SAND2010-3655C

Advanced Conceptual and Numerical Methods for Modeling Subsurface Processes Regarding Nuclear Waste Repository Systems

Sierra Mechanics Overview

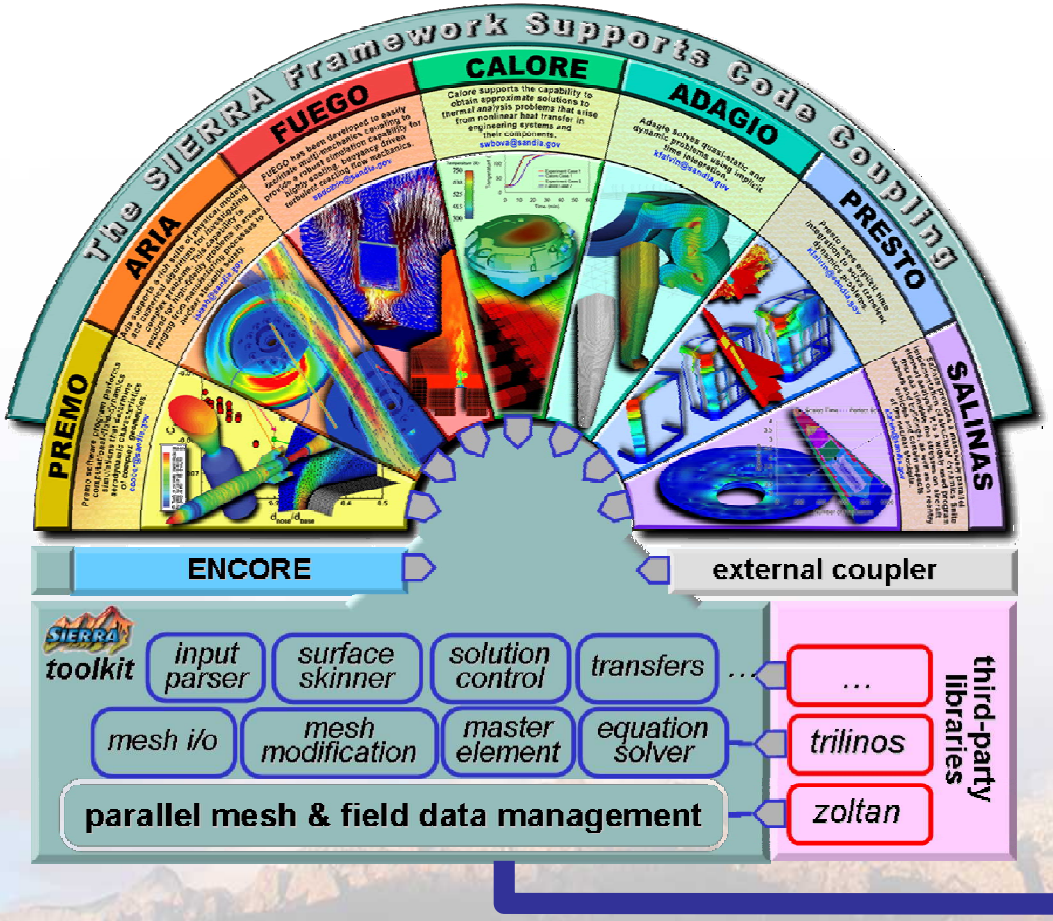
J. Guadalupe Argüello
Computational Structural Mechanics and Applications

Sandia National Laboratories
Albuquerque, NM, USA

SIERRA Mechanics overview

DOE ASCI program funded for ~10 years

Massively parallel multi-physics capabilities for Sandia's engineering science mission



Designed and developed for MP hardware

SNL Red Janus

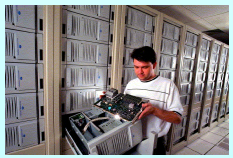
LLNL Blue Pacific

LANL Blue Mountain

LLNL White

LANL Q

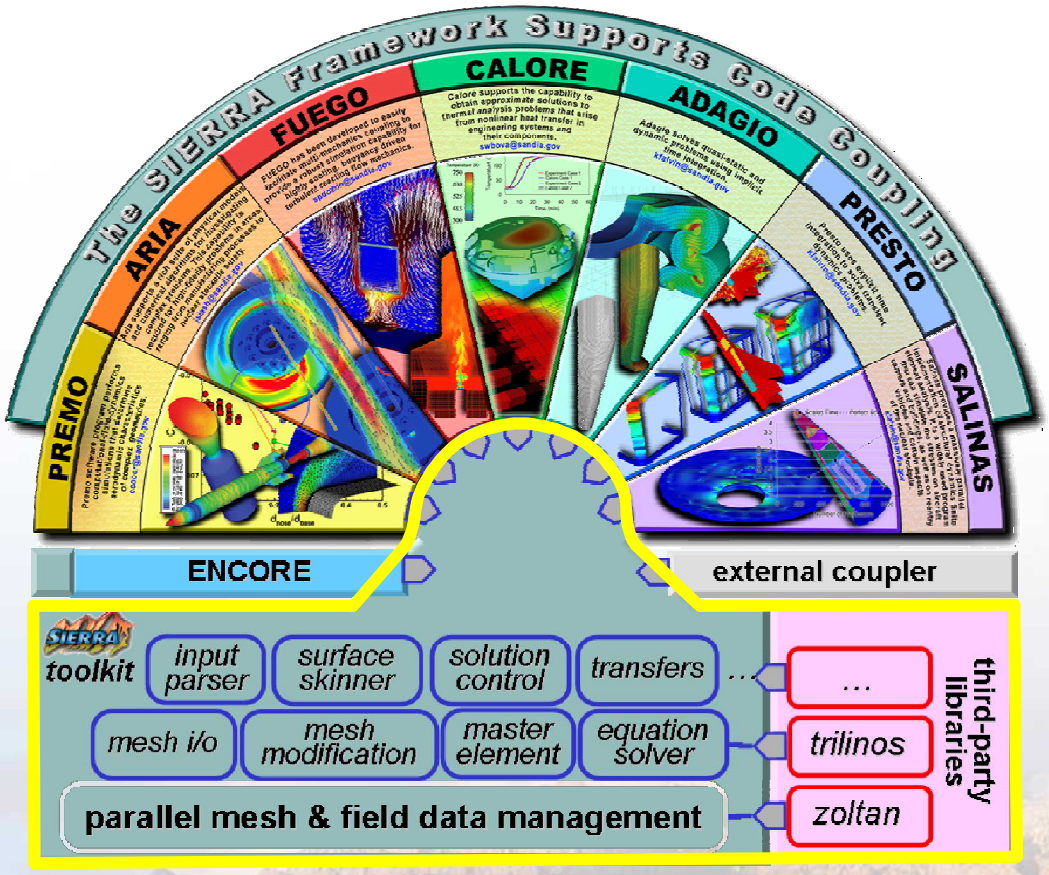
SNL Red Storm



Linux Clusters

SIERRA Mechanics foundation

SIERRA_toolkit
FE application code services



Capabilities:

Mesh & field data management (parallel, distributed)

Services provided to Mechanics applications

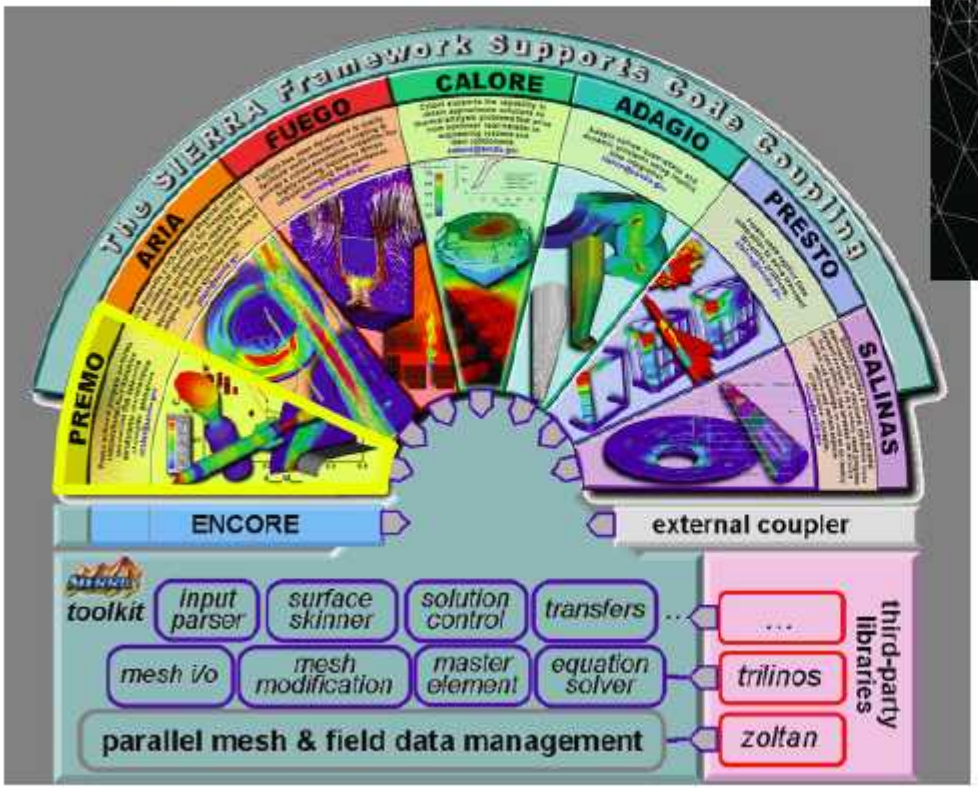
Transfer operators for mapping field variables from one mechanics to another

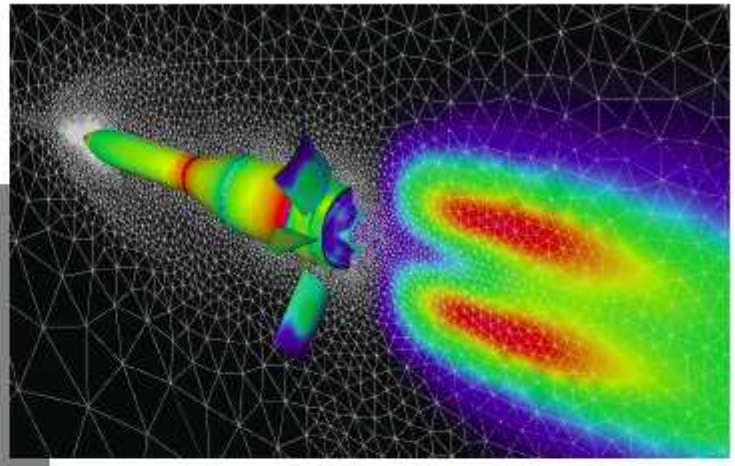
Solution controller for code coupling

Includes third party libraries (e.g. solver libraries, MPI communications package)

Sierra Mechanics modules

SIERRA_Premo
Compressible Fluid Mechanics





Capabilities:

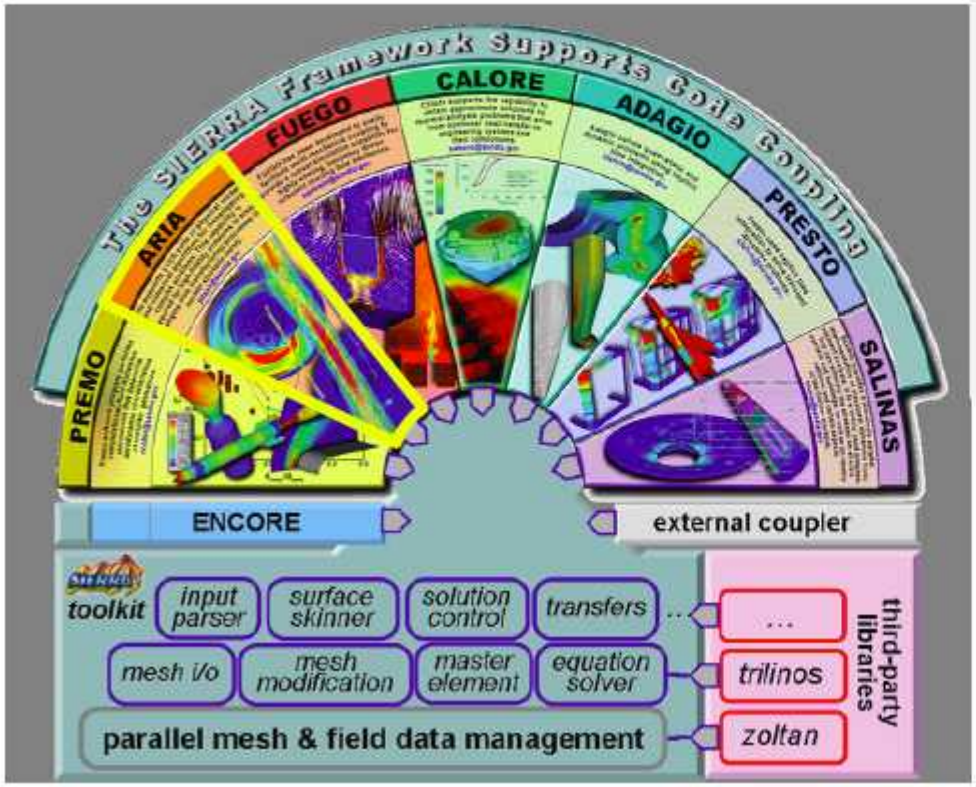
Subsonic through hypersonic

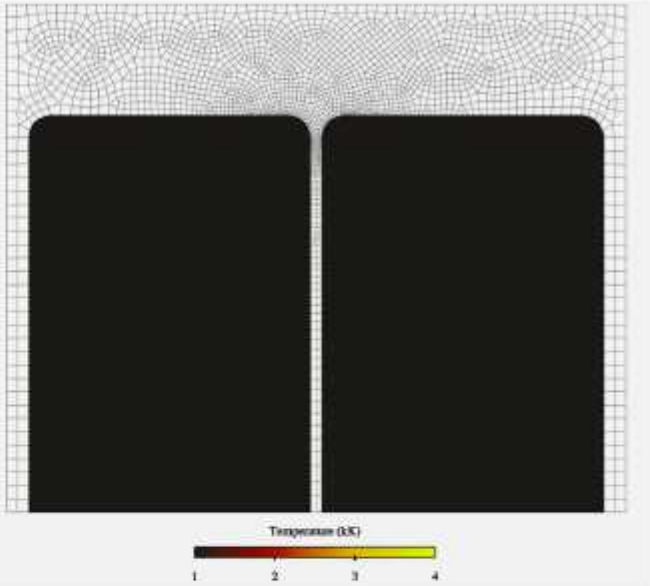
Laminar and turbulent

Unstructured mesh

Sierra Mechanics modules

SIERRA_Aria
Non-Newtonian flow





Capabilities:

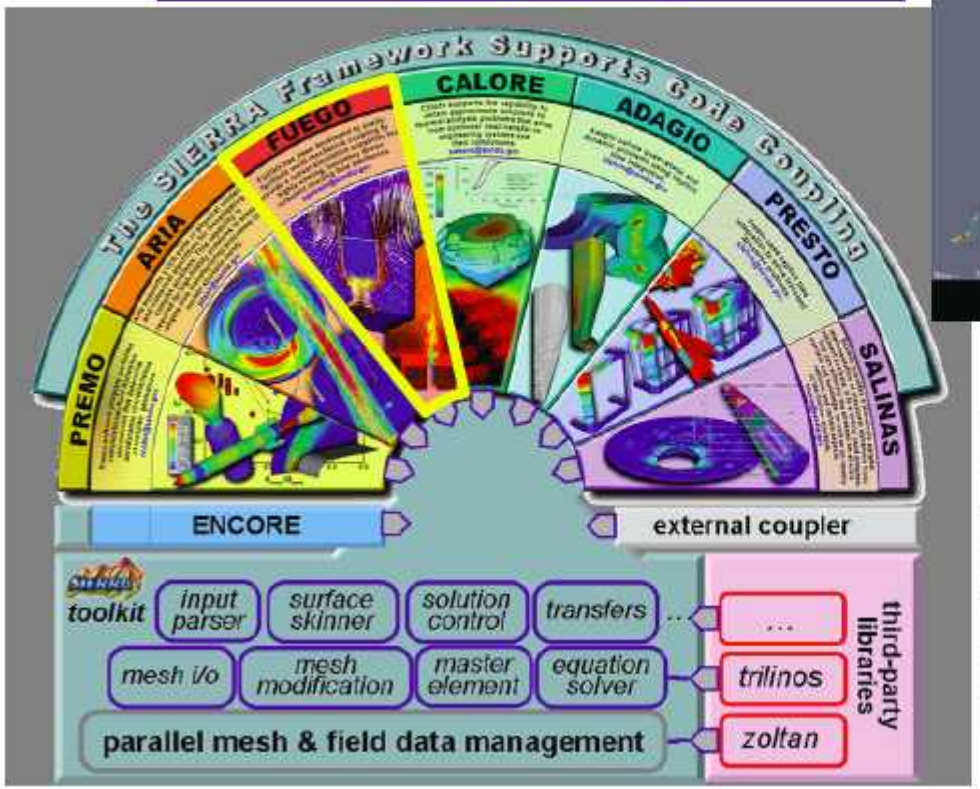
Chemically Reacting flows, Level sets for free-surface tracking

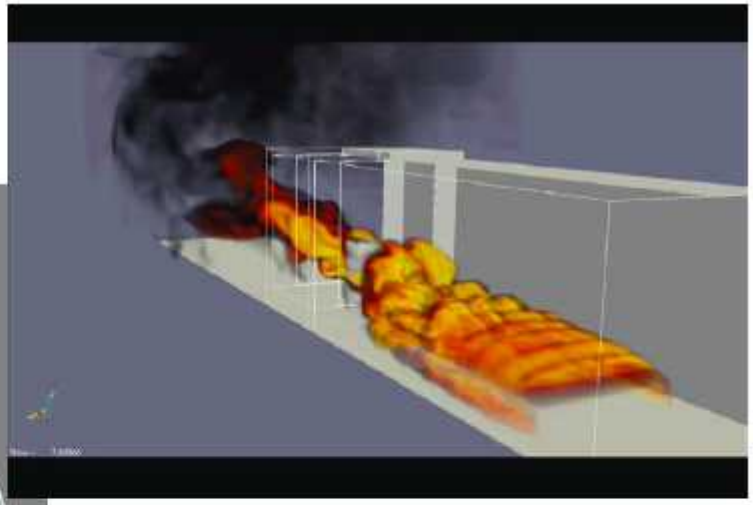
Complex material response

Basic Physics: Navier-Stokes (variable ρ), Energy, Species, Electrostatics

Sierra Mechanics modules

SIERRA_Fuego
Low Mach number, finite-volume fluid dynamics





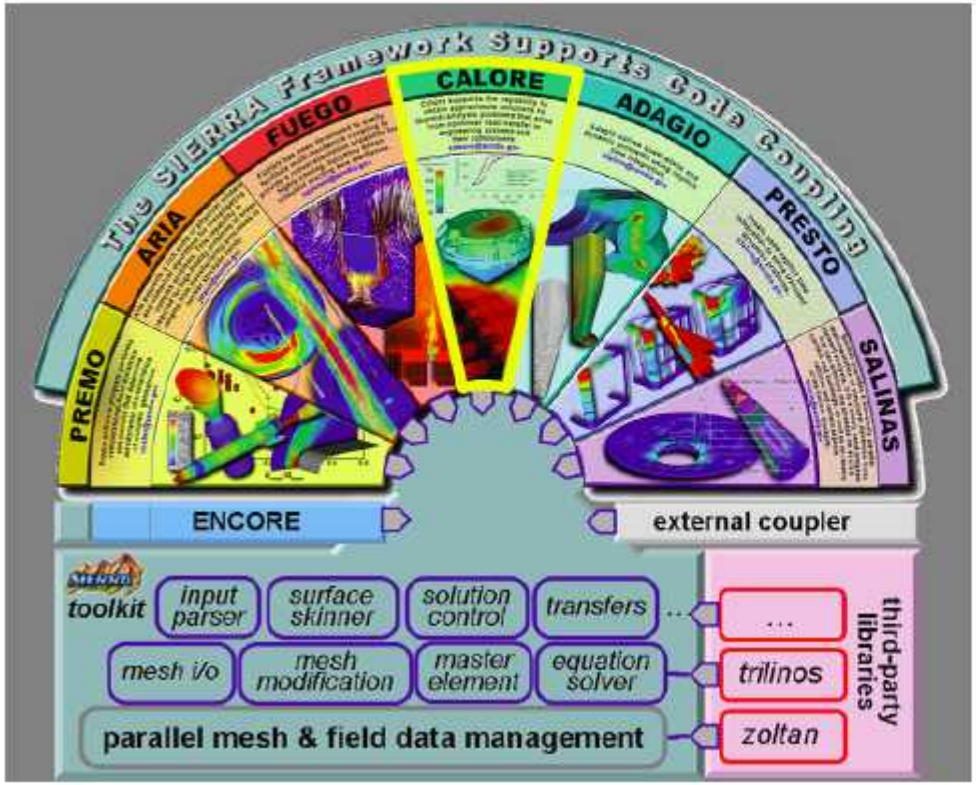
Problem description:
Hydrocarbon pool fire, 400M DOFs, 5000 procs. on Red Storm

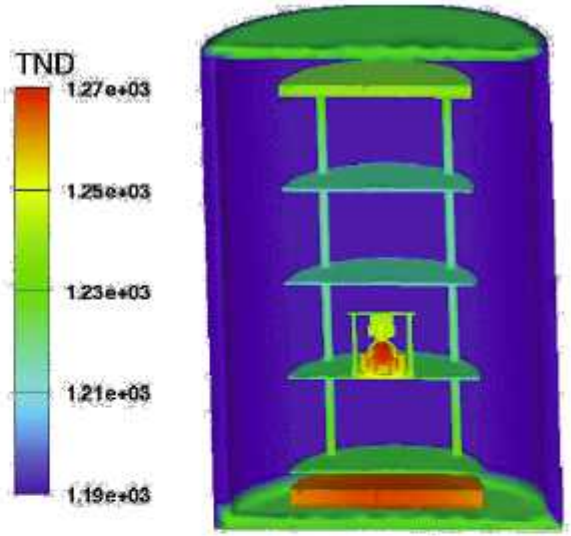
Capabilities:

Turbulent reacting flow with coupling to participating media radiation and heat conduction, RANS and LES-based turbulence models

Sierra Mechanics modules

SIERRA_Calore
Heat transfer





Problem description:

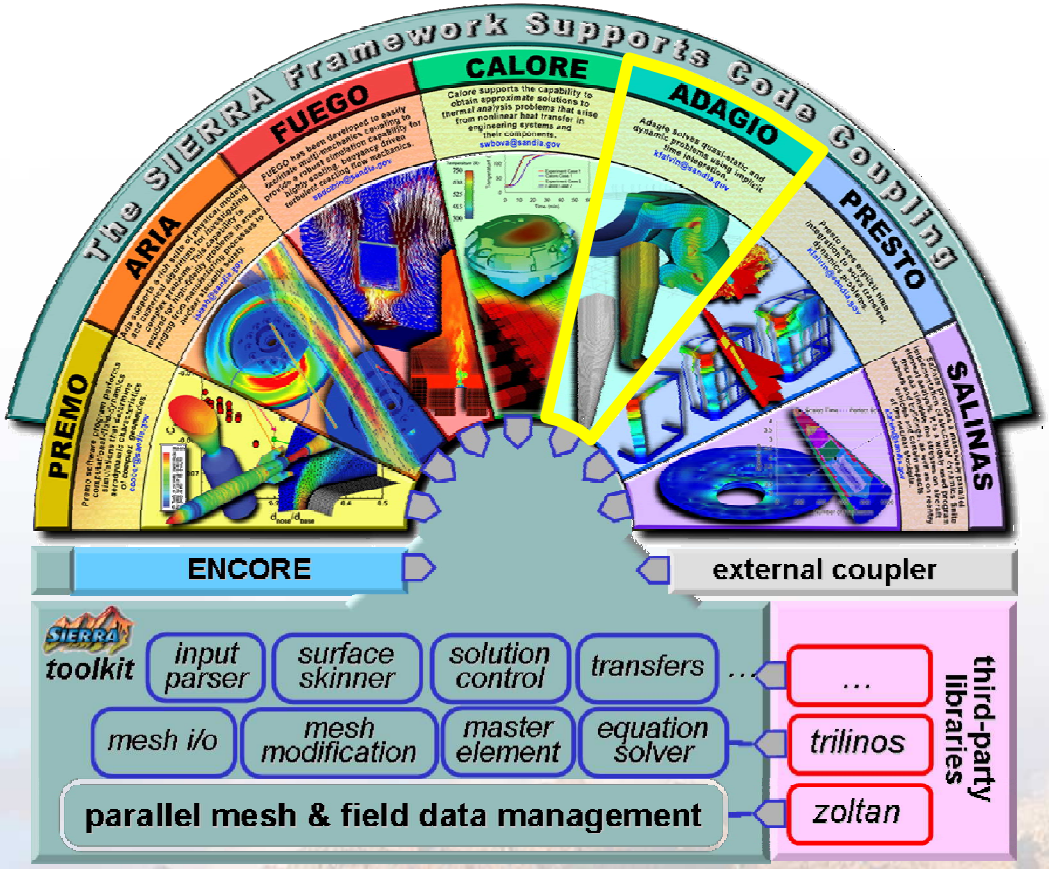
Braze furnace process optimization

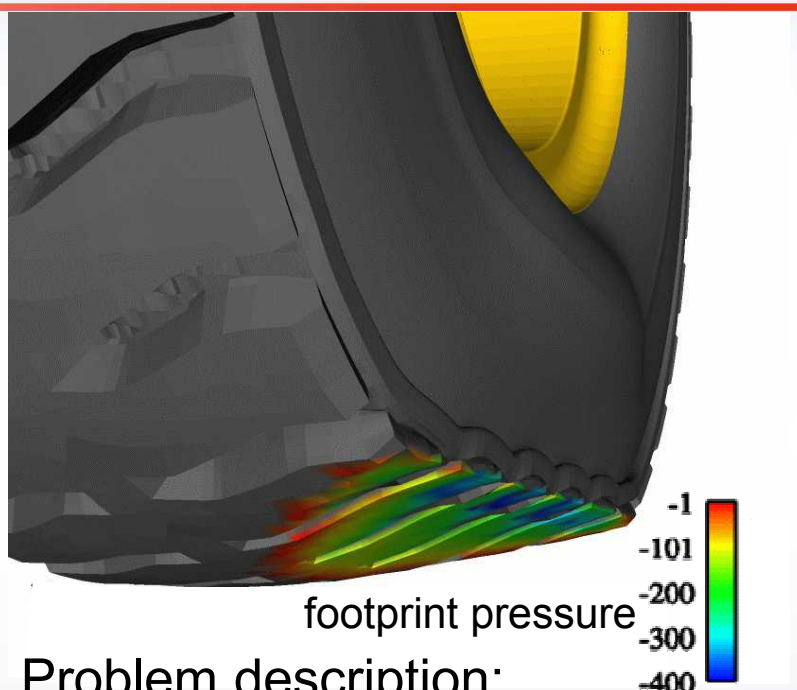
Capabilities:

Steady/Unsteady, conduction, limited convection, chemistry, enclosure radiation, thermal contact, h-adaptivity

SIERRA Mechanics modules

SIERRA_Adagio
Non-linear Solid mechanics





footprint pressure

Problem description:

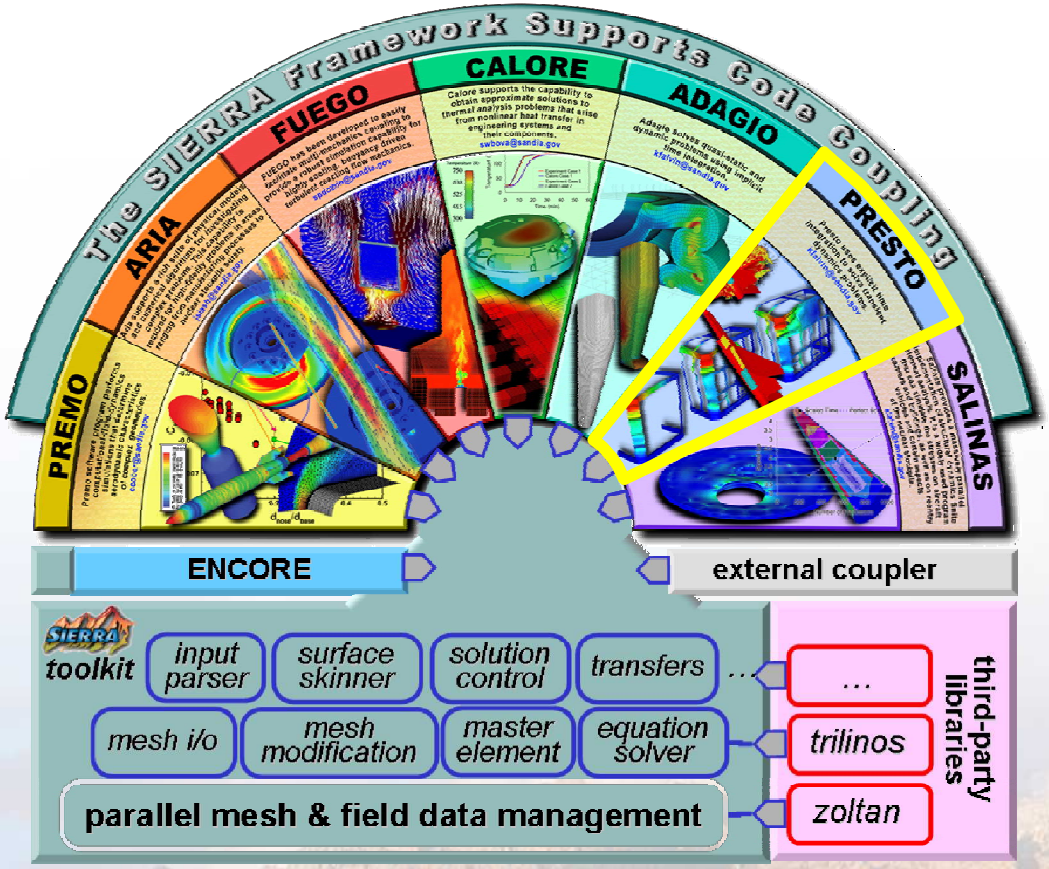
Tire performance modeling, camber angle = 18°

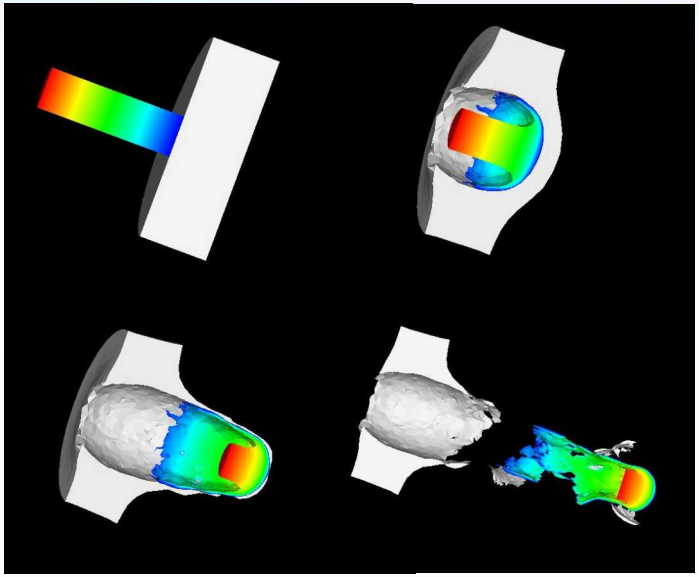
Capabilities:

Quasistatics, implicit dynamics, Parallel non-linear implicit solvers, contact, >50 material models, failure & tearing

SIERRA Mechanics modules

SIERRA_Presto
Non-linear Solid transient dynamics





Problem description:

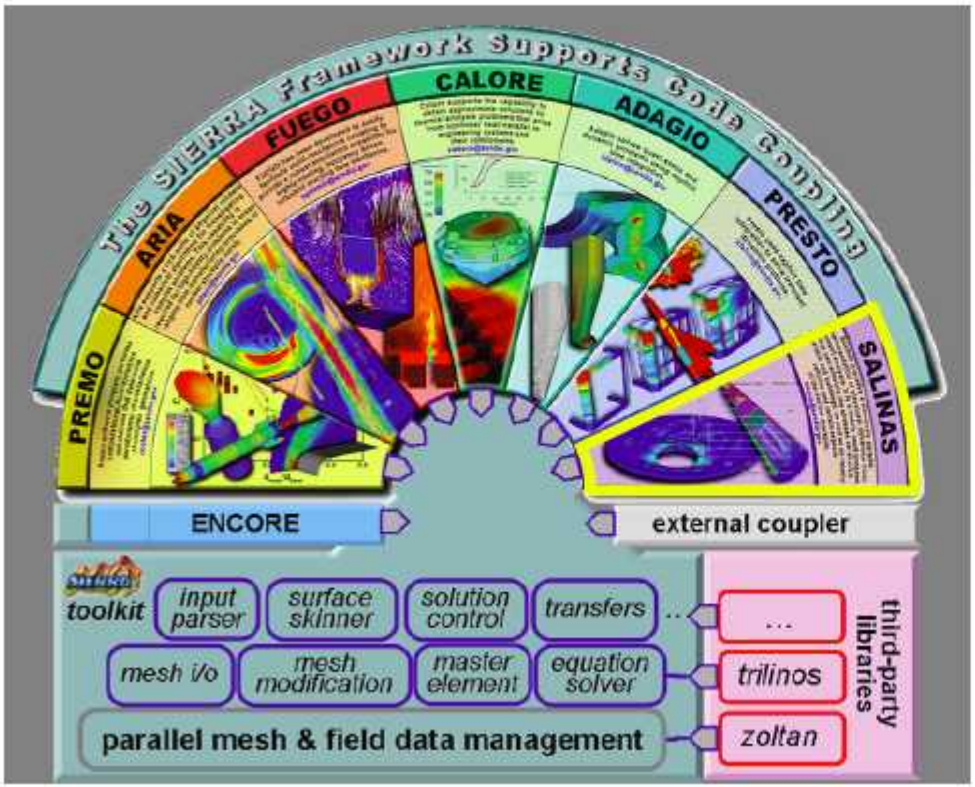
Tungsten rod impacting steel plate @2500 m/s

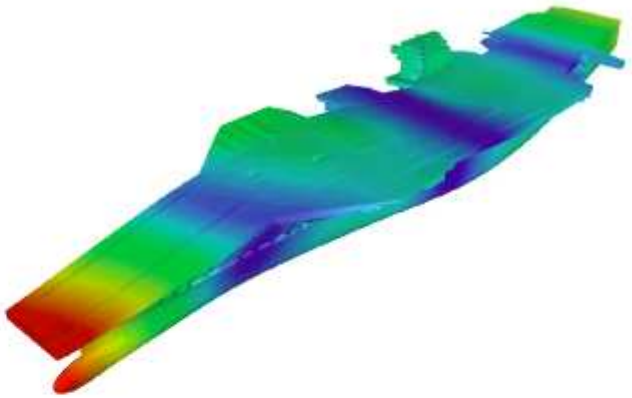
Capabilities:

Explicit time integrator, Nodal-based tet w/ remeshing, particle methods, cohesive surface elements, spot welds, contact, material failure

SIERRA Mechanics modules

SIERRA_Salinas
Linear structural dynamics





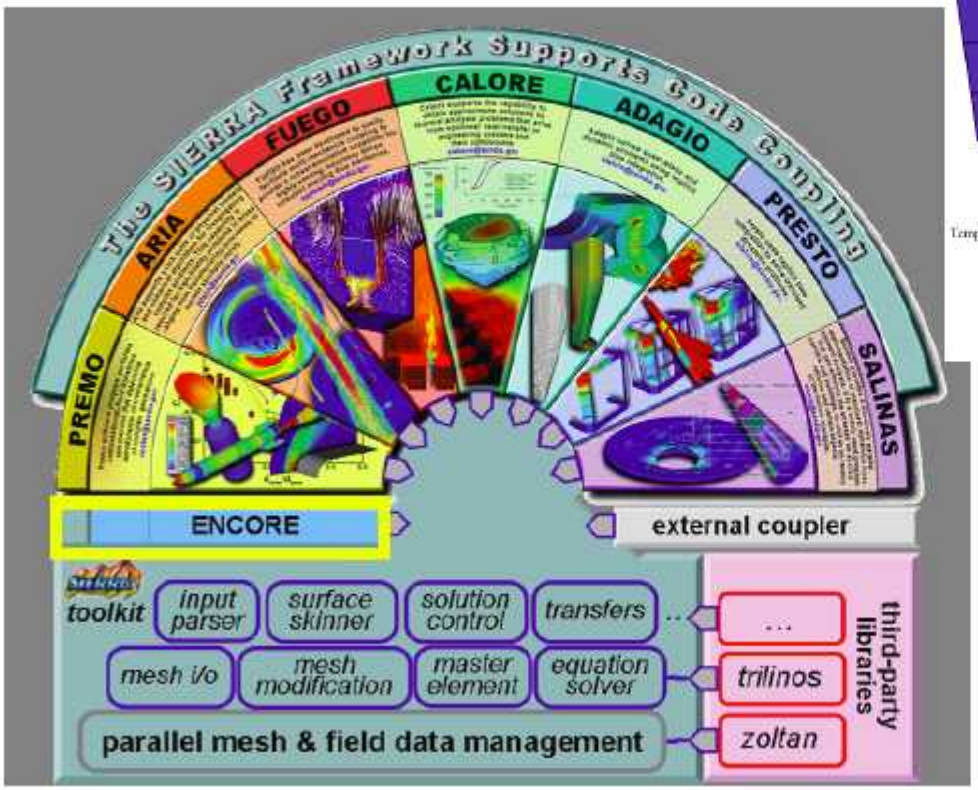
Problem description:

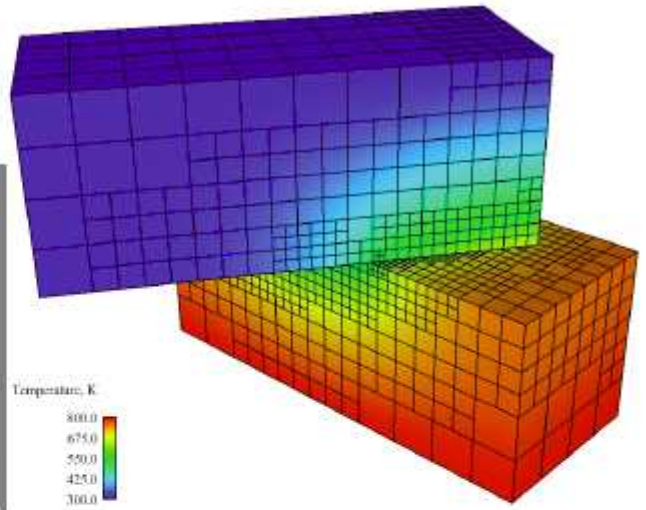
Modal solution of complex structures (1000's of material regions, offset shells and beams)

2.0M DOFs, solved on 64 processors

Acoustics

SIERRA_Encore
FE Mapping, FE error measures





Capabilities:

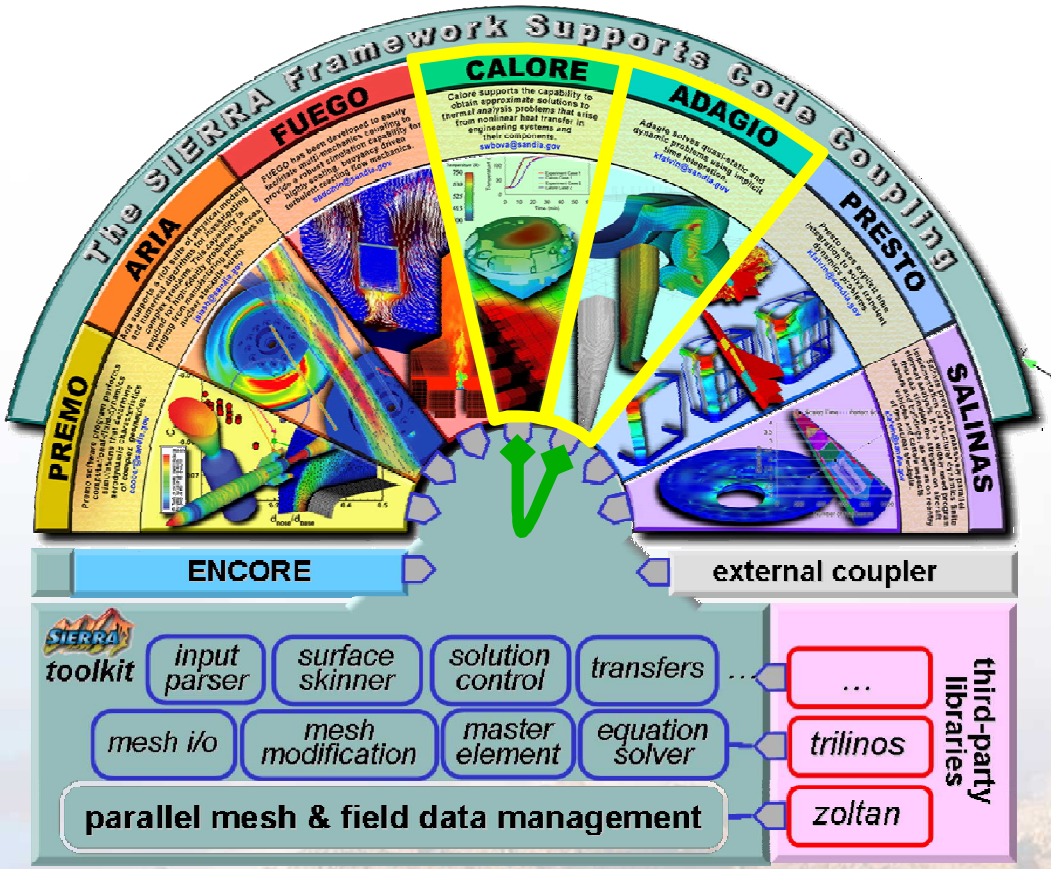
General mapping capability between dissimilar meshes (on same geometry, e.g. adaptivity)

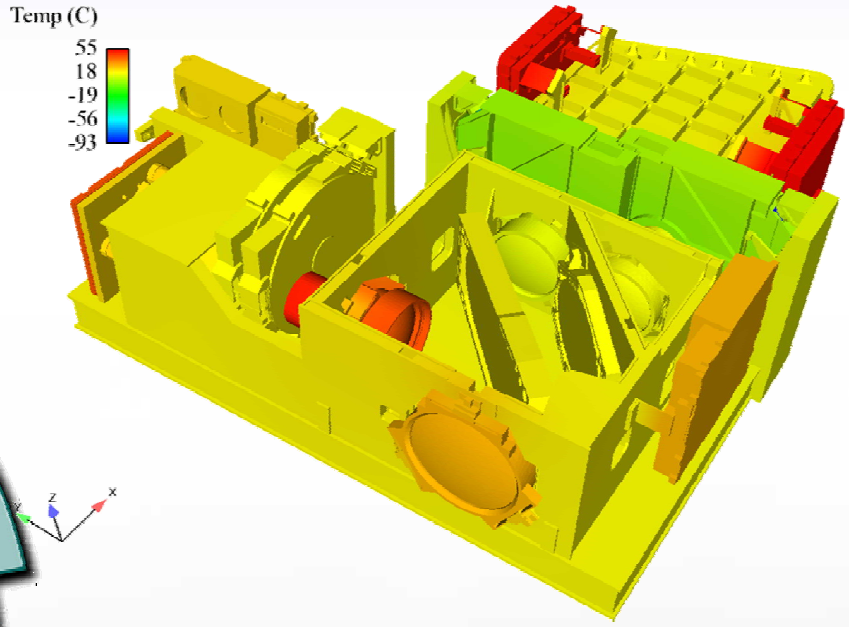
Differencing multiple FE solutions (convergence)

Error measures (field variable norms)

SIERRA Mechanics coupling

SIERRA_CoupledMechanics
Thermo-Structural Response



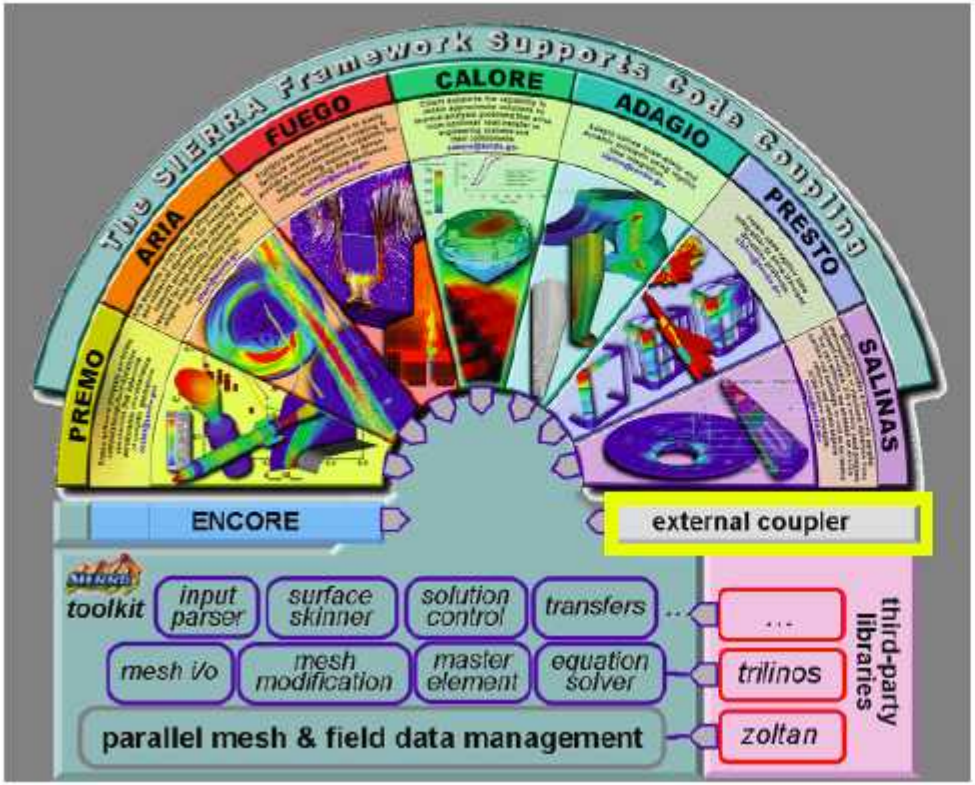


Problem Description:

Coupled Thermal-Mechanical response of a satellite Optical Bench Assembly subjected to normal orbital thermal loads and onboard heating

SIERRA Mechanics coupling

SIERRA_external coupler
coupling to external codes



Capabilities:

Accesses parallel mesh & field data management for transfer of SIERRA nodal and element variables to/from external code

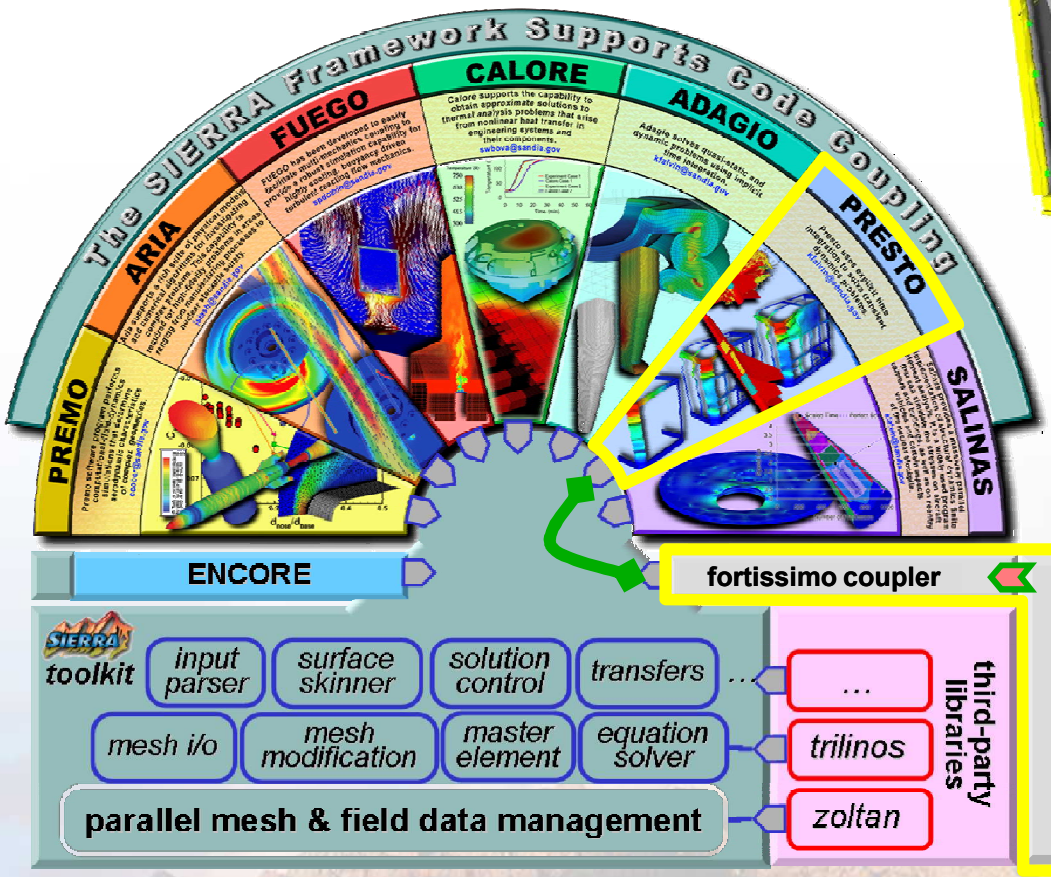
Implementation specific to external code as needed

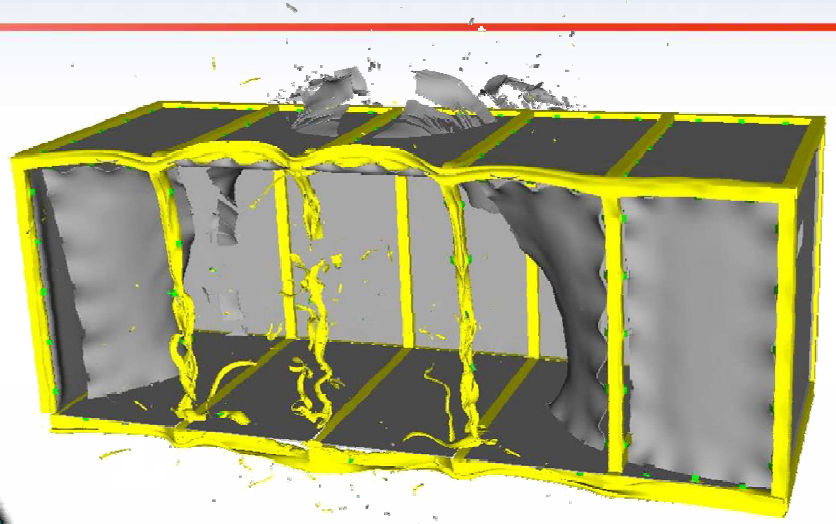
Note: this allows access to all SIERRA mechanics modules

SIERRA Mechanics coupling

SIERRA_Fortissimo

Blast-Structural failure Response





Problem Description:

1-way & 2 way coupled Blast-Structural calculation of a welded 5-panel steel box, predicting failure mechanics & post failure response

CTH capability: shock-hydrodynamics

SIERRA_SolidMechanics : capabilities (sampling)

Single Physics

- Nonlinear stress response (w/ multi-level solvers)
- Interface problems (contact)
- Large deformations (remeshing, particle methods)
- Failure Modeling (explicit & implicit)
- Multi-scale modeling

Coupled Physics

- quasistatics↔transient dynamics↔structural dynamics
- Thermal-structural response (e.g. residual stress)
- Fluid Structure Interaction
- Energy deposition (e.g. blast-structure, ground shock)

Analysis Process: Overview

As in all analysis processes, there are three steps:

1. Pre-processing: Mesh & input file creation
2. Analysis execution: Running the analysis
3. Post-processing: Visualizing the results

These steps are not *yet* in a single UI for SolidMechanics

Analysis Process: Pre-Processing

Pre-processing must produce two inputs for analysis execution:

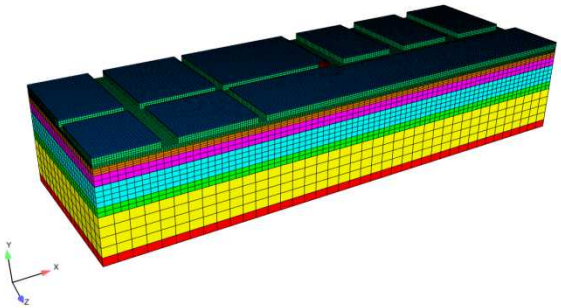
1. mesh file
   …is a binary file(s) in the Genesis/Exodus format derived off of HDF5

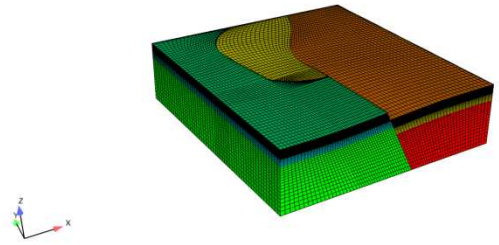
2. input file
   …is a text file

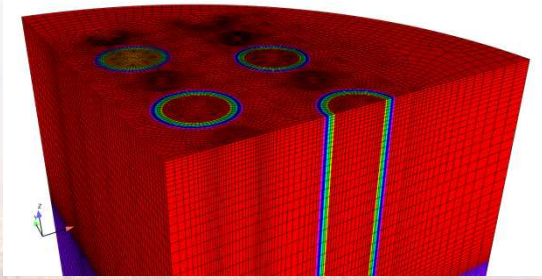
Analysis Process: Pre-Processing

Mesh file generation

- ✓ Cubit (licensed)
  Sandia's current meshing tool
- – Patran (licensed)
  can convert from other codes:
  LSDyna, Abaqus
  Sandia's "exodus preferences"
  are available
- – SEACAS (available from Sandia)
  Sandia legacy meshing tools –
  Fastq, Gen3D, Grepos,
  SPHGEN, etc.







Analysis Process: Pre-Processing

Input file generation

- emacs, xemacs, vi, or any other plain text editor
- SierraEditor is a context specific input file editor that understands Sierra-specific input (using XML file) will highlight, do word completion, provides on-demand help, etc…)
  - Cubit plug-in (under development) understands Sierra-specific input (using HTML file) will allow selection & use of nsets, ssets, etc… in generating input file

Analysis Process: Input File Anatomy

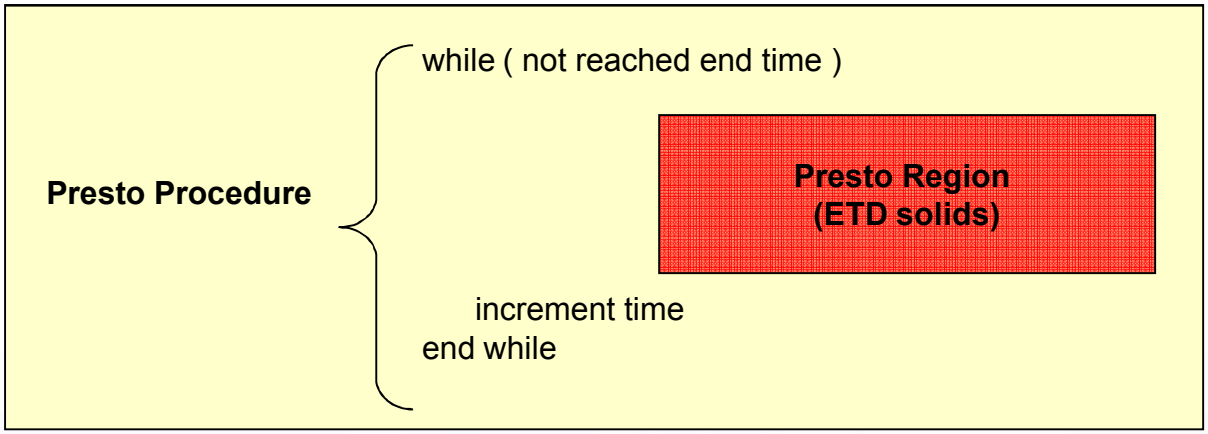
The input file for all Sierra codes follow a structure adopted to facilitate loose multi-physics coupling

Primary parts:

- a **procedure** advances time from a start to end time by driving **regions** and **transfers** data between them (loose coupling)
- a **region** computes physics for a single time increment – can be single-physics or multi-physics (tight coupling)
- a **transfer** maps nodal/element state data between **regions**

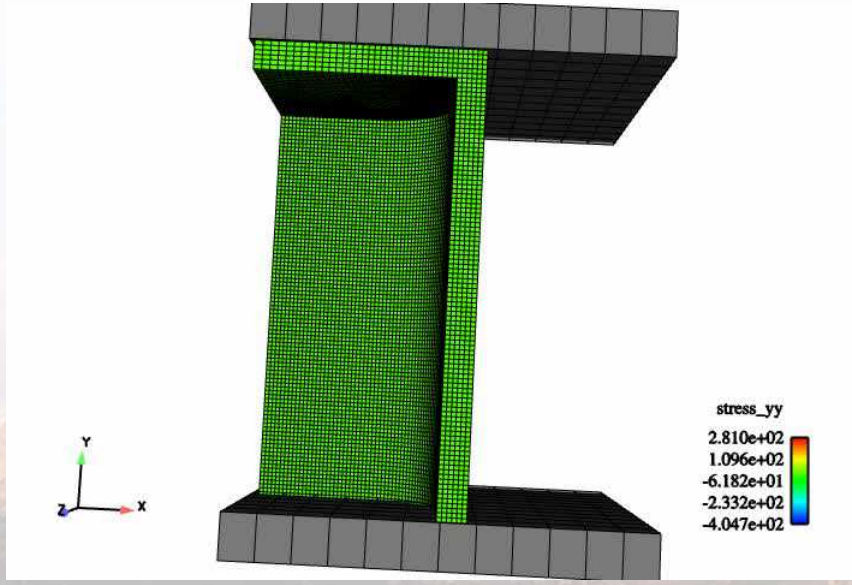
Analysis Process: Input File Anatomy

Example: Explicit transient dynamics analysis (single physics)



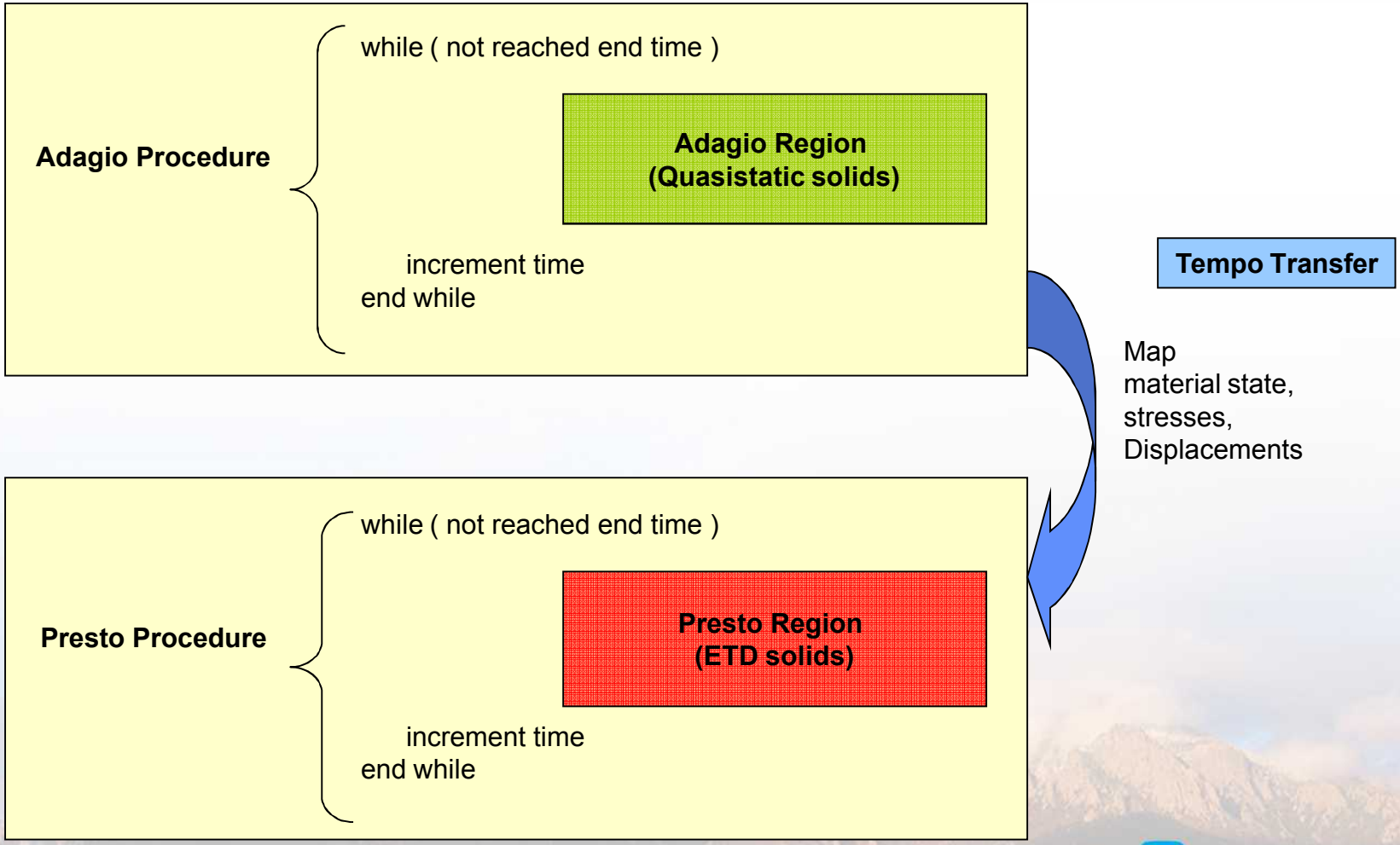
Analysis Process: Input File Anatomy

```
begin sierra <my_analysis_name>
  … functions, materials, mesh file to read …
  begin presto procedure <my_procedure_name>
    … time control definition …
    begin presto region <my_region_name>
      … presto physics definition …
    end
  end
end
```



Analysis Process: Input File Anatomy

Example: Pre-load analyses (a one-time “hand-off”)

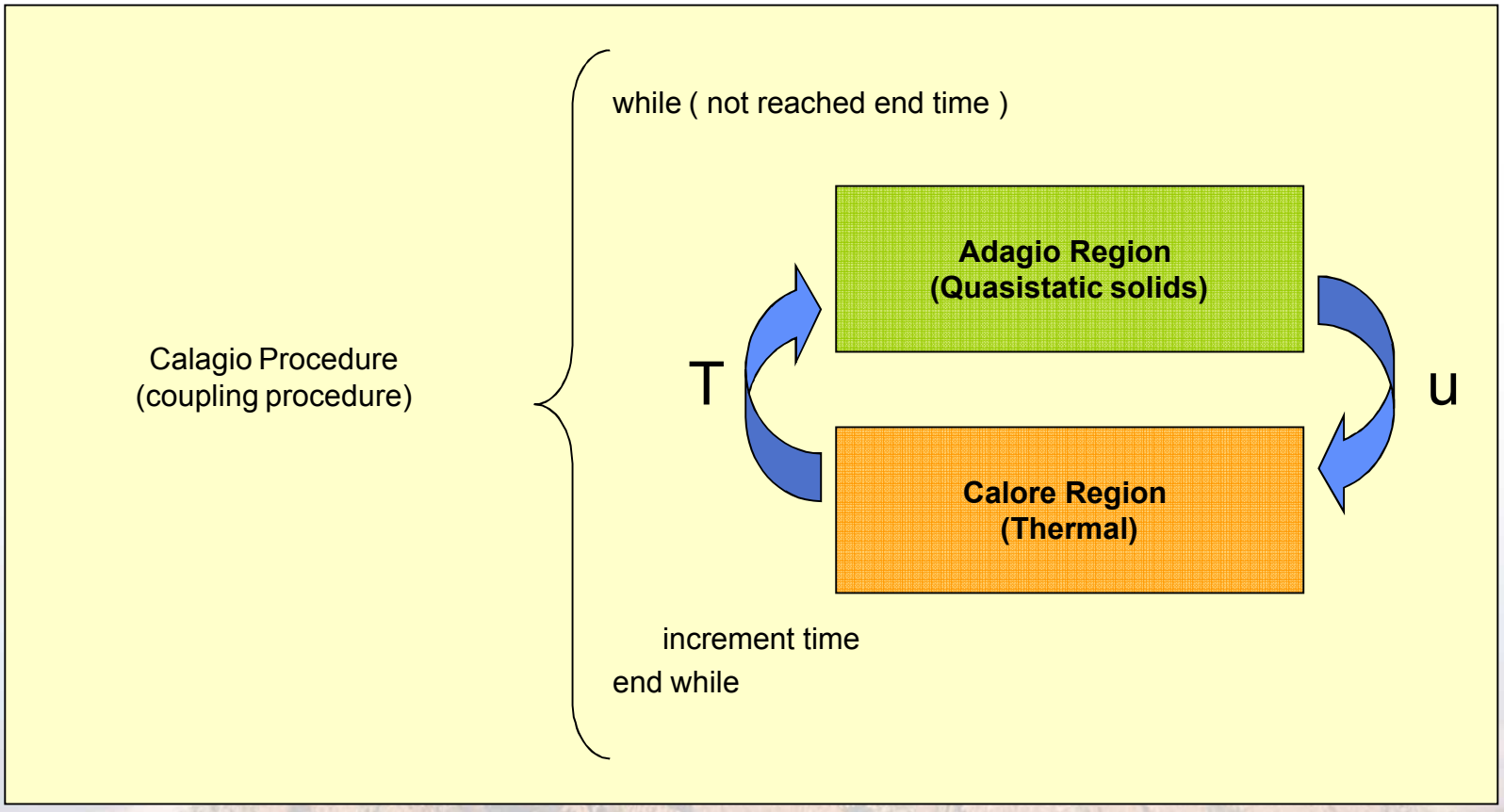


Analysis Process: Input File Anatomy

```
begin sierra <my_analysis_name>
  … functions, materials, mesh file to read …
  begin adagio procedure <my_procedure_name>
    … time control definition …
    begin adagio region <my_region_name>
      … adagio physics definition  …
    end
  end
  begin presto procedure <my_procedure_name>
    … transfer description …
    … time control definition …
    begin presto region <my_region_name>
      … presto physics definition  …
    end
  end
end
```

Analysis Process: Input File Anatomy

Example: Thermal-Structural modeling (a loose coupling)



Analysis Process: Input File Anatomy

```
begin sierra <my_analysis_name>
  … functions, materials, mesh file to read …
  begin calagio procedure <my_procedure_name>
    … time control definition …
    begin adagio region <my_adagio_region_name>
      … adagio physics definition …
    end
    … transfer description …
    begin calore region <my_calore_region_name>
      … calore physics definition …
    end
  end
end
```

Analysis Process: Analysis Execution (running Adagio, Presto)

Execution is typically done through the “Sierra” script:

sierra -V <version**> adagio -i <**input file**> -j <**num procs**>**

sierra -V <version**> presto -i <**input file**> -j <**num procs**>**

-V <version>	must be a known version that was part of the install
adagio , presto	refers to the Sierra mechanics (physics) module to execute
-i <input file>	refers to the ASCII text file
-j <num procs>	refers to the number of processors the code execution will use

Analysis Process: Analysis Execution (running Adagio, Presto)

Execution is typically done through the “Sierra” script:

sierra -V <version**> adagio -i <**input file**> -j <**num procs**>**
sierra -V <version**> presto -i <**input file**> -j <**num procs**>**

Other flags:

-Q <queue> (for batch submittal)

-T <time for queue> (for batch submittal)

-X <location of exe>/<name of exe> (using a specific executable)

-x <project with exe> (for developers)

-a (run aprepro input file pre-processor)

-h (help page)

Analysis Process: Analysis Execution (running Adagio, Presto)

The sierra script conducts several operations:

1. Takes a single mesh file and splits it (slice & spread) into many (one for each processor specified)
2. Writes a submission script to run the code (sierra.sh)
   - If there is a queuing system, submits submission script to queue
   - If there is not, runs script
3. Upon completion, combines parallel results files into a single file

Analysis Process: Analysis Execution (running Adagio, Presto)

During execution, four output files are written

1. Log file (text format, *.log)
   logs execution progress, error messages, etc.
2. Results file (Exodus format, typically *.e)
   mesh with node/element variables at user specified time interval
3. History file (Exodus format, typically *.h)
   global quantities, e.g. KE, or node/element values at user specific points in time
4. Restart file (Exodus format, typically *.rst)
   like results file, but intended to restart/continue the simulation if needed

Analysis Process: Analysis Execution (running Adagio, Presto)

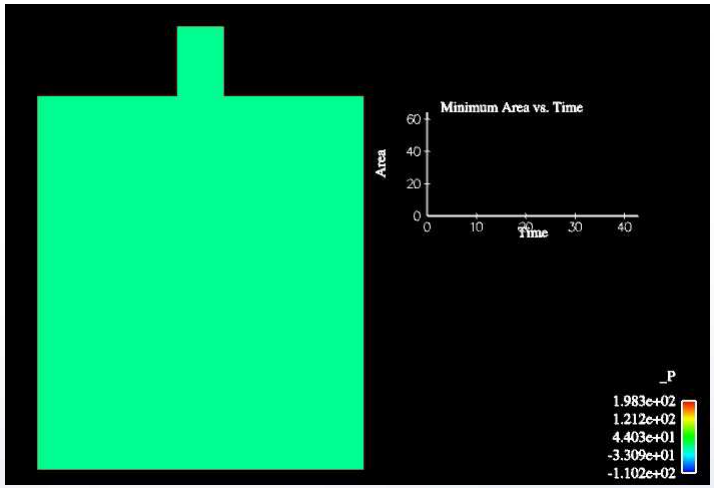
Note: During execution, additional files may be written

1. Queuing system (text format, usually *.e<jobid>, *.o<jobid>)
   contains messages from the queueing system and possibly error output (stderr)
2. Machine output file (*.output)
   Same kind of information as from the queuing system for machine without a queue

Analysis Process: Post-Processing

A variety of tools exist for viewing EXODUS files

- Ensight (licensed)
  Commercial, GUI driven
  available on many platforms, e.g. Linux workstations, windows
- ✓Paraview (no license needed)
  Partially Sandia funded, GUI driven
  available on many platforms, e.g. Linux workstations, windows. Macs
- BLOT (part of SEACAS,
  Sandia legacy tools)
  text input driven
- Patran (licensed)
  Commercial, GUI driven



Questions?