Exceptional service in the national interest

SAND2014-15913PE

Sandia National Laboratories

Sierra Suite of Codes Current and Planned Capabilities

Vicki Porter, Principal Member of Technical Staff

Sierra Solid Mechanics Team

Sandia National Laboratories

Goodyear Innovation Center -- Luxembourg

July 23rd, 2014

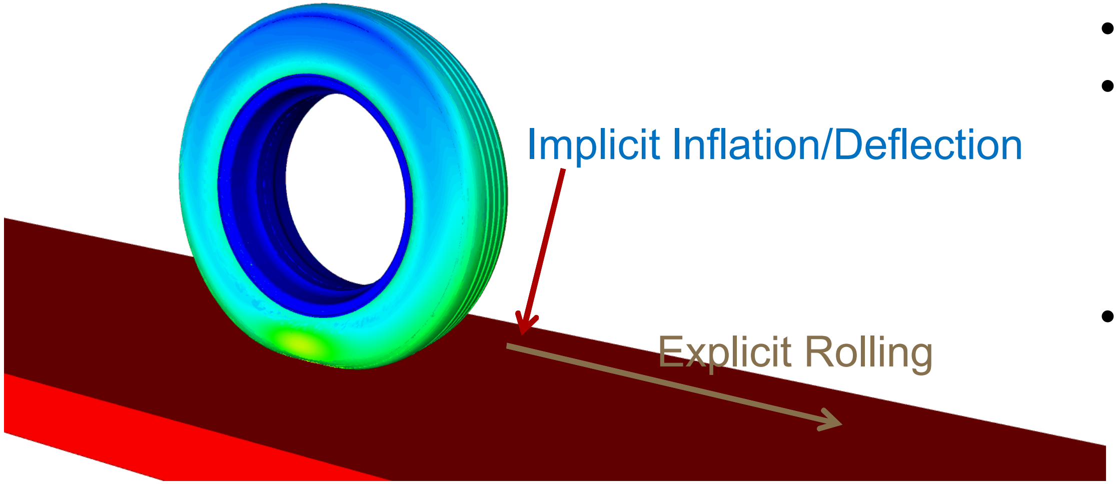
U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Recent Advancements

- Eagle3 deployment for passenger and light truck tires with capability for modeling sipe and groove closures.
- Code performance improvements for explicit dynamics – 2 to 10 times faster on a suite of application problems including one explicit tire rolling from Goodyear.
- Coupling with Structural Dynamics acoustics for noise prediction.
- Collaboration with Gert Rebel on an extended fiber shell to model true layer thickness of belts and plies.
- Advanced user interface that allows definition of variables by the user directly in the input file for post-processing and solution termination.

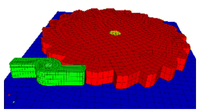
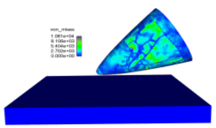
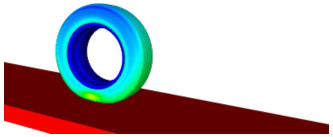
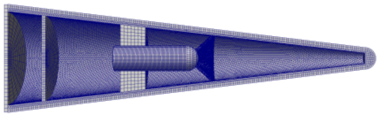
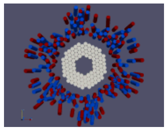
Goodyear Performance Improvements



- 232,561 elements
- Hex elements and rigid bodies, membrane elements, many element blocks and different material models
- ARS Contact, restart, prescribed temperature, prescribed displacement, pressure, implicit/explicit run

Case	4.29.4	4.29.5	Improvement
Implicit Deflection ARS Contact	1.18 Hours →	0.94 Hours	**25%**
Explicit Rolling ARS Contact	18.1 Hours →	14.2 Hours	**27%**
Explicit Rolling ARS Contact +Thermal Strains	34.1 Hours →	15.6 Hours	**118%**

Performance Timings and Speedup

Problem	Key Capabilities	Number of Procs	Original 4.28 (seconds)	VOTD 4.33.2 (seconds)	VOTD 4.33.4 (seconds)	Speedup (total)
	Contact Failure with Element Death	32	4535	883	844	**5.37x**
	Preload Multiple Mechanisms Contact	32	1449	616	689	**2.10x**
	Contact Fracture with Element Death User Derived Output	32	7634	3480	3574	**2.14x**
	ARS Contact Rigid Bodies Embedded Fiber Membranes	16	3928	1082	1085	**3.62x**
	Contact	32	1027	460	499	**2.06x**
	Eulerian Hydrocode Coupling	64	141266	18182	18234	**7.75x**

Performance Improvements

- Improvements made mostly in reduced parallel communications and use of vector computations
  - Positioning for future advanced platforms.
- Other focus areas
  - More efficient and user-friendly nodal based time-step (recently implemented improvements).
  - Ongoing research in algebraic methods that filter higher modes to enable larger stable time step without the need for an actual coarse grid.

Improved Contact for Implicit and Explicit Analyses

- Implicit robustness and ease of set-up for wide-spread contact.
- Improved accuracy for DASH contact – better face projection algorithm.
- Improved interface for friction models drastically reduced implementation complexity for advanced friction models.
- Implicit solver robustness is a focus for coming year.

Implicit Contact Performance Study: Successively Pushing Blocks out of a Jenga Tower

Objective:

Examine the performance and robustness of (near) default contact settings in a simple boundary value problem

Approach:

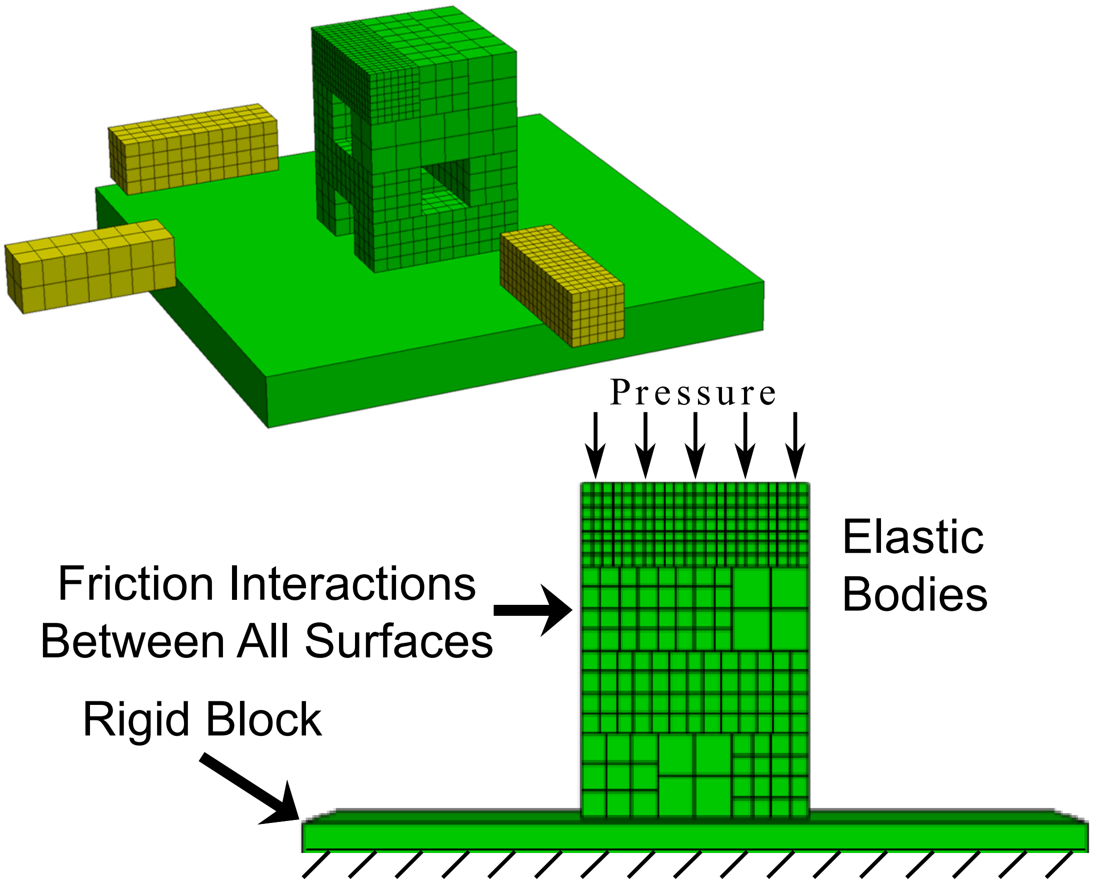
Simulation a suite of Jenga Tower contact simulations with 20 varying applied pressures and friction coefficients (400 total runs)

Robustness: Fraction of simulations passing

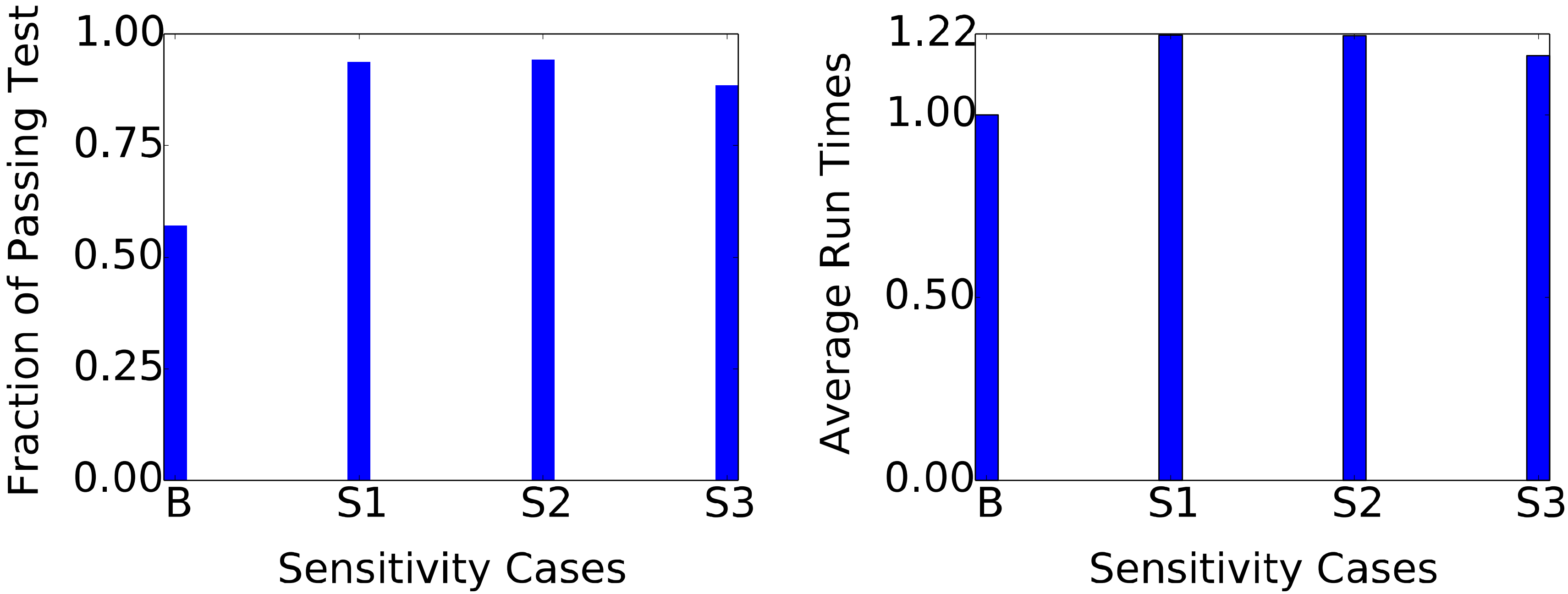
Performance: Wall clock time

Jenga Tower Problem Statement:

Apply different pressures and friction interactions. Push the blocks out 1 at a time.



Robustness and Performance Summary



(a) Robustness Comparison

(b) Performance Comparison

- An increase in the passing fraction (robustness) of ~30% is observed mainly due to turning off the load step predictor.
- A performance hit of ~20% is observed with this change
- Use of smoothing iterations appears to decrease robustness but increase performance

Large Deformation and Failure Modeling

- Extended Finite Elements (XFEM)
  - Implementation of shell failure with XFEM
  - Focus in upcoming year on 3D XFEM
  - Potential applications include tread chipping, chunking, and wear
- Improvements in efficiency and robustness of element death
- Ongoing research for implementation of a polyhedral element.
- Ongoing research on particle methods – Reproducing kernel particle method (RKPM) to remove instabilities in current methods

2-D XFEM Example Problems

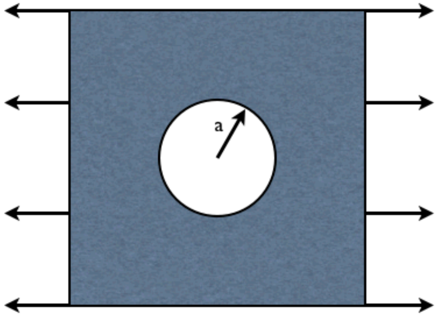
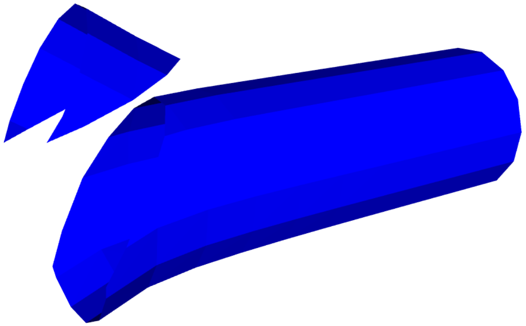


Plate with hole problem

Capabilities tested: Crack nucleation, planar crack growth, cohesive zone insertion



Cylinder Angled Crack Problem

Capabilities tested: Angled prescribed crack and planar crack growth

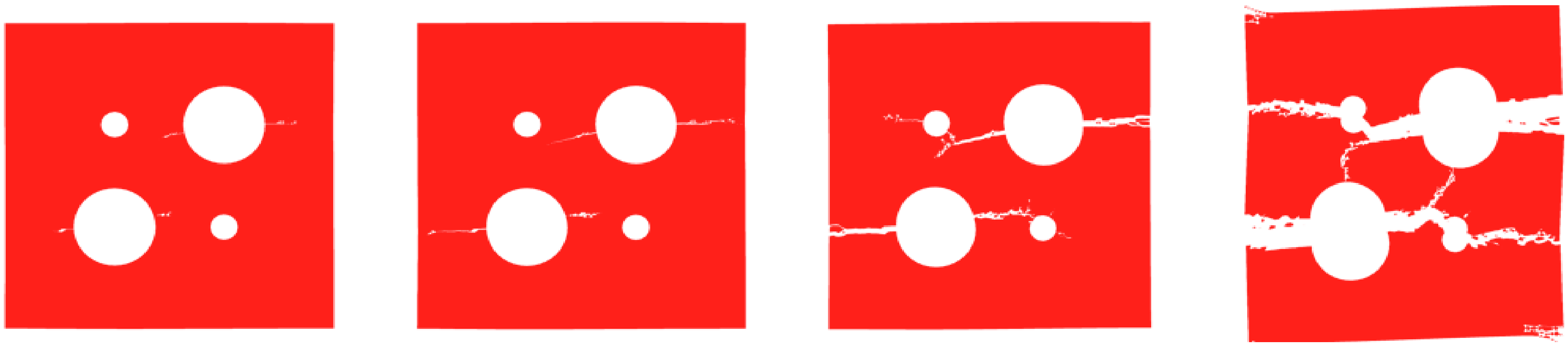
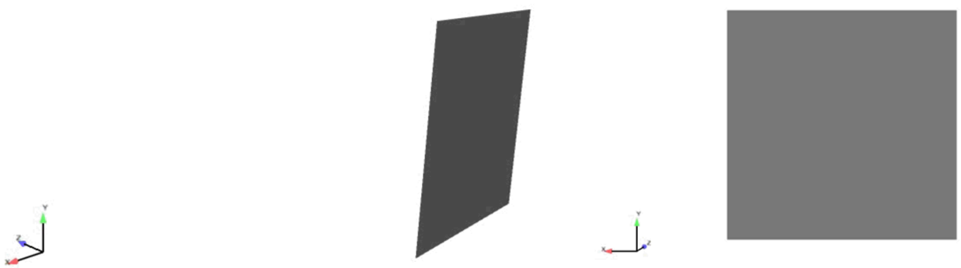


Plate with Multiple Holes Problem

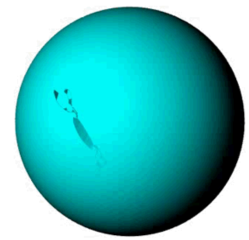
Capabilities tested: Crack nucleation, branching, piecewise-linear crack growth

Modeling Shell Failure with XFEM

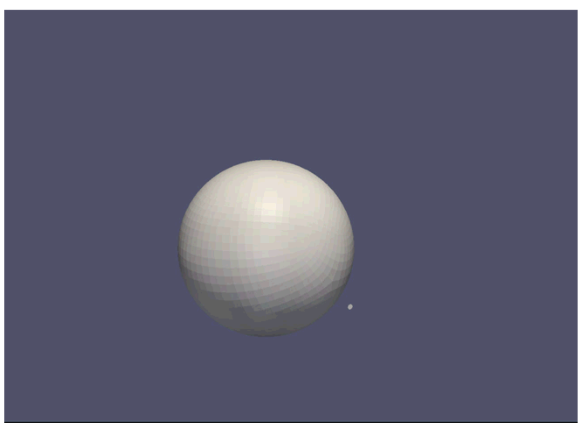
Plate Blast Fracture



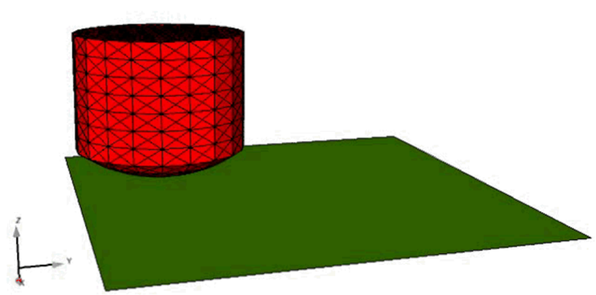
Pressurized Sphere Fracture



Fracture following Projectile Impact



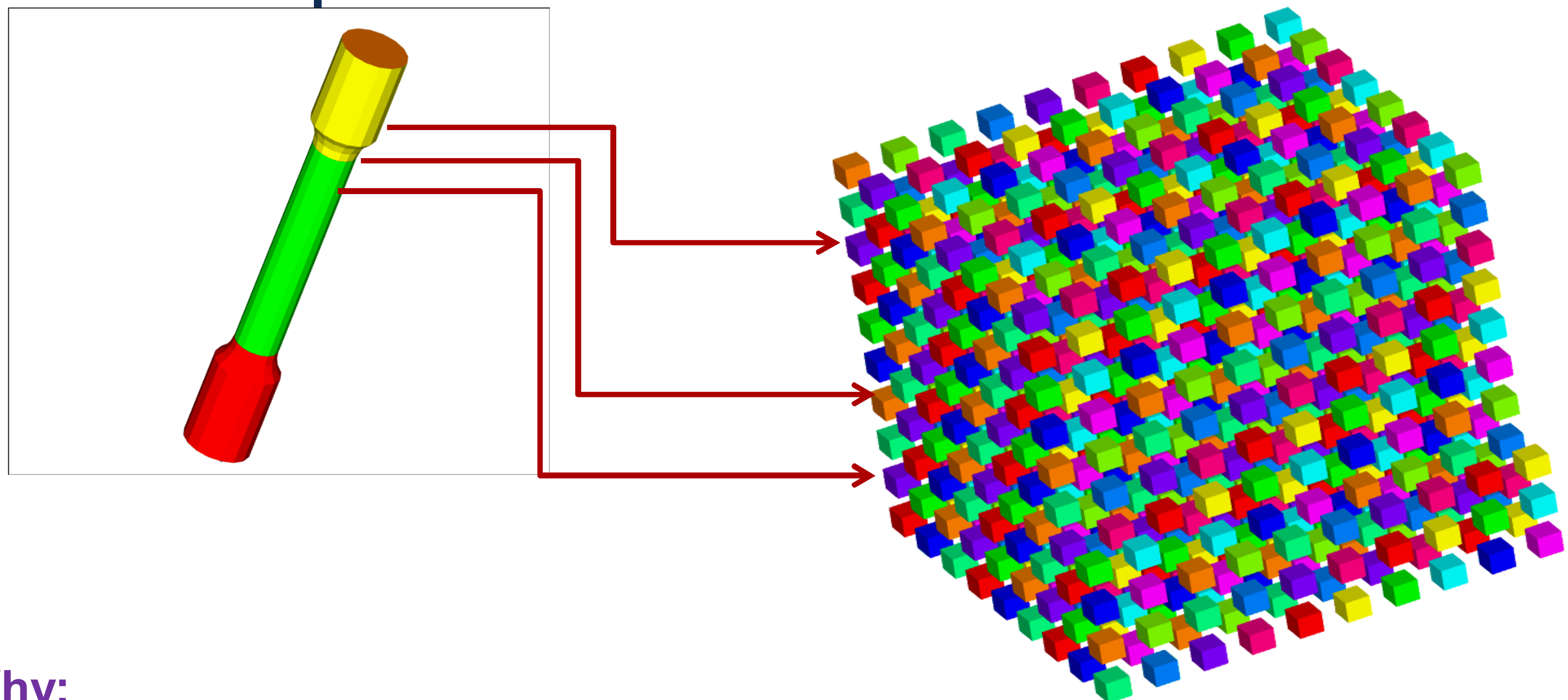
Wear Demonstration



Large Complex Models and Next Generation Platforms

- Multiscale methods
  - Enables much larger scale system models that incorporate local features.
  - Techniques include subcycling, and submodelling, and use of representative volume elements.
- Positioning for exascale models
  - Transition from rigid framework to more flexible Sierra Toolkit
  - Recent implementation of local vectorization on chip
  - In-situ visualization
- Polyhedral elements for ease of meshing and large deformation and failure modeling.

Multiscale Modeling Representative Volume Elements



Why:

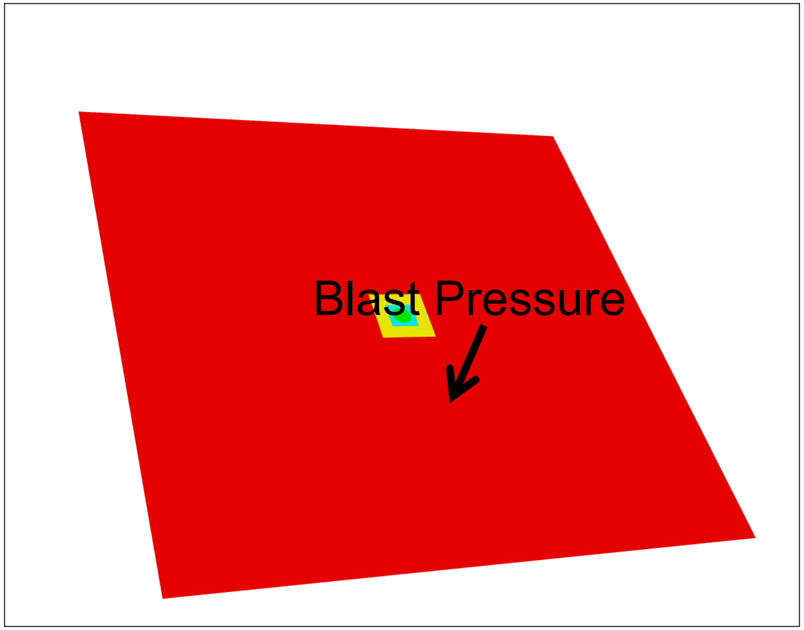
Each RVE represents a material point with a complex material such as a composite

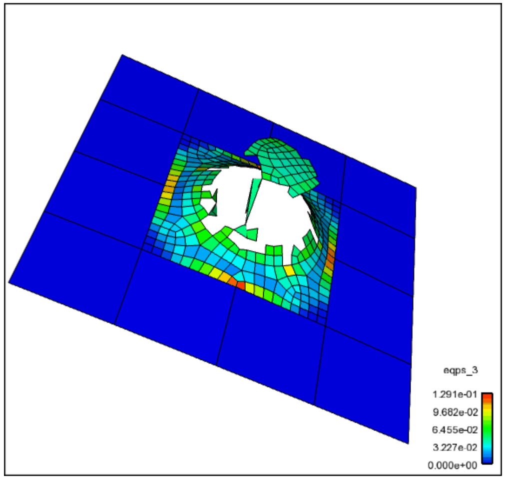
The RVE is solved for the local behavior and then this is assembled to the global problem.

800 RVE problem (~100,000 total elements) was unusably slow

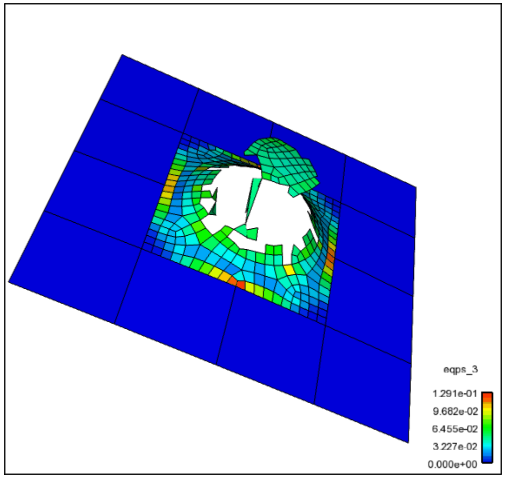
RVE region has ~70K elements, but ~6K different boundary conditions and ~10K individual node buckets.

Multiscale Modeling
Subcycling (shell example)





No Subcycling



With Subcycling

1) Rotational DOFs added to transfer operations

2) Confirmed can run shell MPCs and element death in fine region

Num Elem Coarse: 1584	DT Coarse: 7.64e-7
Num Elem Fine: 404	DT Fine: 5.09e-8

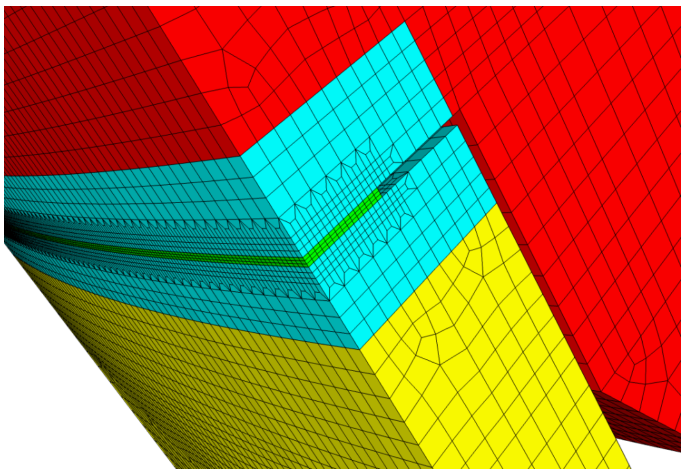
No Subcycling: 66.7 sec
With Subcycling: 22.1 sec

Subcycling Speedup Achieved: 300%
Maximum Theoretically Obtainable: 340%

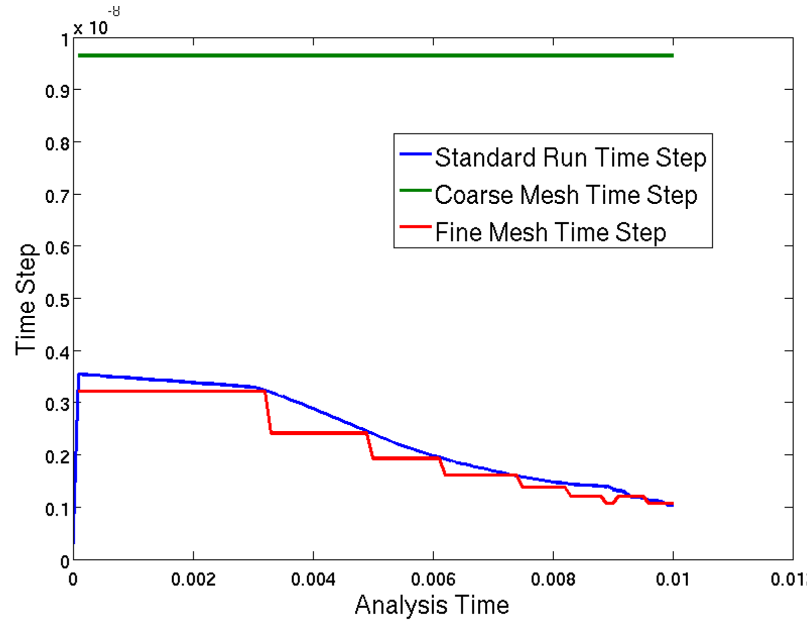
Result Comparison: Identical

Multiscale Modeling
Subcycling: Automated Setup

Sandia National Laboratories



Weld Model

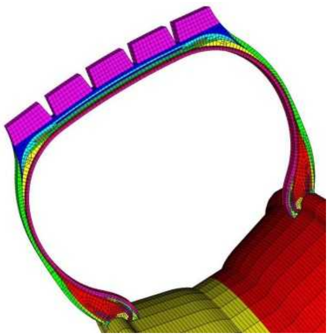


Run Type	Runtime (24 Processors)
Standard	36.9 hours
Subcycling w/o Rebalance	30.3 hours
Subcycling with Rebalance	21.6 hours

Speedup Obtained: 1.71x : Theoretical Speedup: ~2x ?

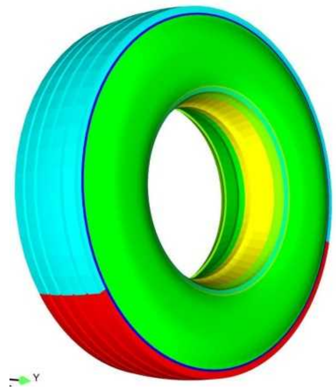
Multiscale Modeling
Goodyear Use of Submodelling

Complicated process involving multiple transfers and restarts on various meshes and combinations of meshes. One stage of restart/transfers was failing in rigid body initialization.

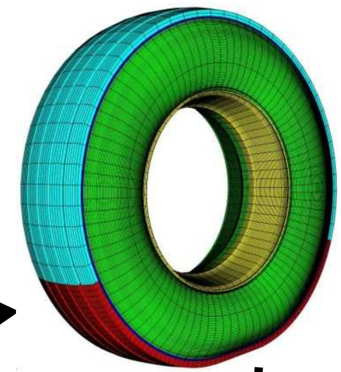


Stage 1: Wedge Inflation

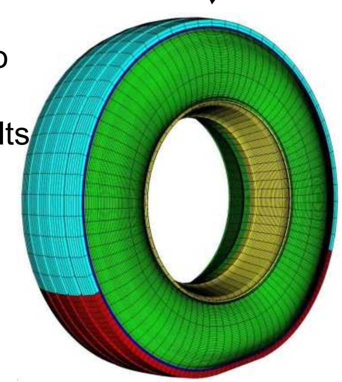
Transfer: Revolve wedge to 3D.



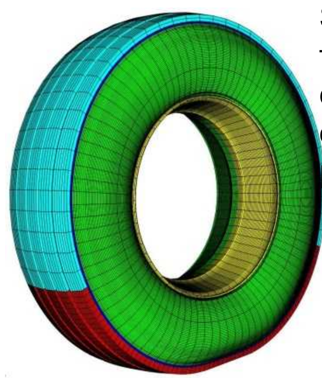
Stage 2: Restart from revolved inflation results to do deflection against flat road. Pieces of tread MPC'd to carcass and each other.



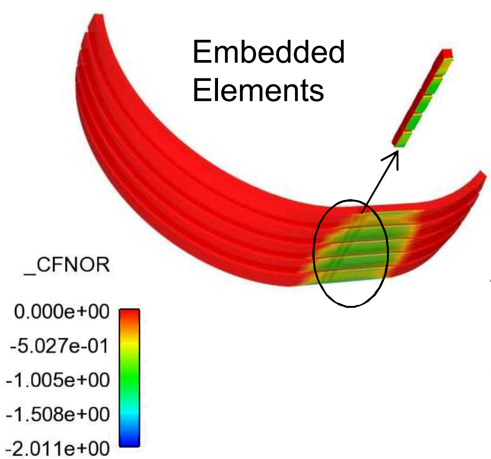
Stage 3: Restart from deflect for initial quasistatic roll for 42 degrees.



Stage 4: Restart to do fine stepping to 82 degrees. Output results only on small tread piece.



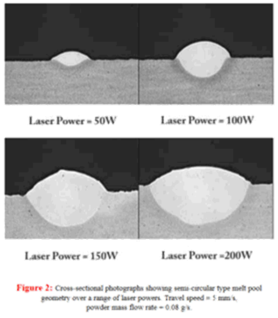
Stage 5: Restart from Stage 4 applying BC from tread output file. Also has embedded submodel for single tread pitch.

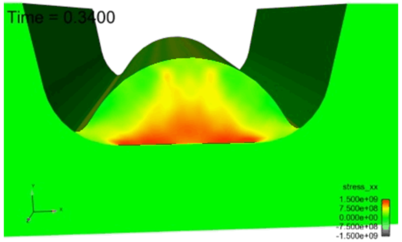


Material Deposition Modeling

Goal: To model additive manufacturing

- Current capabilities that may be applied:
  - Model fluid material only in Aria
  - Use adaptive remeshing and/or nodal-based tets to model material in fluid state
  - Fixed grid with xfem_volume_fraction < 1.0 for elements containing the fluid/solid interface
  - Switch element blocks from “active” to “inactive” periods
- Capabilities requiring research & development:
  - XFEM to represent fluid-surface interface
  - Level Set to model evolution of fluid-surface interface





Context for Sierra's transition to "Toolkit"

Sandia National Laboratories

Significant changes in hardware have always implied significant changes in engineering-application code architecture

comments:

Pre-1990's : Vectorization Paradigm

Hardware: vector processors
Code design pattern: functions operating on a workset of data

Vestiges still present in current code.

1990's: Distributed Parallel Paradigm

Hardware: change from vectorization to multiple caches
Code design pattern: on-processor computations + parallel assembly
Sierra's Framework provided parallel services that encapsulated these design patterns
Sierra applications were re-written to use the framework

We have been using it effectively for over a decade, but transition was costly.

2010's: MPI+X+Y paradigm

Hardware: several variations on threading, SIMD, GPUs
Design patterns: minimize data movement, understood to be work-unit based, optimal data layout is different for different hardware
Need to revisit MPI scalability – but now at much larger scale
Need to discover optimal mix of work for MPI and/vs. threads
Sierra's Toolkit is a componentization of basic parallel services that are in the Framework
Transition intended to be transparent to the user community.

Several code architectures being explored in the broader community

…but not transparent to the developer community.

Sierra Transition to "Toolkit"

Component usage in applications:

Address a longstanding concern of inflexibility of the Sierra Framework (which is understood as a problem with frameworks in the broader community as well)

Solving larger problems:

Support larger problem sizes (64 bit INT global IDs, Load balancing)

Performance and scalability on distributed MPI-based hardware:

Demonstrate scalability of foundational & application algorithms to 10^5 – 10^6 cores (beyond current use cases)

Field types & memory layout options:

Memory layout flexibility for optimal performance depending on specific hardware (e.g. left justified, right justified, tiling)

Agility to change algorithms to suit hardware:

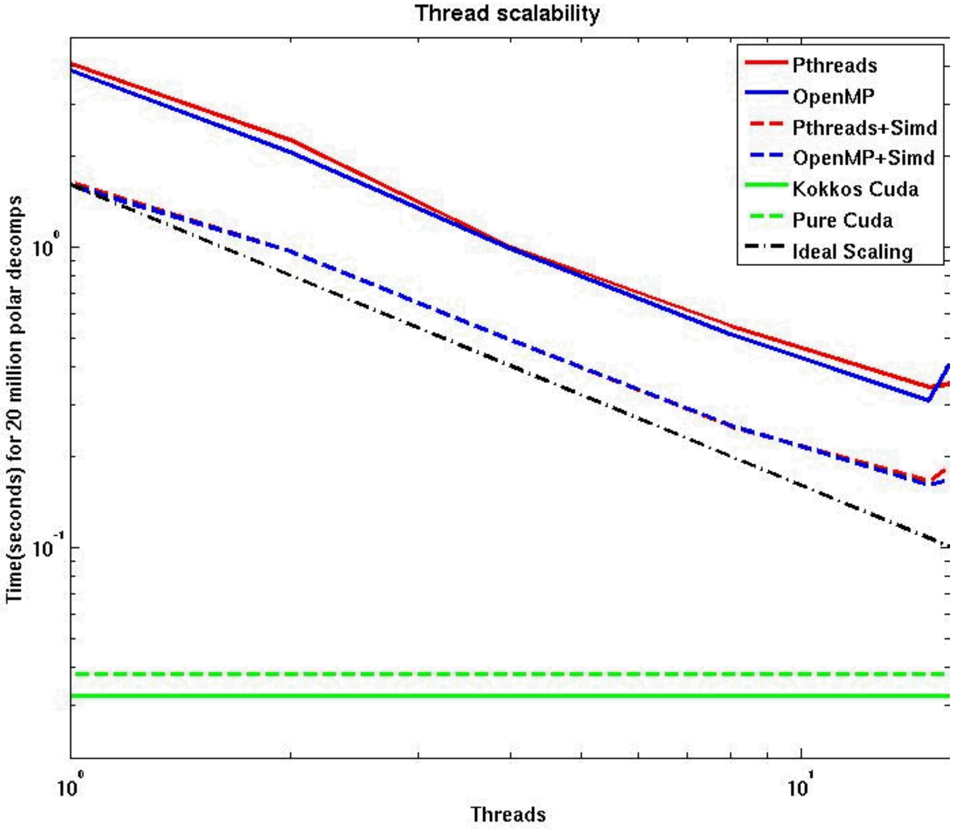
Adapt to the different ways that hardware achieve performance -- search algorithms, solvers, mesh modifications, and others (examples already exist).

Initial Work toward Next Generation Platforms

- SIMD (Single Instruction, Multiple Data) instructions for math and internal force calculations (Intel and AMD) implemented in Sierra/SM to utilize vectorization on the chip.
- Many similarities with programming for a GPU
- Resulted in ~15% overall improvements for Goodyear performance benchmark problem.

Comparison of Parallel Techniques:

- PThreads utilization of multiple threads on a single core
- OpenMP: easy but limited MPI parallel capability
- SIMD – local vectorization on a chip (up to length 4 currently, 8 on the horizon)
- GPUs – large numbers of threads on graphics chips or accelerators



Plots of MTK polar decomposition routine

Summary

- Current year accomplishments
  - 2-5 times improvement in explicit dynamics performance on a variety of problems (within +10% abaqus runtimes)
  - XFEM for shells including contact on cut surface
  - Fiber shell representing actual ply/belt thickness (Gert Rebel)
  - Initial coding for advanced platforms
- Targets for coming year
  - 3D XFEM with possible applications to wear and durability.
  - Further implementation of coding for next generation platforms
  - Implicit solution robustness (contact, stabilization methods)
- On-going Research
  - Continued advancements in modeling for pervasive fracture and failure.
  - Improved techniques for large (exascale) modeling
  - Multi-scale methods including algebraic coarse grids for larger explicit time steps
  - Polyhedral elements
  - Particle methods (RKPM) possibly applicable to snow and mud modeling