Ratel - New Foundations of Computational Mechanics for the Exascale Era

Jeremy L Thompson

University of Colorado Boulder

jeremy@jeremylt.org

1 May 2025

Interrupt Me!

Please interrupt with questions/comments

(Yes, especially you students!)

Overview

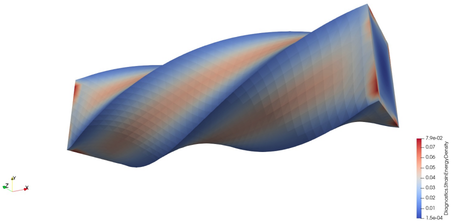
1. libCEED
2. Ratel
3. Mentoring
4. Questions

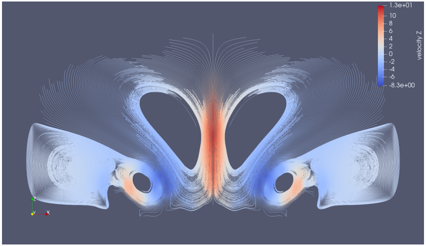
libCEED Team

libCEED Repo: https://github.com/CEED/libCEED

Developers: Ahmad Abdelfattah, Zach R. Atkins, Valeria Barra, Natalie Beams, Jed Brown, Jean-Sylvain Camier, Veselin Dobrev, Yohann Dudouit, Leila Ghaffari, Sebastian Grimberg, Tzanio Kolev, David Medina, Will Paznel, Thilina Ratnayaka, Rezgar Shakeri, Stan Tomov, James Wright III, Jeremy L Thompson

Background





libCEED solid mechanics (left) and fluid dynamics (right) mini-apps

- Physics based simulations widespread in science/engineering
- Intuition: FEM solves equations with piecewise polynomial solution
- libCEED supports FEM-like simulations on modern hardware

libCEED Projects

Several projects built using libCEED

- Ratel - solid mechanics FEM and iMPM (PSAAP)
- HONEE - fluid dynamics FEM & differential filtering (PHASTA)
- MFEM - various applications, libCEED integrators (LLNL)
- Palace - quantum circuit design, MFEM + libCEED (Amazon)
- RDycore - FV river dynamical core, PETSc + libCEED (SciDAC)

Top 500

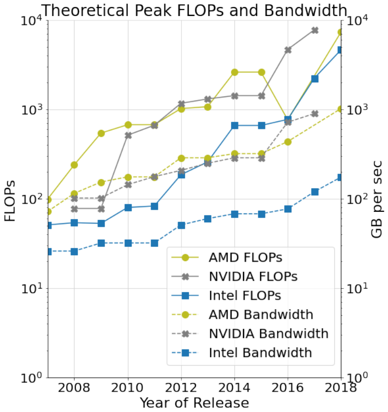
Machine	HPL	HPCG
Fugaku	442.01 PFLOPs	16.01 PFLOPs
Frontier	1,353.00 PFLOPs	14.05 PFLOPs
Aurora	1,012.00 PFLOPs	5.61 PFLOPs
LUMI	379.70 PFLOPs	4.59 PFLOPs
Alps	434.90 PFLOPs	3.57 PFLOPs

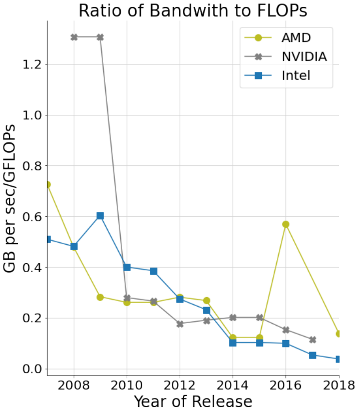
Top 500 Machines for HPCG with HPL peak FLOPs

HPCG closer to representative FLOPs for simulations

Difficult to realize peak FLOPs with CG on modern machines

Modern Hardware



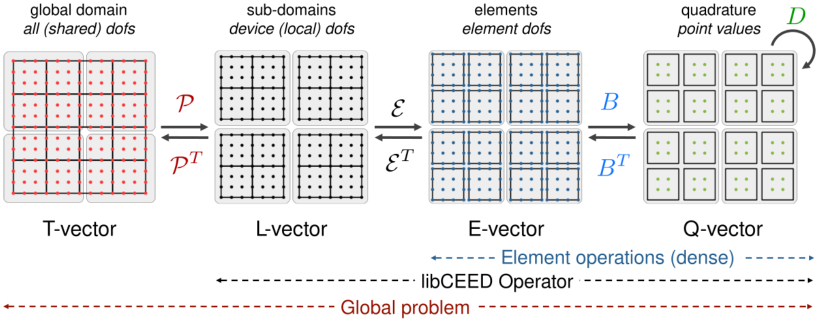


Memory bandwidth is improving slower than FLOPs [2]

Mirrors difference between Top 500 HPL vs HPCG benchmarks [1]

Matrix-Free Operators from libCEED

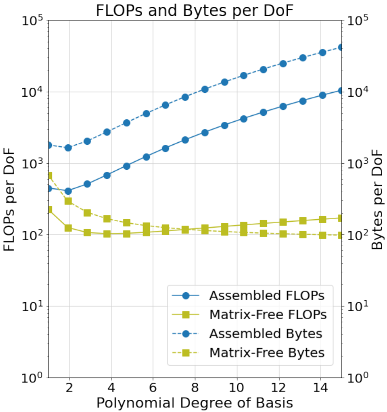
$$A = \mathcal{P}^T \mathcal{E}^T B^T D B \mathcal{E} \mathcal{P}$$

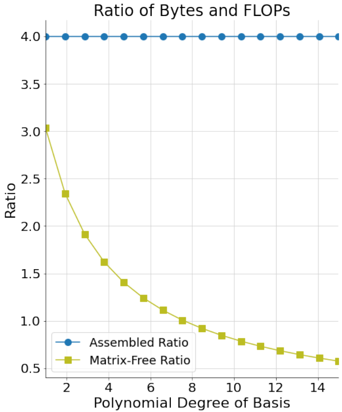


libCEED provides matrix-free operator evaluation on various hardware

Matrix-free operators apply these steps instead of populating a matrix

Benefits of Matrix-Free

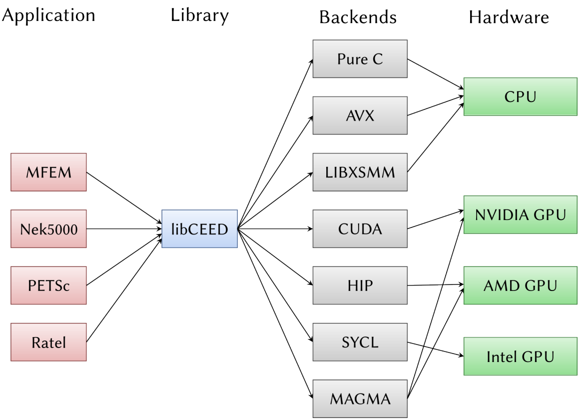




Requirements for matrix-vector product with sparse matrix vs matrix-free for screened Poisson $\nabla^2 u - \alpha^2 u = f$ in 3D

Matrix-free representations using tensor product bases better match modern hardware

Performance Portability



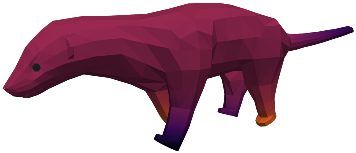
libCEED's design naturally allows multiple hardware implementations

Design Implications

Using matrix-free operators drives design decisions

- Direct solvers are out (assembled matrices, $\mathcal{O}\left(n^2\right)$)
- Iterative solvers are in (Krylov methods, etc)
- High order $=$ high accuracy & bad condition numbers
- Preconditioning is needed for fast convergence

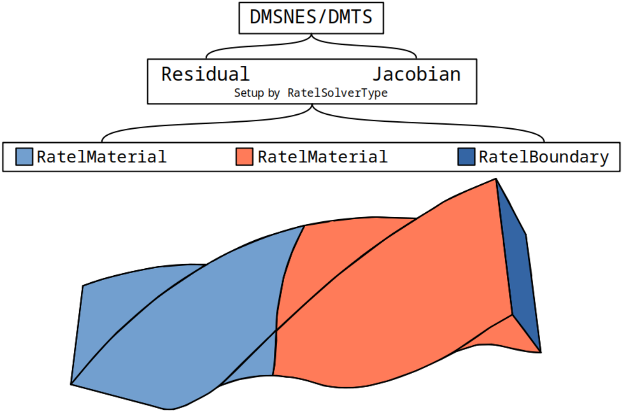
Ratel Team



Ratel Repo: https://gitlab.com/micromorph/ratel

Developers: Zach R. Atkins, Jed Brown, Fabio Di Gioacchino, Leila Ghaffari, Zach Irwin, Rezgar Shakeri, Ren Stengel, Jeremy L Thompson

Basic Design



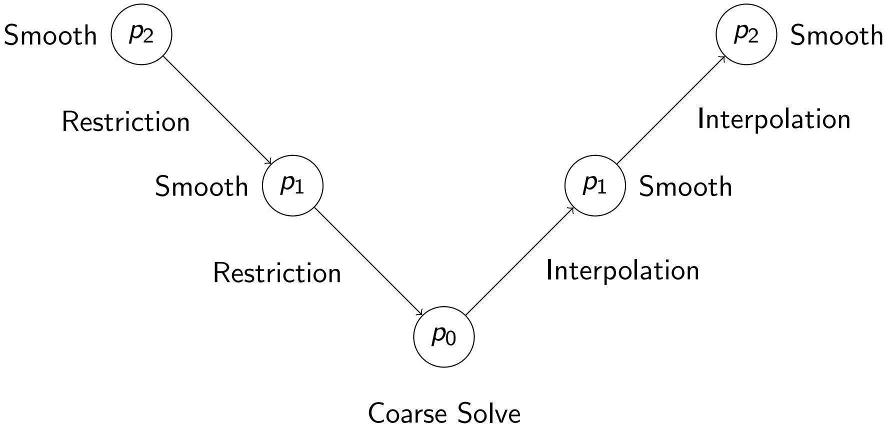
Each material region sets up part of the non-linear and linear equations

Preconditioning Support

Iterative solvers need preconditioning

- $Ax = b$, slow for ill-conditioned (high-order)
- $P^{-1}Ax = b$, faster for good $P^{-1} \approx A^{-1}$
- Need to balance setup costs and preconditioner effectiveness
- Assembly of diagonal, etc needed

p-multigrid

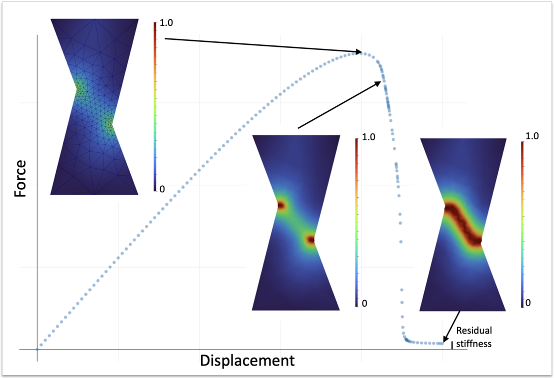


libCEED provides matrix-free p-multigrid

Too General, Need Specifics

Ok, lets look at some specific simulations

Example - Linear Damage




```
$ bin/ratel-quasistatic -options_file examples/ymls/ex02-
    quasistatic-elasticity-linear-damage-compressiveshear-
    AT2-face-forces.yml
```

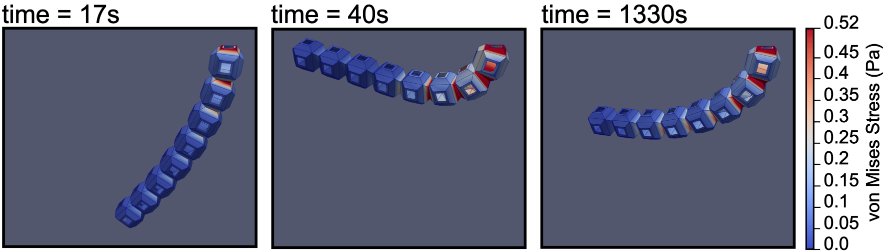
Quasistatic simulation of compressive shear for generic brittle material

Material Modeling

Modeling costs significant effort

- Derivation and derivatives, multiple places to have bugs
- Automatic differentiation tools can help
- Targeting engineering formulation directly to simulation

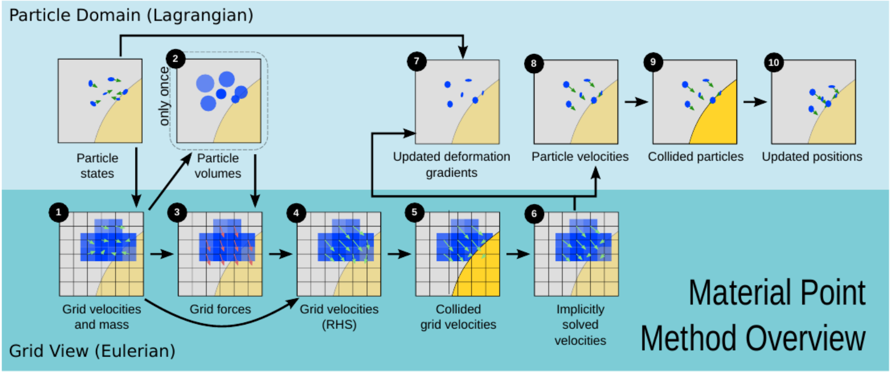
Example - Dynamic Pendulum




```
$ bin/ratel-dynamic -options_file examples/ymls/ex03-dynamic
    -elasticity-schwarz-pendulum-enzyme.yml
```

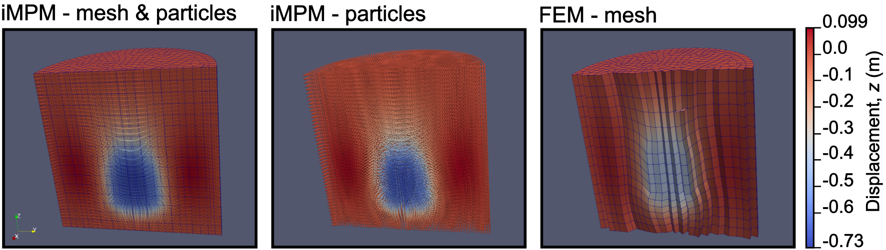
Dynamic simulation of Neo-Hookean Schwarz-P "pendulum" with Enzyme

What is MPM?



- Continuum based particle method with background mesh for gradients
- Extension of FLIP (which is an extension of PIC)
- Used in rendering for the movie *Frozen*

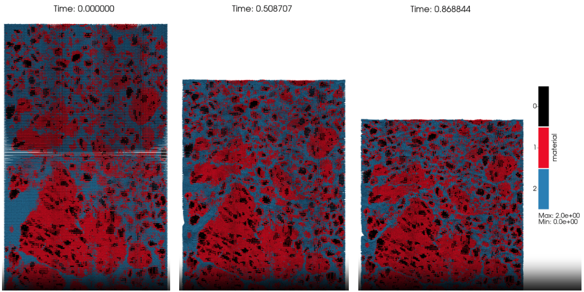
Example - MPM Sinker




```
$ bin/ratel-quasistatic -options_file examples/ymls/ex02-
    quasistatic-elasticity-mpm-neo-hookean-damage-current-
    sinker-cylinder.yml
```

FEM, iMPM simulations of dense sinker in "foam" validation problem

Example - Press Simulation



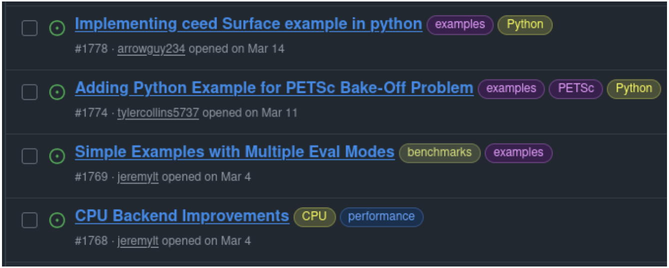
Compression of mock HE grains, binder mixture

Dev Best Practices

Open Source development best practices provide mentoring framework

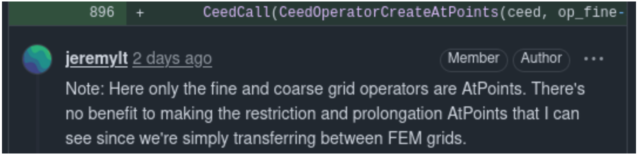
- Issues/planning act as literature review/research plan
- PR/MRs provide feedback about the work
- Testing verifies correctness, to a degree
- Documentation gives an opportunity to show understanding

Planning



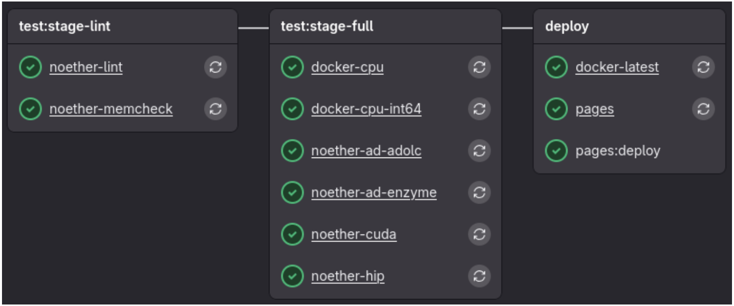
- Issues focus the effort and coalesce conversations
- Planning publicly (Issues, Slack/Zulip) records decisions
- Good example: https://gitlab.com/micromorph/ratel/issues/270

Review



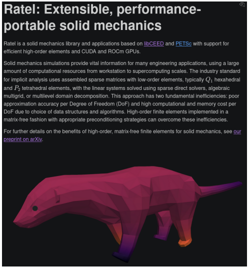
- All code can be strengthened with review and feedback
- PR/MRs provide tangible assets to guide any discussion
- Public review lets multiple people comment; prevents surprises

Testing



- Tests communicate intended usage and verify correctness
- New logic must have tests, code coverage is a **guide**
- Static/dynamic analysis (ASAN, clang-tidy, ...)

Documentation



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.

- Another avenue to communicate/verify understanding
- Documentation can also be used to start papers/dissertations
- Facilitates planning new features, on-boarding new members

Questions?

libCEED Repo: https://github.com/CEED/libCEED
Ratel Repo: https://gitlab.com/micromorph/ratel

Grant: Predictive Science Academic Alliance Program (DE-NA0003962)

Ratel - New Foundations of Computational Mechanics for the Exascale Era

Jeremy L Thompson

University of Colorado Boulder

jeremy@jeremylt.org

1 May 2025

Hans Meuer, Erich Strohmaier, Jack Dongarra, Horst Simon, and Martin Meuer.
Top 500 list, 2020.

Karl Rupp.
CPU-GPU-MIC comparision charts, 2020.

Ratel - New Foundations of Computational Mechanics for the Exascale Era

Jeremy L Thompson

University of Colorado Boulder

jeremy@jeremylt.org

1 May 2025