Powering Innovation That Drives Human Advancement

Driving Pre & Post processing of LS-DYNA with LS-PrePost and PyDyna

Philip Ho & LS-PrePost Team

May 20, 2025

Outlines

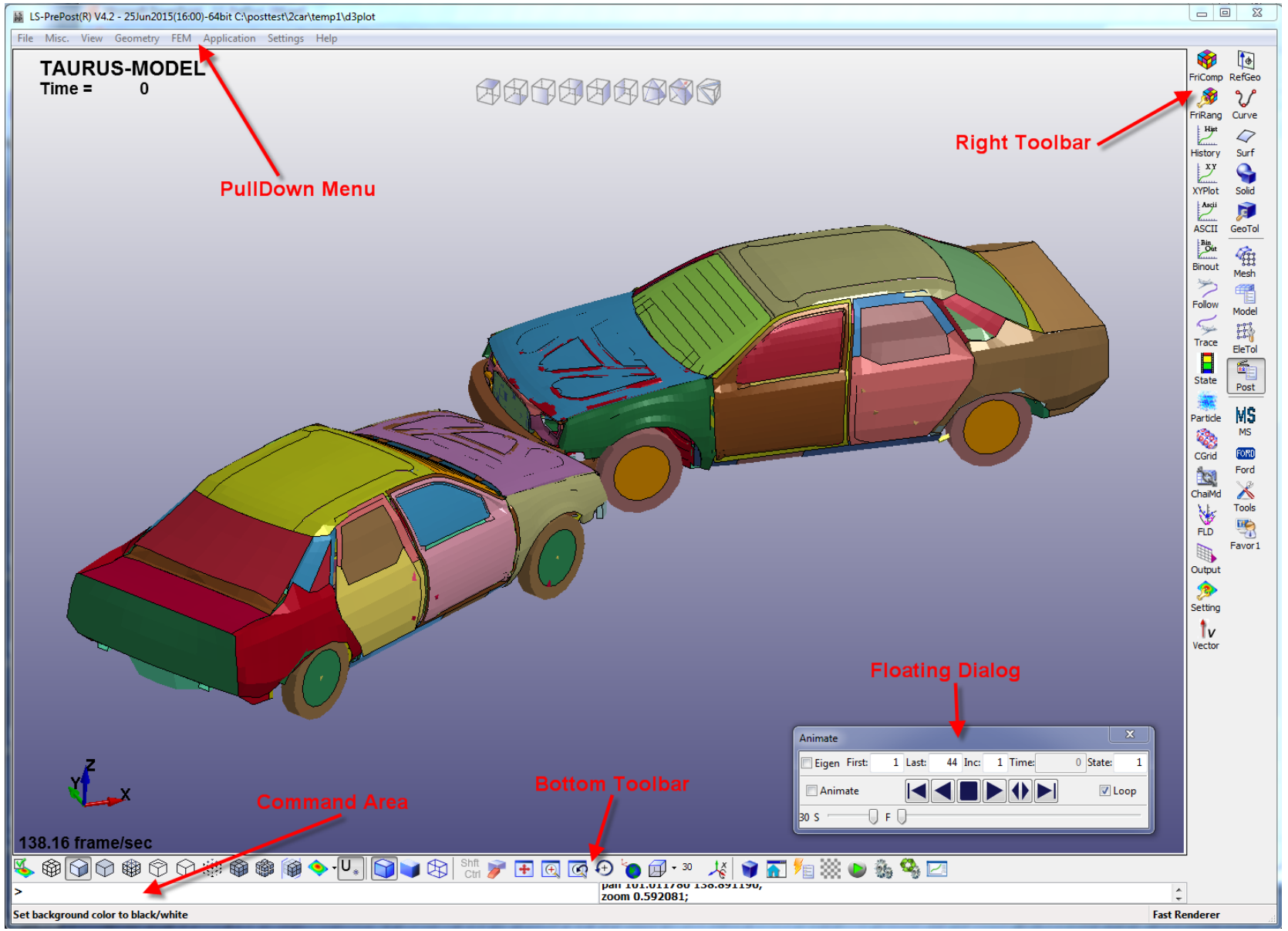
- Overview of LS-PrePost
- Solution Explorer
- Post-Explorer
- PyDyna
- PyDFP

LS-PrePost Overview

- LS-PrePost is an advanced pre and post-processor designed for LS-DYNA and is delivered free with LS-DYNA
- LS-PrePost does not need LS-DYNA license key, it can be freely download and loaded onto any computer that meets the hardware requirements to run it
- Supported systems are Windows 10/11; 64bit Linux (Centos 7/8, Opensuse 15, Ubuntu)

LS-PrePost Overview

The user interface is designed to be both efficient and intuitive. The latest OpenGL technology is utilized to achieve fast rendering for graphics and XY plotting



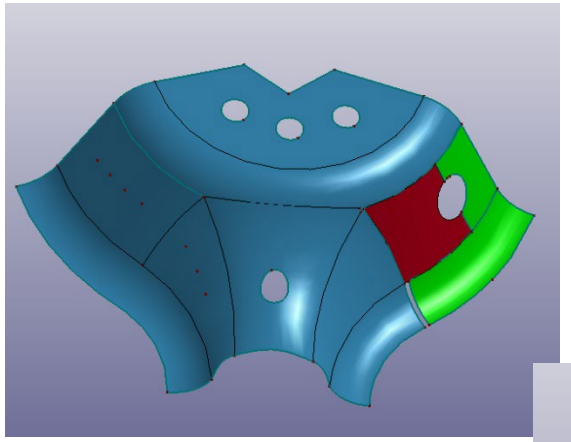
LS-PrePost Overview

There are 6 major functions in LS-Prepost

- Geometry engine
- Meshing
- Finite Element Model Building
- Keyword Manager
- Post-processing
- Applications

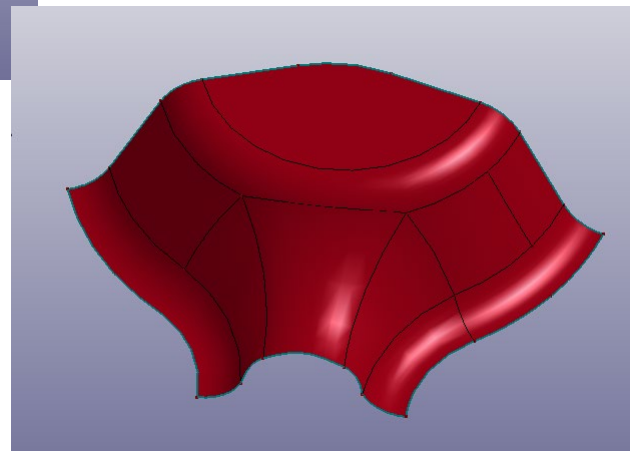
LS-PrePost – Geometry Engine

- Allows user to build geometry entities or to read geometry data from IGES or STEP formatted file
- Provides capabilities to repair and clean geometry entities before meshing, for example:
  - Stitch surfaces together
  - Trim entities again each other
  - Repair overlapping
  - Fill small holes
  - Eliminate small features like fillets surfaces
  - Suppress unwanted common edges

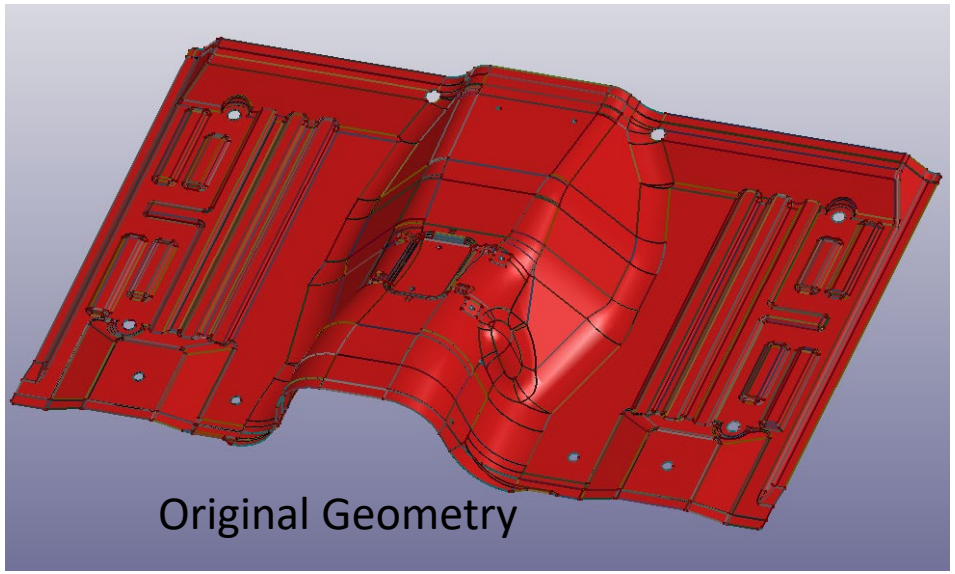


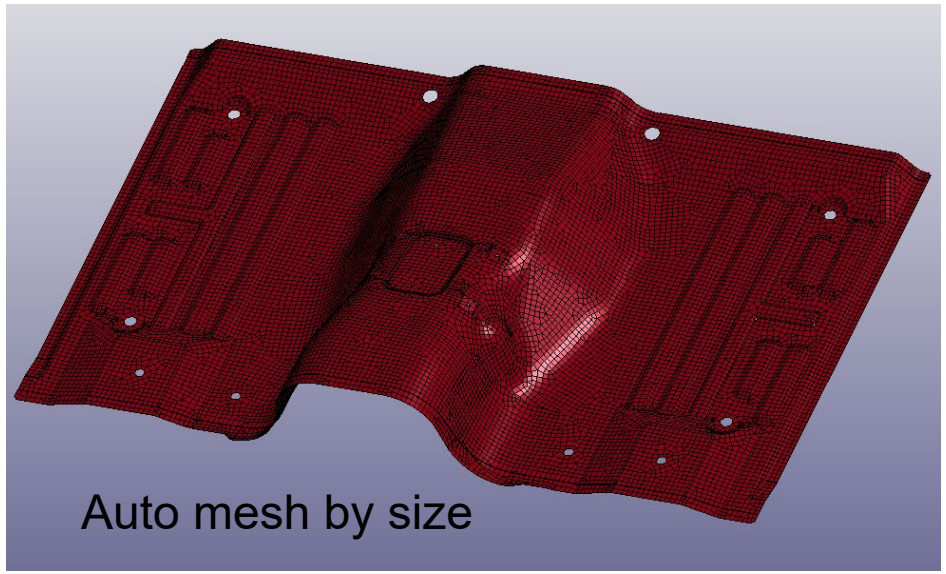
A raw geometry

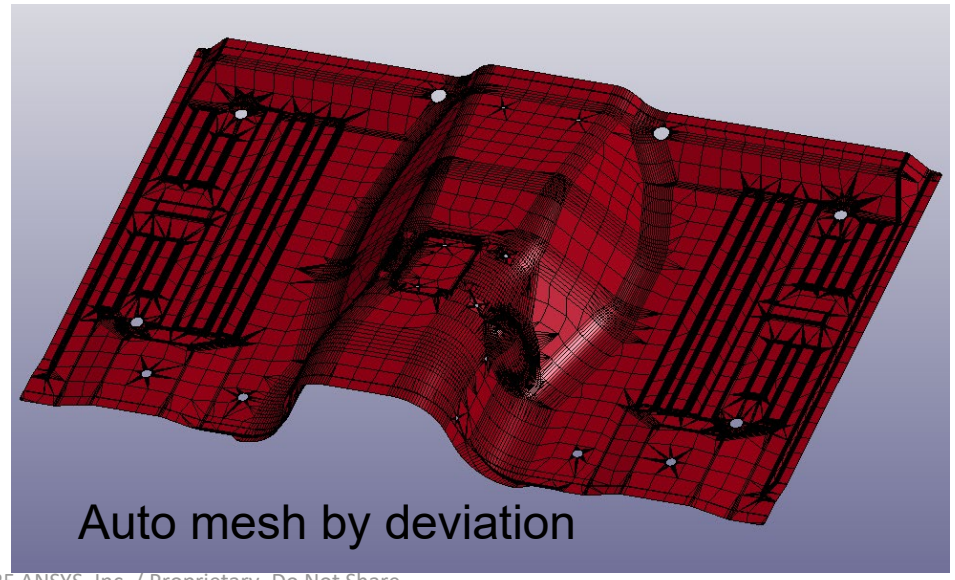
A cleanup geometry

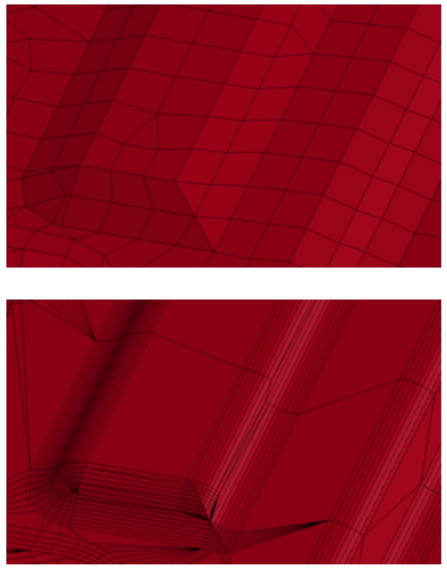


LS-PrePost – Shell Meshing





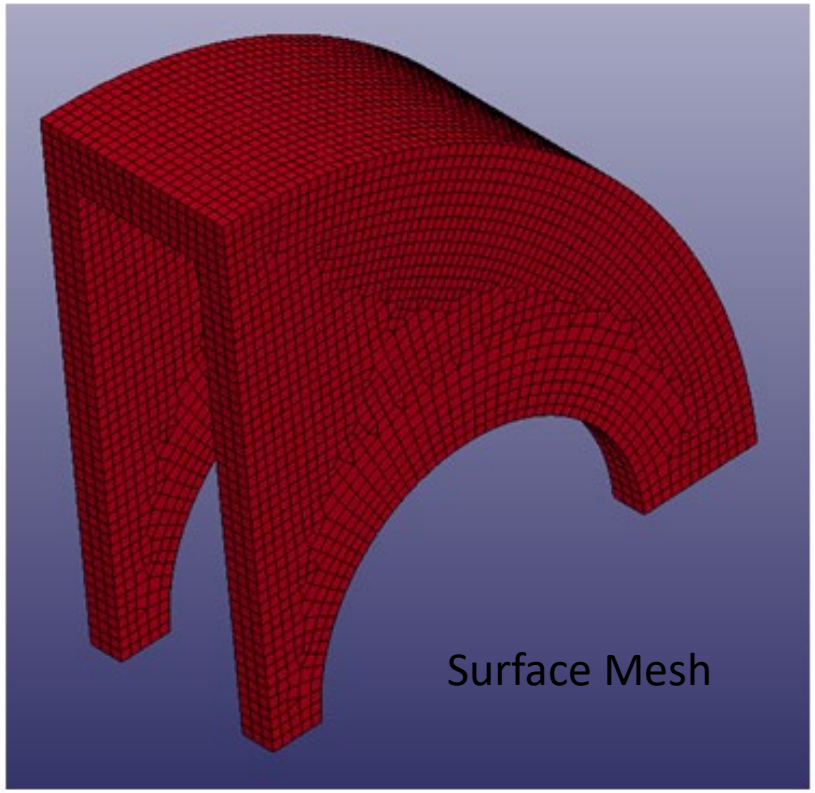


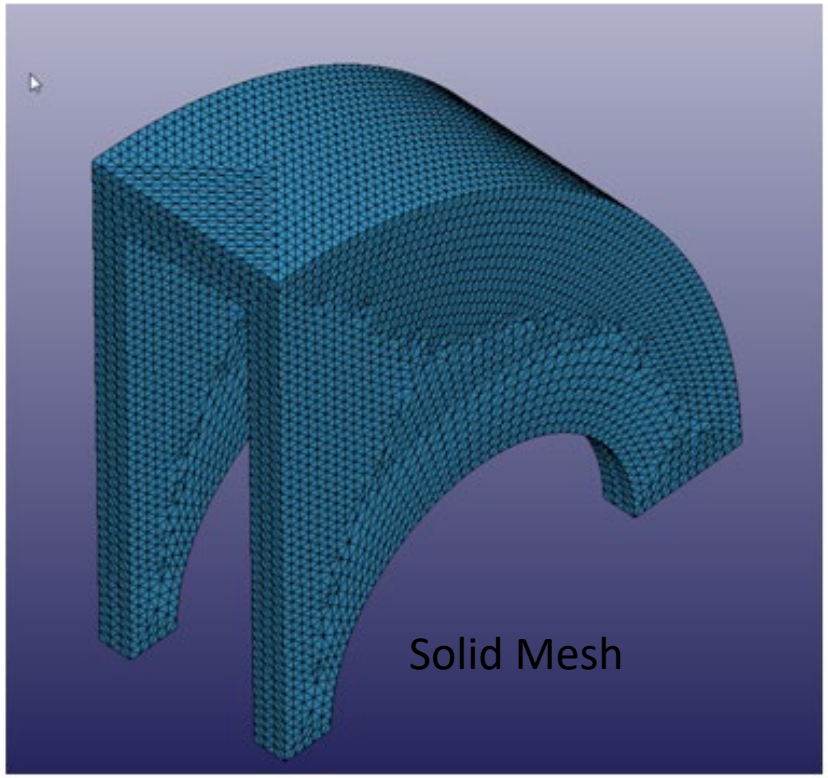


Detailed difference at the curvature

LS-PrePost – Solid Meshing

- Mesh surface of a watertight volume with shell elements, then use automatic Tetrahedron meshing to create tetrahedron elements





LS-PrePost – LS-DYNA Model Builder

Create and modify a complete LS-DYNA model

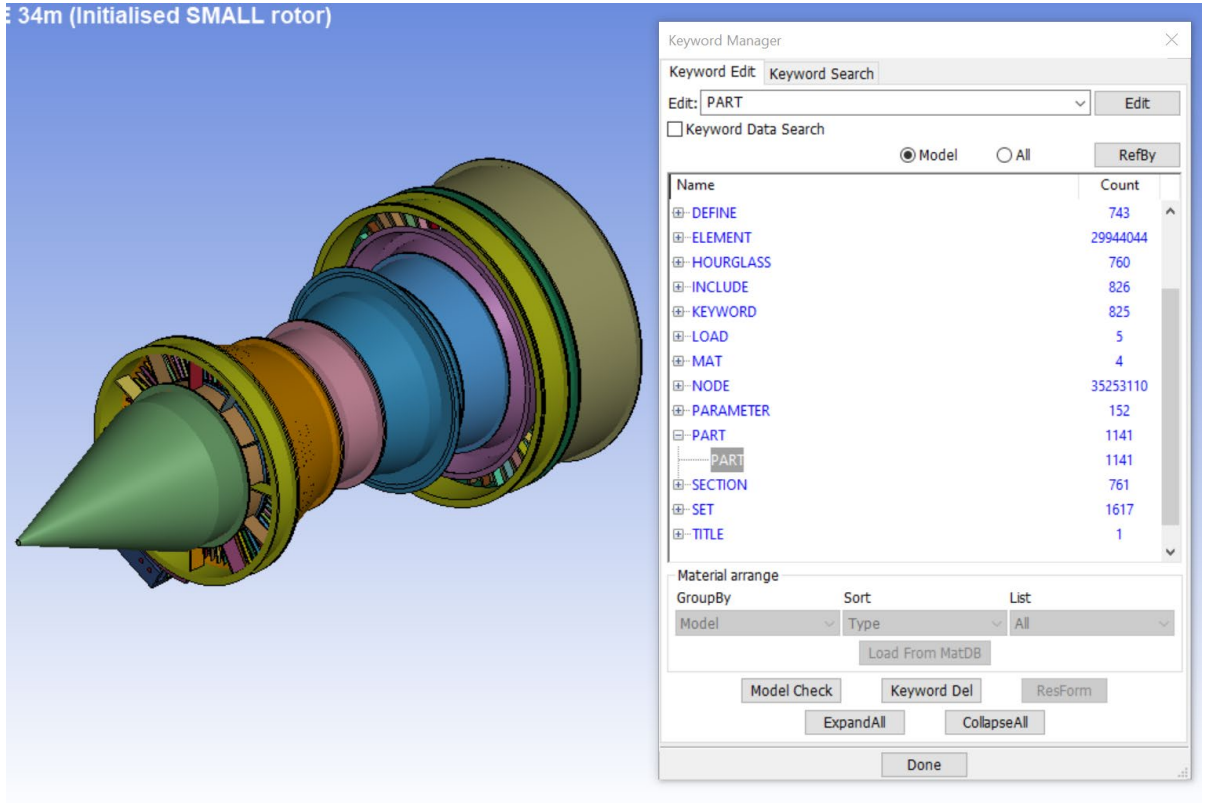
- LS-DYNA entities data creation and modification
  - Coordination systems
  - Set data
  - Constrained data
  - Contact definitions
  - Initial conditions
  - Define curves
  - Rigid Walls
  - Loading

LS-PrePost – LS-DYNA Model Builder

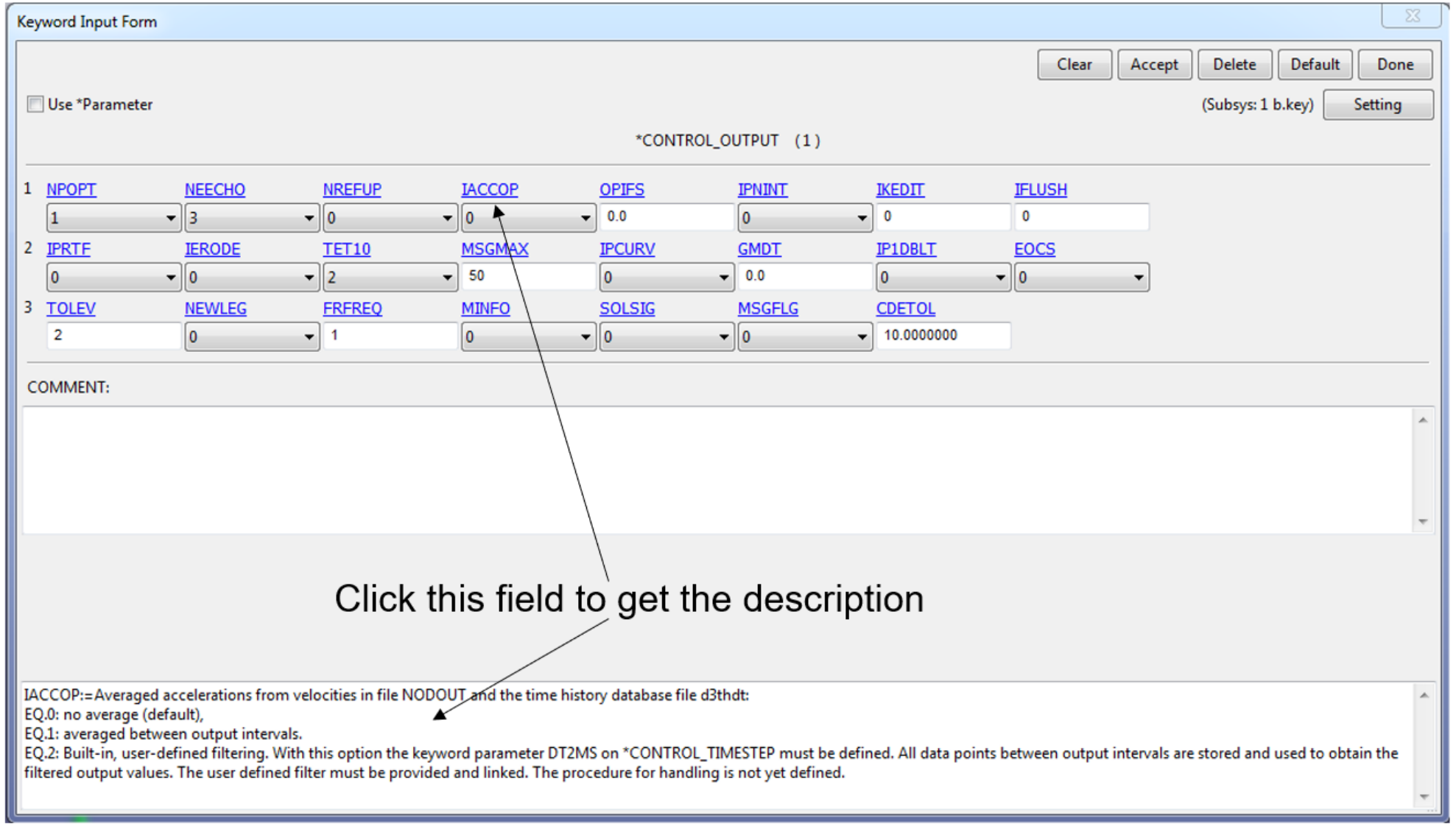
- LS-DYNA non-graphical keyword data creation and modification
  - Material data
  - Execution control data
  - Physical properties
  - Misc. parameters
  - Output control
- Keyword data deck checking – to ensure model is acceptable for LS-DYNA

LS-PrePost – Keyword Manager

- Most completed and up-to-date LS-DYNA keyword support in the industry
- Keyword data is represented by tree structure, and each has its own form
- Each Keyword field has its own description which eliminates the need of user manual most of the time
- Extensive keyword data checking to ensure model is acceptable by LS-DYNA

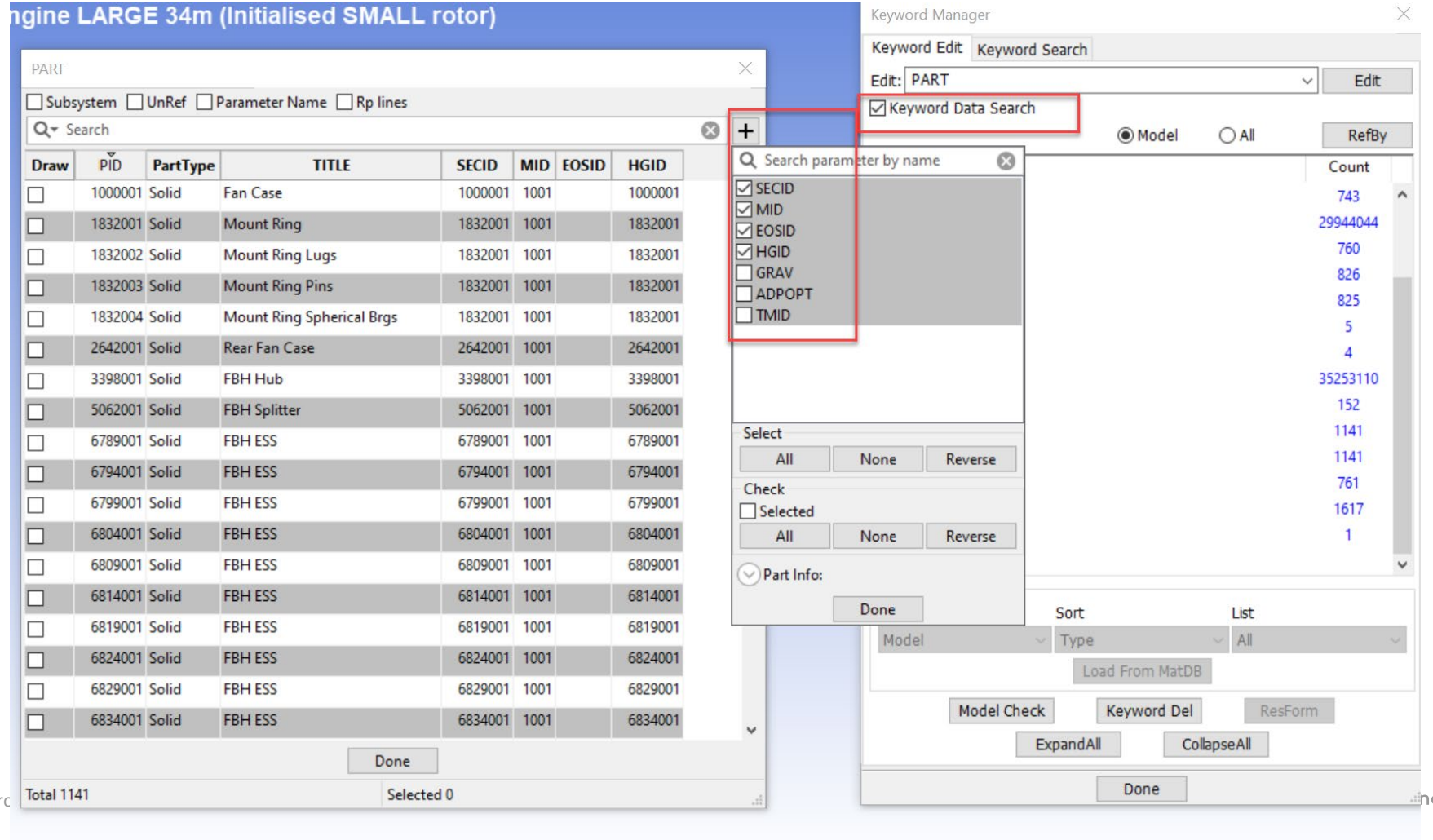


LS-PrePost – A Typical Keyword Form



Keyword Data Search Manager

- Popup “Keyword Data Search Manager” when “Keyword Data Search” is checked, or else open “Keyword Input Form”

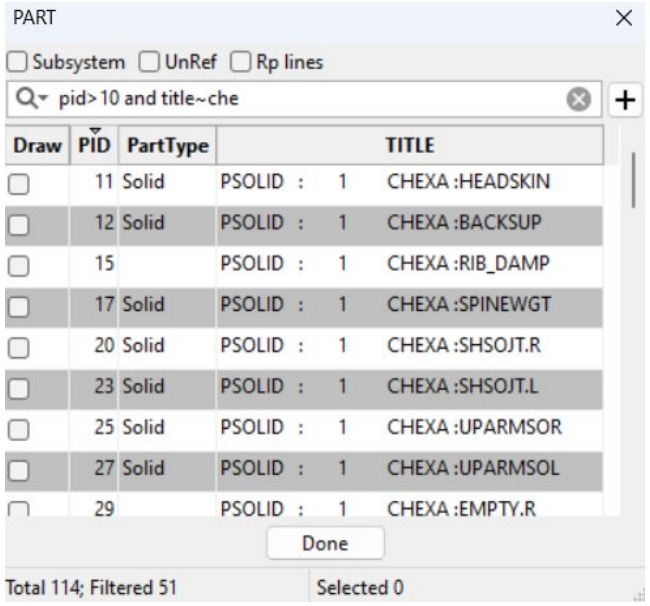


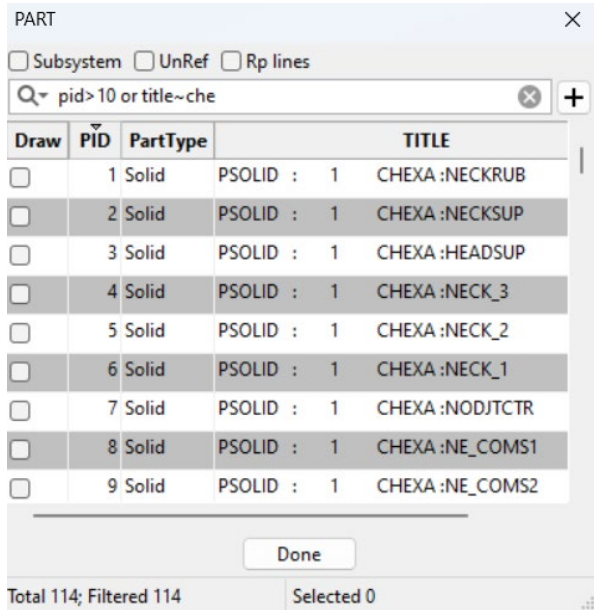
Keyword Data Search Manager

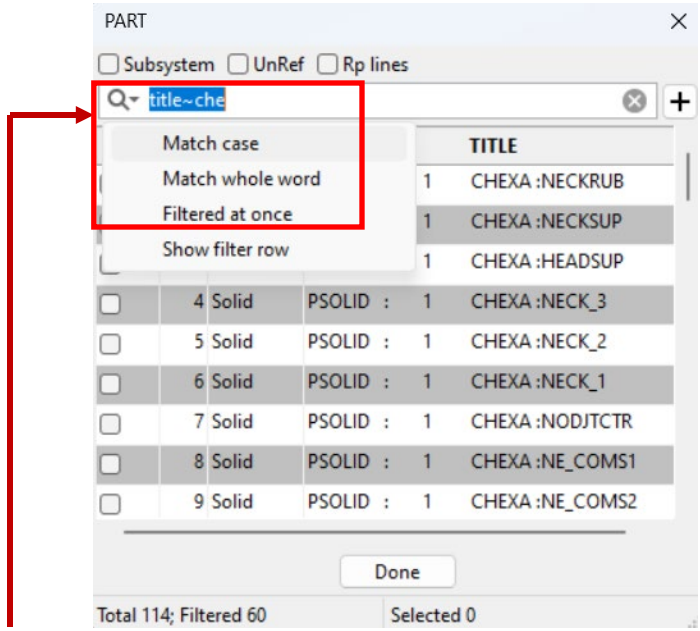
- Supported operations in “Keyword Data Search Manager”:
  - List information of subsystem that the keyword belongs to
  - List if the keyword is unreferenced
  - List keyword data of repeatable card
  - List toggled parameters
  - List part information
  - Display keywords as filtered parameter values
  - Sort displayed keywords
  - Draw keyword entities
  - Right click menus

Supported operations on “Keyword Data Search Manager”

- Display keywords as filtered parameter values



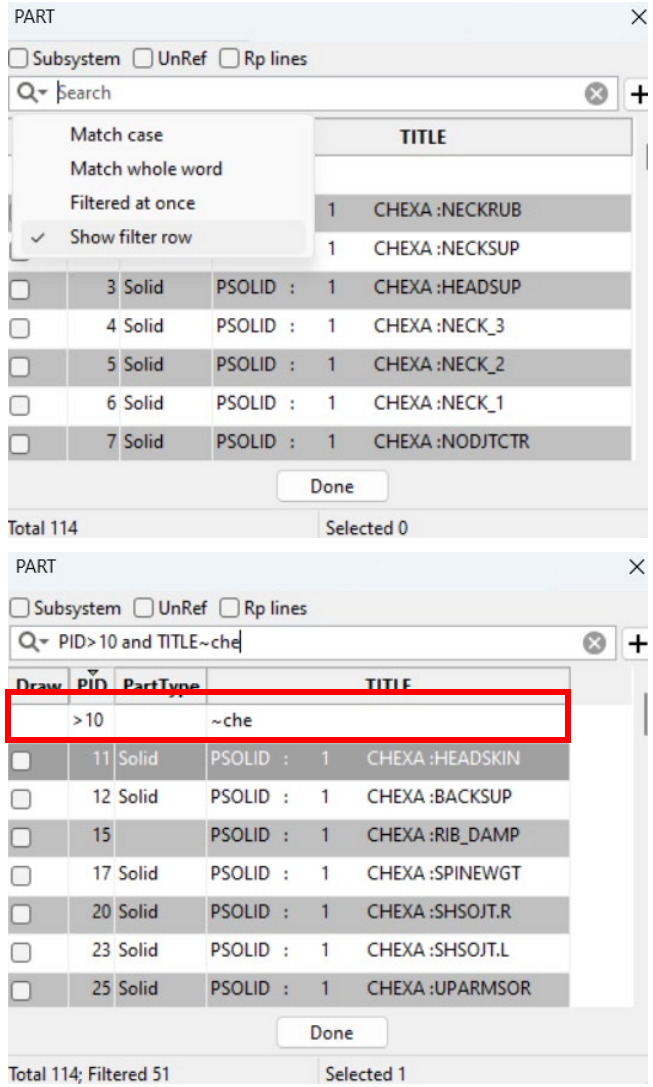




  - Click at the front area of the search widget to set match conditions
    - “Filtered at once” means filtered keywords once the text of search widget changed

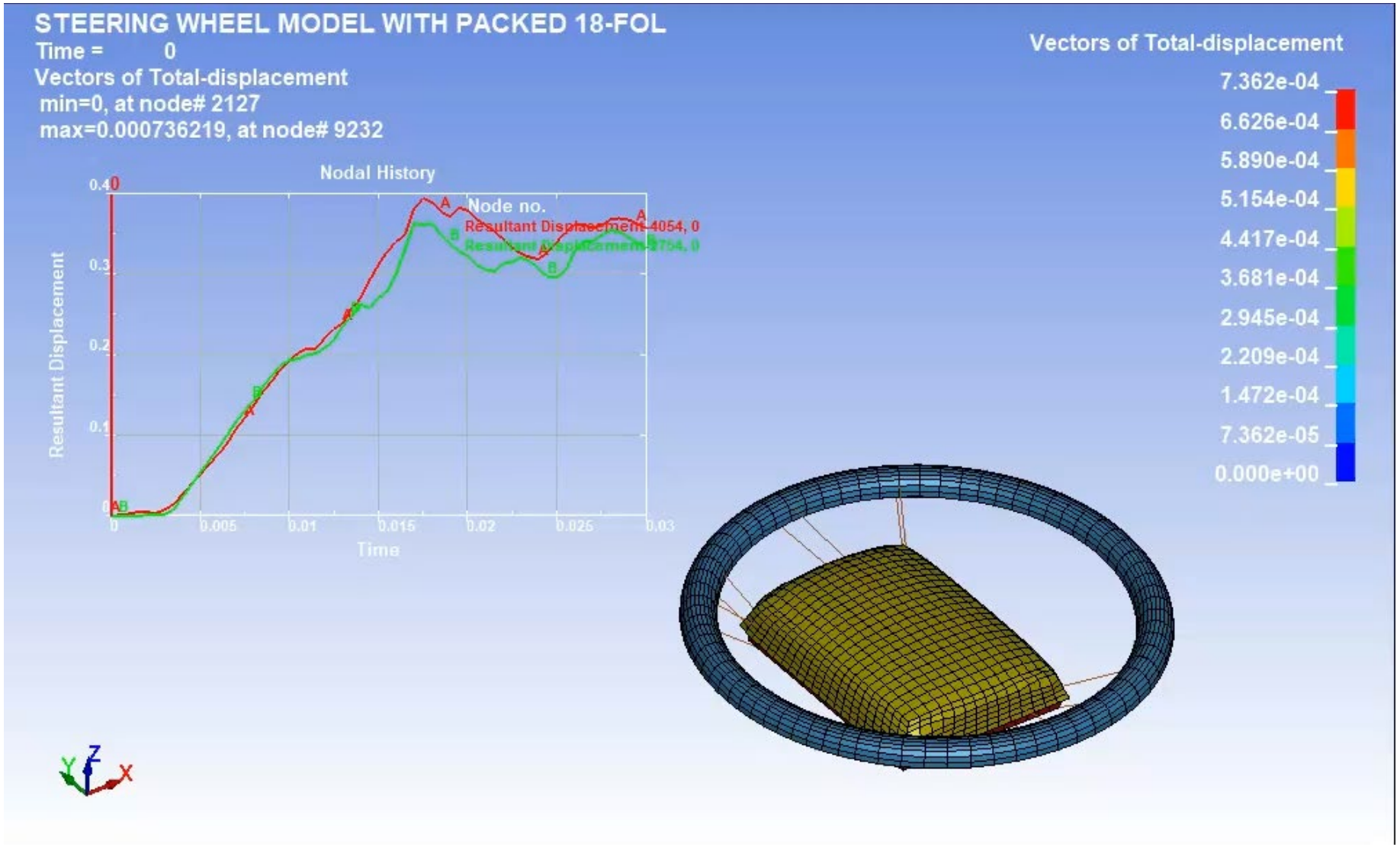
Supported operations on “Keyword Data Search Manager”

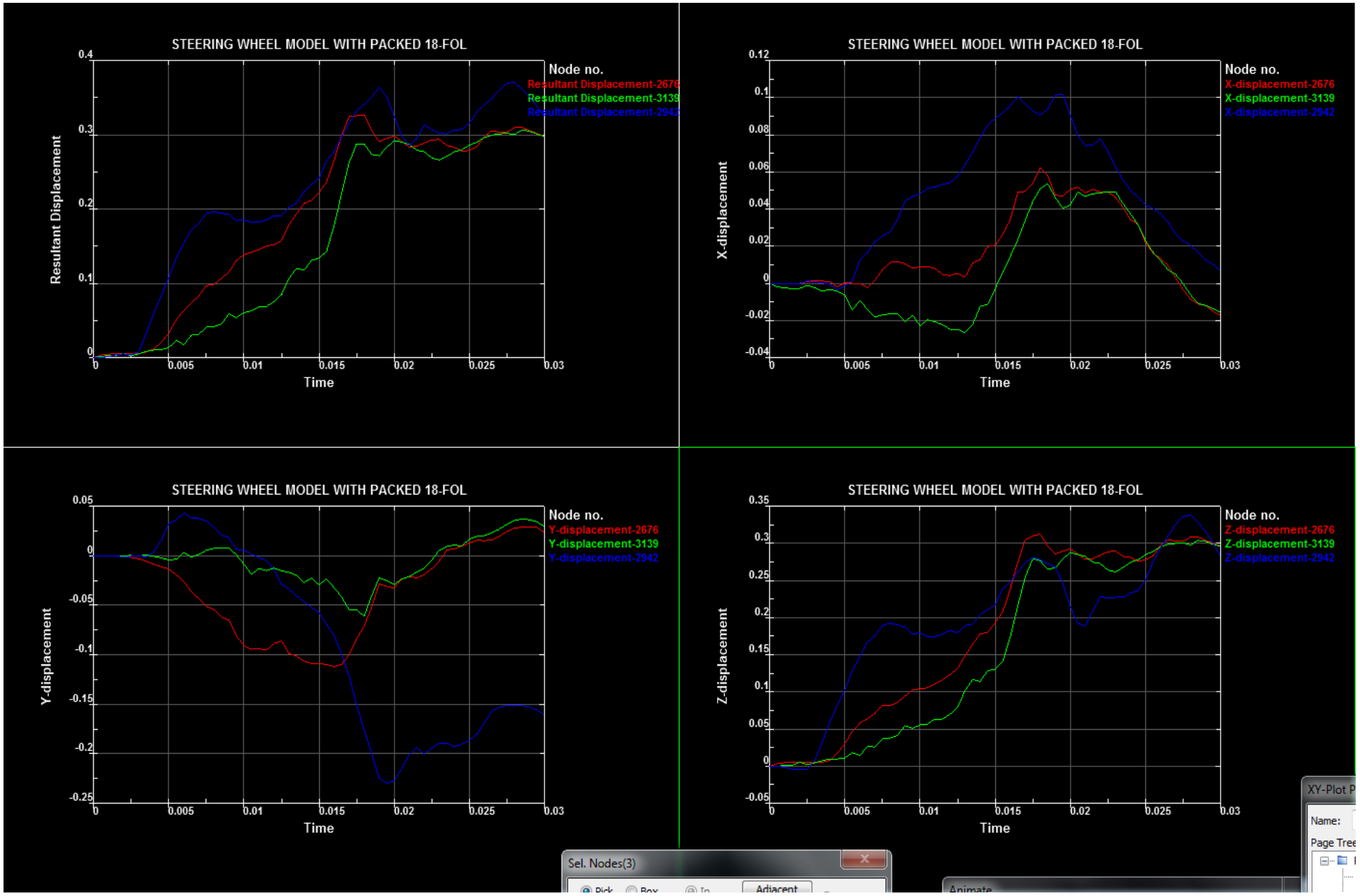
- Display keywords as filtered parameter values
  - Toggle the menu “Show filter row” to display the filter row in the grid
  - Input filter value of the parameter
  - Supported symbols of parameter value filter: >, GT, >=, GE, <=, LE, !=, NE, =, EQ, <, LT, ~
  - Supported symbols of combined conditions: “and”/“&&”, “or”/“||”



Post-Processing

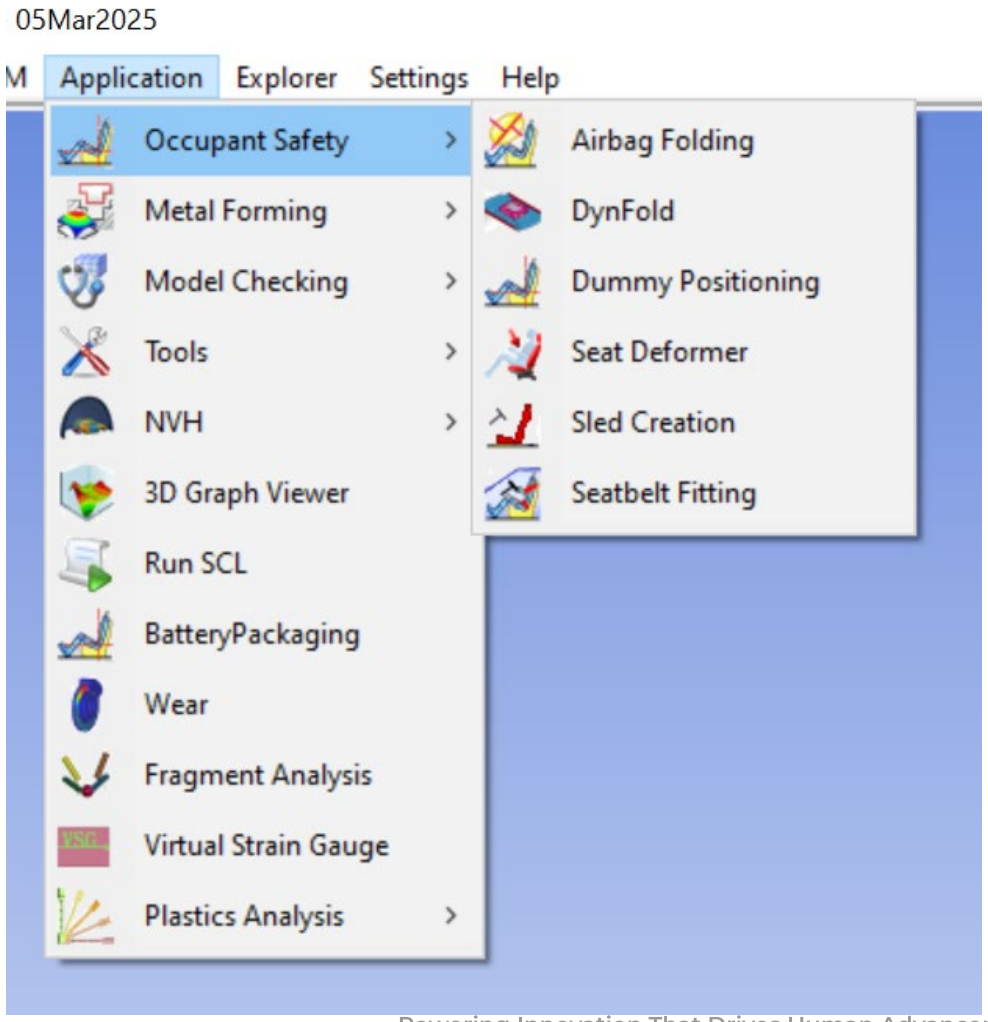
- Animation
- Vector plot
- Fringe contour
- XY history





LS-PrePost – Applications

- An application - Many operations being grouped together into one interface for a special purpose
  - Airbag folding
  - Dummy positioning
  - Seatbelt fitting
  - Seat Deformer
  - Sled Creation
  - Battery Packaging
  - Wear Analysis
  - Fragment Analysis
  - Virtual Strain Gauge



Powering Innovation That Drives Human Advancement

Solution Explorer

Solution Explorer

Multiphysics

- Structure
- ICFD
- Structured-ALE
- Thermal
- EM
- Drop Test
- ……

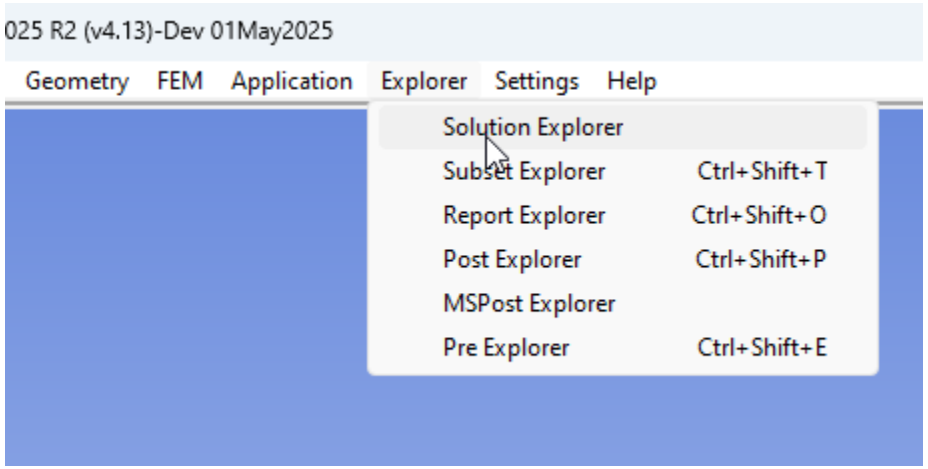
Translation Tools

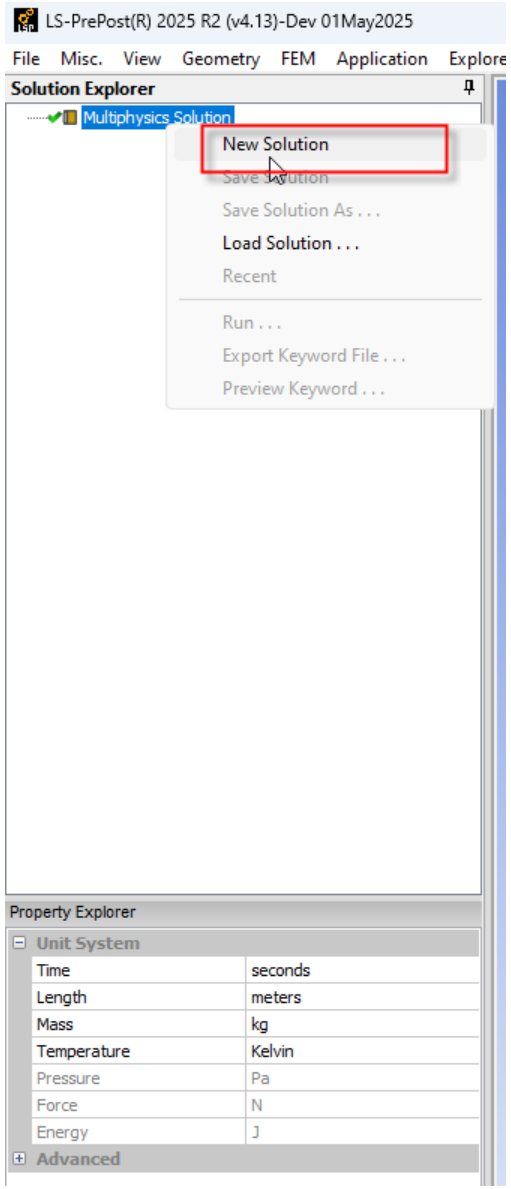
- LS-DYNA input data
- Tree structure & Properties
- Material databases

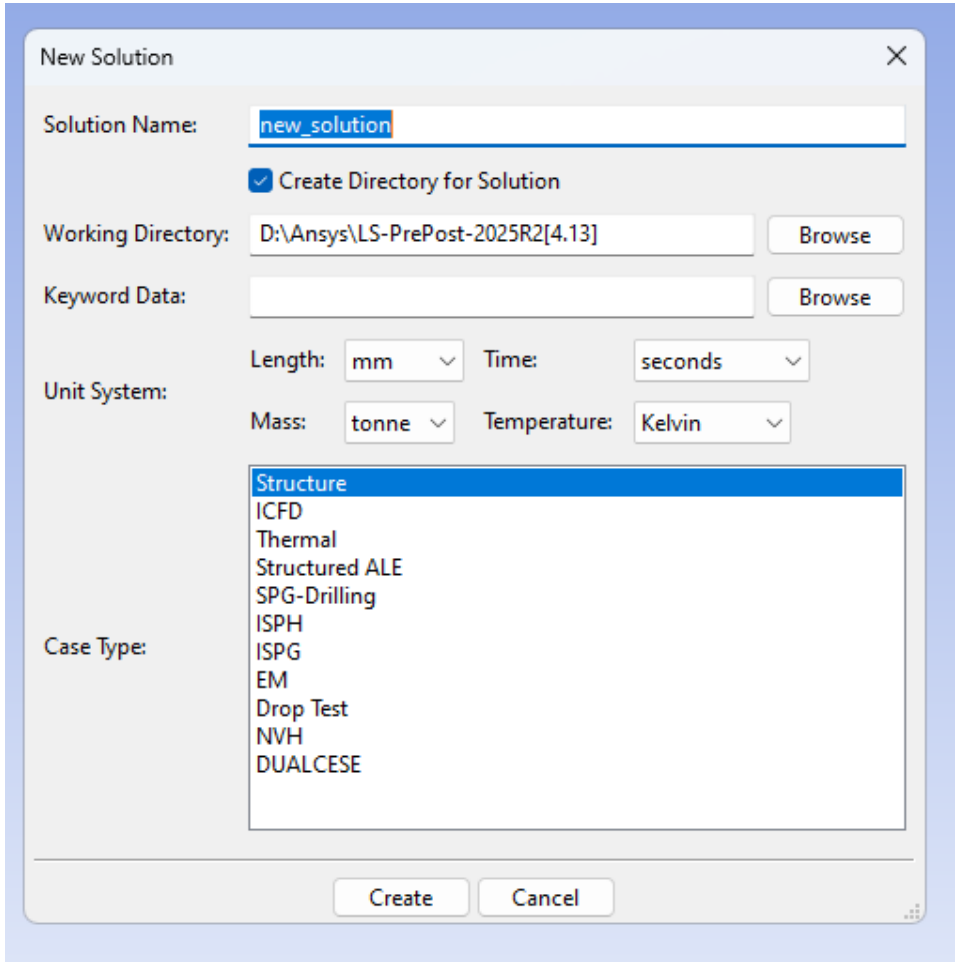
One Flow Line

- No LS-DYNA input data file editing
- Whole flow line based on tree structure and friendly to engineers

Solution Explorer Entrance

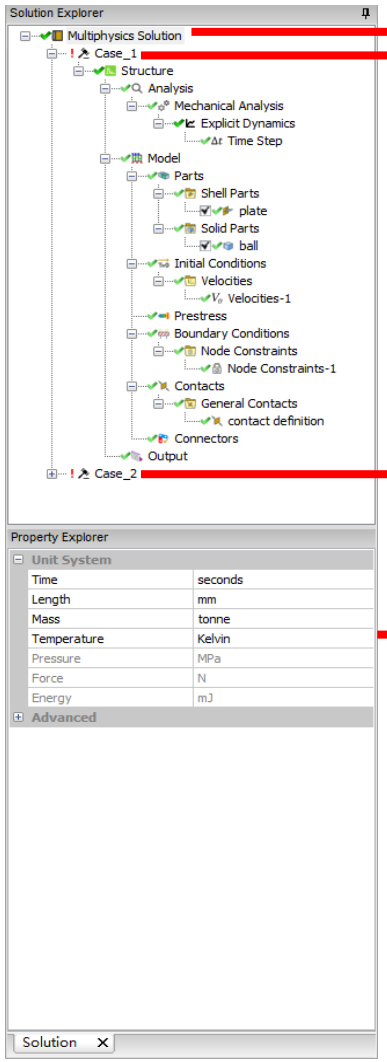






Folder	LS-DYNA Input	Case
• Solution Name • Working Directory	• Mesh data (Part , Element data, Node data, Material…) • Unit System	• Structure • ICFD • … • EM • Drop Test

Solution Explorer Tree Structure



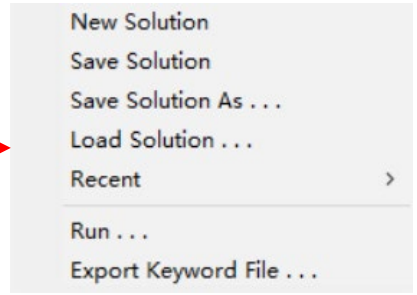
Top Level

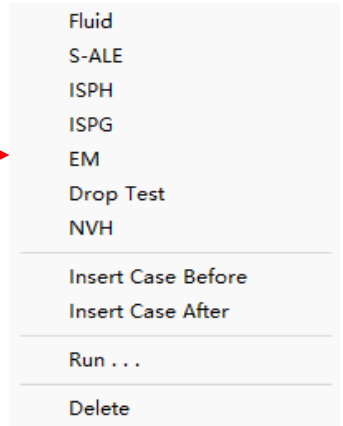
Case Level

Structural Tree Structure

Case Level

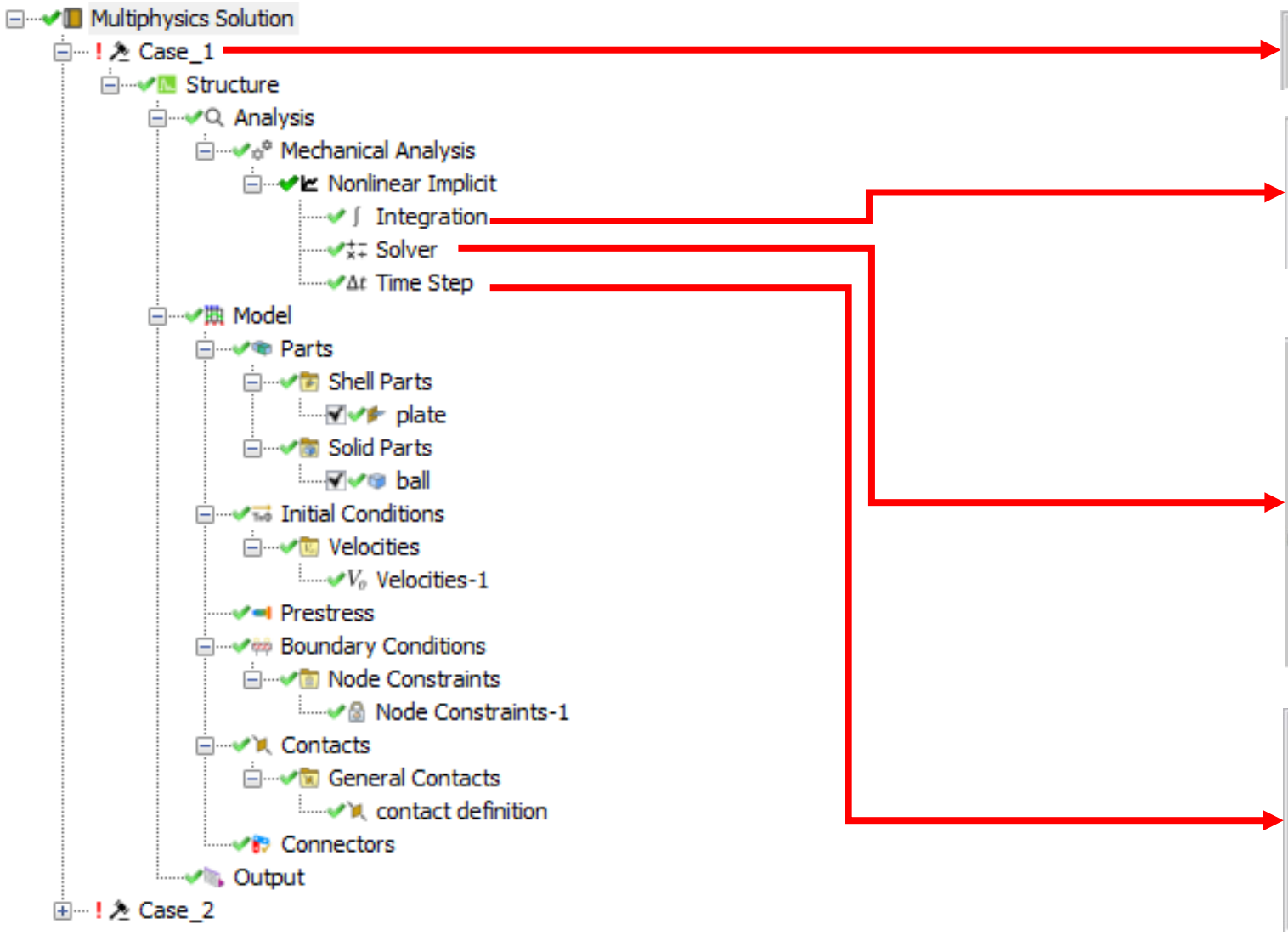
Property Explorer





1. Different tree item has different properties.
2. Edit the properties to change analysis data.
3. Many tools can be designed to the properties.
4. The descriptions of the properties are friendly to engineers.

Solution Explorer Tree Structure



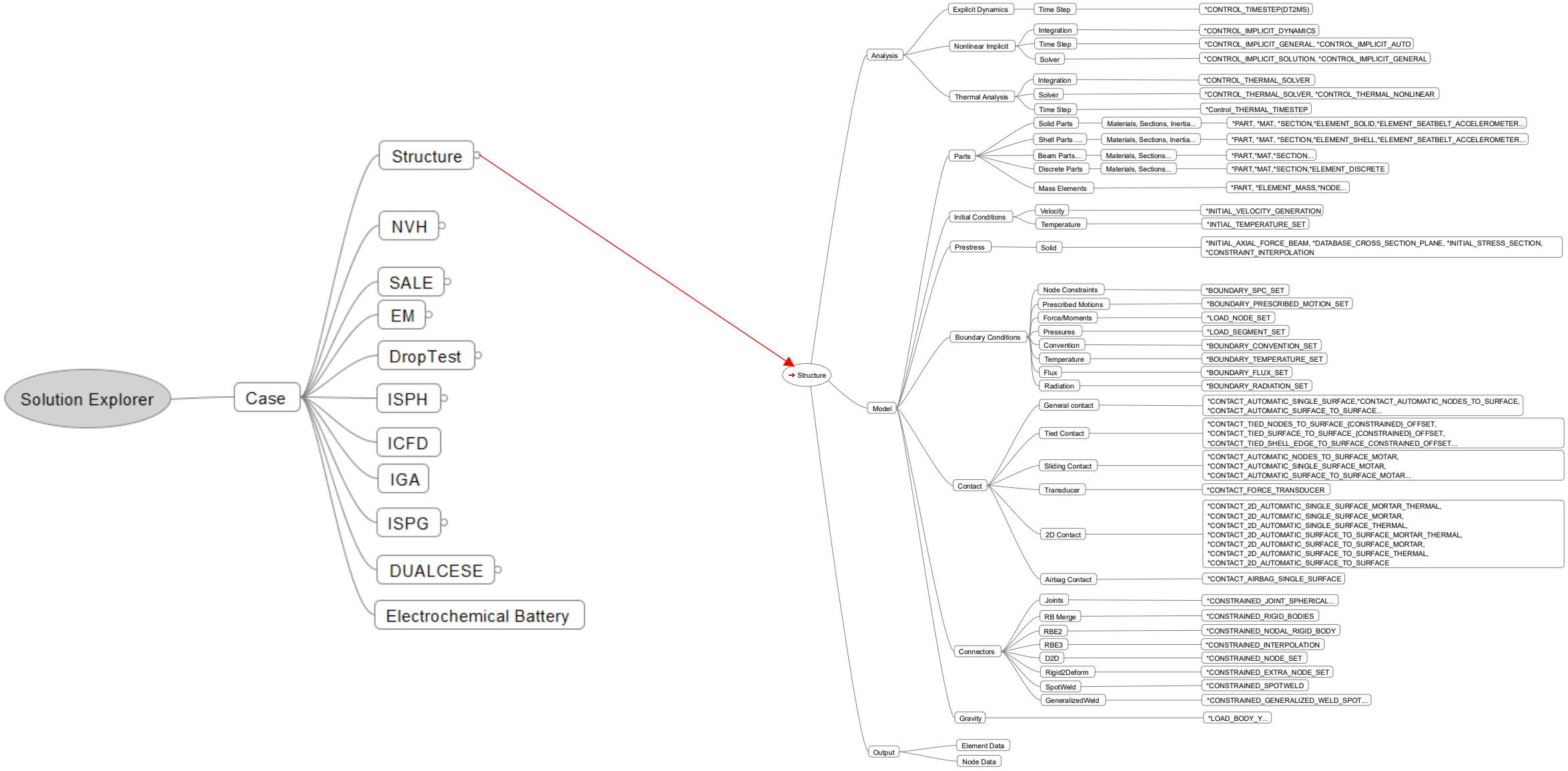
End Time	1 s
Database Plot Interval	0.01 s

Types	Dynamics
Methods	Newmark Damped
Gamma	0.6
Beta	0.38

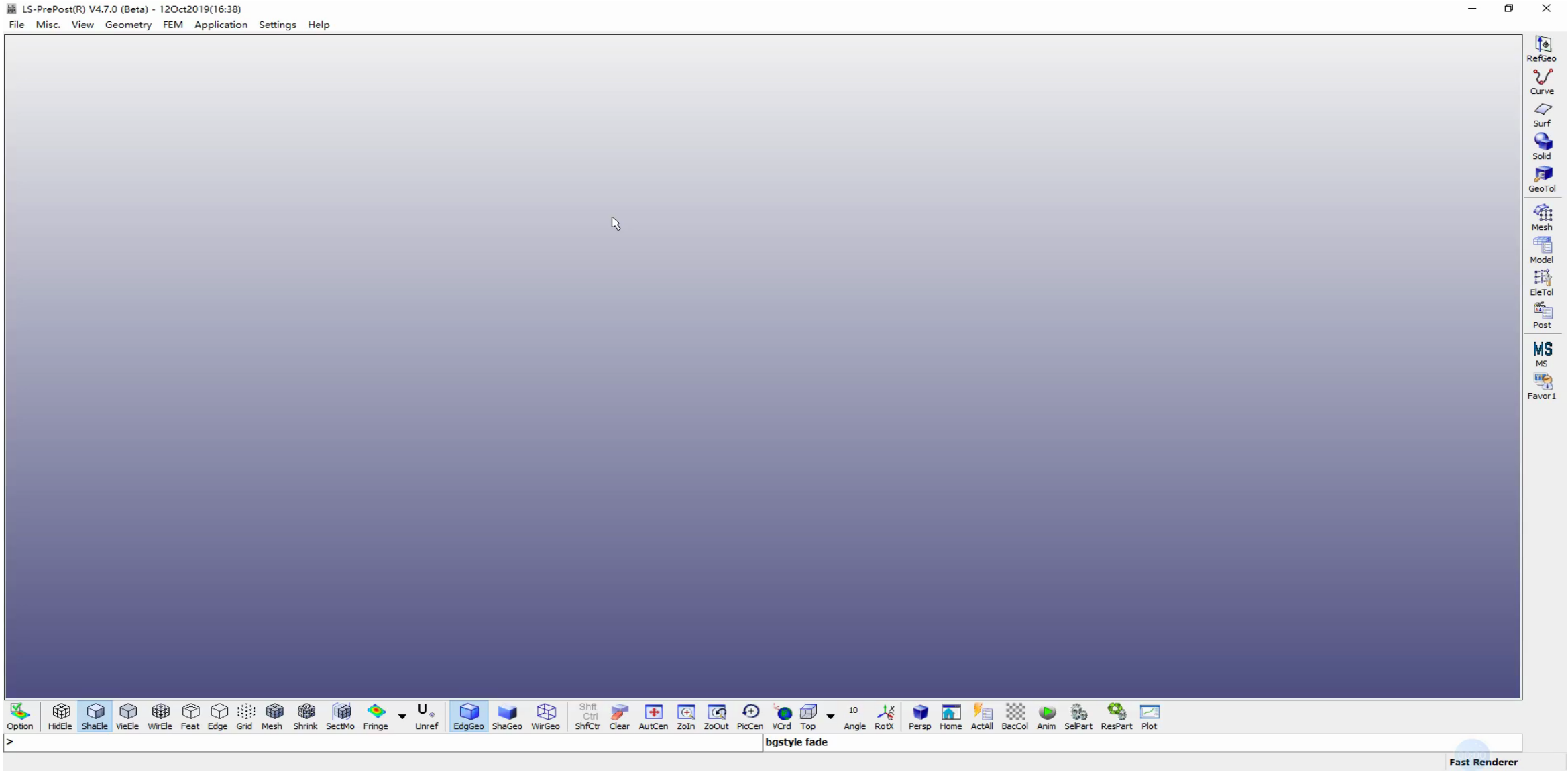
Solution Method	BFGS Slightly Nonlinear
Reformations Before Retry	15
BFGS Iterations Before Reform	11
Tolerance Type	Recommended
Relative Displacement	0.001
Advanced	
Geometric Stiffness Matrix	☐
Linear Solver	Symmetric
Use residual tolerances	☐

Initial Size	0.01 s
Step Size Option	Maximum Time Step Size
Maximum Time Step Size	0.05 s
Change Strategy	Aggressive
Increase T.S. if iter <	100
Decrease T.S. if iter >	110

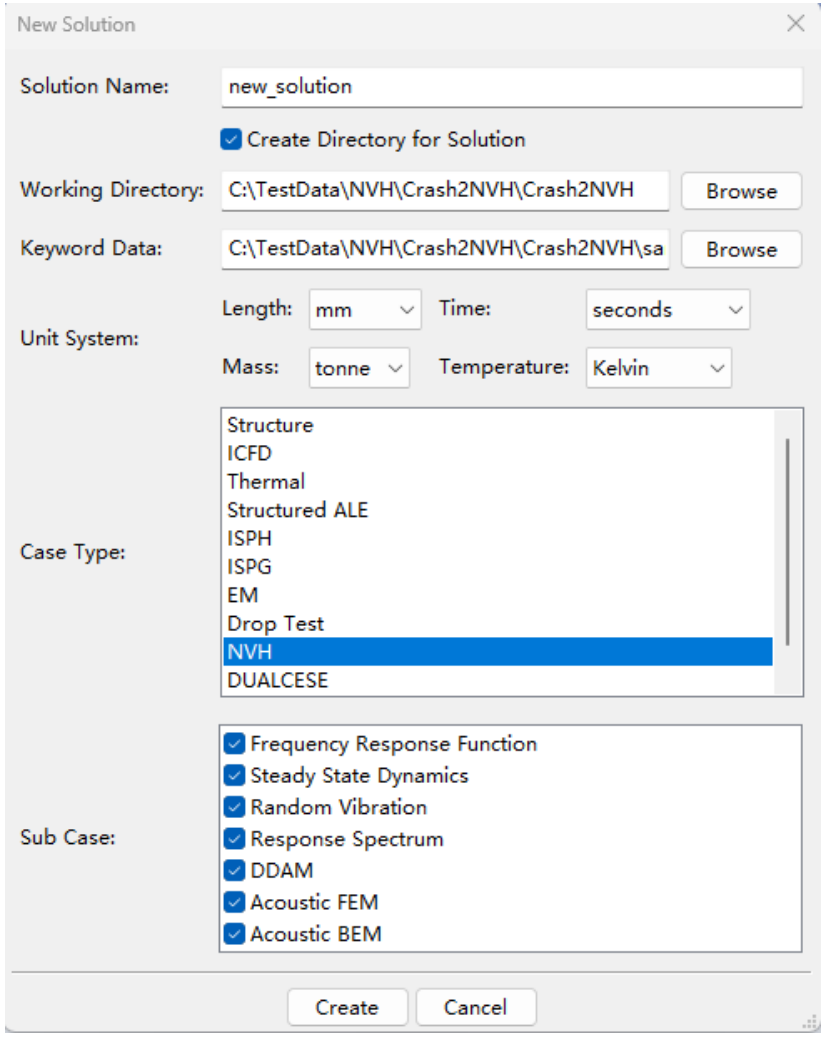
Maps of LS-DYNA input data from tree objects



Solution Explorer Demo Video

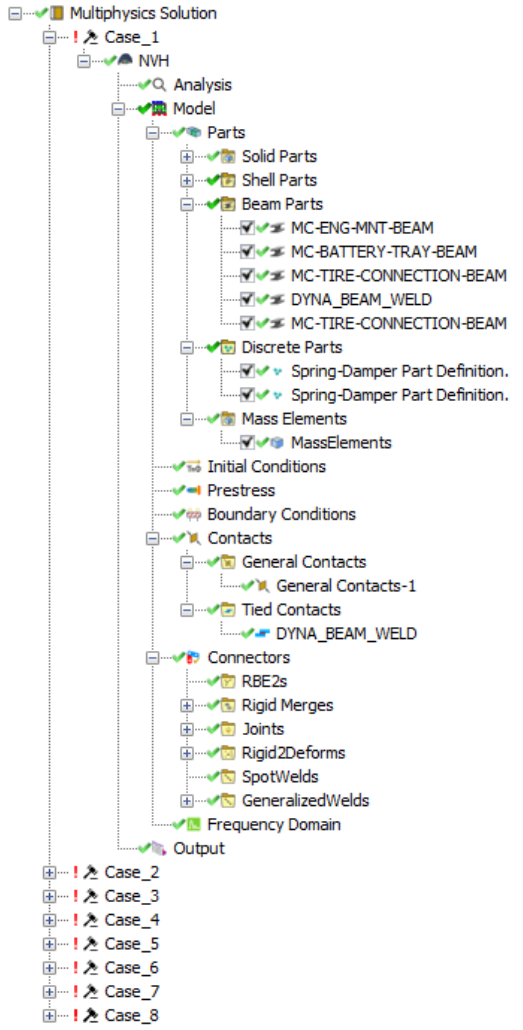


Solution Explorer Example Highlights (NVH)

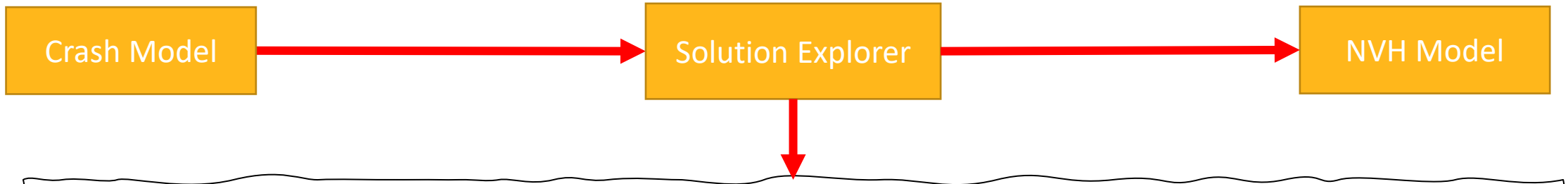


Click “One Button” to create multiple solution trees at one time:

Case_1: Eigen Value
Case_2: Frequency Response Function
Case_3: Steady State Dynamics
Case_4: Random Vibration
Case_5: Response Spectrum
Case_6: DDAM
Case_7: Acoustic FEM
Case_8: Acoustic BEM



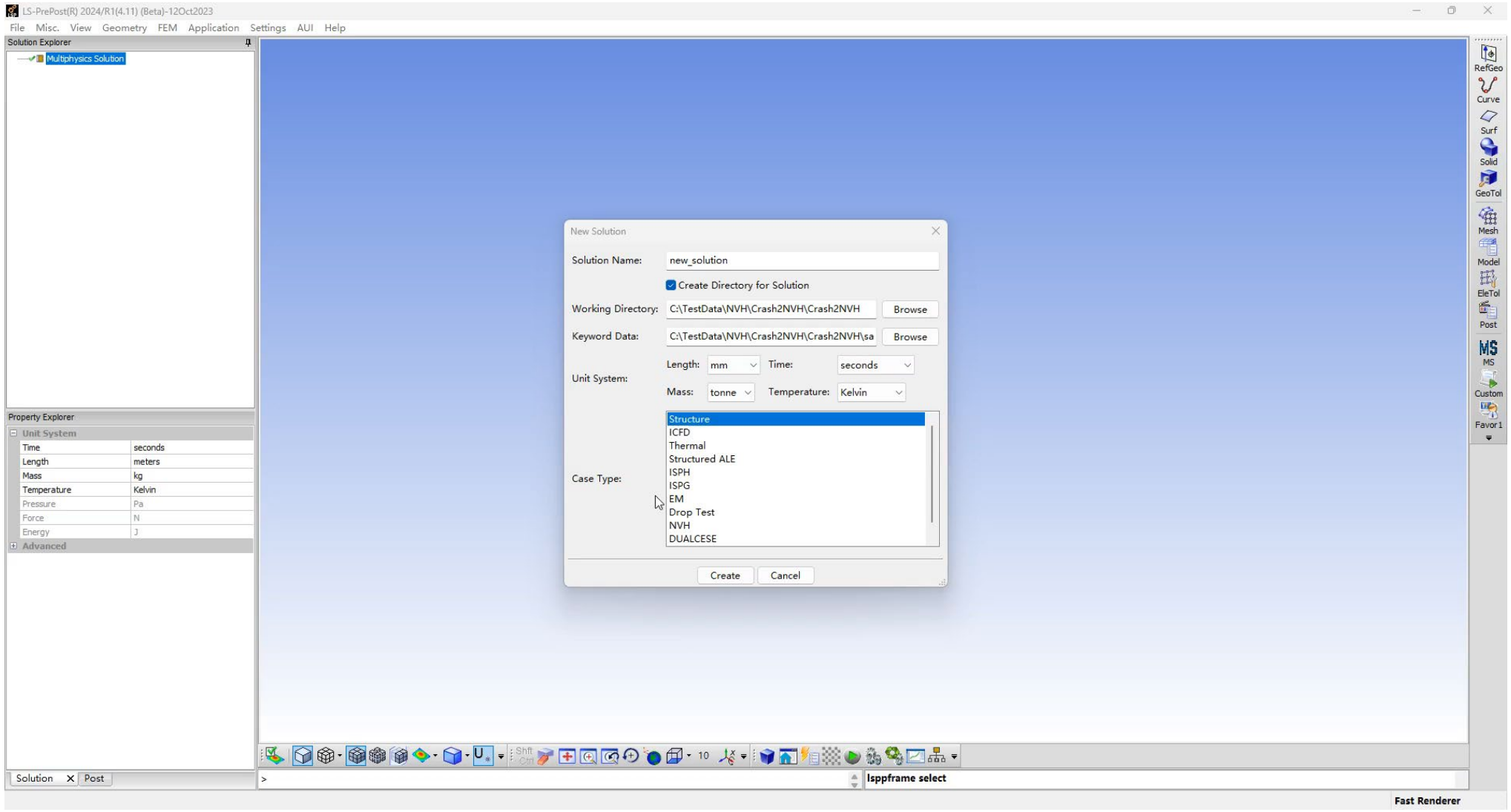
Solution Explorer Example Highlights (NVH: conversion rules)



1. **Disable**: Airbag, Gravity...
2. **Contact**: tied contact
   a) TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET
      TIED_SURFACE_TO_SURFACE_CONSTRAINED_OFFSET
   b) TIED_NODES_TO_SURFACE_OFFSET TIED_SURFACE_TO_SURFACE_OFFSET
   c) TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET
   d) TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET
3. **Materials:** changing to linear material
   Keep : *MAT_ELASTIC, *MAT_BLATZ-KO_RUBBER, *MAT_NULL, *MAT_RIGID, *MAT_HONEYCOMB,
   *MAT_CRUSHABLE_FOAM,*MAT_SPOTWELD,*MAT_SPRING_ELASTIC,*MAT_DAMPER_VISCOUS...
   Change:*MAT_PIECEWISE_LINEAR_PLASTICITY to MAT_ELASTIC
4. **Section:** changing to linear elements
   Linear element tyepe: **Shell** 18,20, and 21; **solid** 18.
5. **Keep:** ”Boundary Conditions” and ”Connectors” like joints, spotwelds, rbe2s, rbe3s and so on.

Solution Explorer Example Highlights (NVH)

Click “One Button” to convert crash model to nvh model.

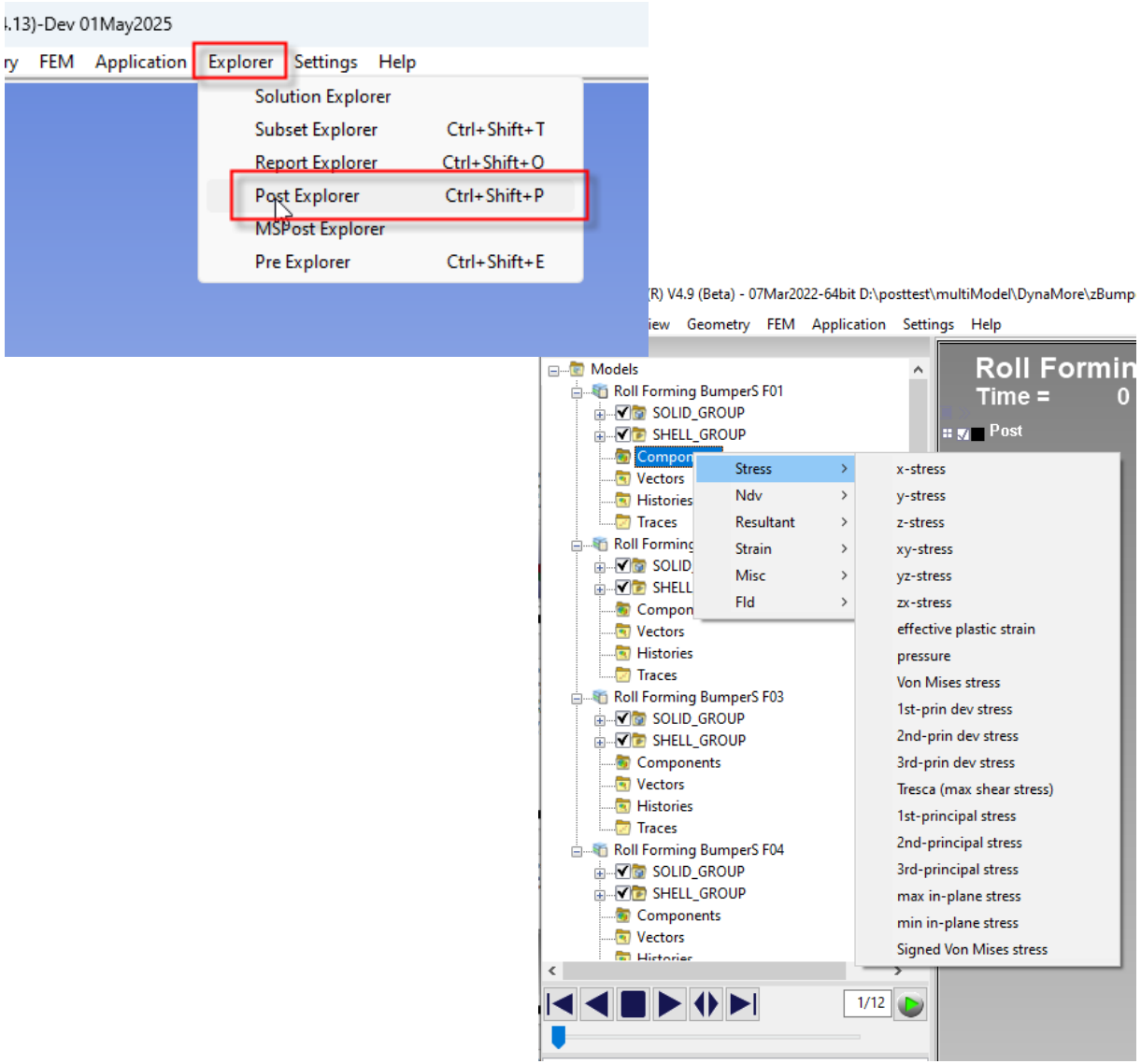


Powering Innovation That Drives Human Advancement

Post-Processing with Post Explorer

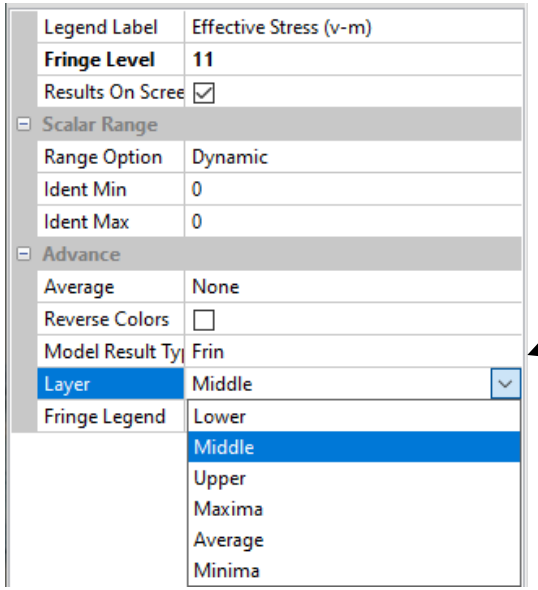
Post-Processing – Introduction of Post Explorer

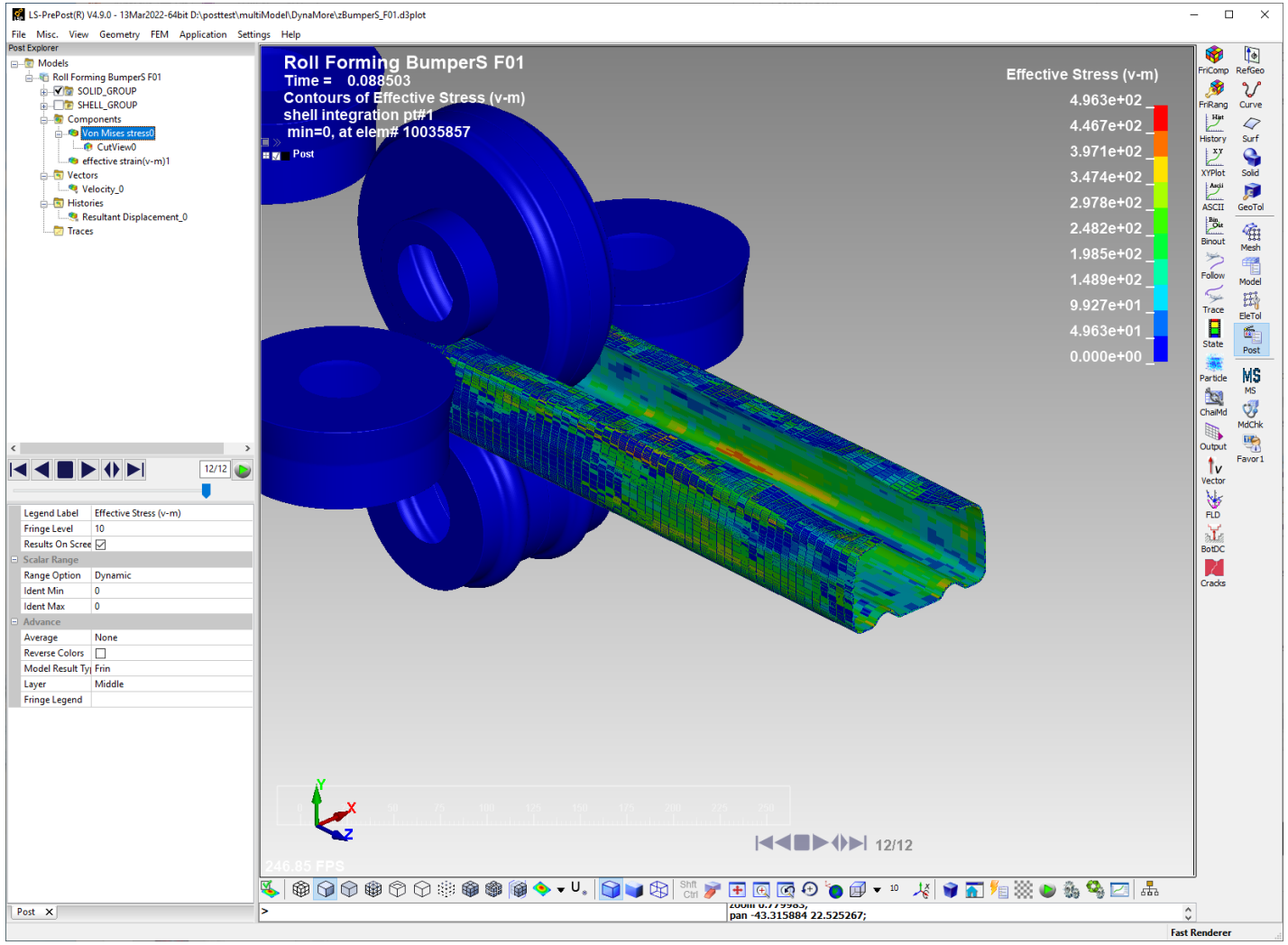
- A new paradigm of LS-DYNA post-processing in LS-PrePost
- It is feature tree-based operations
- To activate: pull down menu “Explorer”->”Post Explorer”
- Multiple models can be loaded and listed
- Each model has its own Components, Vectors, Histories, and Traces main objects
- Right click on these objects to select sub-object



Post-Processing – Introduction of Post Explorer

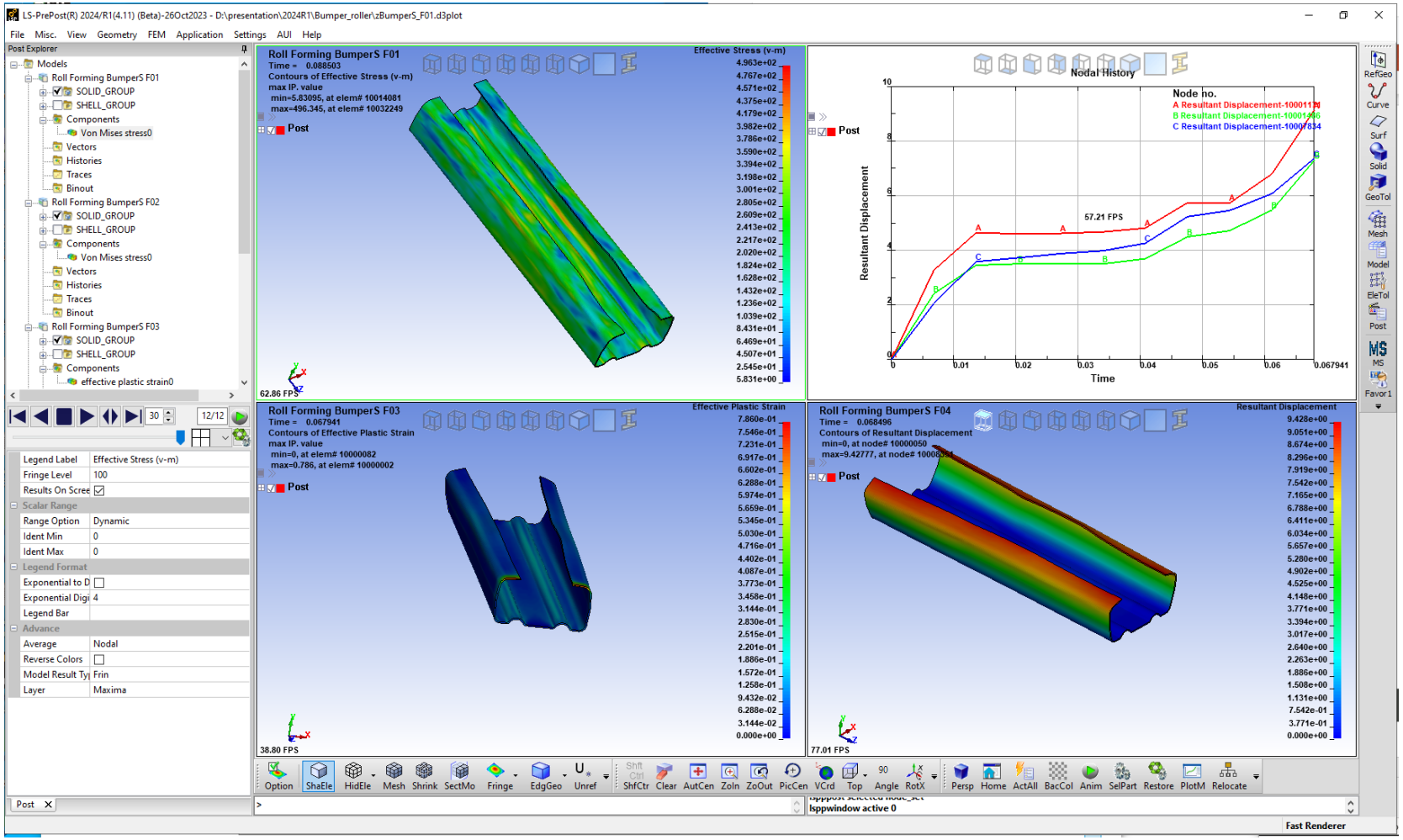
- Each selected object has its own property tree
- Many quantities associated with this object can be changed on the property tree





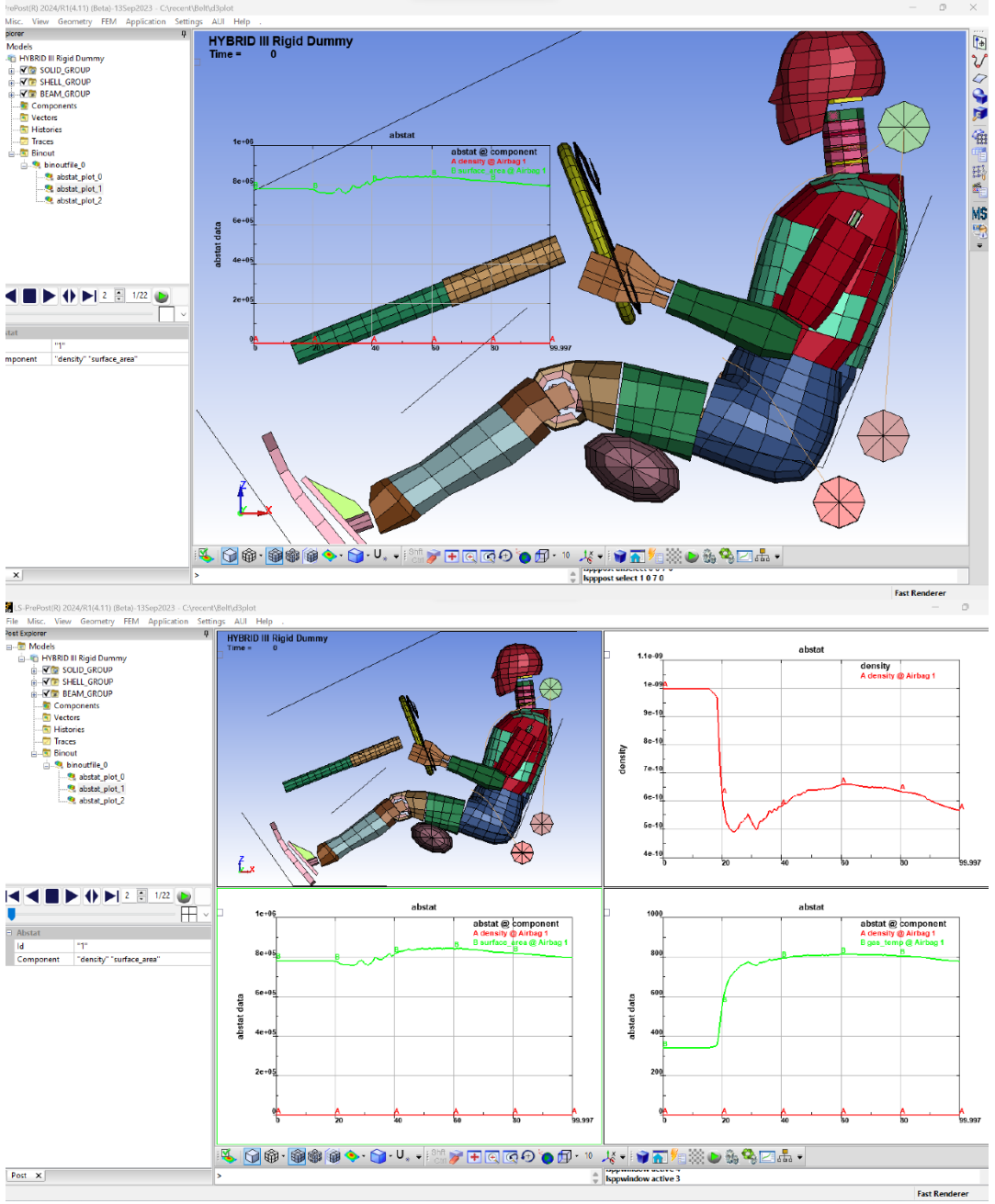
Post-Processing – Introduction of Post Explorer

- Multiple models can be shown on Split windows
- History data (XY plot) can also be drawn on one of the windows



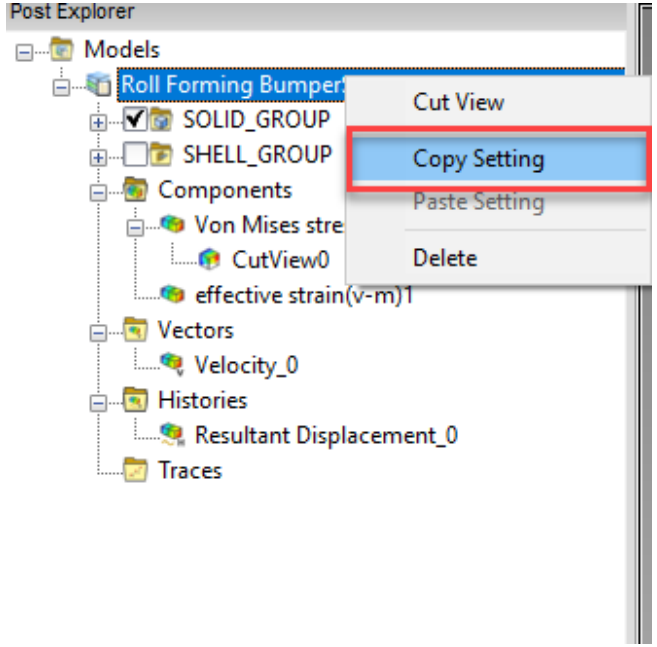
Post-Processing – Post Explorer

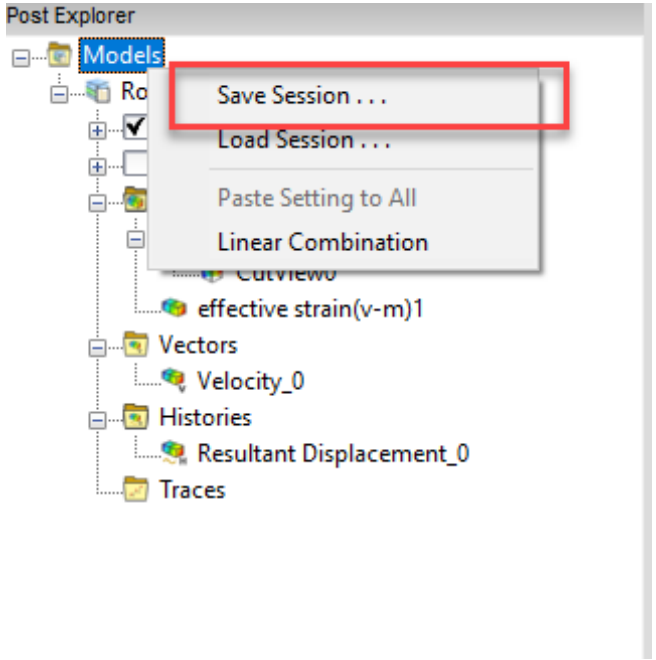
- Binout Support
- Binout history data is one of the object on the Feature tree
- Not all Binout branches are supported for now
- ASCII history data interface will come in the future



Post-Processing – Introduction of Post Explorer

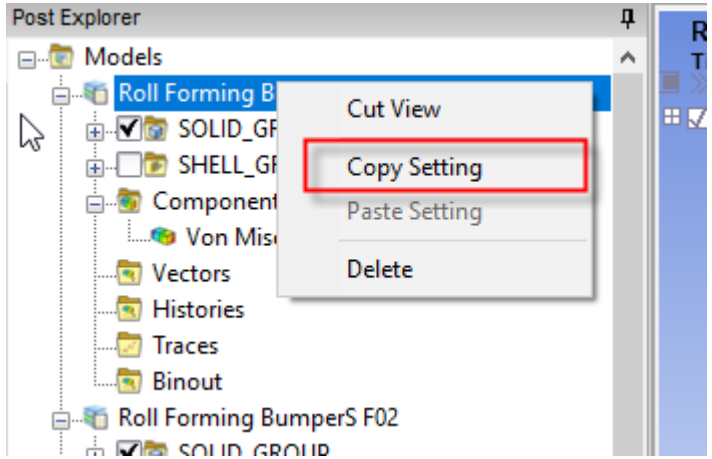
- Many objects with their properties can be setup in the feature tree
- The Settings for one model can be copied and pasted to other model (if multiple modes have been loaded)
- All settings for the session can be saved to file and be loaded back in the future

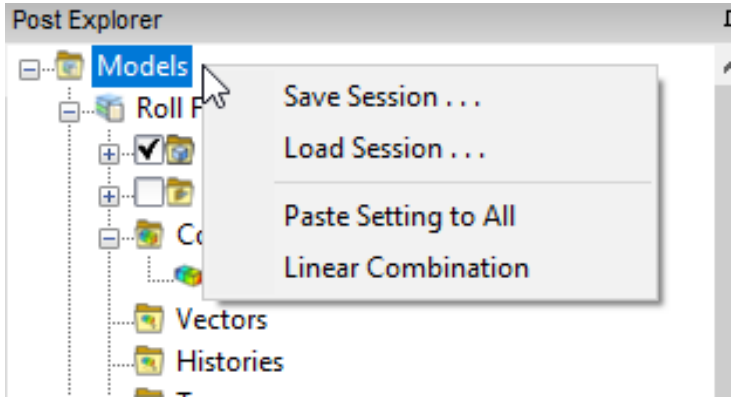




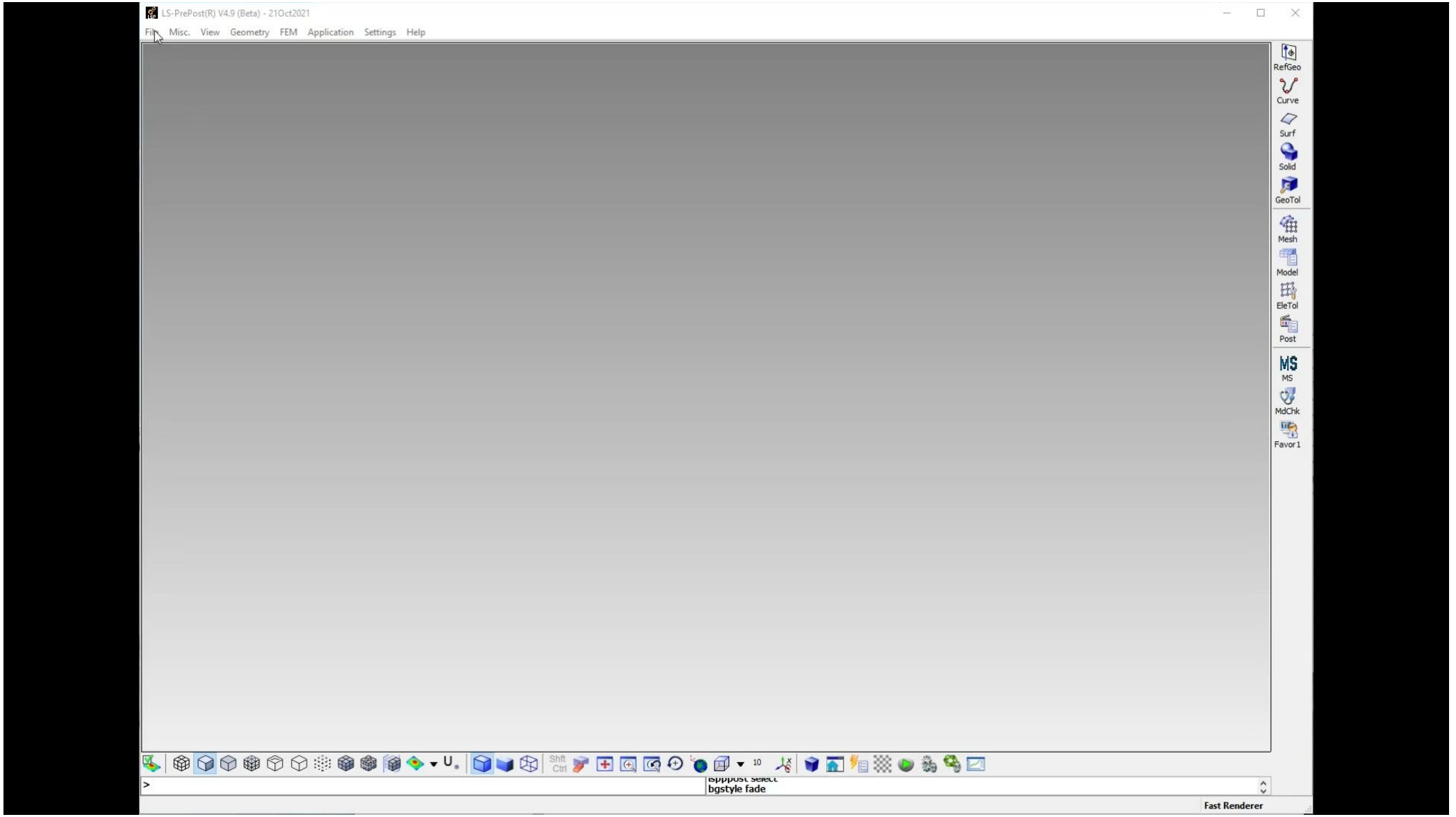
Post-Processing – Introduction of Post Explorer

- Many objects with their properties can be setup in the feature tree
- The Settings for one model can be copied and pasted to other model (if multiple modes have been loaded)
- All settings for the session can be saved to file and be loaded back in the future





Post-Explorer Demo



Powering Innovation That Drives Human Advancement

Work progress of PyDYNA & PyDPF

pyDyna

PyDYNA is a Pythonic package for providing a more convenient and complete way to build an Ansys LS-DYNA input deck, submit it to the Ansys LS-DYNA solver, and finally postprocess the results

https://github.com/ansys/pydyna
https://dyna.docs.pyansys.com

PyDyna

```
git clone https://github.com/pyansys/pyDyna
cd pyDyna
pip install -e .
```

```
unzip ansys-dyna-core-v0.3.5-wheelhouse-Linux-3.8.zip -d wheelhouse
pip install ansys-dyna-core -f wheelhouse --no-index --upgrade --ignore-installed
```

PyDPF

https://github.com/ansys/pydpf-post

Pre

```
from ansys.dyna.core.pre.dynasolution import DynaSolution
solution = DynaSolution(hostname)
solution.open_files(fns)
solution.set_termination(termination_time=10)
ballplate = DynaMech(AnalysisType.NONE)
solution.add(ballplate)
solution.download(serveroutfile,downloadfile)
```

Solver

```
import ansys.dyna.core.solver as solver
dyna=solver.DynaSolver(hostname,port
dyna.push("./output/ball_plate.k")
dyna.start(4)
dyna.run("i=ball_plate.k memory=10m ncycle=20000")
```

Post

```
from ansys.dpf import core as dpf
ds = dpf.DataSources()
ds.set_result_file_path(r'./d3plot', 'd3plot')
resultOp = dpf.Operator("lsdyna::d3plot::stress_von_mises")
result = resultOp.outputs.stress_von_mises()
```

Docker Service

Pre-Service

docker compose up -d

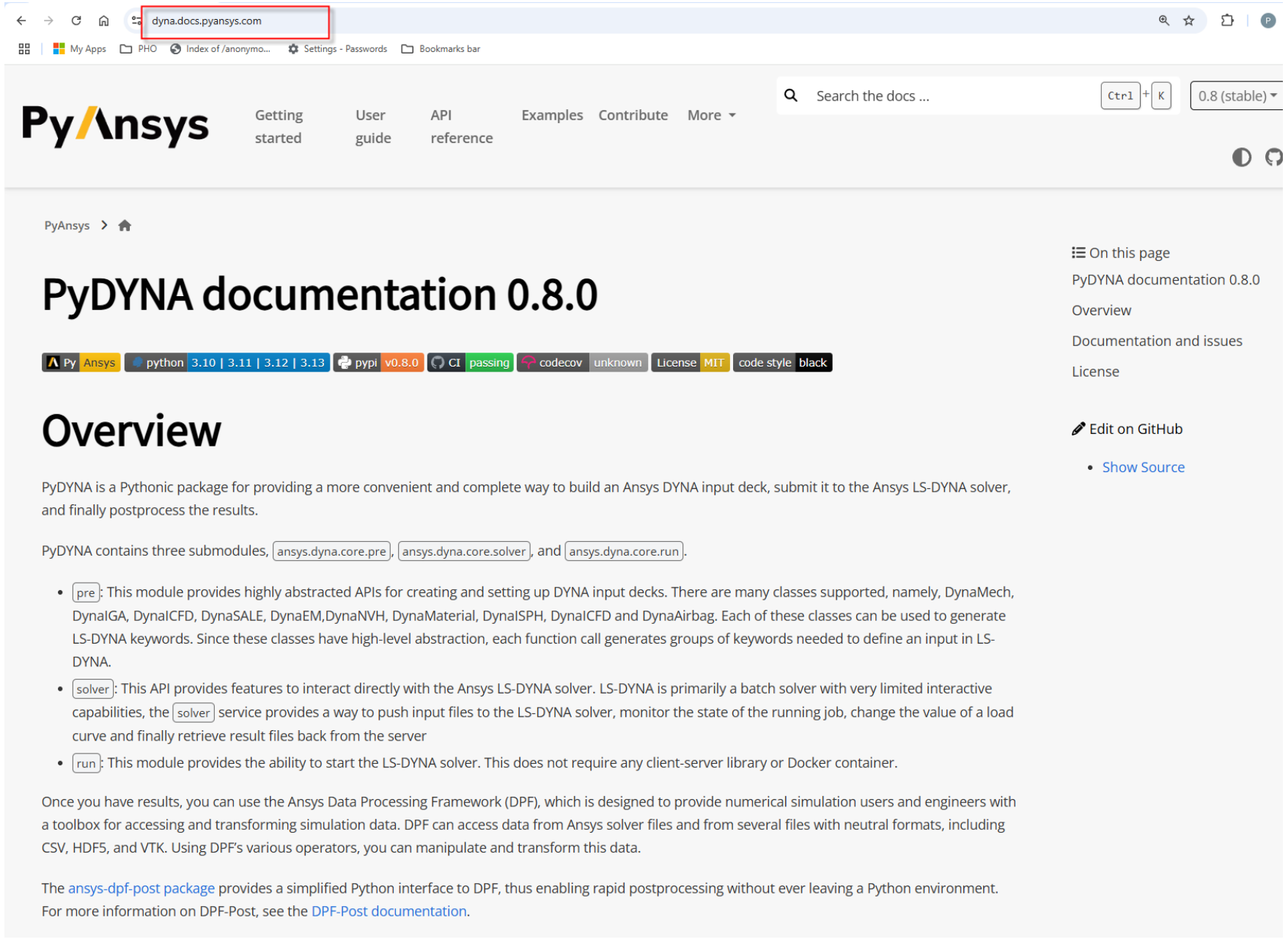
gRPC Architecture

Solver-Service

docker compose up -d

DynaSolution

DynaMech
DynaIGA
DynaICFD
DynaSALE
DynaEM
DynaAirbag



Documentation and issues

Documentation for the latest stable release of PyDyna is hosted at PyDYNA documentation.

For examples on how to use PyDYNA, see Examples in the PyDYNA documentation.

In the upper right corner of the documentation's title bar, there is an option for switching from viewing the documentation for the latest stable release to viewing the documentation for the development version or previously released versions.

On the PyDYNA Issues page, you can create issues to report bugs and request new features. On the PyDYNA Discussions page or the Discussions page on the Ansys Developer portal, you can post questions, share ideas, and get community feedback.

To reach the project support team, email pyansys.core@ansys.com.

License

PyDYNA is licensed under the MIT license.

PyDYNA makes no commercial claim over Ansys whatsoever. This library extends the functionality of Ansys LS-DYNA by adding a Python interface to LS-DYNA without changing the core behavior or license of the original software. The use of the interactive control of PyDYNA requires a legally licensed local copy of LS-DYNA.

For more information on LS-DYNA, see the Ansys LS-DYNA page on the Ansys website.

PyDYNA Workflow

- build an Ansys DYNA input deck
- submit it to the Ansys LS-DYNA solver

interact directly with the Ansys LS-DYNA solver

postprocess the results

For more information about PyDYNA: PyDYNA documentation 0.8.0 PyDYNA

PyDYNA - Start Server locally

Update server package automatically

Start pydyna locally	Windows	Linux
ansys-pydyna-pre-server.zip	Download server package automatically	Download server package automatically
ansys-pydyna-solver-server.zip	Installed the ANSYS locally Download server package automatically	Installed the openmpi package Download server package automatically

PyDYNA - Start Server in Docker

Setup pydyna-pre docker.

Ensure that Docker is installed(Linux: Docker Engine, Windows:
Docker Desktop)

File name	Description
Dockerfile	Build docker image
Linux-binaries.zip	Include library and .proto file,dowload from: https://github.com/ansys/pydyna/releases/download/v0.3.1/linux-binaries.zip
Docker-compose.yml	“docker compose up –d”
README.rst	How to start pydyna-pre container

For more information about PyDYNA: PyDYNA documentation 0.8.0 — PyDYNA

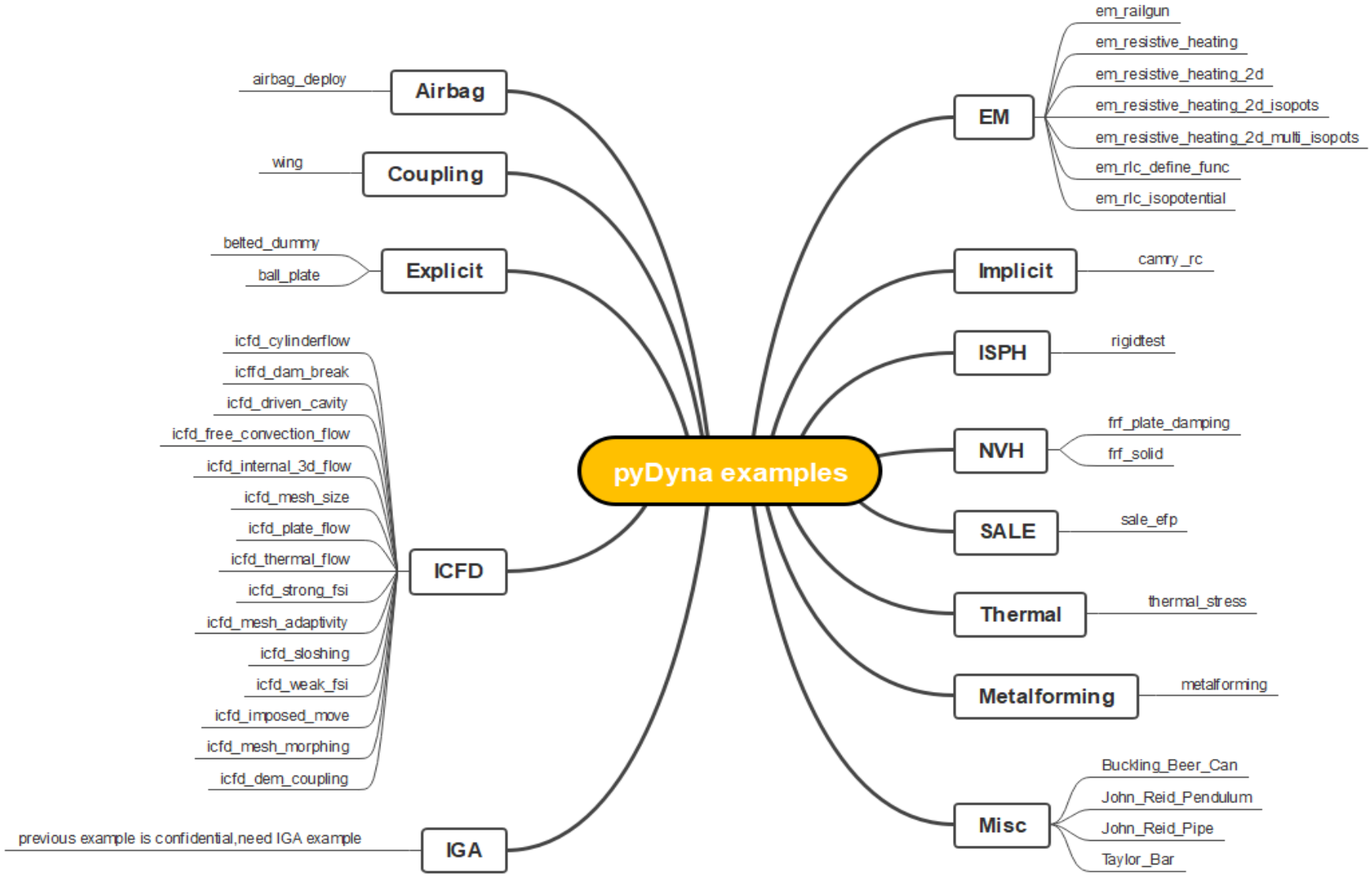
PyDYNA - Start Server in Docker

Setup pydyna-solver docker.

Ensure that Docker is installed(Linux: Docker Engine)

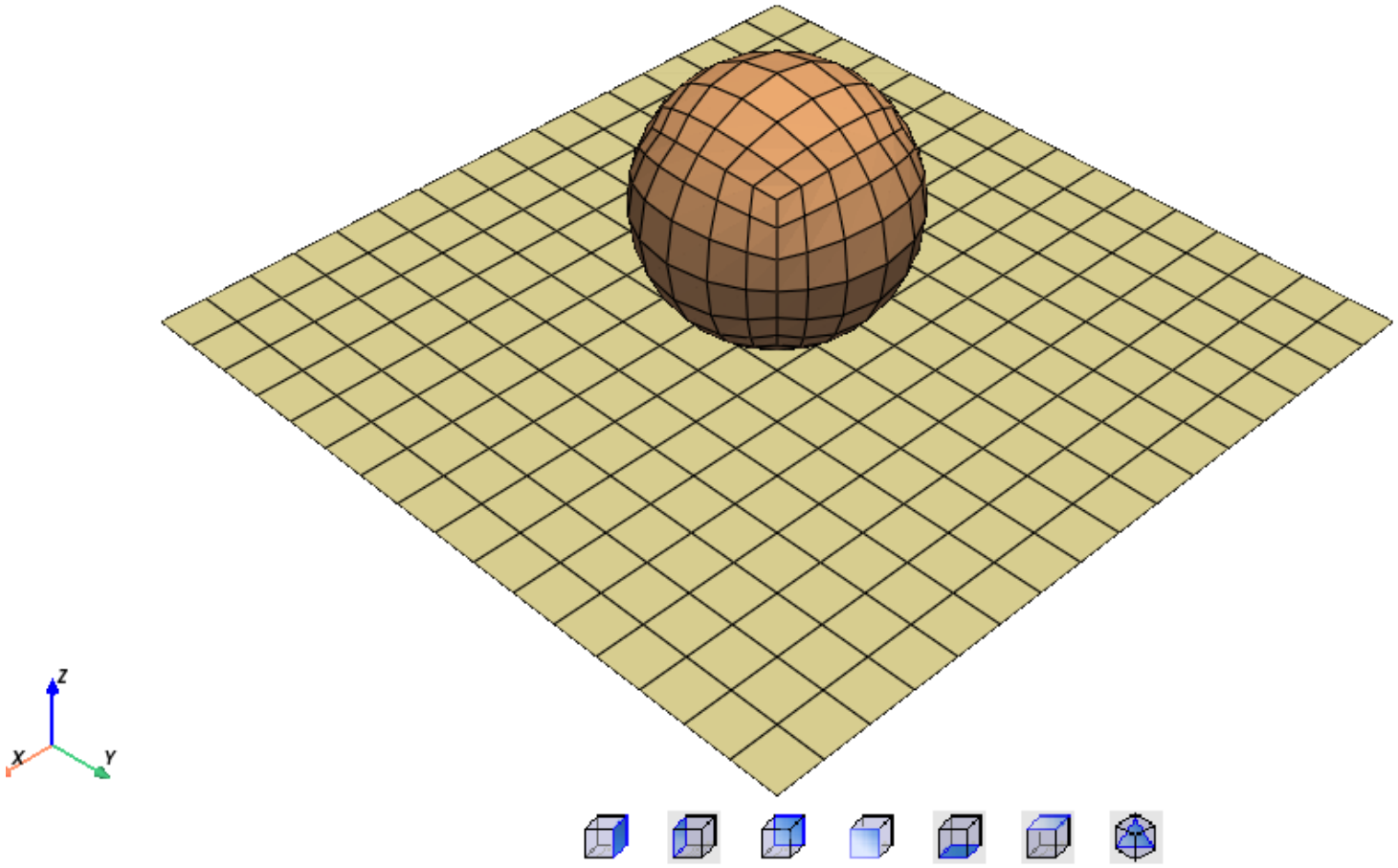
File name	Description
Dockerfile	Build docker image
mppdyna_docker_centos7.zip	Include libraries and .proto file, download from: https://github.com/ansys/pydyna/releases/download/v0.3.1/linux-binaries.zip
docker-compose.yml	“docker compose up –d”
do_build	./do_build
README.rst	How to start pydyna-solver container

PyDYNA – Pre Examples

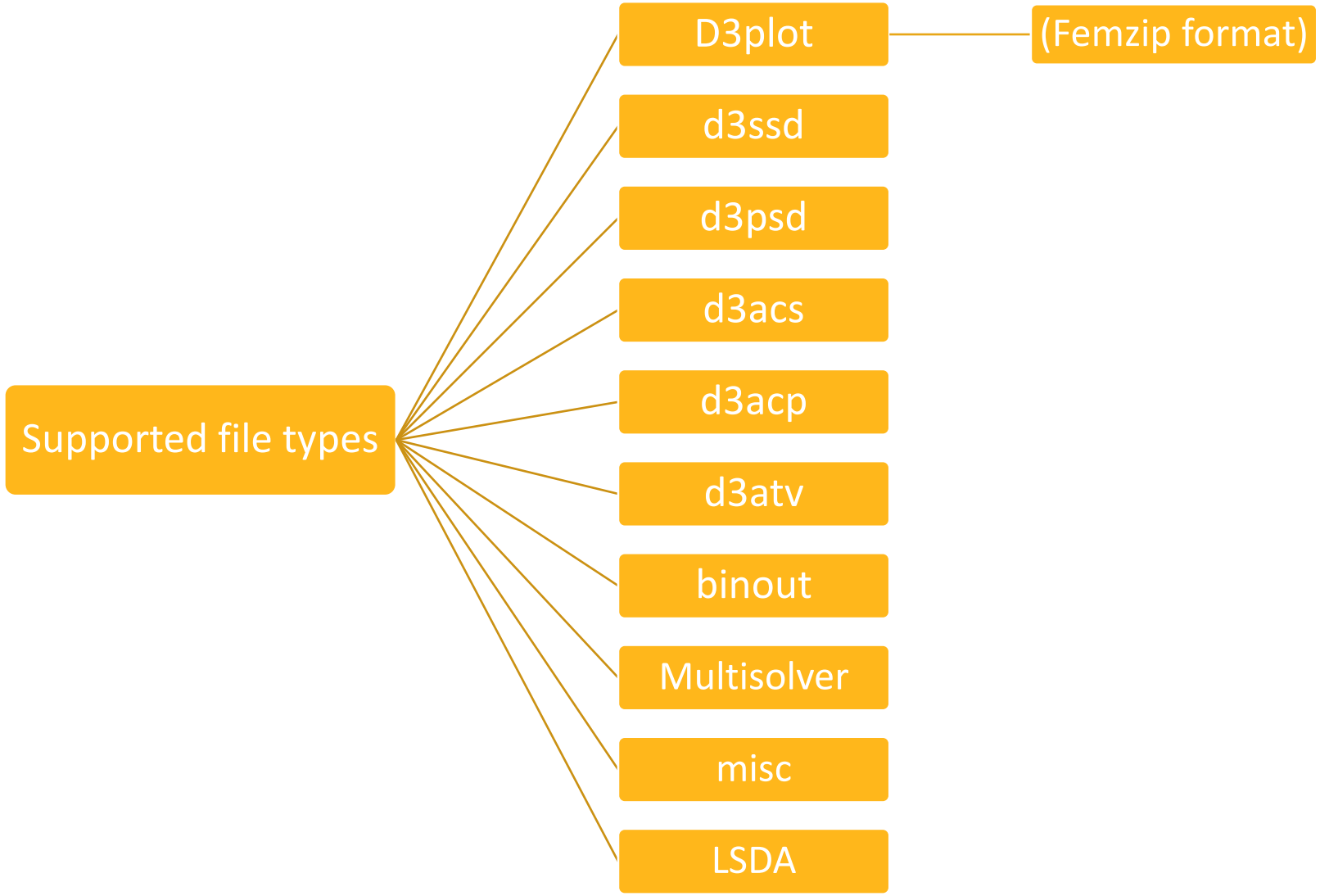


PyDYNA

Draw model: Support draw model with pyvista



PyDPF LSDYNA plugin



PyDPF LSDYNA plugin – Node, Element, global, Shell

Node variable

- initial coordinates
- Coordinates
- Velocity
- Acceleration
- Temperature
- Heat Flux
- node mass scaling
- contact penetration
- relative contact penetration
- contact energy density
- node residual force
- node residual moment
- node temperature divide time

Element variable

- Stress
- Von Mises Stress
- Strain
- Von Mises Strain
- effective plastic strain

Global variable

- Global Kinetic Energy
- Global Internal Energy
- Global Total Energy
- Global Velocity

Shell variable

- Thickness
- Shell X Bending Moment
- Shell Y Bending Moment
- Shell XY Bending Moment
- Shell X Shear Force
- Shell Y Shear Force
- Shell X Axial Force
- Shell Y Axial Force
- Shell XY Axial Force

PyDPF LSDYNA plugin – Shell, Beam, Beam Ipt

Shell variable

- Thickness
- Shell X Bending Moment
- Shell Y Bending Moment
- Shell XY Bending Moment
- Shell X Shear Force
- Shell Y Shear Force
- Shell X Axial Force
- Shell Y Axial Force
- Shell XY Axial Force

Beam variable

- Beam Axial Force
- Beam S Shear Force
- Beam T Shear Force
- Beam S Bending Moment
- Beam T Bending Moment
- Beam Torsional Moment

Beam Ipt variable

- Beam Axial Stress
- Beam RS Shear Stress
- Beam TR Shear Stress
- Beam Axial Plastic strain
- Beam Axial Total strain

PyDPF LSDYNA plugin – d3ssd, d3psd, d3acs, d3acp, d3atv

d3ssd

- Displacement
- Velocity
- Acceleration
- Stress
- Strain

d3psd

- Displacement
- Velocity
- Acceleration
- Stress
- Strain

d3acs

- Sound Pressure Level
- Acoustic Pressure
- Acoustic Intensity
- Normal Velocity

d3acp

- Sound Pressure Level
- Acoustic Pressure
- Acoustic Intensity

d3atv

- Sound Pressure Level
- Acoustic Pressure

PyDPF LSDYNA plugin – Multi-Solver,Misc

Multisolver (EM,ICFD)

- Fluid velocity
- Scalar potential
- Cur density
- EM Cur
- Level Set
- Viscosity
- Fluid temperature

Misc

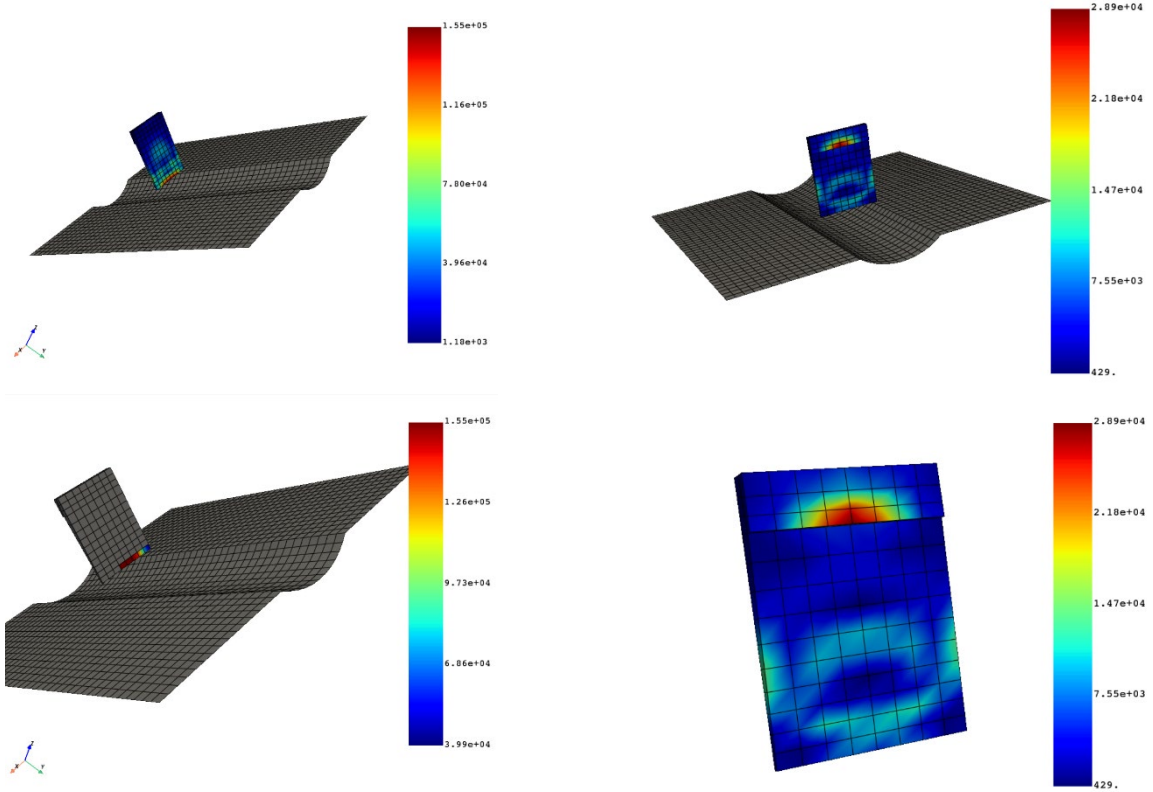
- History variable
- Erosion data
- Part IDS

PyDPF LSDYNA plugin - Binout branch

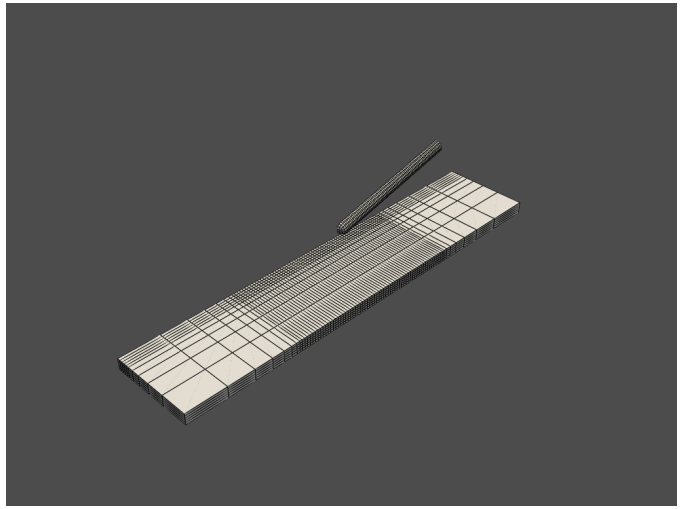
glstat	matsum	reforc	icvout
rwforc	secforc	swforc	nodout
abstat	sbtout	jntforc	spcforc
bndout	rbdout	sleout	Elout (under developing)

DPF (Data Processing Framework) LS-DYNA PlugIn

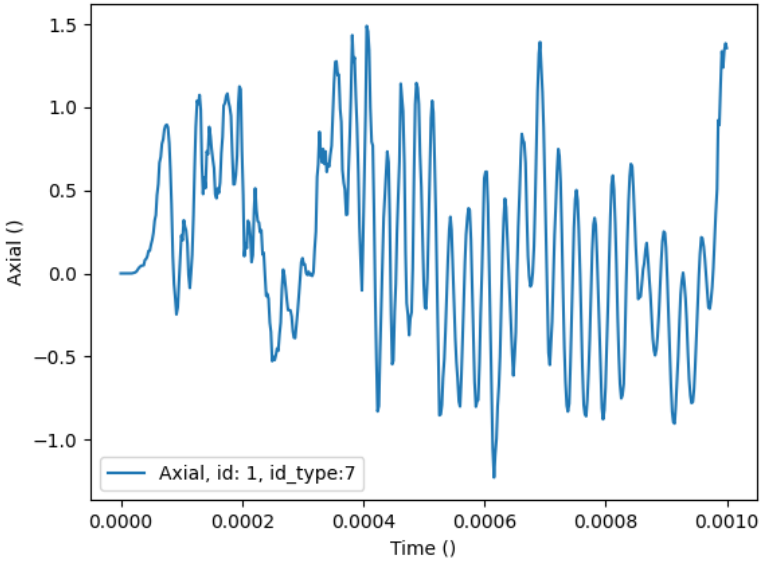
- **Fringe Results (scoping, by part)**



- **Adaptive Model**



- **Binout**



Thanks!