. The limitation of this integration scheme is that stresses are never calculated directly on the surface, which means that your shell stresses driven by a bending load will always be slightly under reality. If a Lobatto Quadrature is used, then one can be assured that the stress calculation will exactly capture the stresses on the surface of the shell element; albeit with less accuracy in its calculation of plastic strain through-thickness in the element.

19.8.2.1 Classic Tradeoff: To Gauss or To Lobatto – that is the Question?

This question can only be answered by the simulation engineer doing the work. If high-quality stresses are the objective of the analysis and the majority of the structure stays in the linear range or plasticity is limited, then Lobatto Quadrature with 5 points (*nips*=5) can be an effective choice. There is no one "rule-of-thumb" since Gaussian and Lobatto each have advantages and disadvantages.

Gaussian Through-Thickness Integration Scheme	Lobatto Through-Thickness Integration Scheme

Gaussian Quadrature Points

Point	1 Point	2 Points	3 Points	4 Points	5 Points
#1	.0	-.5773503	.0	-.8611363	.0
#2		+.5773503	-.7745967	-.3399810	-.9061798
#3			+.7745967	+.3399810	-.5384693
#4				+.8622363	+.5384693
#5					+.9061798
Point	**6 Points**	**7 Points**	**8 Points**	**9 Points**	**10 Points**
#1	-.9324695	-.9491080	-.9702896	-.9681602	-.9739066
#2	-.6612094	-.7415312	-.7966665	-.8360311	-.8650634
#3	-.2386192	-.4058452	-.5255324	-.6133714	-.6794096
#4	+.2386192	.0	-.1834346	-.3242534	-.4333954
#5	+.6612094	+.4058452	+.1834346	.0	-.1488743
#6	+.9324695	+.7415312	+.5255324	+.3242534	+.1488743
#7		+.9491080	+.7966665	+.6133714	+.4333954
#8			+.9702896	+.8360311	+.6794096
#9				+.9681602	+.8650634
#10					+.9739066

Location of through thickness Gauss integration points. The coordinate is referenced to the shell midsurface at location 0. The inner surface of the shell is at -1 and the outer surface is at +1.

Lobatto Quadrature Points

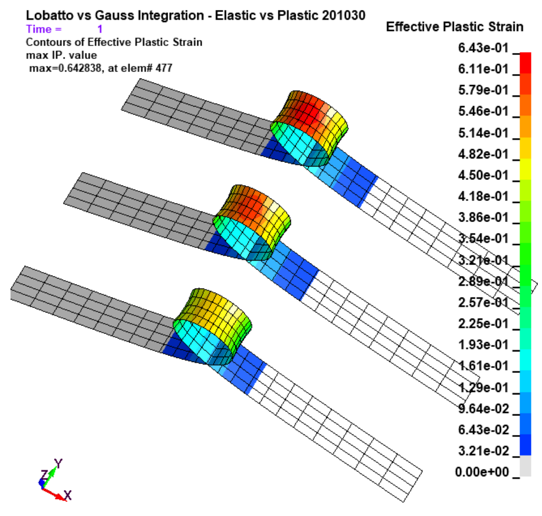
Point	1 Point	2 Points	3 Points	4 Points	5 Points
#1			0.0	-1.0	0.0
#2			-1.0	-0.4472136	-1.0
#3			+1.0	+0.4472136	-0.6546537
#4				+1.0	+0.6546537
#5					+1.0
Point	**6 Points**	**7 Points**	**8 Points**	**9 Points**	**10 Points**
#1	-1.0	-1.0	-1.0	-1.0	-1.0
#2	-0.7650553	-0.8302239	-0.8717401	-0.8997580	-0.9195339
#3	-0.2852315	-0.4688488	-0.5917002	-0.6771863	-0.7387739
#4	+0.2852315	0.0	-0.2092992	-0.3631175	-0.4779249
#5	+0.7650553	+0.4688488	+0.2092992	0.0	-0.1652790
#6	+1.0	+0.8302239	+0.5917002	+0.3631175	+0.1652790
#7		+1.0	+0.8717401	+0.6771863	+0.4779249
#8			+1.0	+0.8997580	+0.7387739
#9				+1.0	+0.9195339
#10					+1.0

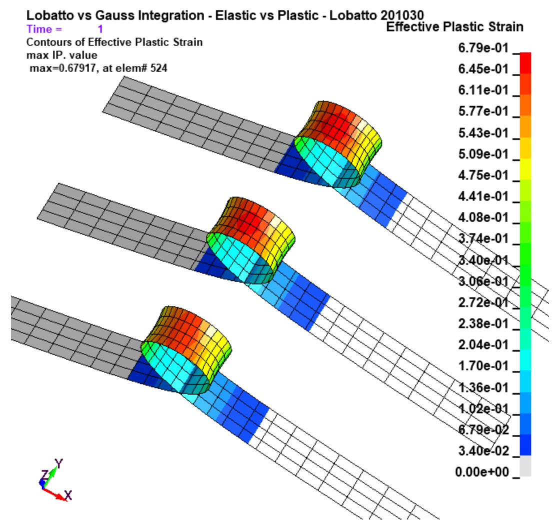
Location of through thickness Lobatto integration points. The coordinate is referenced to the shell midsurface at location 0. The inner surface of the shell is at -1 and the outer surface is at +1.

19.8.2.1.1 An Example of Gauss vs Lobatto in Action

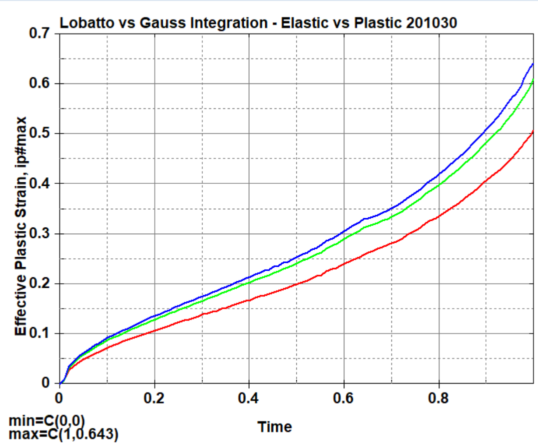
A simple example is shown of a shell undergoing bending using the two forms of through-thickness integration. Lobatto reports 68% versus 64% max plastic strain. There is no free lunch in LS-DYNA and although the plastic strain is higher in *THIS* example, it still only reflects one measure of the analysis result – plastic strain. Overall, the default is Gauss since it provides the best "capture" of plastic strain thru the thickness of the shell element.

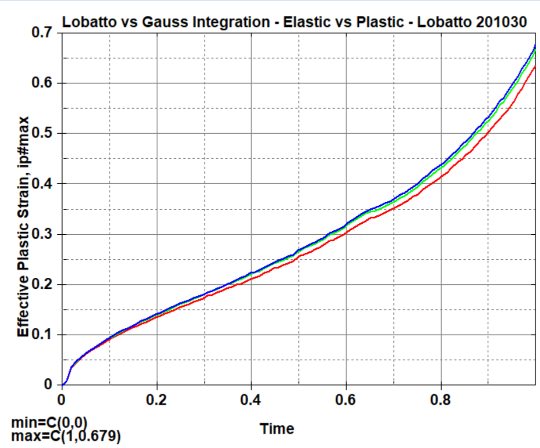
Gauss 3, 5 & 7 *nips*	Lobatto 3, 5 & 7 *nips*
Max Plastic Strain: 64%	Max Plastic Strain 68%





Plastic Strain	Plastic Strain





19.8.3 New Keyword Commands Used in this Section for Static Nonlinear Implicit Analysis

Keywords for Nonlinear Implicit Analysis		
Keyword	**Card Variable**	**Description**
*CONTROL_ACCURACY	*iacc*=1	The *iacc*=1 is new and is something specific for implicit to account for the solution potentially taking large steps through the solution. This card setup is a recommended standard for all implicit analysis work. While *osu*=1 is standard for rotating equipment, it is not necessary for standard implicit work unless one has large rotations.
*CONTROL_IMPLICIT_AUTO	*iauto*=1, *iteopt*=? & *dtmax*=?	When in doubt, leave everything in default. However, if you know how the solution might behave, one should control it. By setting DTMAX to a specific value we force the implicit solution to only advance at steps no greater than this value. Another setting that has helped to improve the solution speed is setting iteopt=200. This forces the solution to iterate thru 200 steps prior to cutting down the time step (the purpose of _IMPLICIT_AUTO (RTM)).
*CONTROL_IMPLICIT_DYNAMICS	*imass*=1, *gamma*=0.60 & *beta*=0.38	Adds dynamics with numerical damping to stabilize the implicit solution. If this technique is used, the kinetic energy (KE) should be checked at the end of the solution to verify that it is very low as compared to the internal energy. Damping only removes energy from the solution when there is motion or kinetic energy. If the KE is near zero than the dissipative energy (fictional energy removed by damping) will also be zero. *Note: make sure your material cards have mass!* Also please note that for a true dynamic analysis, reset Gamma and Beta to their defaults. These settings (0.6 / 0.38) are to simulate a damped response and their only purpose is to provide convergence stability to a nonlinear solution.
*CONTROL_IMPLICIT_SOLUTION	*abstol*=1e-20, *nlprint*=2, *dnorm*=1 & *nlnorm*=4	*nlprint* is for extra information. As the implicit solver matures and improves, yesterday's settings are not always the best. A standard override is to set *abstol*=1e-20 where the default is 1e-10. As of this writing (Oct. 2021), *dnorm*=1 and *nlnorm*=4 are recommended.

19.8.4 Workshop: 25 – Implicit Nonlinear Material Analysis

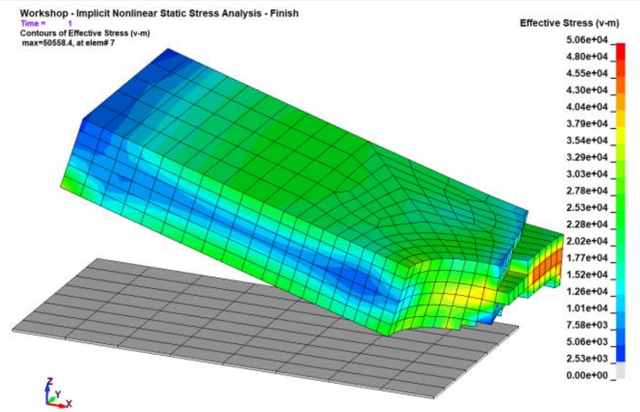
Introduction: A simply supported beam (quarter-symmetry) is has a ramped pressure load that ends at time=1.0. The material of the beam is 6061-T6 using a material law from Varmit Al's database (Aluminum material 175). The aluminum material fails at a tensile plastic strain of 15%. However, we know from materials science that the failure strain under compression would be closer to 3x or to keep it simple 45%.

Objective: Review the Keyword deck (Workshop - Implicit Nonlinear Static Stress Analysis - Start.dyn) and fill out the necessary information to enable an implicit analysis. *One might want to review the prior page for what should be added.*

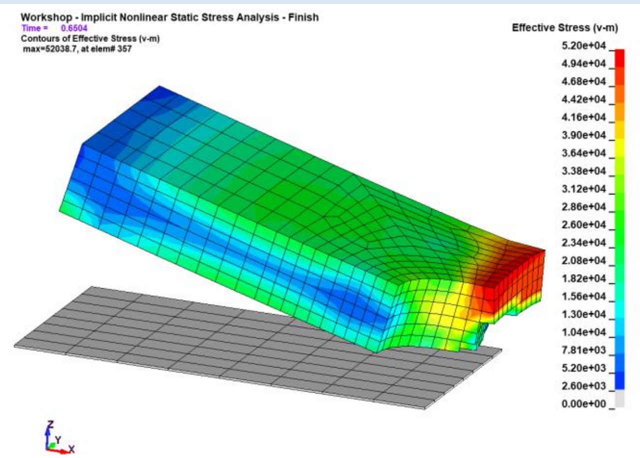
Tasks:

- Open in a text editor "Workshop - Implicit Nonlinear Static Stress Analysis - Start.dyn" and inspect the keywords.
- To capture enough timesteps, within *CONTROL_IMPLICIT_AUTO, set *dtma* =0.01
- Since we have plasticity, *CONTROL_OUTPUT, *solsig*=2
- For implicit, we like to have a fully integrated element (*SECTION_SOLID, *elform* =-2 or -18) and also dump out all 8 integration points into the d3plot database (*DATABASE_EXTENT_BINARY, *nintsld*=8).
- To control element erosion, we are using *MAT_ADD_EROSION. To set these values, one might have to read the manual (RTM) or look at the workshop on material failure.
- Also, one will notice we have a rigidwall in the model.
- Lastly, there is a finish deck that runs.

Standard – Failure Strain 15%



With *MAT_ADD_EROSION

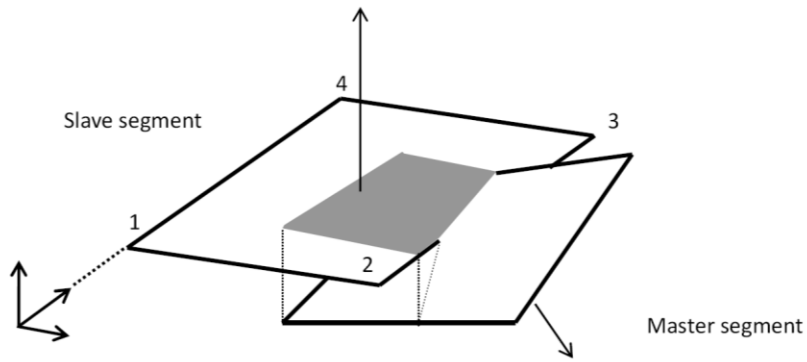


19.9 Contact

19.9.1 General Comment and Focus on Mortar Contact

LS-DYNA was developed specifically to solve contact problems (see Class Reference Notes / History of LS-DYNA). We are not going to talk at length about contact theory since sources are available via the LSTC Theory Manual and many fine application notes (see Reading Assignments / Mortar Contact for Implicit Analysis).

Contact can be effortlessly implemented, or it can be bewitching in complexity. A reasonable treatment of contact is a multi-day course in itself. For this reduced treatment for LS-DYNA implicit, we will focus only on Mortar contact and just a bit on _TIED applications.



Efficient Contact Modeling

Whenever possible, interferences between parts should be avoided. It is standard contact practice that any initial interference is removed (nodes are shifted) and as such, sharp stress spikes can occur where parts/plates overlap.

Setting up contact surfaces appropriately that account for plate thickness can be time consuming. If necessary, contact thickness can be overridden within the *CONTACT Keyword card or tracked via *ignore*=1. This is especially important with implicit since the solution stability depends upon clean contact.

"Mortar contact is a penalty-based segment-to-segment contact with finite element consistent coupling between the non-matching discretization of the two sliding surfaces"

Thomas Borrvall, DYNAmore Nordic AB

In other words, no soft=2 setting is needed and pretty-much all other varied and sundry contact settings can be left un-touched.

Analyst's Note: Mortar contact does not apply an end extension. For example, in standard _AUTOMATIC_ contact, shell edges are extended as a half-cylinder with a radius equal to ½ the thickness of the shell. Thus, when modeling shell edge to shell face contact, one must allow for the plate thickness and the shell edge extension to avoid initial contact problems. In _MORTAR_ contact, no shell edge extensions are used and the edge of the shell is the edge of the shell.

19.9.2 General Mortar Contact Types

For general contact, the list is short and simple:

1. *CONTACT_AUTOMATIC_SINGLE_SURFACE_MORTAR
2. *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR

How Mortar contact treats penetrations is a subject of some discussion since it is one known mechanism that can destroy convergence. From the Keyword Manual – Implicit Section (and now would be a good time to do some reading):

> ***Initial Penetrations***
>
> As mentioned above, initial penetrations are always reported in the message file(s), including the maximum penetration and how initial penetrations are to be handled. The IGNORE flag governs the latter and the options are

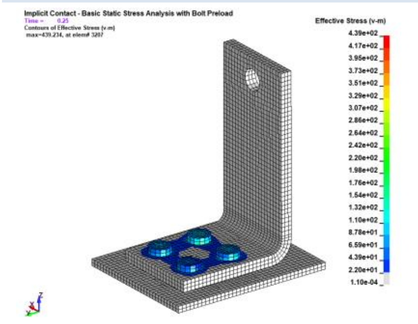
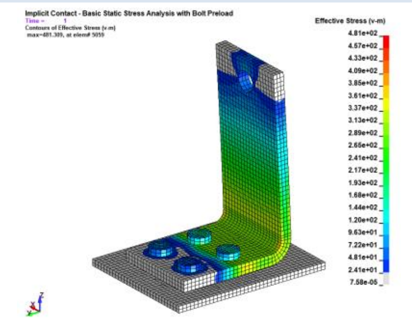
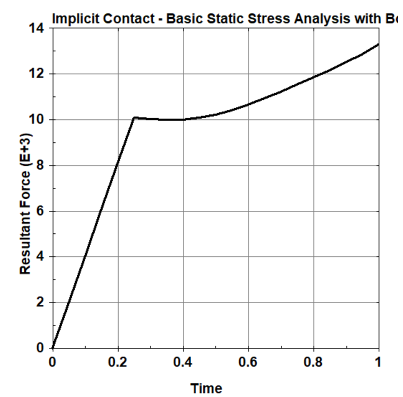
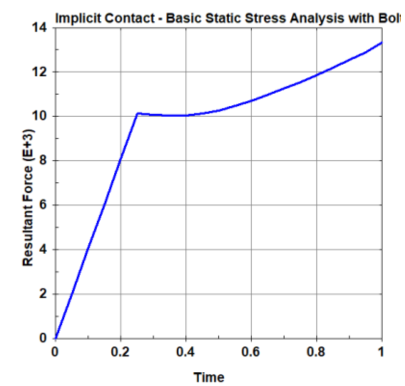
IGNORE < 0	Same functionality as the corresponding absolute value, but contact between segments belonging to the same part is ignored completely
IGNORE = 0	Initial penetrations will give rise to initial contact stresses, i.e., the slave contact surface is not modified
IGNORE = 1	Initial penetrations will be tracked, i.e., the slave contact surface is translated to the level of the initial penetrations and subsequently follow the master contact surface on separation until the unmodified level is reached
IGNORE = 2	Initial penetrations will be ignored, i.e., the slave contact surface is translated to the level of the initial penetrations, optionally with an initial contact stress governed by MPAR1
IGNORE = 3	Initial penetrations will be removed over time, i.e., the slave contact surface is translated to the level of the initial penetrations and pushed back to its unmodified level over a time determined by MPAR1
IGNORE = 4	Same as IGNORE = 3 but it allows for large penetrations by also setting MPAR2 to at least the maximum initial penetration

Most of these options are self-explanatory but *ignore*=4 provides the ability to handle initial interpenetrations. A dedicated workshop will cover this behavior since it is quite useful for interference fits.

19.9.3 Workshop: 26A – Implicit Contact – Static Stress Analysis with Bolt Preload

Objective: Get familiar with contact and being able to check your model by investigating the contact forces generated between the two surfaces pulled together by a bolt preload and then by the "lever" load (Y-axis).

Model Description: A simple plate structure is bolted together with the bolts given a 100 MPa bolt preload and then a horizontal force is applied along the top edge of the L-Section. The analysis finishes at full load when time = 1.0.

At End of Bolt Preload	Finished
	
RCFORCE (Resultant Force)	**SECFORCE (Resultant Force)**
	

19.9.3.1 Bolt Preload Discussion via Solid Elements

The bolt preload is done by two keyword cards: (i) *DATABASE_CROSS_SECTION_PLANE_ID and then (ii) *INITIAL_STRESS_SECTION. The bolt preload stress is defined by a curve. The concept is that the bolt preload is applied at the beginning of the analysis and then held as a "strain". Once the strain is set (aka bolt preload stress) at the end of the bolt preload curve, the keyword action has completed. Think of this as a mechanical operation where one tightens up the bolt and then walks away. The bolt stays "tightened" but may still come loose if subsequent loads cause the bolt strain to change.

Workshop Tasks

1. Open: "Workshop - Implicit Contact - Static Stress Analysis with Bolt Preload - Start 201031.dyn" and take a look. Your job is to set up the contact keywords and also enable contact forces between the L-Section and the Base Plate to be extracted (don't forget that you'll need to request that contact forces be written out to the *DATABASE ascii file. All the other keywords have been set.
2. When the run has finished, use LSPP to review the stress results. One will note that the bolt preload is from 0.0 to 0.25 load (it is a static stress analysis – time is just load). After the bolt preload has been set, a force is applied to the top of the L-Section.
3. Using LSPP, looks the Post / ASCII plots for RCFORCE and SECFORC. They should look the same since you are grabbing the same force balance information.

*If you have some time, inspect the finish deck and note the variable option in the _IMPLICIT_AUTO, iteopt field. Try it out and see if your run goes faster. Then ramp up the load by 2x (see *DEFINE_CURVE, sfo=2) and watch it struggle to converge.*

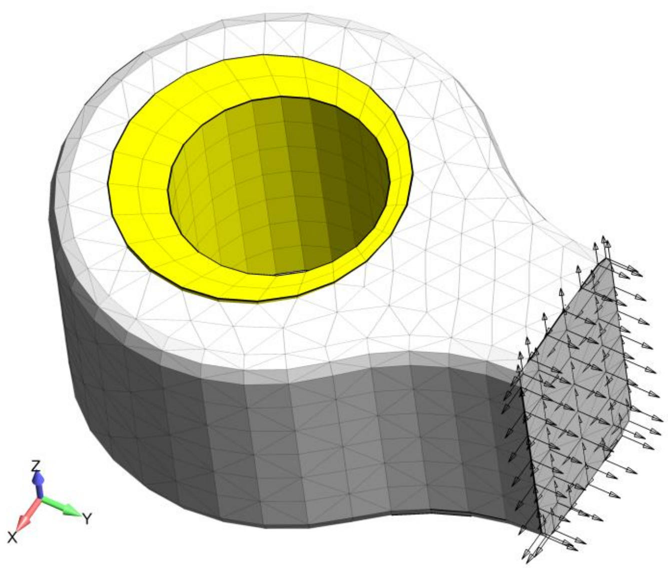
19.9.4 Workshop: 26B - Contact - Shrink Fit Analysis

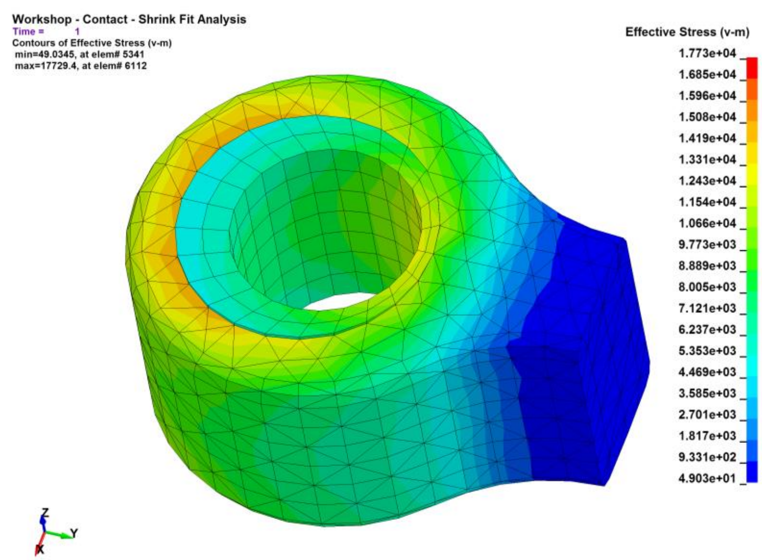
Objective: We explore how IGNORE=4 works for MORTAR contact to handle interference fits

Problem Description: A sub-section of a larger model is used to illustrate how contact can be used to develop an interference fit. No load will be used and the analysis will be run with only the geometric interference fit of 0.0035" is used to drive the analysis. The analysis finishes at time=1.0. Thus one might say that the interference load application is from time 0.0 to 1.0).

Tasks: Setup Mortar Contact with the correct parameters to enforce the interference action

- Review settings and set the appropriate values within the *CONTACT_AUTOMATIC_ card (ignore, mpar1 and mpar2 values);
- Run the model and see if you get lucky;
- Add friction and re-analyze;
- Remove or comment out (i.e., place a "$" in front of the Keyword) the *CONTROL_IMPLICIT_DYNAMICS keyword.



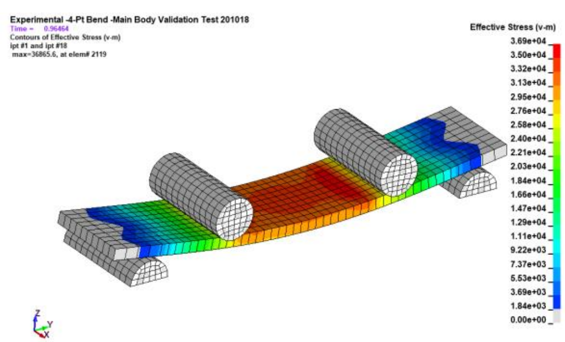
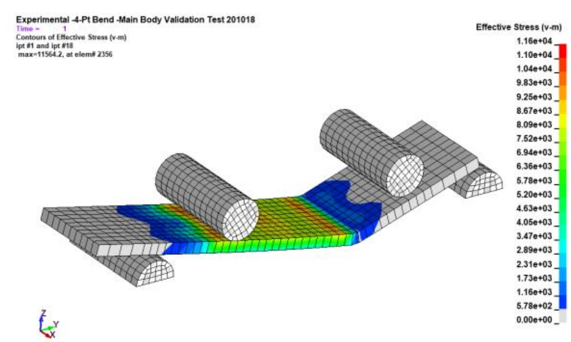
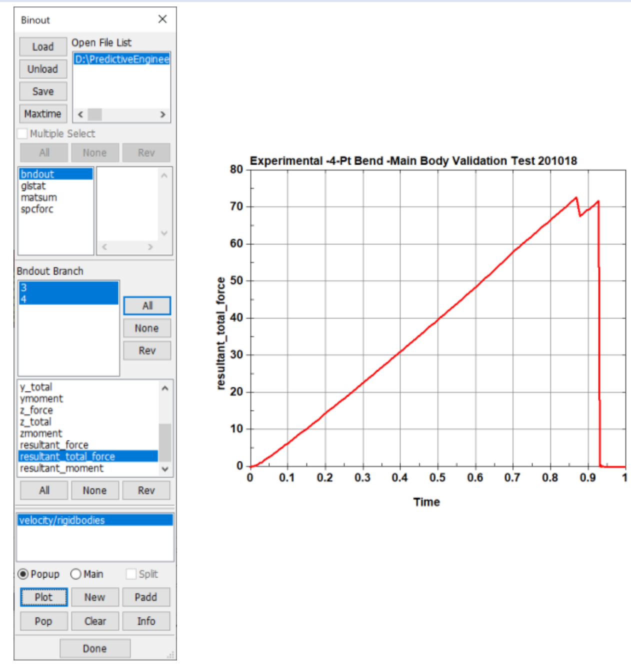


19.9.5 Workshop: 26C – 4pt Bend Composite Bend Test

This is a complex subject and could easily take a couple of hours to just “touch” the subject. The purpose of this workshop is show how a composite test specimen can be setup for a four-point bend test. The composite laminate has 18 plies and is loaded to failure.

Workshop Tasks:

- Inspect keyword deck: “Implicit Nonlinear 4-Pt Bend Experimental Composite Test.dyn”;
- Note how rigid bodies are used to apply the load independently of each other;
- Inspect composite definition with *PART_COMPOSITE;
- Run model and review results. Create XY plot of failure force (bndout – resultant_total_force).

At Max Load	Failure	Force vs Displacement (time=displacement)
		

Analyst’s Note: Let’s see if you can make a cross-plot showing load versus displacement? The displacement can be obtained from the LSPP Post / History / Nodal / Resultant Displacement

19.9.6 Tied Contact for Mesh Transitions, Welding and Gluing

Given the idealization difficulty of system modeling, the ability to tie together different mesh densities (e.g., hex-to-hex or tet-to-hex), snap together parts along a weld-line or just glue sections together (e.g., plate edge to a solid mesh) is an amazingly useful ability and implicit LS-DYNA provides a very complete Tied Contact tool box to work with.

The emphasis of this course to provide an overview of the basics to get started efficiently with LS-DYNA, a short list of recommended *KEYWORDS for implicit Tied Contact are presented.

It is also suggested that this would be good time to review the implicit section of the Keyword Manual under “tied” (Appendix P). Keep in mind that it is better to use fewer options but understand more completely what each command does. For example, the _SURFACE option tells LS-DYNA to only look at the surface of the MST that is provided within the Keyword Card. For example, if a Part is specified, it only looks at the surface of this Part as eligible for Tied Contact and nothing in its interior.

When a Solid Mesh is Used (Translational DOF Tied)

*CONTACT_TIED_NODES_TO_SURFACE (planer connection)

*CONTACT_TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET (When the mesh is offset)

When a Shell Mesh is Used or attaching beams to a shell mesh (All Six DOF Tied)

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET

General Comments

- _TIED (no _OFFSET) implies that the *surfa* surface is co-planer or if it isn't, it'll move it to be adjacent to the *Surfb* surface. And, BTW, it'll only move it if is “close” to the *Surfb* surface (Read The Manual (RTM)). One can override the default search distance by specifying a negative search distance on the SST MST fields. Please note that one has to specify both SST and MST with a negative number for the search distance. When in doubt – RTM.
- _OFFSET will not move your *surfa* surface but you may need to expand your search distance to get things to tie together. This is where setting negative values on SST and MST come in handy. Be aware that on the _NODES side that if the distance is too large you can pick up interior notes if you specify the SST using a PART.

New School: Two (2) Tied Contact to do it all

**CONTACT_TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET with CARD E option IPBACK=1 (to tie to rigid bodies) – 3 DOF (solid) tie*

**CONTACT_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET with CARD E option IPBACK=1 (to tie to rigid bodies) – 6 DOF (shells) tie*

19.9.7 Checklist for Implicit Nonlinear Contact Analysis in LS-DYNA

- ☑ *CONTACT_AUTOMATIC_ ……….._MORTAR should always be used and contact part interference should be checked and if required IGNORE=1 set.
  - ☑ CONTACT_TIED_…... be thoughtful in the use of the appropriate tied relationship based on the degree-of-freedom to the parts to be tied, i.e., solid elements – 3 DOF while shell elements – 6 DOF.
  - ☑ *CONTACT_TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET with CARD E option IPBACK=1 (to tie to rigid bodies) – 3 DOF (solid)
  - ☑ *CONTACT_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET with CARD E option IPBACK=1 (to tie to rigid bodies) – 6 DOF (shells)
  - ☑ *DATABASE_RCFORC to dump out contact forces between SURFACE_TO_SURFACE contacts or use *CONTACT_FORCE_TRANSDUCER to specify a particular Part within a SINGLE_SURFACE contact definition
  - ☑ For contact pressures and more, use the three part setup for INTFOR file information via (i) SPR and MPR on the contact keyword card, (ii) *DATABASE_BINARY_INTFOR to request output intervals and finally (iii) specify the Interface Force s=*output file name* (Advanced option on LS-DYNA MPP Program Manager for Windows)

----- And from the prior section -----

- ☑ *CONTROL_ACCURACY with *iacc*=1
- ☑ *CONTROL_IMPLICIT_AUTO with *iauto*=1 and *dtmax*=0.1 or use a "curve" to define solution keypoints
- ☑ *CONTROL_IMPLICIT_DYNAMICS with *imass*=1, *gamma*=0.60 and *beta*=0.38 for damped dynamics to assist in solution stability, e.g., when buckling occurs. *Note: If a true transient dynamic analysis is required, then the values of gamma and beta should be set to their defaults.*
- ☑ *CONTROL_IMPLICIT_GENERAL with *imflag*=1 & *dto* at 0.1 or some thoughtful initial timestep value
- ☑ Check element formulation: (i) Shells *elform*=-16 with *nip*=5; (ii) Bricks *elform*=-18 (mild plasticity) or -2 (full-on plasticity); (iii) 10-node Tets *elform*=16 (4-node tets *elform*=13) and Beams *elform*=4 with QR=4
- ☑ *CONTROL_IMPLICIT_SOLUTION (*dctol*=1e-20, *dnorm*=1 & *nlnorm*=4) and *nlprint*=2 to print out convergence information to the screen
- ☑ *CONTROL_OUTPUT with *shlsig*=2 and *solsig*=2 for solid element extrapolation of integration point stresses to the nodes
- ☑ If the analysis is not dynamics based, a recommendation is to set load curve to end at Time=1.0 and Termination=1.0 for consistency
- ☑ *DATABASE_EXTENT_BINARY to recover integration point data for shells, solids and beams via *maxint*=-3, *beamip* > 0 and *nintsld*=8.
- ☑ In post-processing of stress data, fringe shell and solid results using the "Mid" setting since it aligns the closest to standard FEM post-processing

19.9.8 But My Boss Says That Using "Dynamics" is Wrong for a Static Solution?

This is a real story. Yes – turning on *CONTROL_IMPLICIT_DYNAMICS means you have a full-on dynamic solution. That means one is solving for mass x acceleration and inertia effects are real and can be present. Since every simulation is different, it would be a fool's errand to suggest that one can dismiss dynamic effects by just stating that the solution is critically damped (gamma=0.60 and beta=0.38). Therefore, to demonstration that your quasi-static solution does not suffer from erroneous stress amplification effects, one can show a plot of internal energy (IE) and kinetic energy (KE). Additionally, our practice is to apply any nonlinear implicit static load slowly to dissipate any KE effects. Additionally, we will apply the load, e.g., from 0 to 1.0 and then extend the solution out to 1.1 at constant load. Obviously, if we see any SPC force variation or noticeable stress variation (<1%), we would conclude that we should be a bit more careful on the load application. Otherwise, we can deem it a pure nonlinear implicit static solution identical to our implicit pseudo-static solution.

Okay – why do we prefer to run all our nonlinear implicit solution using *CONTROL_IMPLICIT_DYNAMIC? At the end of the day, they just run better. It encourages a smoother convergence and handles any initial mechanisms prior to contact and for buckling, it just plain solves the impossible. Our preference arrives due to experience rather than pure numerical purity. As engineers, we need accurate solutions, quickly and efficiently. By adding a bit of *dynamics*, we achieve these targets.

19.10 Rigid Body Usage

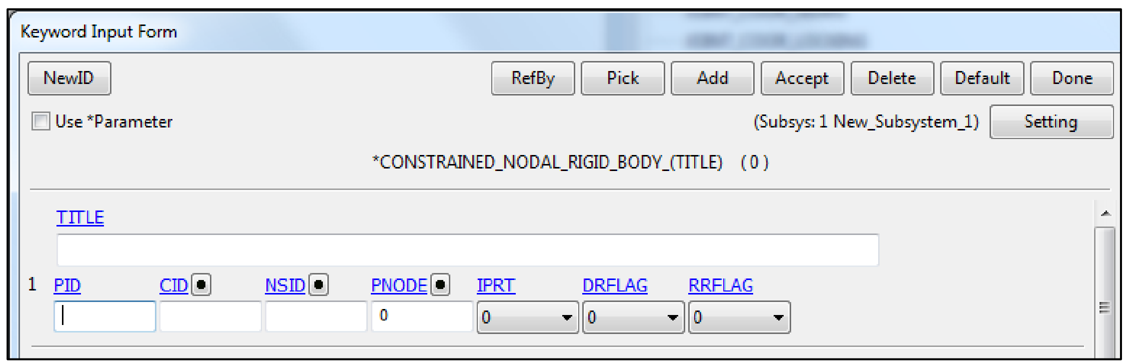
19.10.1 RBE2 (Nastran) to CNRB

In Nastran, a rigid-body element is a multi-point-constraint (MPC) where the constraint relationship is enforced within the stiffness matrix. This operation is done at the start of the Nastran analysis and is only applicable to a linear, elastic analysis since it ties together stiffness terms. Given this background, a MPC is completely incompatible with a nonlinear analysis. In LS-DYNA, MPC's are translated into *CONSTRAINED_NODAL_RIGID_BODY (CNRB's) and otherwise have nothing in common with a MPC – they are completely nonlinear in capability. A CNRB is described and is treated as a "separate body" with six DOF. It ties all the nodes together into a rigid body that can translate and rotate as a rigid body.

From the Keyword Manual:

The first node in the nodal rigid body definition is treated as the surfb for the case where DRFLAG and RRFLAG are nonzero. The first node always has six degrees-of-freedom. The release conditions applied in the global system are sometimes convenient in small displacement linear analysis, but, otherwise, are ***not recommended****.*

They are not tricky to use if one keeps it simple and never use releases or that is to say, always keep all six DOF engaged. Our experience is that if you release one of the DOF's, the model won't converge due to mechanisms.



Extra Credit: There is a RBE2 to CNRB example file in the Instructor Led Workshops / RBE2 to CNRB / CNRB file folder that demonstrates that a CNRB preserves spatial relationships and can be used to transfer nodal loads.

19.11 Nonlinear Transient Dynamic Analysis (implicit): Initialization to Transient Dynamic

Load initiation refers to getting the right stress and deformation shape of the system prior to the application of the principal load case. In this example, we will show the process for performing a nonlinear transient dynamic stress analysis using the implicit method.

19.11.1 What is a Satisfactory Implicit Time Step for a Transient Event?

A transient time step, whether explicit or implicit, is often defined by the load event and the system frequency of interest. For example, if the load event occurs over 100 μs as a half-sine wave, then a time step of 20 μs might be sufficient to capture the load application (five points on the curve); but that is just the load application and it doesn't address the system's response. This is where it can be tricky and performing a normal modes analysis on the system can lend insight into how the system would respond or dynamic vibrate upon load application. For example, if the first mode of the system is 1000 Hz, then its response could be numerically captured with a time step of 100 μs (i.e., frequency 1 kHz – one cycle very 1 ms and then 10 points per cycle). If this was our loading and system of interest, then a time step of 20 μs would work well and capture the loading response 1 to 1 and with a 5 to 1 "overkill" on the system response. Conversely, if our loading occurred over 5 ms, the loading step could be 1 ms, but given the system's response, we would want to keep the time step at 100 μs. These observations hold true whether the analysis is run using the explicit or implicit method. Given that the explicit method usually has a very small time step, it is often not necessary to think about managing the time step toward the loading and system response.

Important Reminders:

- Think about the timing of the load application;
- It is not just the 1st natural frequency of the system but the natural frequency of interest that you want to capture. If the 1st mode is 500 Hz but the frequency of interest is 5000 Hz, then the time step must be set to that frequency. The implicit time step determines the frequency content of the analysis;
- Although Nyquist's digital sampling rule indicates that 2 data samples is sufficient to capture a frequency it implies that a frequency can be measured and says nothing about understandings its mechanical response. Using 10 points at the frequency of interest ensures that one will capture the complete frequency wave to a reasonable measure. Of course, once a solution is in hand, one can play around with a smaller time step to determine the sensitivity of the analysis to the time step.

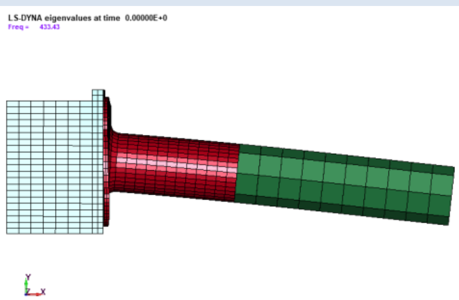
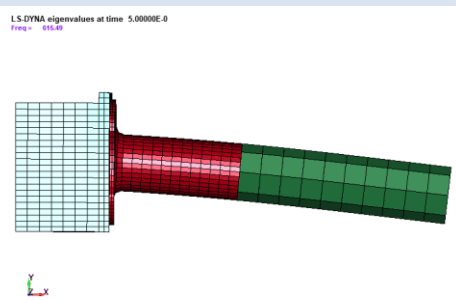
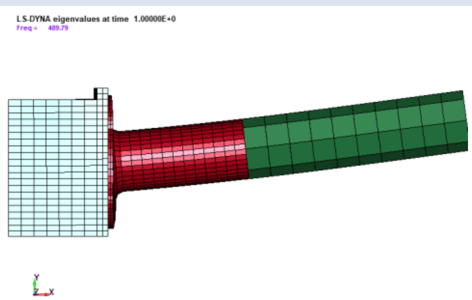
19.11.2 Workshop: 27 – Implicit - Nonlinear transient Dynamic Analysis

Introduction: In the Workshop below, the model will be given a bolt preload and then loaded and released. The bolt preload is done from 0.0 to 0.5 and then an end tip force is applied from 0.5 to 1.0. This can be termed the static stress initialization of the structure. At this point (time = 1.0), the load is removed and the structure is allowed to vibrate. Our objective is to capture the transient dynamic response of the structure once the load is released. We will solve this problem using the implicit method. To determine the implicit time step we will only need to consider the system response since load application is done statically. The dynamic system response can be evaluated by performing an Eigenvalue or Normal Modes analysis. Once we know the structure's dominate response (1st mode), we can calculate an appropriate trial implicit time step. Please note the use of the word "trial" since one should consider this initial time step as on the top end and smaller values should be also investigated to determine the structure's sensitivity to the time step. Please keep in mind that with an explicit analysis, the time step is usually several orders of magnitude smaller than what might be needed to resolve the system's dynamic response (albeit outside of crash and ballistic events).

Model Introduction: The keyword deck has (Workshop - Implicit - Nonlinear Transient Dynamic - Start – 201102.dyn) has been setup for you to fill-in the necessary values. The idea of this analysis is to perform a complete analysis from start-to-finish. The first part is to determine the natural frequencies of the structure in its final loaded state. Once we have this information, we can set the time step for the transient analysis.

19.11.2.1 Part I: Normal Modes Analysis (Eigenvalue) with Load Application

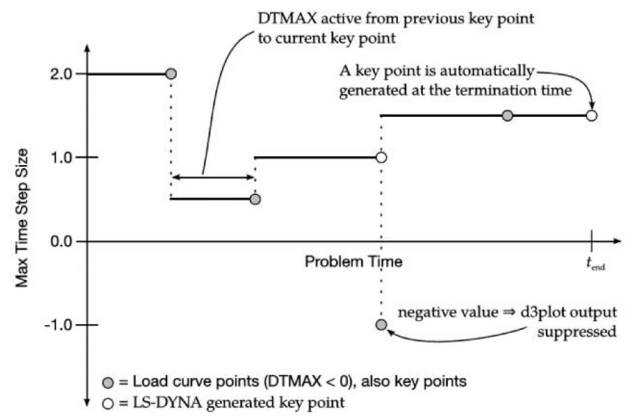
- Bolt-Preload: Setup model to for bolt preload analysis by defining the bolt preload at 1,000 (the units of the model are English). This is done thru the *INITIAL_AXIAL_FORCE_BEAM where scale=1000. Note that the load curve (see *DEFINE_CURVE_TITLE, *lcid*=2) stops at 0.5;
- Apply force load at the end of the beam element using *LOAD_NODE_POINT, *sf*=1000;
- Define contact whether SINGLE_SURFACE or SURFACE_TO_SURFACE as you wish. Since this is implicit, don't forget _MORTAR;
- Setup intermittent eigenvalue (normal modes) analysis using *CONTROL_IMPLICIT_EIGENVALUE, *neig*=-3 (yes, it is a curve definition since it is negative 3). Verify that *DEFINE_CURVE_TITLE, *lcid*=3 exists and has the right settings of 0,10 / 0.5,10 / 1.0, 10;
- Set termination time to 1.0, save Keyword file to new name and LS-Run'it;
- Once it has run, load the d3eigv1 file and record the 1st mode frequency, repeat this step for d3eigv2 and d3eigv3;
- Create an XY Plot of the bolt (beam element) axial bolt force. If you don't know how, watch the video showing the LSPP steps;

d3eigv1: 1st Mode – 433 Hz	**d3eigv2: 1st Mode – 615 Hz (see *Read Me Note*)**	**d3eigv3: 1st Mode 490 Hz**
		

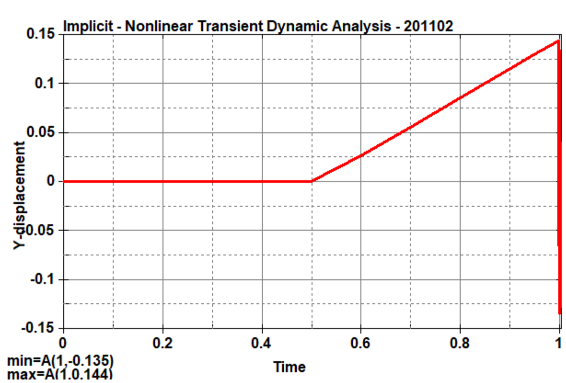
19.11.2.2 Part II: Implicit Nonlinear Transient Dynamic Analysis

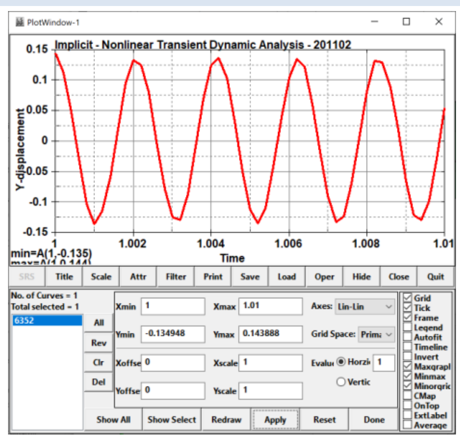
- Since the time step doesn't depend upon the load application (all load application was done statically), it then just depends upon the system's mechanical response. Our goal is to capture its 1st mode response. At its final loaded state, the 1st mode is 490 Hz. At this frequency, the time step is calculated as 1/(490 x 10) = 0.204 ms or 2.04e-4 seconds. To keep it simple, we'll round down to 0.2 ms. The implicit timestep in this analysis is controlled by the dtmax setting in the *CONTROL_IMPLICIT_AUTO keyword. A curve will be used to set the dtmax at 2 ms. This is done by adding a line to the end of *DEFINE_CURVE_TITLE, lcid=4 as 1.0001, 0.0002;
- Termination time (*CONTROL_TERMINATION, endtim) is defined as the initialization time (1.0) plus the transient dynamic run time. To visualize 5 cycles, it would be something like 5/490 or 10 ms; thus, the endtim=1.01;
- The next step is to release the applied force after time=1.0. This is done by *DEFINE_CURVE_TITLE, *lcid*=3 and just remove the "$" and adjusting the spacing. Thus, at time=1.00001, the load goes to 0.0 and the structure will start to vibrate;
- Run the model and load the d3plot file into LSPP. Plot the Y-Displacement of the node at the end of the beam (tip of the model) (LSPP / Post / History / Nodal / {select node at the end of the beam #6352} / Y-Displacement / Plot;
- After this step, perform an "oper" on the curve via the fft_radix and note that the fundamental frequency is 500 Hz and closely matches the eigenvalue 1st mode.

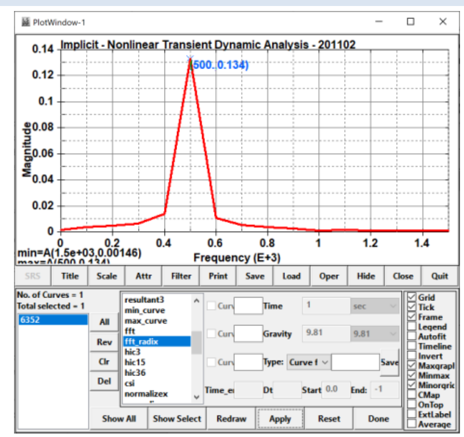
Keyword Manual Vol 1: dtmax explanation



Y-Displacement Results	Results Limited (Scale) from 1.0 to 1.01	Results processed using Oper / fft_radix



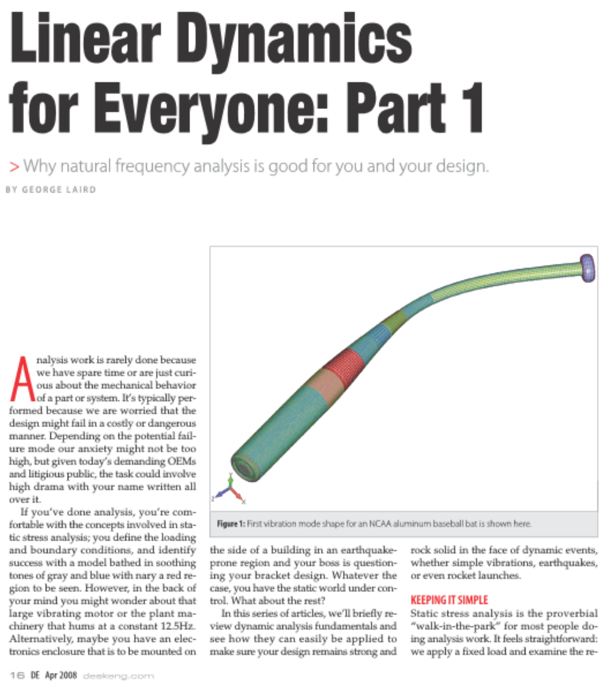




19.12 Linear Dynamics: It Is All About The Normal Modes

Linear dynamics is a class of solutions that are developed based on the assumed linear behavior of the system during the dynamic excitation. In numerical terms, once the stiffness matrix has been calculated, it is not updated during the solution. The linear dynamic solutions that will be discussed within this section are based on a normal modes analysis. A nice basic primer can be found in the Class Reference Notes if you are not sure about the foundation of linear dynamics.

See Class Reference Notes / Linear Dynamics / Super-Simple Vibration Primer



Linear Dynamics for Everyone: Part 1

> Why natural frequency analysis is good for you and your design.

BY GEORGE LAIRD

Analysis work is rarely done because we have spare time or are just curious about the mechanical behavior of a part or system. It's typically performed because we are worried that the design might fail in a costly or dangerous manner. Depending on the potential failure mode our anxiety might not be too high, but given today's demanding OEMs and litigious public, the task could involve high drama with your name written all over it.

If you've done analysis, you're comfortable with the concepts involved in static stress analysis; you define the loading and boundary conditions, and identify success with a model bathed in soothing tones of gray and blue with nary a red region to be seen. However, in the back of your mind you might wonder about that large vibrating motor or the plant machinery that hums at a constant 12.5Hz. Alternatively, maybe you have an electronics enclosure that is to be mounted on the side of a building in an earthquake-prone region and your boss is questioning your bracket design. Whatever the case, you have the static world under control. What about the rest?

In this series of articles, we'll briefly review dynamic analysis fundamentals and see how they can easily be applied to make sure your design remains strong and rock solid in the face of dynamic events, whether simple vibrations, earthquakes, or even rocket launches.

KEEPING IT SIMPLE

Static stress analysis is the proverbial "walk-in-the-park" for most people doing analysis work. It feels straightforward: we apply a fixed load and examine the re-

Figure 1: First vibration mode shape for an NCAA aluminum baseball bat is shown here.

16 DE Apr 2008 deskeng.com

Linear Dynamics for Everyone: Part 2

> Vibration analysis can show detailed structural behavior under dynamic loading.

BY GEORGE LAIRD

In part I of this series (*DE* April 2008, p. 16), we explained the concept that every structure has natural frequencies of vibration (eigenvalues) and that these natural frequencies have specific deformation shapes (eigenmodes or normal modes). We also took a swipe at how one would use this information in the structural design world by noting that excitation frequencies outside of a structure's first couple of eigenvalues means it will behave statically stable. We now want to expand upon this theme and demonstrate how this simple form of analysis can be leveraged to uncover how your structure might behave under dynamic loading.

THE DOMINATORS: MODES WITH MASS

An interesting fact about normal modes analysis is that we can associate a percentage of the structure's mass to each mode. With enough modes, you get 100 percent of the mass of the structure, though for complex structures this can mean hundreds of modes. The common thought is that if you capture 90 percent of the mass of the structure that will be good enough. For now, we'll start classically and then show what this concept means in a real-world engineering situation.

We use the supported beam because it is simple to visualize, simple to formulate, and best of all, simple to draw on a white board. In Figure 1 (*next page*), we show a quick example of the first three modes of a simple supported beam along with the percentage of mass associated with the mode. The first mode dominates with 82 percent of the mass of the beam swinging up and down. The second and third modes contribute a little bit of mass but nothing like what we saw in the first mode.

All of these modes operate in one direction. In the real world, the mass fraction is associated with all six degrees of freedom within a particular natural frequency. What this means is that if we excite this first mode in the vertical direction (the direction of the mass fraction), then the structure will move with 82 percent of its mass behind this mode. If we think like Newton and realize that F=m*a, then we can visualize the dominance of this mode and the huge forces that can be generated at resonance.

PEA POD TRANSPORT

Let's leverage this information in a couple of typical engineering problems. Manufacturers use vibrating conveyors to move materials ranging from pea pods to lumps of coal. One such vibrating conveyor is shown above. It moves pea pods within a food-processing plant using a vibratory motor that creates a sinusoidally varying force that is aligned down the axis of the conveyor (*y*-axis). This force causes the conveyor to swing forward and up on its fiberglass laminate springs.

When operating at its resonant frequency, the conveyor tosses the pea pods forward and upward in a gentle swinging fashion. The material transport rate is determined by its operating frequency and the length and angle of the fiberglass spring laths.

A fundamental problem with this type of conveyor is that during startup as the vibratory motors spin up to speed, nonoperating modes get excited, often causing the conveyor to tear itself apart before it reaches the target operating frequency. Our eigen analysis of the conveyor shows that it has to pass through three modes before reaching its operating frequency at 18Hz. Table 1 shows a brief summary of the data

The motor mount for this vibrating conveyor is the mass of flexible metal plates hanging off the end of the conveyor. Yellow elements are fiberglass laminate springs; the motor is not shown.

62 DE May 2008 deskeng.com

Linear Dynamics for Everyone

> Part 3: Extracting real quantitative data to anticipate everything from earthquakes to rocket launches.

BY GEORGE LAIRD

If you've kept up with this series of articles, you now know more about the dynamic behavior of structures than 95 percent of your peer group within the design and engineering world. And after reading about how vibration analysis reveals key information about structural behavior (see *DE*, April and May 2008), the terms "natural frequencies, normal mode shapes, mass participation factors, and strain energy" have become integral to your vocabulary.

Up to this point the discussion has centered on qualitative terms about the mechanical response of structures due to dynamic loading. In this last part, we'll show how to extract real quantitative data (i.e., displacements and stresses) from a simple normal-modes analysis.

DOING IT ON THE CHEAP

The dynamic response of a structure is derived from its individual normal modes. If you hit your structure, its dynamic response is formed by the summation of its individual modes. Mathematically, we know that each one of our normal modes has a frequency, a mode shape, and a bit of mass associated with that shape (a mass participation factor). All of this data is derived from the basic equation of motion:

$$ma+kx=0$$

From this equation, the standard linear dynamics solution can be derived as:

$$v=\sqrt{K/m}$$

where v is the frequency or eigenvalue of the system. Since no forces are involved in this equation we can't have any real displacements or stresses.

If we want real data, we need real forces as in: $(ma+kx=F)$.

The brute-force approach is to solve the model in the time domain. A time-based displacement, force, or acceleration load is inserted into the model and then the computer makes a few thousand solves. At some later date, we then wade through piles of output data (remember, we are doing a complete solve at each time step) to figure out what went wrong, when, and where. This can be a daunting task and is often just plain impractical.

If the loads are frequency-based (displacement, force, or acceleration as a function of frequency), then the door is wide open to all sorts of very numerically efficient solution strategies. First you perform a normal modes analysis and then apply a

This is a satellite FEA model showing instrumentation attachment points (black squares) for idealized mass elements as defined by a center of gravity connected by rigid links (orange lines). The model is driven with an input power spectral density (PSD) function.

18 DE June 2008 deskeng.com

19.12.1 Normal Modes Analysis

Normal modes analysis requires a stiffness and a mass matrix. One calls it “normal modes” since it is the normal or natural response of a structure or system to excitation. However, a normal modes analysis does not look at the forced response of the structure due to loading but its natural response that might occur upon loading. These leads to a naming convention where the terms “natural frequency” or “normal modes” are used to denote the inherent or natural response of the system.

To calculate the normal modes of a system, one solves an Eigenvalue problem based on this matrix. Please note, since there is no excitation, the mode shapes indicate the permissible mode of deformation and not any specific magnitude; that is why it is good to know the underlying equations.

Eigen Equation	Frequency Relationship	Eigen Mode Extraction
$([K] - \omega^2[M])\phi = 0$	$\omega = \sqrt{\frac{K}{m}}$	$\phi^T[K]\phi = [\omega^2]$

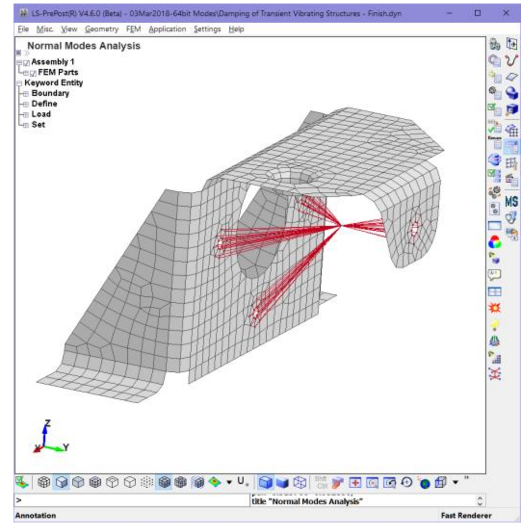
In FEA terms, this is simple but when one has thousands of natural frequencies it can be difficult to extract out all the roots. The key takeaway is that it assumes that your structure behaves linearly during the vibratory response and that is a good assumption for most structures that vibrate elastically.

Keywords for a normal modes analysis are given below.

Keyword	Field Name	Discussion
_IMPLICIT_EIGENVALUE	NEIG=?	Although other settings can be useful, NEIG is the main setting to define the number of eigenvectors to extract.
_IMPLICIT_GENERAL	IMFLAG=1	Standard linear analysis

*Analyst Note: That is it for settings. When LS-DYNA reads that a standard Eigenvalue analysis is requested, it sets the required defaults and the analysis is ready to go. This means it is not necessary to set other standard *CONTROL_ keyword cards.*

19.12.1.1 Workshop: 28 - Normal Modes Analysis

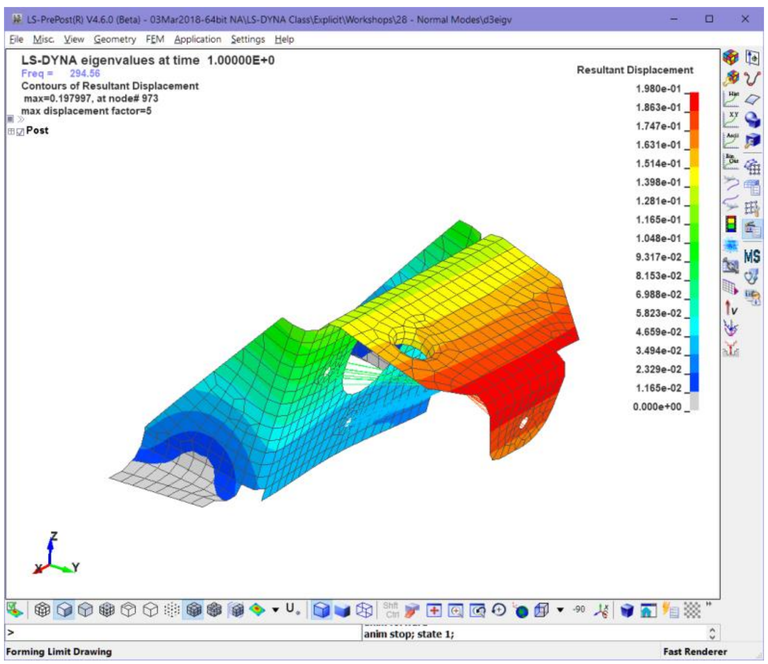


Objective: Take a Keyword deck and set it up to run a normal modes analysis

Problem Description: A bracket is to be analyzed for normal modes. The Keyword deck has been generated but it lacks the right setup. The student gets to apply the correct Keyword settings.

Tasks: The Keyword deck "Normal Modes – Start.k" is your starting point. One should edit this deck to create a running model. Given the prior descriptions above, no further listing is needed. After running the model, open up the text file eigout and take a look at its contents.

Knowledge Gained: Basic analysis is pretty simple. But there are some secrets. The element formulation can influence the calculated normal modes. Since it is linear, one should use *elform*=21.

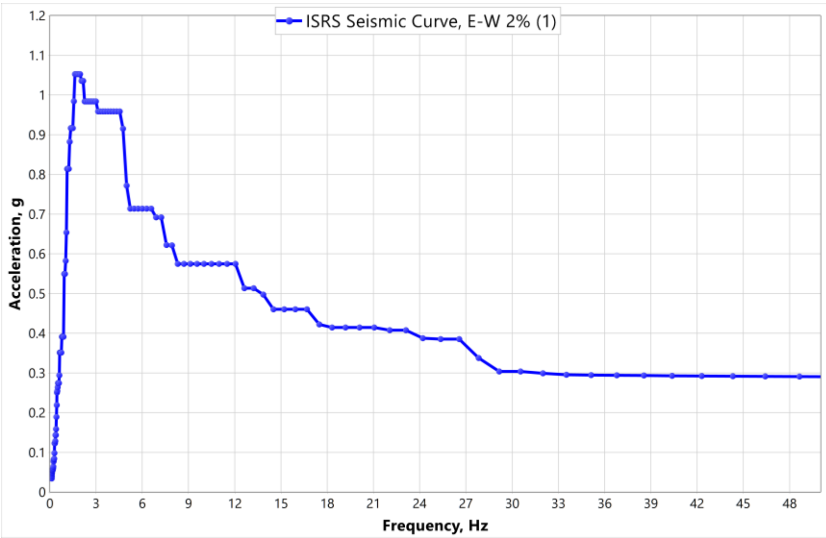
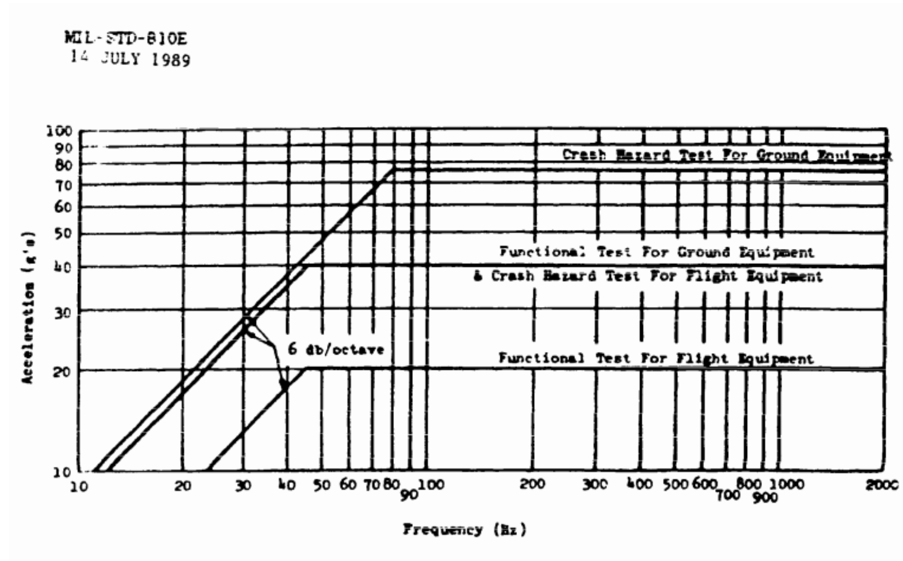
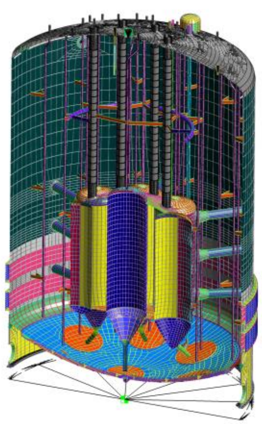
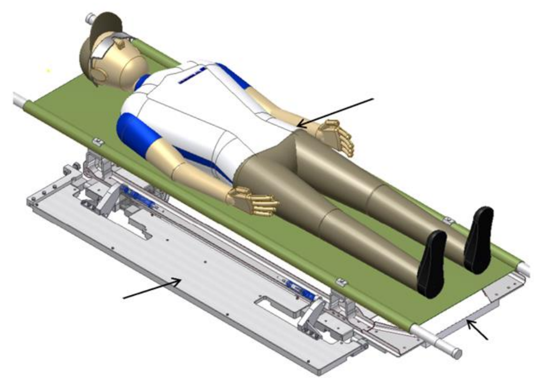



```
Normal Modes Analysis - Super Simple
                       ls-dyna mpp.124727 d          date 03/02/2018

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+00

(all frequencies de-shifted)

                           |------ frequency -----|
      MODE    EIGENVALUE     RADIANS        CYCLES        PERIOD
         1   3.425471E+06   1.850803E+03   2.945644E+02   3.394843E-03
         2   2.754894E+07   5.248708E+03   8.353579E+02   1.197092E-03
         3   3.536632E+07   5.946959E+03   9.464879E+02   1.056538E-03
         4   4.657841E+07   6.824838E+03   1.086207E+03   9.206351E-04
         5   1.374527E+08   1.172402E+04   1.865936E+03   5.359241E-04
         6   1.603425E+08   1.266264E+04   2.015322E+03   4.961986E-04
         7   2.734267E+08   1.653562E+04   2.631726E+03   3.799788E-04
         8   3.155929E+08   1.776494E+04   2.827377E+03   3.536847E-04
         9   3.855595E+08   1.963567E+04   3.125114E+03   3.199884E-04
        10   4.588777E+08   2.142143E+04   3.409327E+03   2.933131E-04


MODAL PARTICIPATION FACTORS

       MODE       X-TRAN          Y-TRAN          Z-TRAN          X-ROT           Y-ROT           Z-ROT

         1   0.234557E-03   -0.599350E-01   0.516048E-01   0.180014E-02   -0.520591E-05   -0.265710E-04
         2  -0.569352E-01    0.351520E-02   0.376884E-02   0.254942E-03    0.371594E-03    0.227060E-02
         3  -0.591407E-02   -0.594683E-01  -0.325598E-01  -0.300773E-02    0.287621E-04    0.744643E-04
         4   0.670942E-01   -0.196855E-02   0.311782E-03  -0.638429E-04    0.306631E-03    0.179950E-02
         5   0.513604E-04    0.154681E-01   0.643504E-01  -0.254471E-02   -0.551003E-04   -0.374652E-04
         6   0.460547E-02   -0.125631E-02  -0.244294E-02   0.434920E-04   -0.795643E-03    0.467199E-03
         7   0.203180E-02    0.170437E-02  -0.621061E-02  -0.150485E-03   -0.438943E-03    0.160452E-03
         8   0.123897E-02   -0.557844E-02   0.304694E-02  -0.128287E-02   -0.381271E-03    0.214756E-03
         9   0.489319E-02   -0.602439E-03   0.609455E-03  -0.339623E-03    0.255261E-03   -0.144242E-02
        10  -0.271448E-02   -0.204441E-02  -0.100466E-02  -0.928683E-03    0.230865E-03    0.120187E-03


MODAL EFFECTIVE MASS


       MODE            X-TRAN                     Y-TRAN                     Z-TRAN
                 Eff. Mass    Accum. %      Eff. Mass    Accum. %      Eff. Mass    Accum. %
```

Extra Credit: Change element formulation to elform=16 and finally back to 21 – compare results.

19.12.2 Response Spectrum Analysis or Shock Response Analysis

This type of analysis is part of the MIL-STD-810E and is also used in seismic analysis. It is also known as a poor man's transient analysis since it uses the structures normal modes to form the overall response of the system. It is also easy to post process the results since a summed stress response is provided by the code.

Seimic Response Spectrum (Earthquake)	MIL-STD 810E Shock Spectrum
	
	

19.12.2.1 Workshop: 29 - Response Spectrum Analysis of Bracket

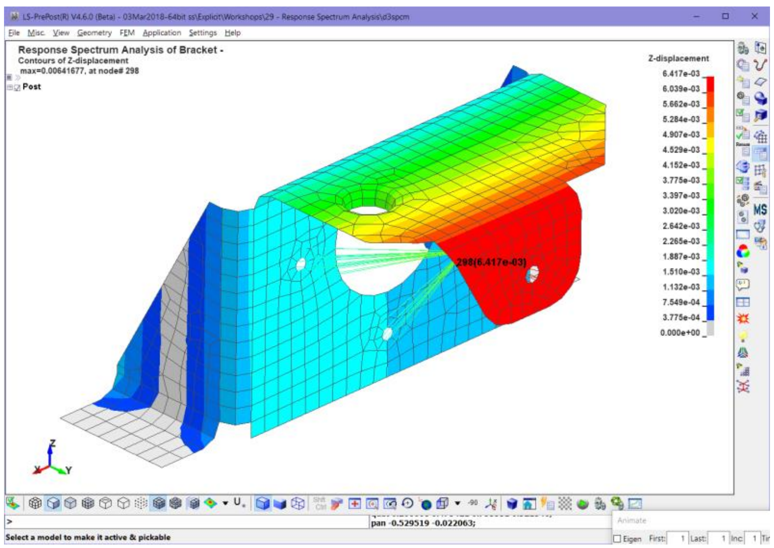
Objective: Leverage existing literature example to modify an existing deck and create a response spectrum analysis using the Crash Hazard Test shock spectrum.

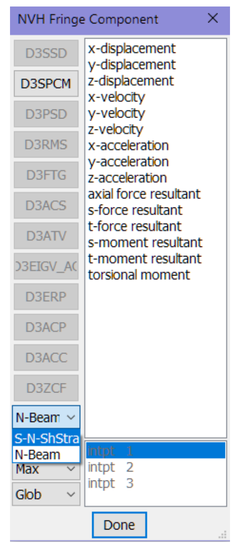
Tasks: Review the response spectrum description given in Frequency_domian_analysis_guideline.pdf on pages 26 to 32 (see Class Reference Notes / Linear Dynamics / User Guide for Linear Dynamics). Modify the Keyword deck provided in the file folder for this workshop. If you get lost and it doesn't work, a –Final deck is provided. There are some tricks – like the MSTRES variable in the _EIGENVALUE keyword.

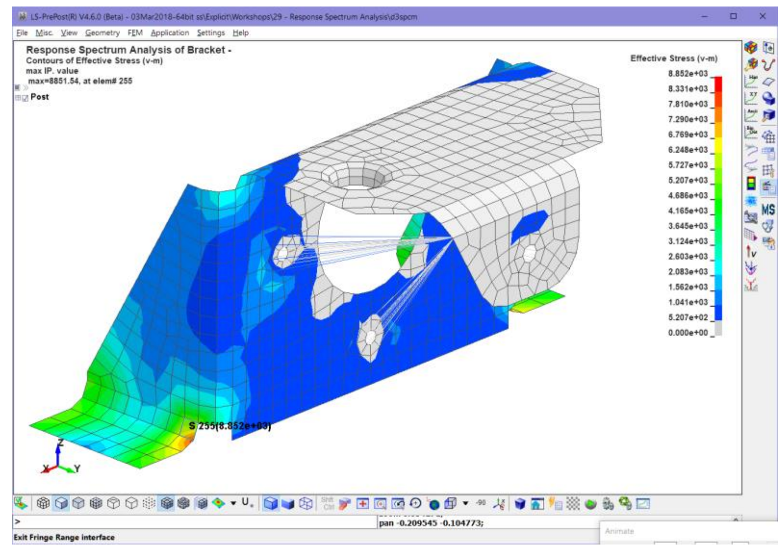
Special Notes: The spectrum is in g and thus needs to be scaled within the units of the model. The mode is in English units, thus the scale factor is 386. We are doing a base excitation, but we could also excite the structure by a node set. In this run we'll use base method and set LCTYP=1. *The base excitation excites the structure through the SPC constraint set; i.e., wherever these is a constraint in the excitation direction is the location where the excitation is applied.*

Knowledge Gained: The ability to move forward using examples clawed out from the literature and feel confident that almost anything can be done in LS-DYNA with enough sweat equity.

Load d3spcm binary file into LSPP and Contour Displacement and Stress (toggle NVH Fringe Component field to S-N-Sh)





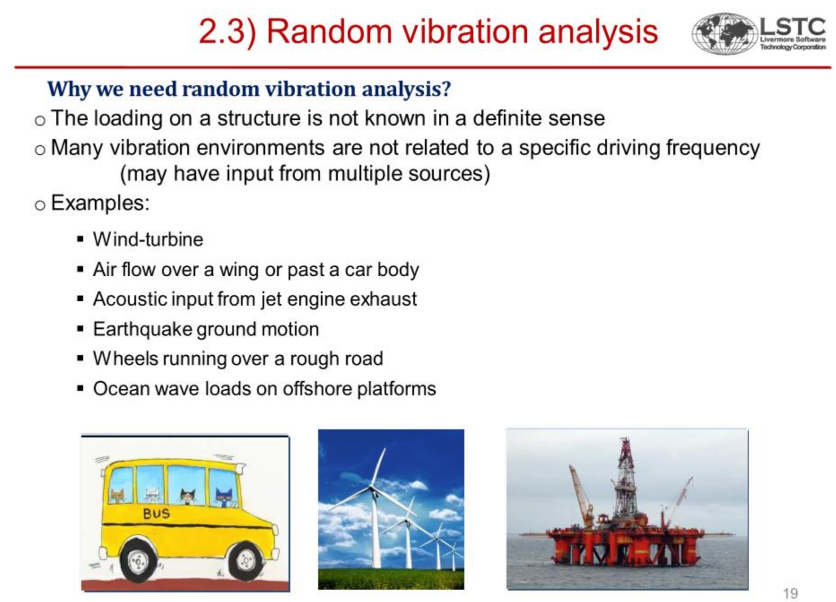


Extra Credit: Change element formulation from elform=21 to =20 – better or worse? Change MCOMB from SRSS formulation to CRC – what does it mean? If you need to analyze multiple directions

19.12.3 PSD Analysis

This is a simple example to demonstrate the robustness of the LS-DYNA code to handle PSD analysis. This FEA consulting project was done initially in Nastran and is now setup to run in LS-DYNA. The model is presented as a template such that the student could develop their own PSD analysis. A compendium of information is given in the Class Reference Note / Linear Dynamics.

No Need to Re-Invent the Wheel – see Recent Developments in Vibration Acoustic and Fatigue Solvers in LS-DYNA October 2017



This analysis leverages the User Guide Material that was noted in the prior workshop.

The files of interest are (see Class Reference Notes / Linear Dynamics or Workshop: 30):

- ✓ Frequency_domian_analysis_guideline.pdf
- ✓ M-10.MANUAL_FREQUENCY_DOMAIN_RANDOM_VIBRATION.pdf
- ✓ S-11.sample.random.vibration.k

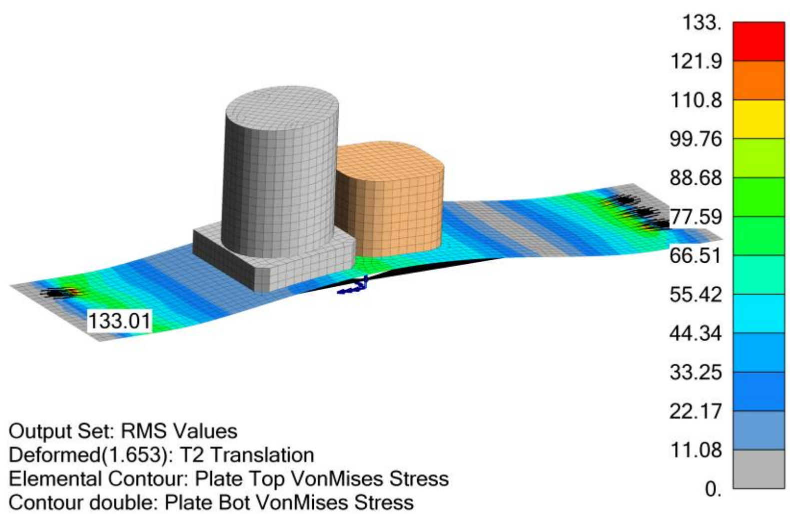
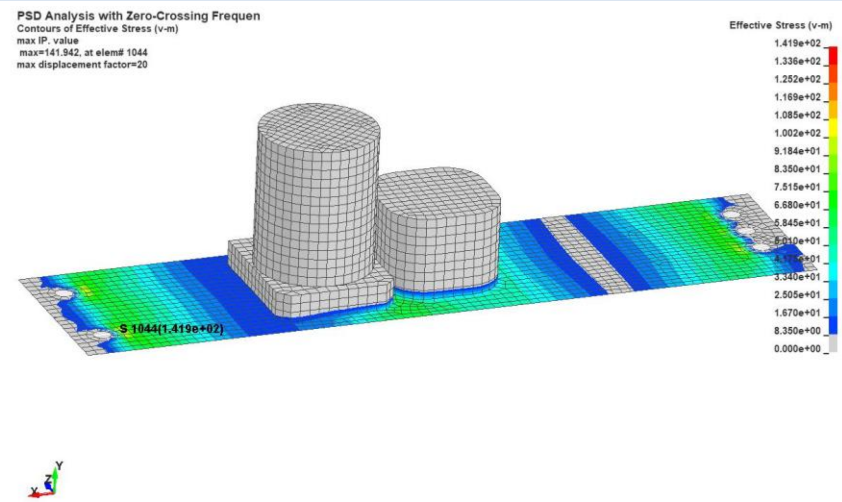
19.12.3.1 Workshop: 30 - PSD Analysis and Zero-Crossing Frequencies

This workshop will require the understanding of how these three Keywords work within a standard linear dynamics Keyword deck (aka, normal modes analysis).

Keyword	What They Do
*FREQUENCY_DOMAIN_RANDOM_VIBRATION	Defines the setup for a PSD analysis. See Manual for description
*DATABASE_FREQUENCY_BINARY_D3RMS	Requests that a binary database be written for the PSD displacements and stresses
*DATABASE_FREQUENCY_BINARY_D3ZCF	Requests that a binary database be written for the PSD Zero-Crossing frequencies

Tasks:

- Inspect the Keyword deck (PSD Analysis with Zero-Crossing Frequencies - Start.k). Units are in N, mm, s and tonne.
- Update Keyword deck to create PSD analysis with two binary result files – D3RMS and D3ZCF.
- Investigate results and see if they make sense. Look at Eigenvalues, PSD RMS and the Zero-Crossing Frequencies.

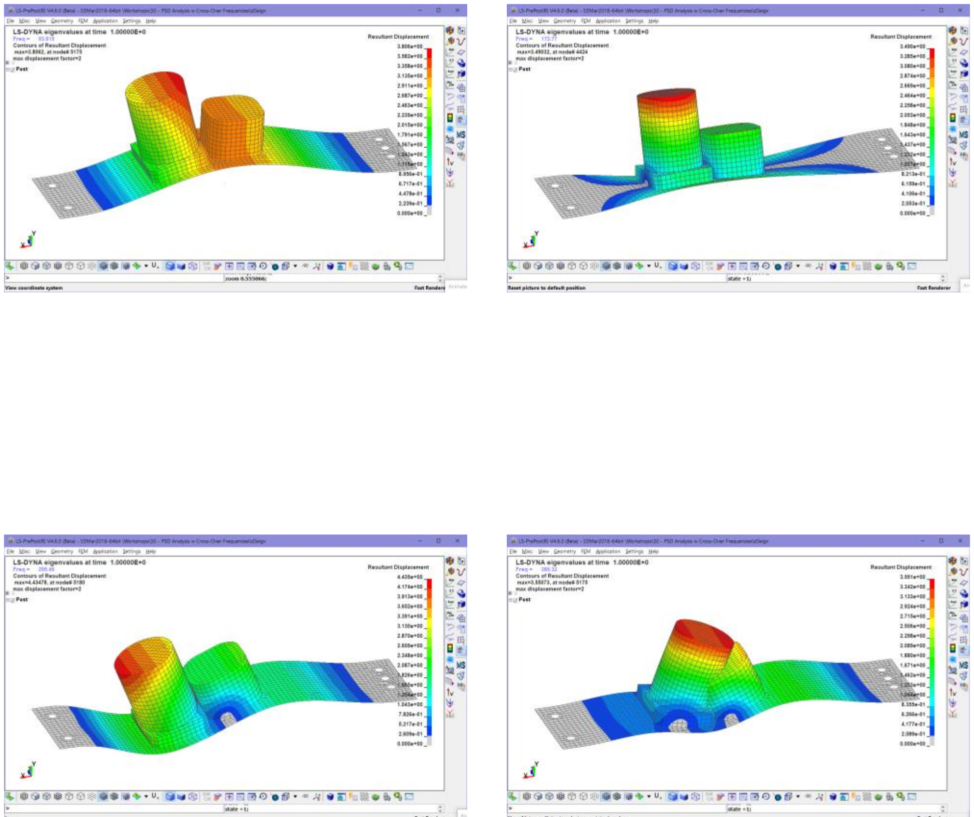
Nastran PSD Result	LS-DYNA PSD Result
	

Analyst's Note: PSD computation and RMS computation in LS-DYNA are totally independent. The reason is to provide a consistent RMS result not depending on the output frequency (PSD) resolution. MS is computed by integration on the input PSD and also FRF transfer functions, not on the output PSD. A higher order spline interpolation is used to get the semi-analytical RMS solution. (Explanation courtesy of Yun Huang)

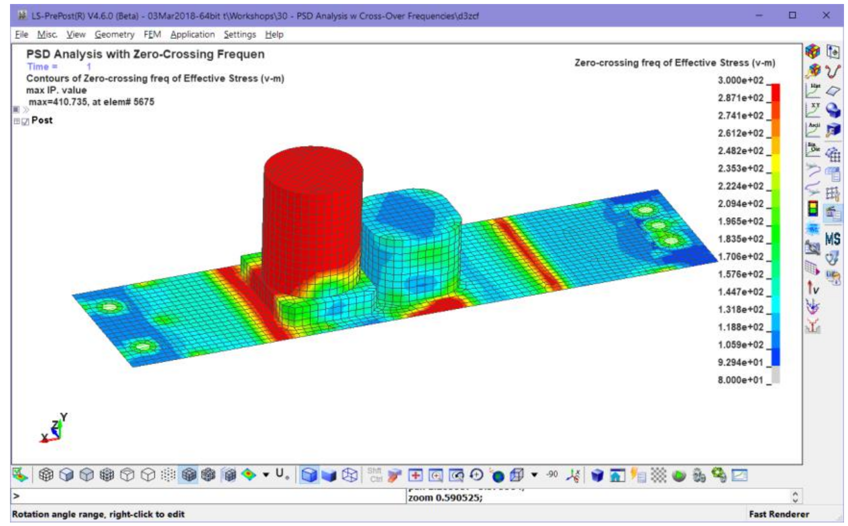
19.12.3.1.1 PSD Zero-Crossing Frequencies

Zero-crossing frequencies represent the RMS frequency at which the RMS stress occurs at that location. It is a great tool for debugging your model and understanding what frequencies can harm your structure. The PSD RMS plot shows the average one sigma stress in the structure but it doesn't indicate what dominant frequency caused that damage.

Normal Modes



Zero-Crossing Frequencies and Modal Effective Mass



MODAL EFFECTIVE MASS

MODE	X-TRAN Eff. Mass	X-TRAN Accum. %	Y-TRAN Eff. Mass	Y-TRAN Accum. %	Z-TRAN Eff. Mass	Z-TRAN Accum. %
1	7.649743E-07	1.03%	6.664397E-05	89.61%	1.820908E-09	0.00%
2	6.333918E-11	1.03%	3.271119E-09	89.61%	3.722702E-05	50.06%
3	2.378791E-05	33.01%	2.804888E-06	93.38%	4.381928E-10	50.06%
4	7.637307E-08	33.12%	2.635044E-06	96.93%	5.899920E-11	50.06%
5	1.056079E-09	33.12%	6.829478E-10	96.93%	3.248503E-06	54.42%
6	1.754176E-05	56.70%	7.669717E-08	97.03%	1.190403E-09	54.43%
7	3.709131E-10	56.70%	1.829353E-12	97.03%	4.499370E-06	60.48%
8	2.485563E-09	56.71%	1.072811E-11	97.03%	1.694850E-05	83.26%

20. IMPLICIT MULTI-PHYSICS: COUPLED THERMAL-STRESS ANALYSIS

This section is a rough draft. The idea is to start with this simple problem and then over time, add additional solution sequences. The mechanics of a nonlinear coupled thermal-stress solution within LS-DYNA is a natural fit. In LS-DYNA, one can use functions (*DEFINE_CURVE) to apply temperature dependent (nonlinear) material properties for elastic modulus, yield stress, coefficient of thermal expansion (CTE) and I'm sure other properties that are buried within the Keyword Manual. Basically, LS-DYNA can solve almost anything in the realm of solid thermal mechanics (conduction and convection from surfaces).

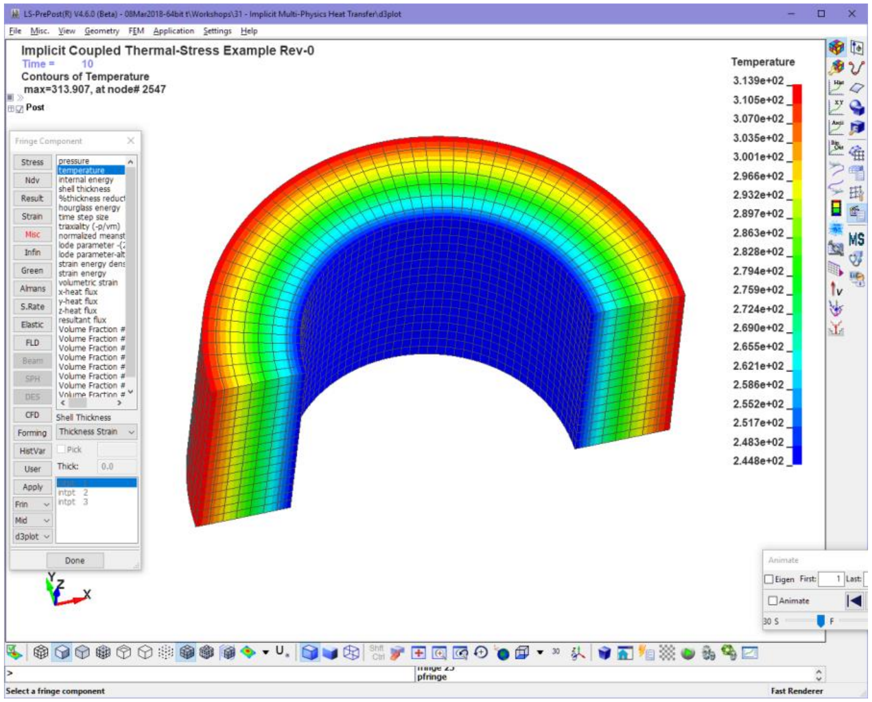
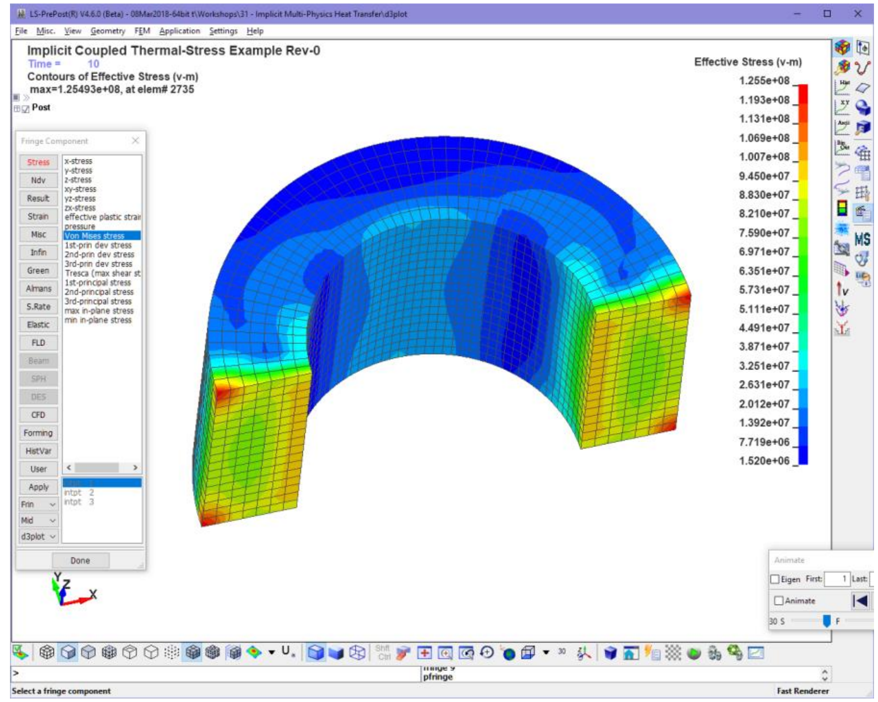
20.1.1.1 Getting Started with Coupled Thermal-Stress Analysis

For this first example, the model couples a convective (heat transfer film coefficients) on the inner and out surfaces to create a temperature gradient. At the end of the transient thermal solution, the temperature gradient is used as a thermal load input to drive the implicit static stress analysis. Stresses are generated based on a constant CTE. The table below provides a brief discussion on the keywords used in the following example.

Keyword	Variable	Discussion
*CONTROL_SOLUTION	*soln*=2	Coupled Structural Thermal Analysis
*CONTROL_THERMAL_SOLVER	*atype*=1	Indicates whether the thermal solution will be state (atype=0) or transient (atype=1)
*CONTROL_THERMAL_TIMESTEP	*ts*=0, *tip*=1 and *its*=0.1	Since the analysis is transient, one needs to set a timestep (its=0.1). Guidance on setting the time step is beyond this little introduction and of course, one should RTM and then verify your model against a standard formula when in doubt.
*MAT_THERMAL_ISOTROPIC		A coupled structural thermal analysis requires thermal and mechanical properties. This can be confusing at first look. One needs to define the thermal characteristics, then add in thermal expansion and finally, the mechanical response to thermal-strain.
*MAT_ADD_THERMAL_EXPANSION		
*MAT_ELASTIC		
Analyst Note: There is tremondous capability within LS-DYNA for handling coupled thermal-stress anlaysis and is only limited by your ability to dig through the Keyword Manual.		

20.1.2 Workshop 31: Couple Thermal-Stress Analysis

The workshop is to inspect the file, read the Keyword Manual and run it. If you can arrive at these plots, that is it. To explore, change the Keywords and see what you get. This section is rough and more to follow.

Temperature Field (Fringe Component / Misc / Temperature)	Stress Field (Fringe Component / Stress / Von Mises Stress)
	

20.2 Implicit Check-Out and Recommendations

Implicit analyses are generally much more difficult to obtain convergence and that is often the downfall of its usage.

For implicit analyses used for the initialization of explicit runs, it is recommended to use the *CONTACT_ birth option (BT) to turn off all non-essential contacts until the start of the explicit run. Likewise use *BOUNDARY_SPC_NODE (or SET)_BIRTH_DEATH to lock down any parts of the structure that are not relevant to the implicit initialization. These two steps will greatly facility a fast and efficient implicit kick-off.

It is also a default “must” recommendation to switch all *CONTACTS_ to the _MORTAR formulation. For more details on the usage of _MORTAR contact, please see the DYNAmore notes in the Class Reference Notes / Implicit Analysis section.

For troubleshooting LS-DYNA implicit analyses don’t be shy about locking down (SPC’ing) large parts of your structure and ripping out contacts and nonlinear material laws. Once you have something running, it is a lot easier to add in complexity step-by-step-by-step than struggle with behemoth that is taking 30 minutes to finally error out.

A little thing to note is that LS-DYNA has near identical element formulations to that of standard implicit codes but that the standard stress contouring within LSPP is via the integration points and not via analysis code extrapolation to the nodal points (i.e., Nastran). The default setting is to only report stresses at the centroid of each element or in explicit mode, most elements are one-point integrated elements and thus this default makes perfect since from the standpoint of numerical and data storage efficiency. In implicit, the default is fully-integrated elements. Hence, one must request that integration point stresses be obtained (*DATABASE_BINARY_EXTENT / *nintsld*=8) and then use _OUTPUT / *solsig*=2. For shells, one uses the LSPP option of Extrapolate 1 (command line window at the bottom left-hand corner within LSPP).

RBE2’s (Nastran Only Concept) to CNRB: One thing to note is that an implicit analysis does not like anything less than all six DOF’s enabled for CNRB’s.

20.2.1 Model Construction Recommendations

- Build contact regions with a very small interference (e.g., 0.01% of characteristic element length of your model). With this small interference one can use LSPP via Application / Model Checking / General Checking / Contact Check to verify that your contact is “contacting” albeit with a very small interference which one may considered insignificance since one has set IGNORE=1 (or another appropriate value) on the contact card to track small interpenetrations. In this manner, one can verify 100% that contact occurring between the desired parts.

20.2.2 Implicit Keyword Cards and Recommendations

Of course, your LS-DYNA solver for all implicit work should be the latest MPP, Double-Precision Development version downloaded fresh off of the LSTC website. Our Keyword Card philosophy is that one should only modify the default settings based on direct, verifiable experience (i.e., a pilot model or a side-by-side comparison) or a written recommendation that is applicable to the model at hand. The temptation to modify card fields is ever present in the wild hope of the right touch but the reality is that your odds are more akin to Las Vegas than engineering practice.

*CONTACT_ -- _MORTAR	This algorithm was essentially developed for implicit analyses	
*CONTACT_TIED_	The new global recommendation (see Keyword Manual Vol 1, Appendix P) is that one should be able to meet all tied requirements with _TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET. At the end of the day, one should avoid adding additional springs (_BEAM_) to any FEA model if it can be avoided. If rigid bodies are present one can add the Card E option *ipback*=1.	
*CONTROL_IMPLICIT_ACCURACY	*iacc*=1	Standard setting
*CONTROL_IMPLICIT_AUTO	*iauto*=1 & *dtmax*=(?)	dtmax can be a curve (negative curve number) to set a fixed interval of solution steps and likewise outputs if the corresponding *DATABASE_BINARY_D3PLOT, *dt* value is set at a value to capture the solution steps. For example, if *dtmax* is 0.01 and a d3plot is desired, then *dt* ≤ 0.01.
*CONTROL_IMPLICIT_DYNAMICS	*gamma*=0.60 & *beta*=0.38	With *gamma*=0.60 and *beta*=0.38 you get quasi-dynamic behavior with lots of damping (see DYNAmore notes). This is a very effective technique to handle mechanisms in your structure at the start of the analysis run or progressive damage analysis in composites. At the end of this quasi-static analysis, one can plot the internal versus kinetic energies to verify that the run has stabilized (i.e., low kinetic energy). *For regular dynamics, set gamma and beta to default values or you will "damp" your solution!*
*CONTROL_IMPLICT_EIGENVALUE	*neig*=(?)	*neig* is the number of Eigenvalues (normal modes) that are to be requested. For an intermittent Eigenvalue analysis while performing an implicit or explicit analysis, one can use a curve number (with a negative sign preceding the curve number).
*CONTROL_IMPLICIT_GENERAL	*imflag*=1 and *dto*=()	*imflag* to get your analysis defined as implicit and then a starting time step value to kick things off.
*CONTROL_IMPLICIT_SOLUTION	*nsolvr*=12 (Default) *abstol*=1e-20, *dnorm*=1, *nlprint*=2 & *nlnorm*=4	See DYNAmore's implicit notes for the full story. *nlprint*=2 for convergence information. *nlnorm*=4 treats translation and rotation equally and is recommended (RTM) or even better negative numbers based on the model units maybe used as in *nlnorm*=-1 (mm) or =-0.03937 (inch)

*CONTROL_IMPLICIT_SOLVER	*rdcmem* < 0.5	In 2021, the LS-DYNA solver will consume all of your system's memory under Windows 10. To avoid memory conflicts with other running programs or just to run simultaneous analyses, one needs to choke off LS-DYNA's access to your system memory. Your mileage may vary but a value of *rdcmem*=0.10 often works well for large problems on our systems with > 64 GBytes of RAM.
*CONTROL_OUTPUT	*solsig*=1 (*MAT_ELASTIC) or *solsig*=2 (nonlinear materials) *tet10s8*=1 (10-node tets)	This command will correctly extrapolate the integration point stresses (see manual) for solid elements. For linear elastic stress analysis, the stresses will align well with standard Nastran type codes. If you use this option, please see *DATABASE_EXTENT_BINARY note about nintsld=8. And if 10-node tets are used, one might want to include the midside nodes into the d3plot database to display their displacements (*tet10s8*=1).
*CONTROL_SHELL	*esort*=2 & *RTM*	If one has mixed meshes of quadrilaterals and triangles, then one should set *esort*=2 to switch *elform*=-16 triangles to *elform*=17. Other _SHELL settings should be reviewed for Lobatto integration or composite usage.
*SECTION_BEAM	*elform*=4	QR=4 for linear analysis (Lobatto) and QR=5 when material plasticity is simulated.
*SECTION_SHELL	*elform*=21 (linear) *elform*=-16 (nonlinear) *nips*=5	For linear analysis, *elform*=21 (with nsolvr=1) with *nip*=3 and *CONTROL_SHELL, intgrd=1 for Lobatto integration. While for everything else use *elform*=-16 /16 with *nip*=5 with default Gauss integration. Note: Volvo has standardized on 16 / 5.
*SECTION_SOLID	*elform*=-18 (hex) linear (*nsolvr*=1) and *elform*=-18 (hex) nonlinear (*nsolvr*=12) *elform*=-2 (hex) (nonlinear) *elform*=16 (10-node tet) *elform*=13 (4-node tet)	For hex elements, linear analysis is *elform*=18 (with *nsolvr*=1) while for nonlinear it is *elform*=-18. If tetrahedrals are used, then it is *elform*=16 for 10-node tets and *elform*=13 for 4-node tets, and of course, default *nsolvr*=12.
*DATABASE_EXTENT_BINARY	*maxint*=-3	A negative number dumps out all integration point stresses for fully-integrated shells (e.g., *elform*=16)
	beamip=>0	>0 toggles *beamip* to write out all beam integration points
	nintsld=8	This dumps out all the integration point stress data for solid elements. Very useful for implicit work if you would like to something that approaches a normal linear stress result.

20.2.3 Convergence Troubleshooting and Solution Speed Optimization

20.2.4 General Troubleshooting

- If singularities exist – just kill'em by using *CONTROL_IMPLICIT_DYNAMICS or SPC sections of your model. Go big and once it converges you can restore sections of the model (remove the SPC's);
- Be careful with loading – sometimes your initial load can blow-up your model (very high deformation or if parts are unconstrained they can shoot off like a bullet);
- Mesh quality at start and during analysis. Don't get in a rush and forget to check your mesh quality (e.g., Jacobian, mesh aspect ratio, time step);
- The convergence criteria are treated as individual criterion, that is, if one is met the solution is considered converged. Thus, although you specify criteria (*dctol*, *ectol*, *rctol* (turned off by default by using a large number (1e10) and *abstol*) however, convergence is approved if any individual criterion is met – which implies a convergence criterion. How do you know if you have "convergence" that is significant to your problem? There is no way to know except engineering judgement. The reality is that one should just crank down the tolerances (recommendation to put *abstol*=1e-20 and then drop dctol from default of 1e-3 to 1e-5 and set *lsmtd*=5) and run it over the weekend.

20.2.5 Convergence – How to Find It

What do we use for general convergence? On the *CONTROL_IMPLICIT_SOLUTION card we'll set *abstol*=1e-20, *dnorm*=1, *nlprint*=2 or 3 (if you really need to debug), *dnorm*=1 and *nlnorm*=4. That is it. As for trouble shooting, there is little value is listing all the tricks when there is an excellent paper within Class Reference Notes / Implicit Analysis / LSTC-DYNAmore Implicit Class at Dearborn 2018 - Must Read for New Implicit Engineers / Nonlinear Implicit Analysis in LS-DYNA - Thomas Borrvall.pdf It is my go to reference and is well worth several reads since it contains so much useful information.

When should you iterate and when should you do a full on reformulation of the stiffness matrix (*CONTROL_IMPLICIT_SOLUTION, *ilimit*=1)? It depends on fast the residual is decreasing. The image below shows a good candidate to just do a full-on reformulation (*ilimit*=1).

***CONTROL_IMPLICIT_SOLUTION, *nlprint*=3**

```
Iteration:    5    displacement          energy              residual
 ----------          not conv.          converged            converged
   norm ratio        2.952E-02          8.742E-04               n/a
 current norm        7.209E-03          5.894E-03            2.197E+02
 initial norm        2.442E-01          6.742E+00            6.162E+02

ITERATION LIMIT reached, automatically REFORMING stiffness matrix...

Iteration:    6    displacement          energy              residual
 ----------          not conv.          converged            converged
   norm ratio        2.407E-04          4.364E-07               n/a
 current norm        5.879E-05          2.942E-06            7.832E+01
 initial norm        2.442E-01          6.742E+00            6.162E+02
```

20.2.6 D3ITER PLOT DATABASE TO TROUBLESHOOT ABNORMAL DISPLACEMENTS

This is my most powerful tool to debug non-convergence in implicit. When one plots the total displacement in LSPP, it will indicate regions in your model that are flying off into space or causing problems.

D3Iter Plot Database

To diagnose convergence trouble which develops in the middle of a simulation, get a picture of the deformed mesh. Adjust the d3plot output interval to produce an output state after every step leading up to the problematic time. An additional binary plot database named "d3iter" is available which shows the deformed mesh during each equilibrium iteration. This output is activated by

D3ITCTL = 1 to activate d3iter plot database

on *CONTROL_IMPLICIT_SOLUTION. View this database using LS-PrePost to detect abnormal displacements. The problem may become obvious, especially as deformation is magnified. If not, there is yet another flag to activate to get the residual forces into both this database as well as d3plot for fringing. Setting

RESPLT = 1 to get residual data to binary databases

on *DATABASE_EXTENT_BINARY will do just that. With this option the residual forces are output to the d3plot and d3iter databases for fringing under the "NdV" menu. This is a great tool for locating areas in the model where the residual forces are not being reduced to a satisfactory level and take appropriate actions.

20.2.7 DON'T FORGET ABOUT THE IMPLICIT TIME STEP FOR TRANSIENT, DYNAMIC ANALYSES

From the LS-DYNA Keyword Manual Vol 1, *DATABASE_GLSTAT:

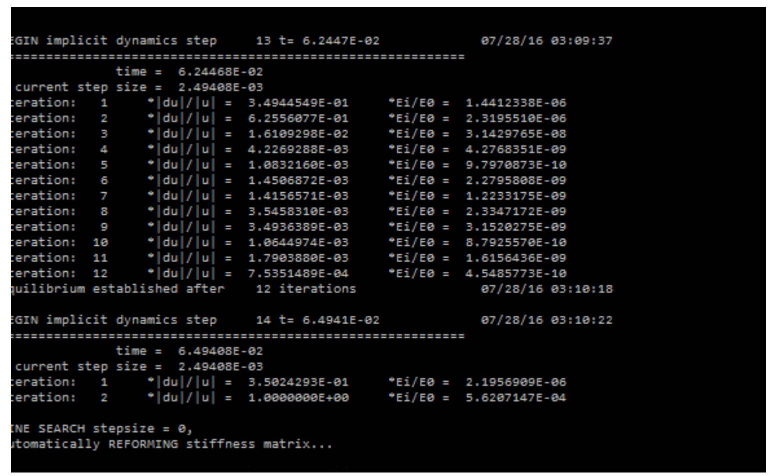
† For implicit, integration errors due to large time steps will result in inaccurate estimates of kinetic and internal energies, regardless of how tight the convergence tolerances are. The "lost" energy arising from this discretization error is accumulated in the eroded kinetic and internal energies, respectively, to render energy balance in the sense described in the introduction. Energy balance in implicit is an implication of having solved the implicit problem to sufficient degree of accuracy, and even though the opposite may not be true it can still be used as an indicator; energy balance *likely* implies a good solution while poor energy balance *definitely* implies a less accurate one.

20.2.8 Comments on LS-DYNA Output Messages and Their Significance

***|du/|u| -** If this value stays pegged at 1.0 then there is a good chance that you have something moving in your structure. Even with _IMPLICIT_DYNAMICS turned on, if something is moving quickly in your model, this value will often stay at 1.00 or maybe drop to 0.9 or 0.8 but quickly bounce back to 1.00. The solution is to kill the analysis and start hunting for something that is un-restrained or not connected or has a missing contact.

***Ei/E0** – If this value stays high (e.g., 1e-4) then you have something that is deforming massively (energy = force*displacement) or poorly formed elements or something that is not behaving in a physical manner (e.g., wrong choice of material law). When this value stays high, the debugging can be difficult. Here's one suggested short list:

- Check element quality (explicit time step & Jacobian)
- Check material formulation and curves that are used to define these materials
- Check CNRB's (implicit only supports full 6-DOF formulation as of this writing)
- Check loads



```
EGIN implicit dynamics step     13 t= 6.2447E-02            07/28/16 03:09:37
=====================================================================
             time =  6.24468E-02
 current step size =  2.49408E-03
eration:   1     *|du|/|u| =  3.4944549E-01     *Ei/E0 =  1.4412338E-06
eration:   2     *|du|/|u| =  6.2556077E-01     *Ei/E0 =  2.3195510E-06
eration:   3     *|du|/|u| =  1.6109298E-02     *Ei/E0 =  3.1429765E-08
eration:   4     *|du|/|u| =  4.2269288E-03     *Ei/E0 =  4.2768351E-09
eration:   5     *|du|/|u| =  1.0832160E-03     *Ei/E0 =  9.7970873E-10
eration:   6     *|du|/|u| =  1.4506872E-03     *Ei/E0 =  2.2795808E-09
eration:   7     *|du|/|u| =  1.4156571E-03     *Ei/E0 =  1.2233175E-09
eration:   8     *|du|/|u| =  3.5458310E-03     *Ei/E0 =  2.3347172E-09
eration:   9     *|du|/|u| =  3.4936389E-03     *Ei/E0 =  3.1520275E-09
eration:  10     *|du|/|u| =  1.0644974E-03     *Ei/E0 =  8.7925570E-10
eration:  11     *|du|/|u| =  1.7903880E-03     *Ei/E0 =  1.6156436E-09
eration:  12     *|du|/|u| =  7.5351489E-04     *Ei/E0 =  4.5485773E-10
quilibrium established after    12 iterations               07/28/16 03:10:18

EGIN implicit dynamics step     14 t= 6.4941E-02            07/28/16 03:10:22
=====================================================================
             time =  6.49408E-02
 current step size =  2.49408E-03
eration:   1     *|du|/|u| =  3.5024293E-01     *Ei/E0 =  2.1956909E-06
eration:   2     *|du|/|u| =  1.0000000E+00     *Ei/E0 =  5.6207147E-04

INE SEARCH stepsize = 0,
utomatically REFORMING stiffness matrix...
```

21. TROUBLESHOOTING IMPLICIT ANALYSES

If you made it this far through the notes, you are truly a dedicated student of LS-DYNA. Prior to tackling this section, it is my hard-core recommendation that one should read Appendix P: Implicit Solver in Keyword User's Manual Volume I. As for myself, I think I have read it more times than I want to admit in public.

21.1 Explicit Always Runs While Implicit Rarely Runs: Why?

As worth repeating, explicit never has to converge since it has all that it needs from the prior wavefront while implicit must search forward in its quest to reduce the residual or converge to an acceptable engineering accuracy.

$Explicit{:}\ \ a^{n+1} = f(d^n, v^n, a^n,\ d^{n-1},\ v^{n-1}, \dots)$	$Implicit{:}\ \ d^{n+1} = f(v^{n+1}, a^{n+1},\ d^n,\ v^n, \dots.)$
(*nothing here to look at…*)	$ma^n + cv^n + kd^n - f^n = Residual$
Explicit has all it needs to jump forward.	Implicit needs to "converge" to find what it needs to jump forward.

21.1.1 What is the Residual?

Mathematically, the residual is the out-of-balance force between the current state and the future state where the "loading" has changed. One has to keep in mind that in the future state or n+1, the loading has changed, whether it is f^{N+1}, or v^{N+1} or a^{n+1} or etc., that creates a force imbalance between the current state and future state. In a static analysis, this force imbalance is addressed by modifying the stiffness matrix whereas in a dynamic analysis, one has to address the whole equation of motion (EOM). To gain insight into this residual reduction process or convergence, LS-DYNA provides two run-time indications given as ***|du/|u|** (displacement norm) and ***Ei/E0** (energy norm). It is these indicators that we seek to lower to some tolerance that we deem acceptable, which in most cases is just the default value.

The most important indicator of convergence is ***|du/|u|**. Let's go back to something simple and remember the basis for linear, elastic static stress analysis:

$[K]\{u\} = \{F\}$ but in nonlinear mechanics we always have a residual and that is in the form of: $[K]\Delta\{u\} = \{R\}$ where the stiffness matrix is updated yielding an updated displacement increment (Δu) and an out-of-balance load or residual force R.

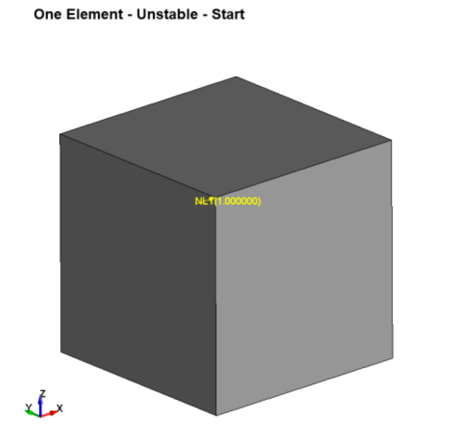
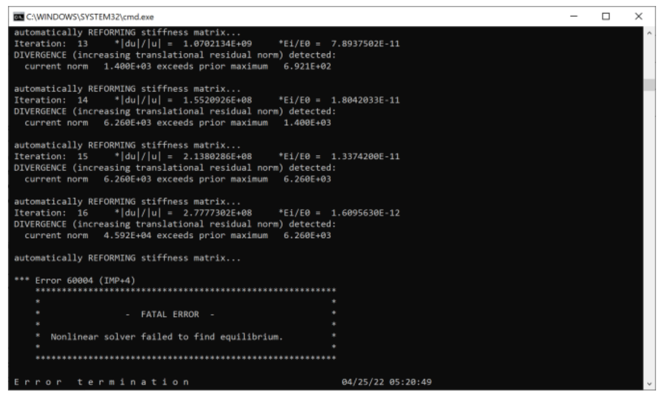
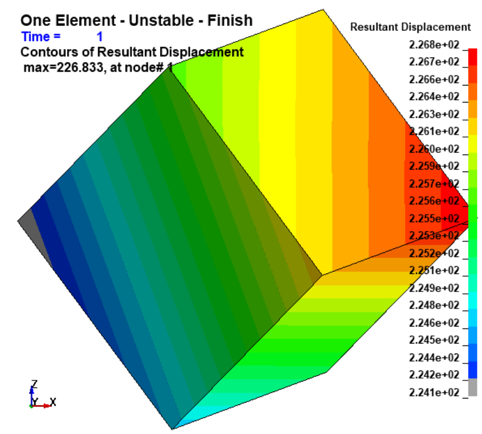
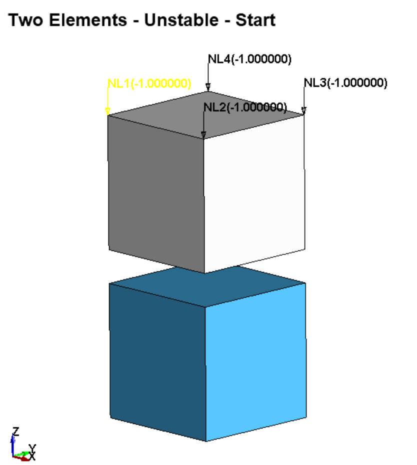
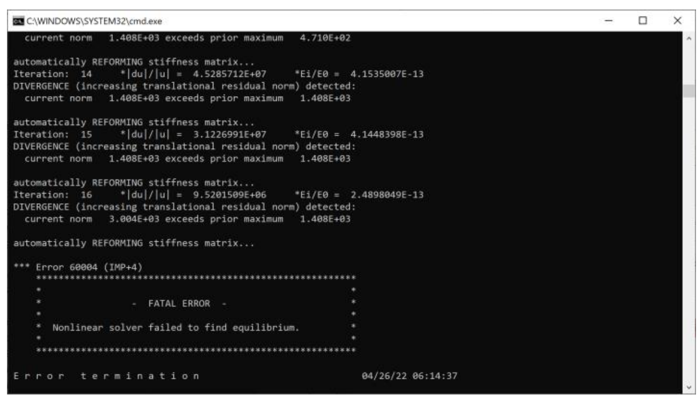
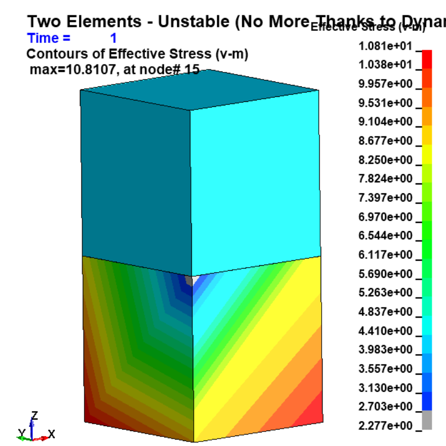
LS-DYNA likes to use three convergence indicators but the reality, it all boils down to reducing the residual force {R}:

$$d = \|\Delta u\|$$
$$residual\ force = \|R\|$$
$$energy\ norm\ (e) = [R^T \Delta u]$$

Analyst's Note: We strive to arrive at zero residual force *but that is an impossible dream; however, we strive to lower it to an acceptable value and the road to that destination passes thru lowering the Δu or* ****|du/|u|***.

21.1.1.1 Workshop: The Basics of Convergence – Displacement Norm *|du/|u|

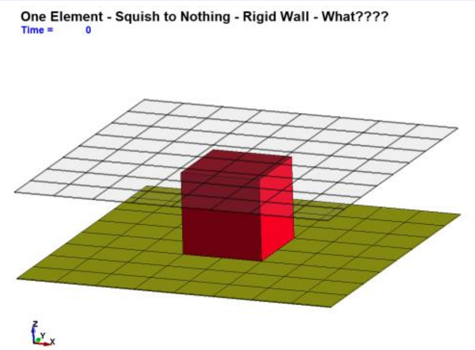
Let's do a confidence builder. It is a one element model where a force is applied to a node. The model is unconstrained. Since it is unconstrained, there is no hope that it will run. The workshop is as simple as applying *CONTROL_IMPLICIT_DYNAMICS and interrogating the results. The takeaway is that the displacement norm can easily indicate if the model is un-constrained since it will have a super high value.

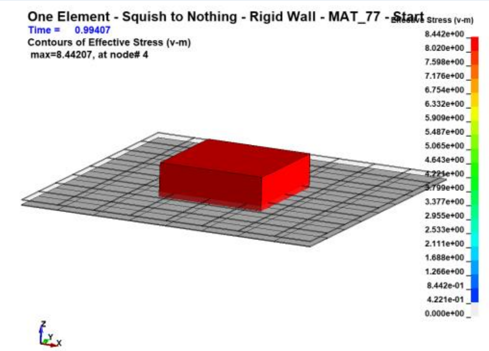
<table>
<tr><th>One Element – No Constraints</th><th>*|du/|u| Blowing Up!</th><th>*CONTROL_IMPLICIT_DYNAMICS</th></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
<tr><th>Two Blocks – Contact Enforced</th><th>*|du/|u| Blowing Up!</th><th>*CONTROL_IMPLICIT_DYNAMICS</th></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
</table>

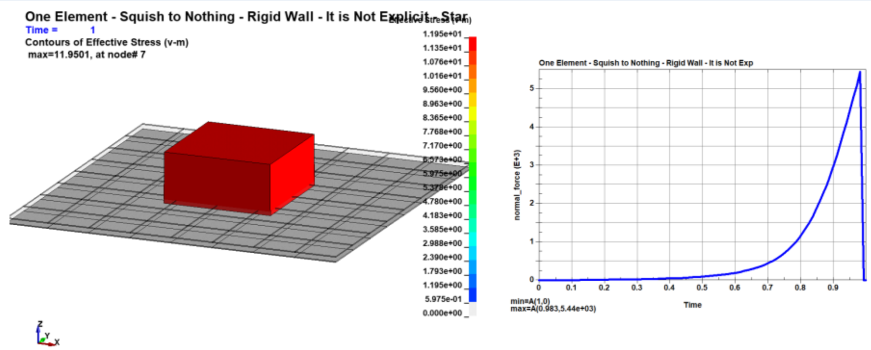
21.2 Why Is Implicit Such a Bad Boy?

One gets relaxed with explicit since it always runs and with this mind set, the expectation is that implicit should behave similarly always run! This series of models all have start and finish Keyword Decks. The Decks are commented with my approach. Sometimes it is just better to read and run then offer a long-written monolog.

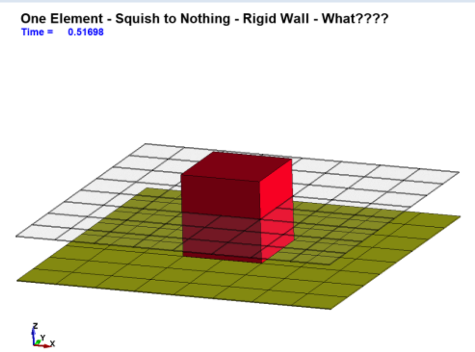
What?	Maybe Material Model?	Implicit Ain't Explicit

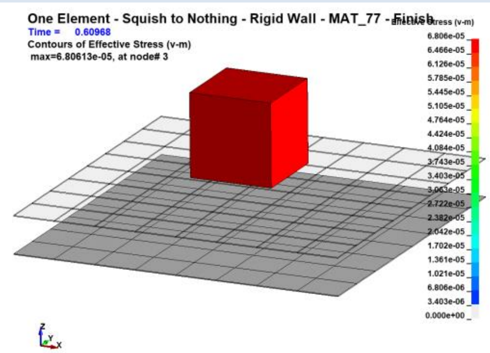


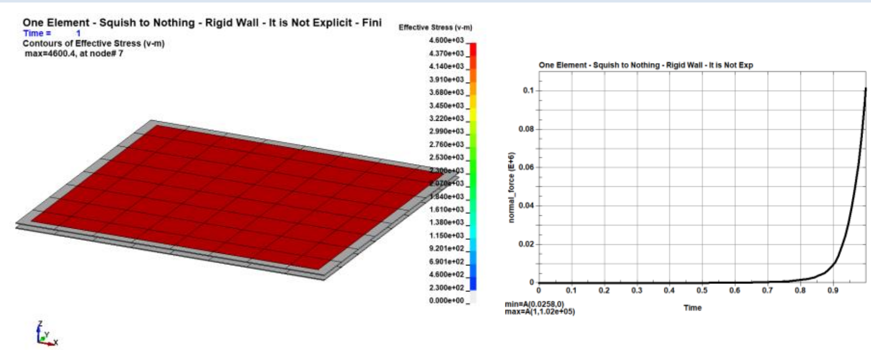




Time Step = Load Step	Let's Try *MAT_77	To get it to Run – Just Look at the Finish Deck







*Analyst's Note: Nonlinear static and nonlinear transient dynamic implicit is the most difficult of all finite element modeling. This model was solved by thinking about how implicit must be handling the contact formulation. Whereas explicit treats "rigidwalls" but setting the nodal velocities to zero upon contact, this approach doesn't work for implicit. Therefore, implicit must rely upon the penalty method and put "springs" between the rigidwall and the deformable material. Since springs are involved, there must be a way to scale their stiffness. Hence, the hunt to look within *CONTROL_CONTACT.*

21.3 Implicit Stability Diagnostics

To be done someday…..

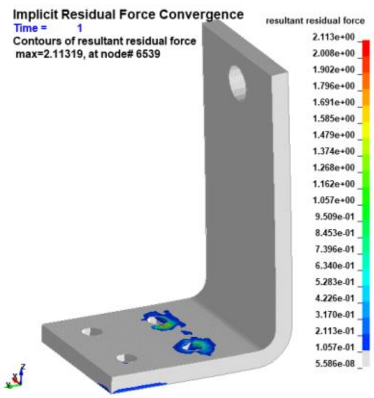
21.4 Implicit Residual Force Convergence

Given that the objective in nonlinear implicit analysis is to reduce the residual force (R) and that a lower value is better, the default settings (albeit with *abstol* = 1.0E-20) generates equivalent results as that with a much tighter tolerance.

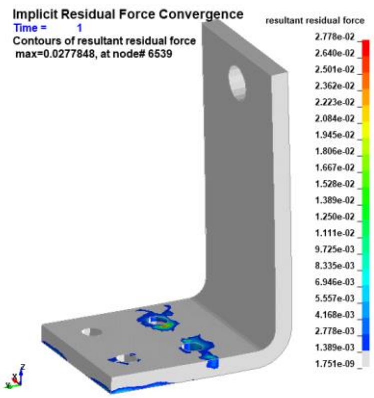
A recommended debugging technique by others (see Appendix P: Implicit Solver) is to contour the residual force by setting *DATABASE_EXTENT_BINARY, *resplt*=1, its usefulness depends on the problem being solved.

But nothing beats *Displacement Control* (i.e., loads as displacement).

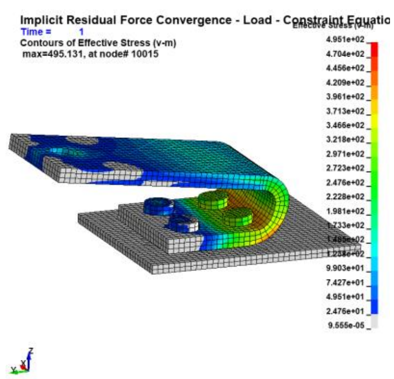
dctol = 1.0E-3 / Residual 2.1



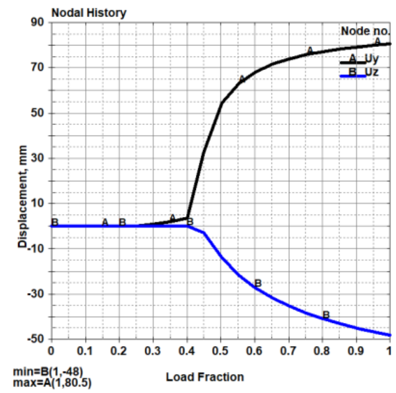
dctol = 1.0E-6 / Residual 0.028



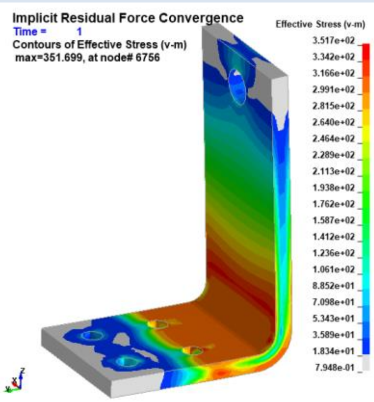
Constraint Equation – 16 min



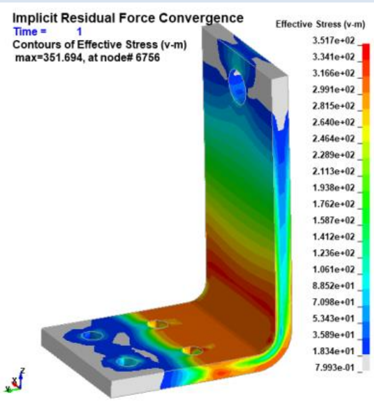
Displacement of Load Point



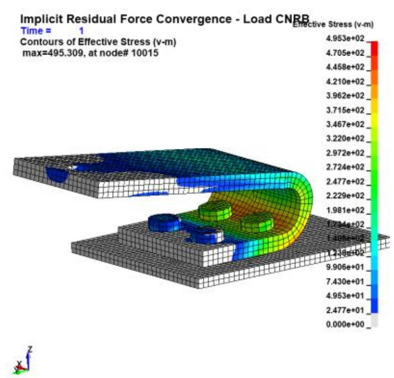
Stress Max. = 352



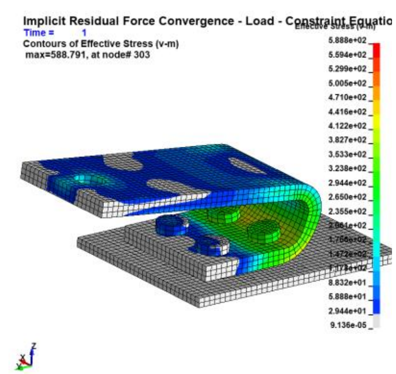
Stress Max. = 352



High Load – CNRB 17 min



Displacement Control – 6 min



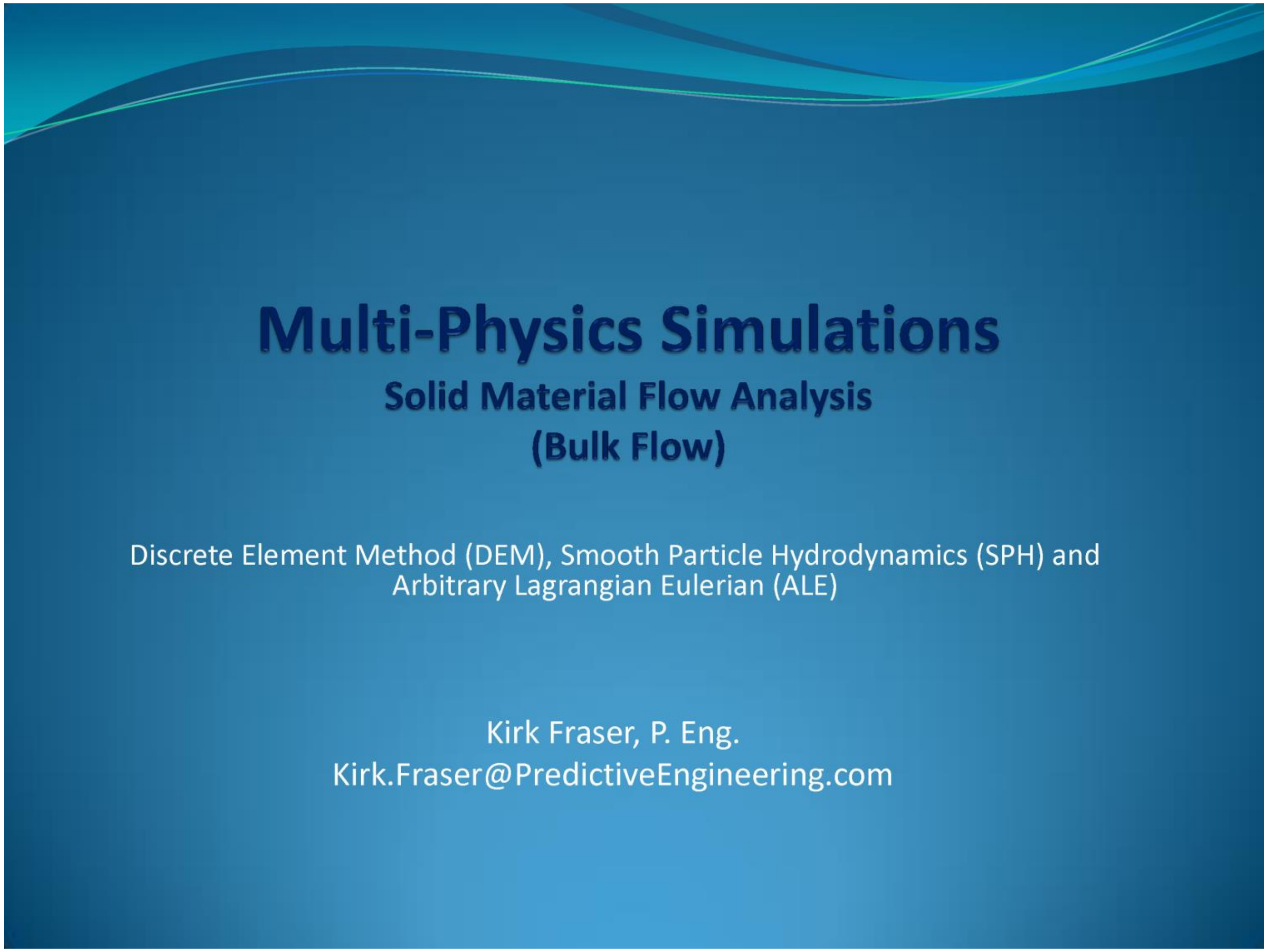
21.4.1 But Can it Go Faster?

A common strategy for nonlinear implicit analyses with minor nonlinearity is to force the solution to not “cut” the load step if convergence problems are encountered. The idea is that during loading, the solution might encounter a difficult spot and automatically cut the load step (*CONTROL_IMPLICIT_AUTO). To recover from this decrease in load step may require many un-necessary iterations. Hence, a suggestion is to set *iteopt* = 200. This is a mixed bag. Sometimes it works great sometimes not-so-much. For example, our prior Displacement Control model with *iteopt* = 200 caused its run time to ***increase*** from 6 to 13 min. At the same time, I have seen this trick work very effectively on problems with mild nonlinearity!

21.5 Deep Dive Into Model Convergence

LS-DYNA lets you peak into the model as it is trying to converge. To see what is happening within a convergence iteration, one sets *CONTROL_IMPLICIT_SOLUTION, d3itctl > 0 (depending on how much information you want) and in *DATABASE_EXTENT_BINARY, *resplt*=1.

22. DISCRETE ELEMENT METHOD



See Class Reference Notes / DEM / Predictive Engineering Discussion of LS-DYNA Meshfree Methods.pptx

23. FLUID STRUCTURE INTERACTION AND MULTI-PHYSICS IN LS-DYNA

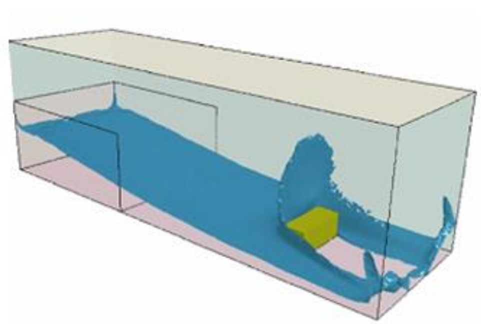
{A bit dusty but still functional for the curious}

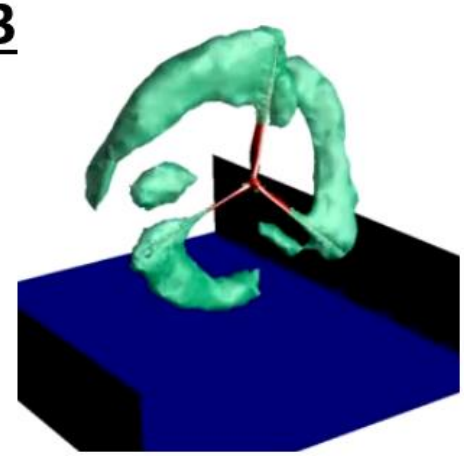
Fluid Structure Interaction with LS-DYNA Multiphysics

Finite Element Analysis Predictive Engineering

LSTC Livermore Software Technology Corporation

August 2013





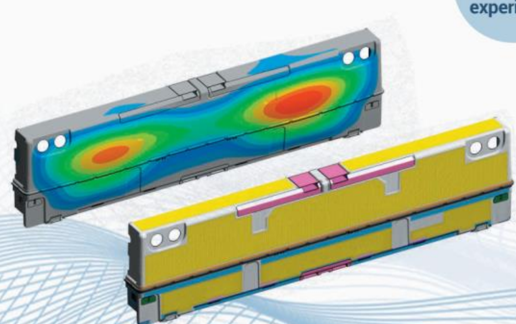
See Class Reference Notes / Multi-Physics / LS-DYNA Multi-Physics.ppsx

Finite Element Analysis
Predictive Engineering

Finite element analysis consulting services, software, training and technical support.

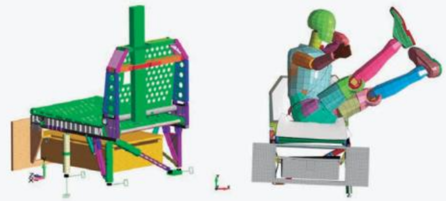
- Composites, Pressure Vessels, Vibration.
- **NASTRAN:** Linear Dynamics.
- **LS-DYNA:** Drop-test, Impact, Burst Analysis.
- **STAR-CCM+:** CFD Thermal/Flow Analysis.

+20 years experience

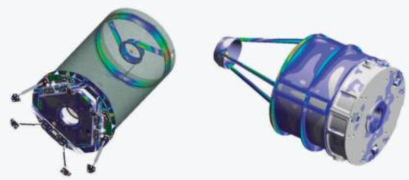


Project Examples

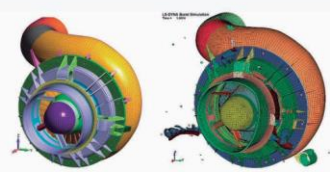
FAA 16G SLED TEST VERIFICATION



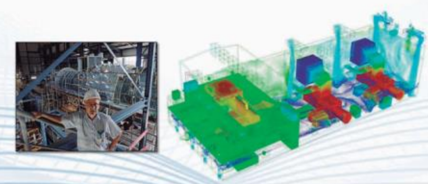
STRESS AND VIBRATION ANALYSIS OF SATELLITES



LS-DYNA TURBINE BURST SIMULATION



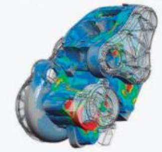
CFD STUDY ON CO-GENERATION POWER PLANT BUILDING



Our Services

FEA

Predictive Engineering brings to bear more than 20 years of finite element analysis FEA consulting experience in solving the most difficult mechanical engineering analysis challenges.
Our validated experience ranges from transmissions to submarines to satellites.

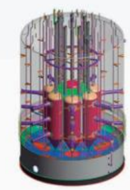


TRANSIENT NONLINEAR

At Predictive Engineering, we pride ourselves on the ability to idealize complex structures and systems into predictive numerical models. Our nonlinear, static and transient dynamic work has been validated against strain-gauges, drop and sled test results, accelerometers, crack growth and fracture and in extreme service environments.

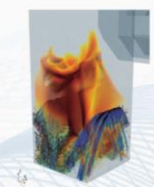
ASME-BPVC

From seismic to buckling to cyclic service (fatigue), Predictive can assist in verifying the most challenging pressure vessel designs. Our hard-earned experience allows us to safely classify tanks and vessels as "fit-for-service" that would typically have required extensive redesign or re-work.



CFD

Our expertise in computational fluid dynamics (CFD) comes from hundreds of thermal-fluid projects in medical, aerospace, marine, HVAC (data centers), civil and automotive. This experience gives us the capability to quickly optimize and provide accurate digital prototypes.



Your comments would be welcomed

On a scale of 1 to 5, where "1" means not satisfactory and a "5" indicates that it was very satisfactory.

How were the class notes and the workshops?	1 2 3 4 5
Did the instructor do a good job in presenting the material?	1 2 3 4 5
Was the pace of the class adequate to learn the material?	1 2 3 4 5
Quality of the experience?	1 2 3 4 5

If you could do one or two things to make it better, what would they be?

General Comments?

When done just tear out this sheet and leave it at your desk. Thank you.

Analyst's Note: If you are taking this class online, it gets tricky to protect your anonymity (if so desired) and I have no real idea on how to do it….but I would value your opinion, positive or negative. This class is a work in progress and it gets better in a large part to your input. So I can take a few hits and it'll just make the class better for the next student. So take a picture and email to Training@PredictiveEngineering.com