
*Center for micromorphic multiphysics porous and particulate materials
simulations within exascale computing workflows*
Multi-disciplinary Simulation Center (MSC)

Ratel Overview

Zachary R. Atkins, Jed Brown, Fabio Di Gioacchino, Leila Ghaffary
Zachariah T. Irwin, Rezgar Shakeri, Ren Stengel, **Jeremy L. Thompson**
19-20 May 2025
University of Colorado Boulder



NNSA
National Nuclear Security Administration



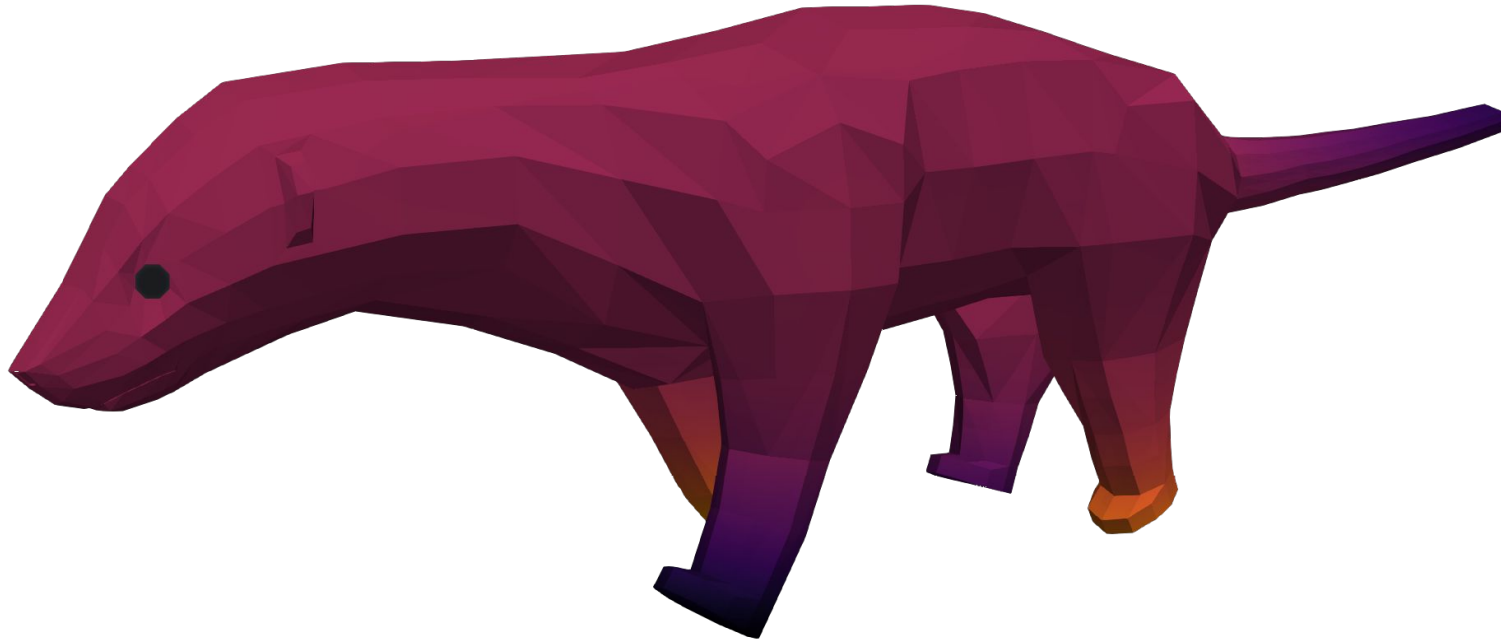
University of Colorado
Boulder



THE UNIVERSITY OF
TENNESSEE
KNOXVILLE



PSAAP III Final Review, 19-20 May 2025
*Center for micromorphic multiphysics porous and particulate
materials simulations within exascale computing workflows*
CU Boulder Multi-disciplinary Simulation Center (MSC)



- Ratel - matrix-free solid mechanics with PETSc + libCEED
- Grew out of libCEED mini-app



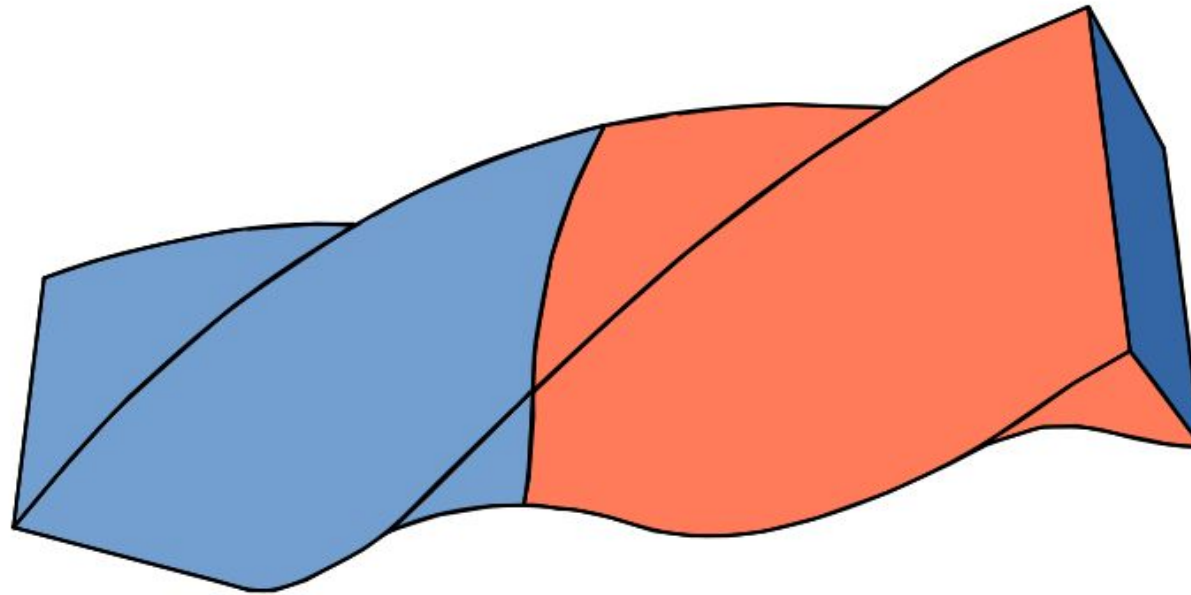
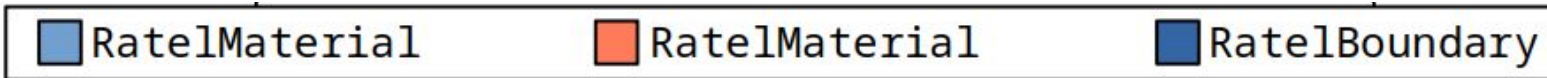
Mesh support

Mini-app

- Single material
- Hex only

Ratel

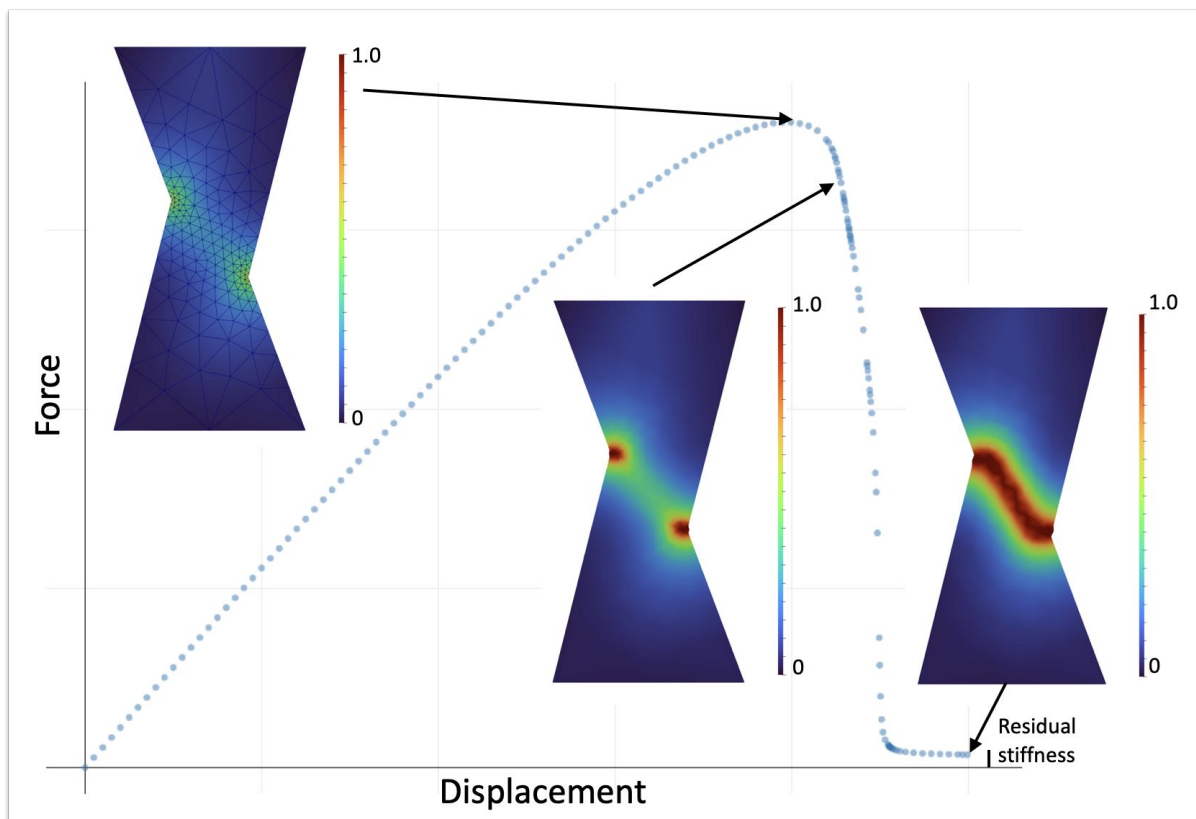
- Multiple materials
- Hex and tet support
- Mixed FEM support



Material Models

Mini-app

- Linear elasticity
- Neo-Hookean hyperelasticity
 - small strain, finite strain



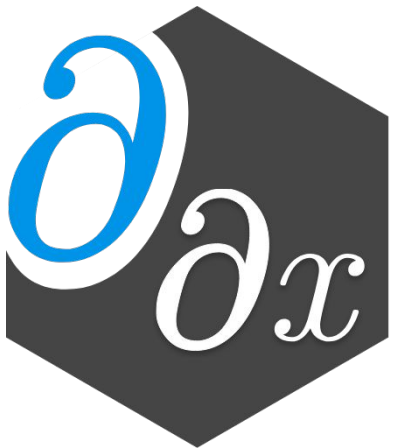
Ratel

- Hyperelasticity
 - Linear, mixed linear, Neo-Hookean, mixed NH, Mooney Rivlin, Ogden, Hencky (some with AD)
- Plasticity
 - Linear, Hencky
- Damage
 - Neo-Hookean, Hencky
- Poromechanics
 - Linear, Neo-Hookean
- Viscoelasticity (Hencky)

- ❑ Material model development is time intensive
- ❑ Jacobians are easy bug opportunity
- ❑ AD speeds up development, can reduce bugs

Ratel

- Hyperelasticity
 - Neo-Hookean, Ogden
- Plasticity
 - In progress
- Enzyme AD, ADOL-C
- Long term - Rust + Enzyme



Solver Modes

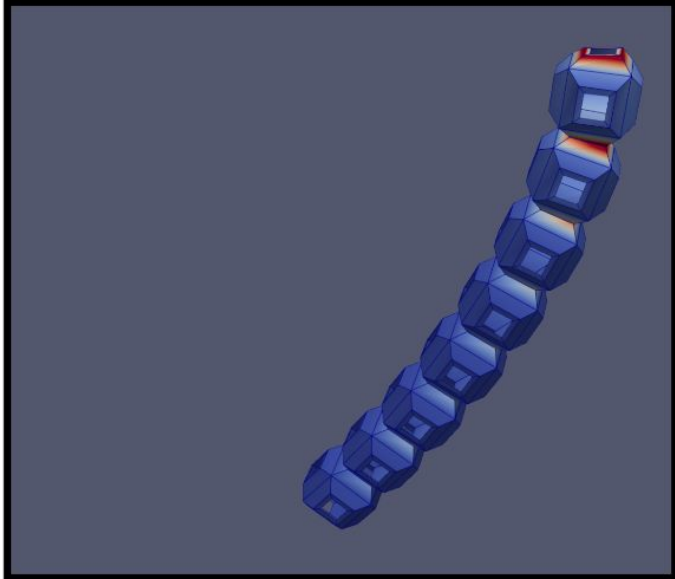
Mini-app

- Static

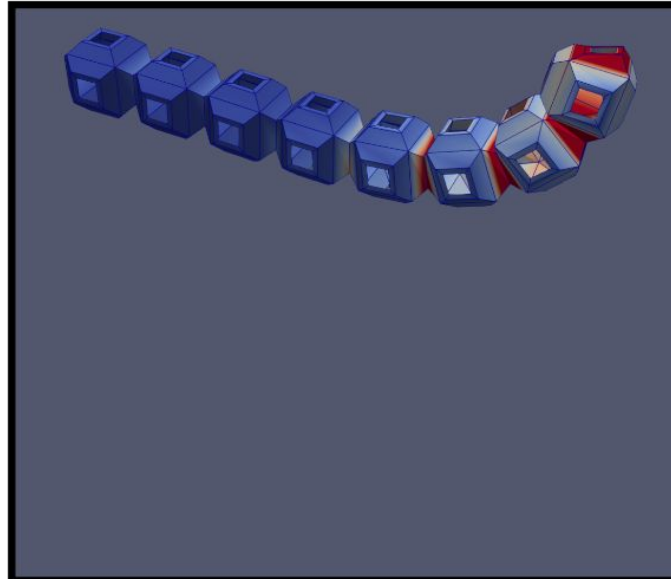
Ratel

- Static
- Quasistatic
- Dynamic

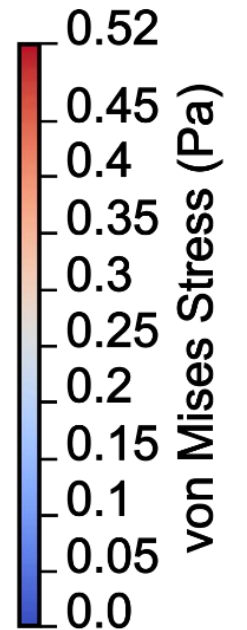
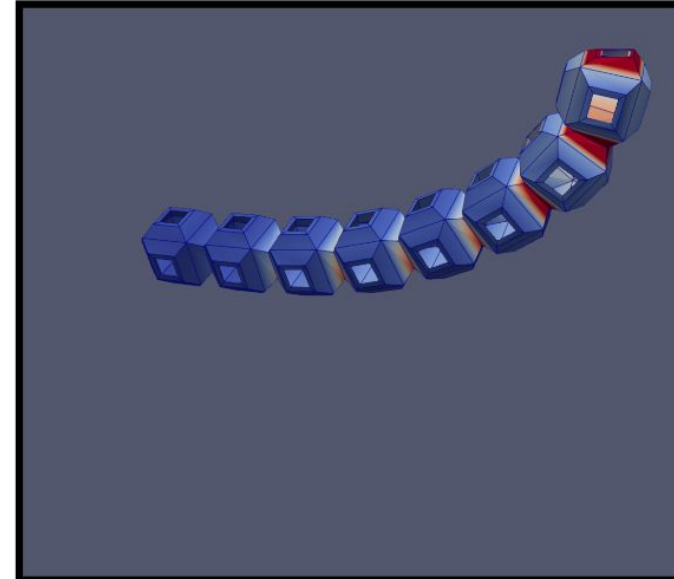
time = 17s



time = 40s



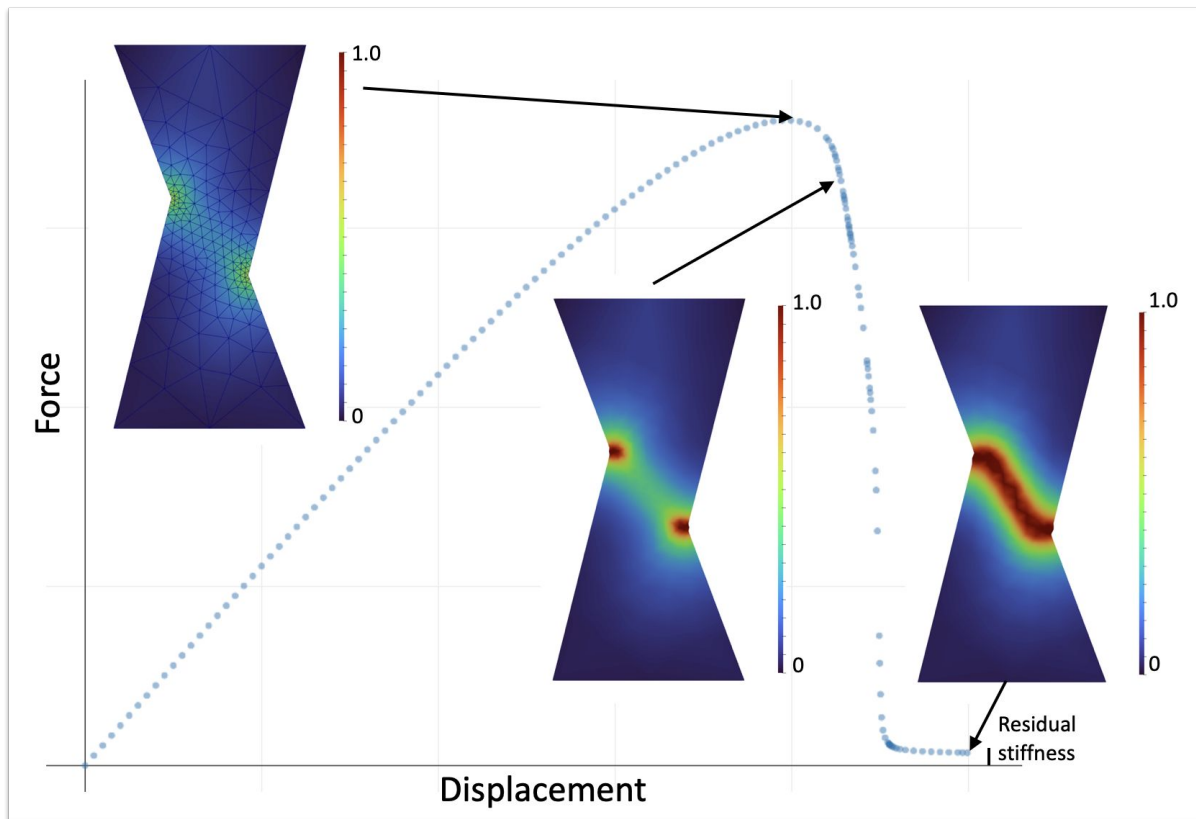
time = 1330s



Boundary Conditions

Mini-app

- Dirichlet
 - clamp
- Neumann (traction)



Ratel

- Dirichlet
 - Clamp, slip
- Neumann (traction)
- Pressure
- Contact
 - Nitsche, penalty
 - Platen, cylinder
- All BCs incremental time varying

Diagnostic Quantities

Mini-app

- Displacement
- Pressure
- Strain energy

Ratel

- Displacement
- Cauchy stress tensor
- Pressure
- Strain tensor invariants
- Strain energy
- von-Mises stress
- Mass density

Numerical Methods

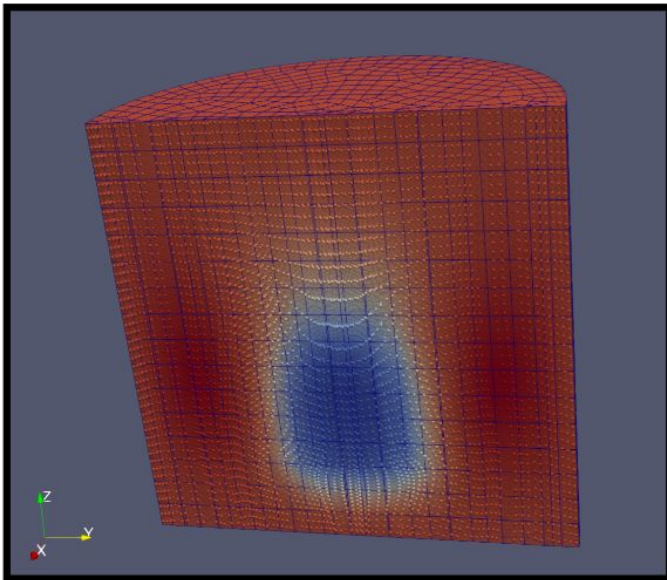
Mini-app

- FEM

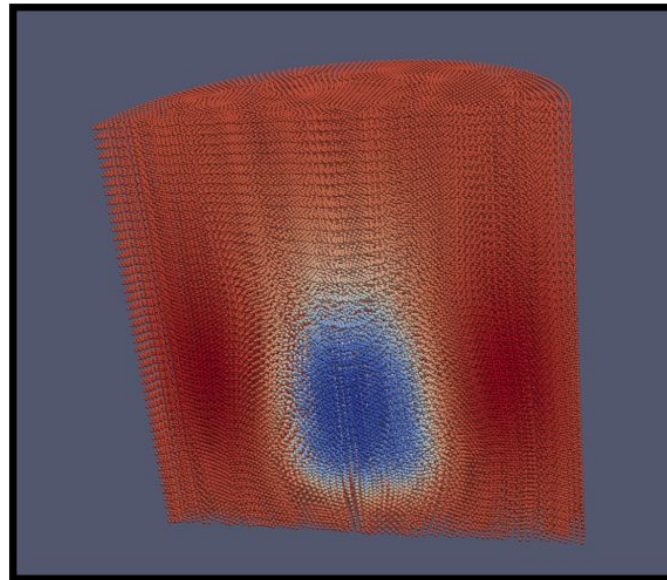
Ratel

- FEM
- Mixed FEM
- iMPM

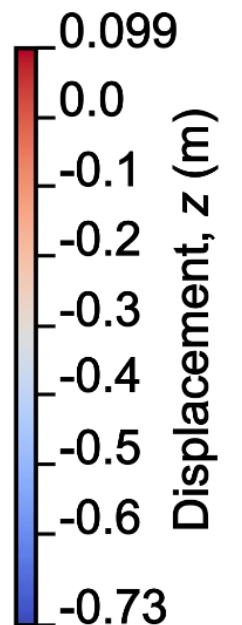
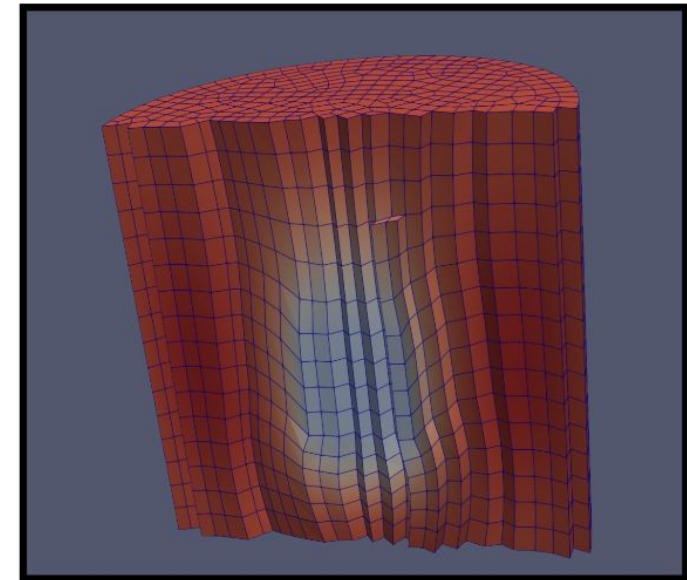
iMPM - mesh & particles



iMPM - particles



FEM - mesh



Preconditioning

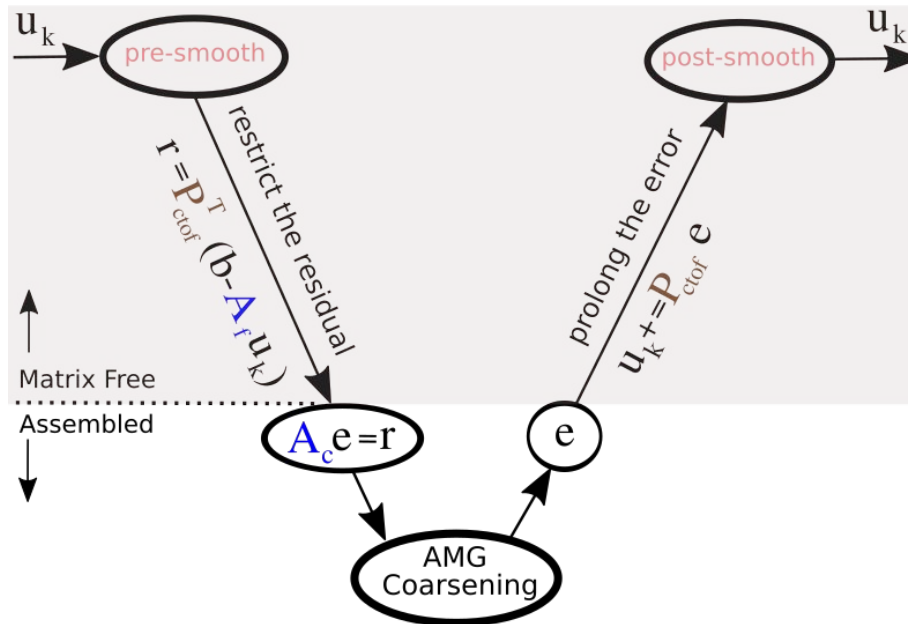
Mini-app

- Jacobi
- P-multigrid

Ratel

- Jacobi (diagonal, block)
- P-multigrid (including iMPM)
- Fieldsplit
- Any PETSc preconditioner


smoother: $u_k += \text{cheby}(u_k; A_f, b, \lambda_{\max})$



Software Engineering

- Theory and API docs
- GPU and CPU CI
- Auto deployed Dockerfiles

test:stage-lint	test:stage-full	deploy
<div>✓ noether-lint</div>	<div>✓ docker-cpu</div>	<div>✓ docker-latest</div>
<div>✓ noether-memcheck</div>	<div>✓ docker-cpu-int64</div>	<div>✓ pages</div>
	<div>✓ noether-ad-adolc</div>	<div>✓ pages:deploy</div>
	<div>✓ noether-ad-enzyme</div>	
	<div>✓ noether-cuda</div>	
	<div>✓ noether-hip</div>	



Ratel 0.4.0 documentation

Q Search

CONTENTS


- [Getting Started](#)
- [Example Applications](#)
- [Using Ratel](#)
- [API Documentation](#)
- [Theory Guide](#)
- [Contributing](#)
- [Code of Conduct](#)
- [Developer Notes](#)
- [Release Notes](#)

Ratel: Extensible, performance-portable solid mechanics

Ratel is a solid mechanics library and applications based on [libCEED](#) and [PETSc](#) with support for efficient high-order elements and CUDA and ROCm GPUs.

Solid mechanics simulations provide vital information for many engineering applications, using a large amount of computational resources from workstation to supercomputing scales. The industry standard for implicit analysis uses assembled sparse matrices with low-order elements, typically Q_1 hexahedral and P_2 tetrahedral elements, with the linear systems solved using sparse direct solvers, algebraic multigrid, or multilevel domain decomposition. This approach has two fundamental inefficiencies: poor approximation accuracy per Degree of Freedom (DoF) and high computational and memory cost per DoF due to choice of data structures and algorithms. High-order finite elements implemented in a matrix-free fashion with appropriate preconditioning strategies can overcome these inefficiencies.

For further details on the benefits of high-order, matrix-free finite elements for solid mechanics, see [our preprint on arXiv](#).

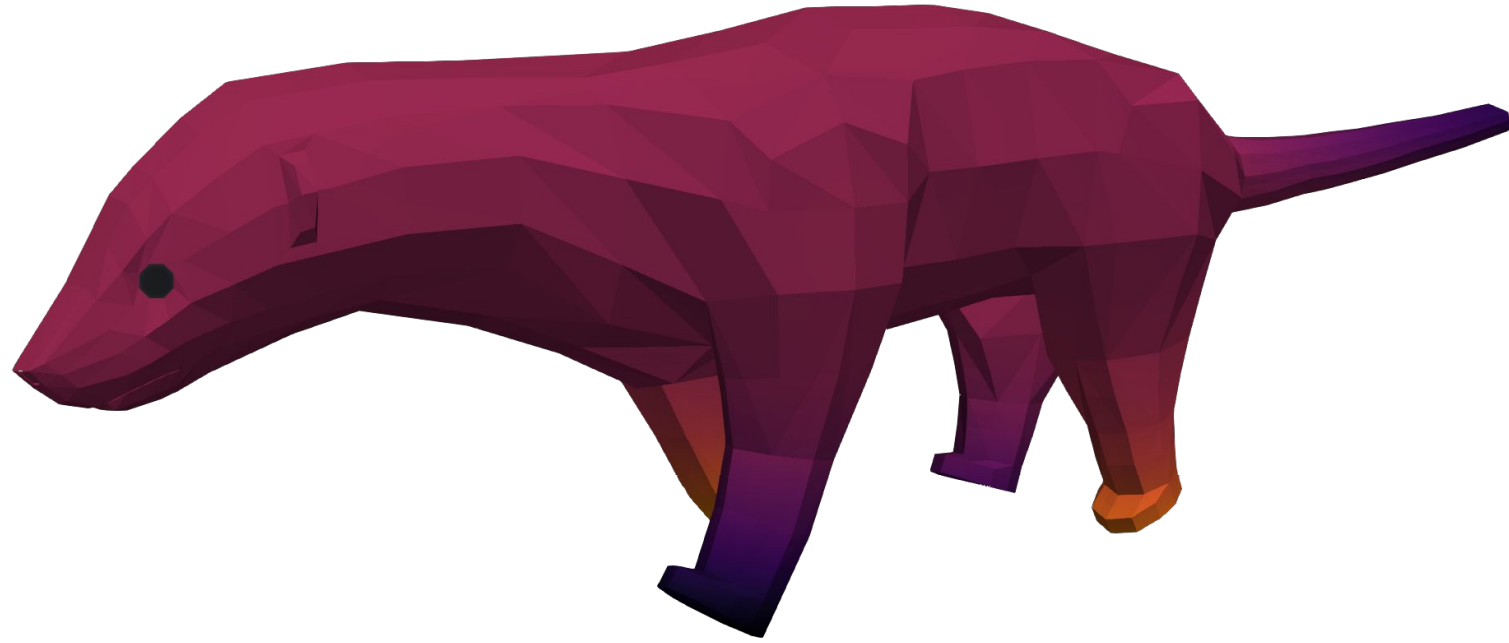


Ratel: Extensible, performance-portable solid mechanics

GitLab-CI	passed	License	BSD 2-Clause	Documentation	latest	coverage	95.65%
-----------	--------	---------	--------------	---------------	--------	----------	--------

Summary

- Huge expansion of material models and BCs
- iMPM and more preconditioners now supported
- Well positioned for further work



Thank you