Matrix-Free MPM on High-Order Meshes with Ratel and libCEED

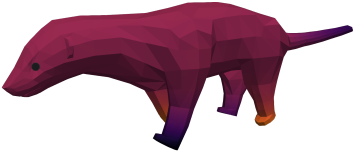
Jeremy L Thompson

University of Colorado Boulder

jeremy@jeremylt.org

15 July 2025

Ratel Team



Repository: 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

The authors acknowledge support by the Department of Energy, National Nuclear Security Administration, Predictive Science Academic Alliance Program (PSAAP) under Award Number DE-NA0003962.

Overview

Ratel - high order, performance portable solid mechanics

Built on libCEED and PETSc

GPU and CPU performance

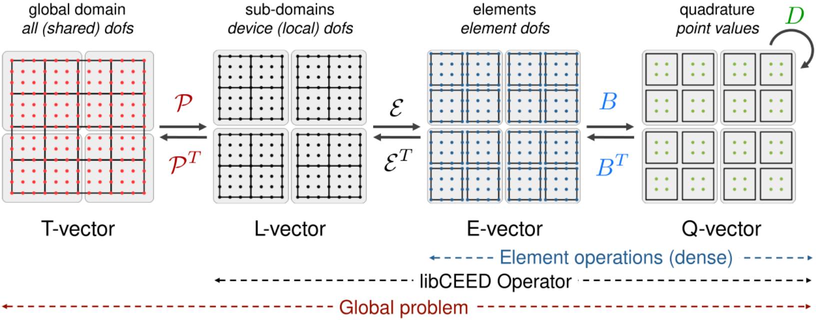
Overview

ECP Roots

- Ratel built directly on results from ECP CEED project
- libCEED provides high-performance operator evaluation
- PETSc provides linear/non-linear solvers and time steppers
- Ratel built from libCEED + PETSc solid mechanics demo app

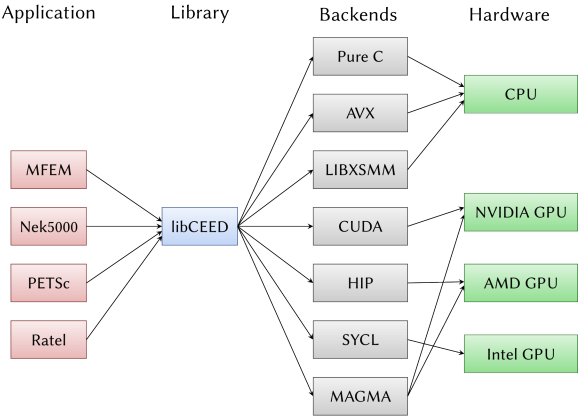
Matrix-Free Operators from libCEED

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



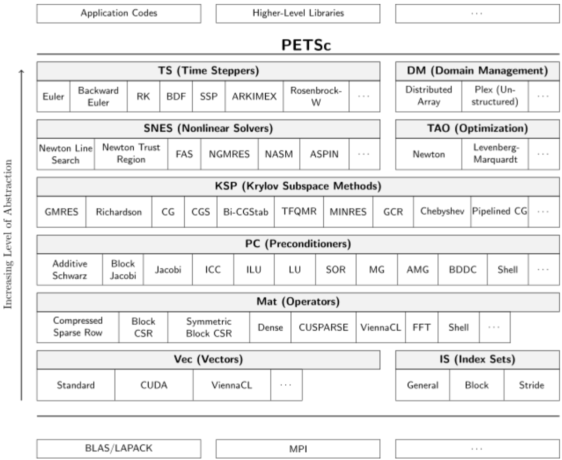
libCEED provides arbitrary order matrix-free operator evaluation

Performance Portability from libCEED



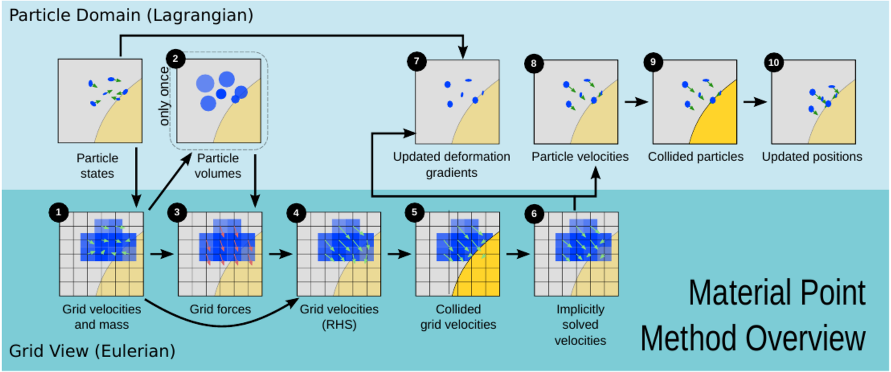
Performance portability with libCEED's matrix-free operators

Extensible Solvers from PETSc



PETSc provides extensible, scalable solvers

What is MPM?



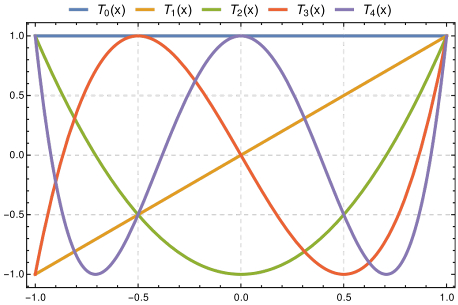
- Continuum based particle method with background mesh for gradients
- Extension of FLIP (which is an extension of PIC)
- Enables large deformation simulations with complex features

MPM vs FEM

MPM can be formulated as very similar to FEM

- Problem on background mesh changes when material points move
- Can be viewed as FEM with arbitrary quadrature point locations
- Natural fit for libCEED matrix-free representation
- Ratel FEM infrastructure provides fast background mesh solves

libCEED Basis Evaluation to Points



- Interpolate from primal to dual (quadrature) space
- Fit Chebyshev polynomials to values at quadrature points
- Evaluate Chebyshev polynomials at arbitrary points

libCEED Basis Evaluation to Points

Interpolation to Chebyshev has same FLOPs as FEM $\mathcal{O}\left(q^4\right)$

- Invert map C^{-1} from quadrature points to Chebyshev coeffs
- Create 1D interpolation matrix $B = CN$
- Tensor product:
  $B = (C \otimes C \otimes C)(N \otimes N \otimes N) = (CN) \otimes (CN) \otimes (CN)$
- Additional cost from evaluation to arbitrary points

libCEED Basis Evaluation to Points

Per point evaluation has higher FLOPs $\mathcal{O}\left(q^6\right)$

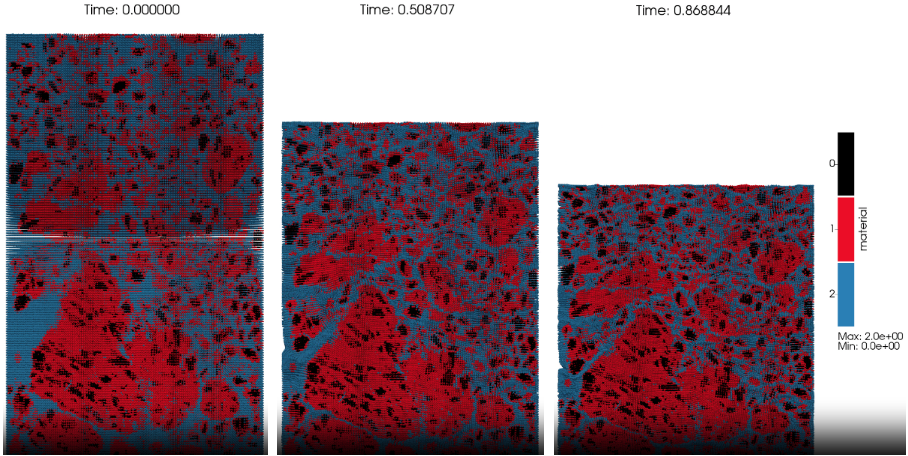
- Recurrence for Chebyshev values at point
  $f_0 = 1,\ f_1 = 2x,\ f_n = 2xf_{n-1} - f_{n-2}$
  $f_0' = 0,\ f_1' = 2,\quad f_n' = 2xf_{n-1}' + 2f_{n-1} - f_{n-2}'$
- Contract pencil of values with element coefficients
- Operation is independent per quadrature point
- $\mathcal{O}\left(q^3\right)$ FLOPs at $\mathcal{O}\left(\hat{q}^3\right)$ points (often $q \approx \hat{q}$)

AtPoints Operator

Final operator very similar to FEM

- $L = \mathcal{E}^T B^T B^{eT} D B^e B \mathcal{E}$ - CeedOperator
- All other operations identical to FEM
- libCEED gives action of local MPM operator
- PETSc responsible for communication between devices
  $$A = P^T L P$$

Sample Run



Confined compression of mock HE material

CEED Benchmark Problems

Performance on CEED BPs

- BP1 - Scalar projection problem
- BP2 - 3 component projection problem
- BP3 - Scalar Poisson problem
- BP4 - 3 component Poisson problem

Bulk of FLOPs are in basis evaluation

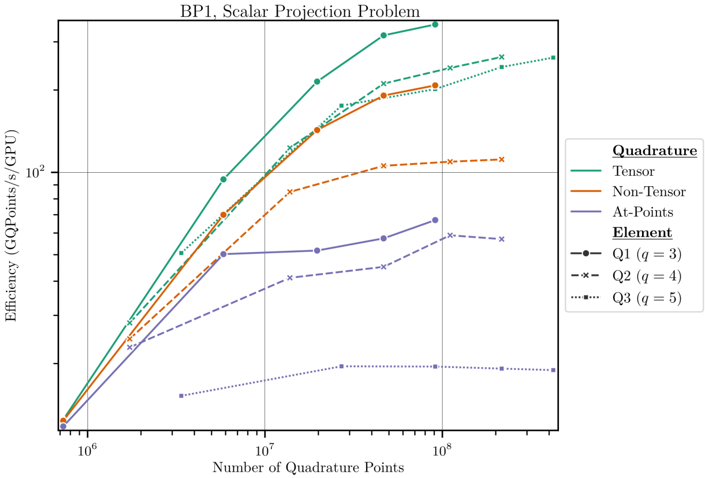
CEED Benchmark Problems

Performance on CEED BPs

- $p = 2, 3, 4$ and $q = p + 1$
- Units cube with 30^3, 60^3, 90^3, 120^3, and 150^3 elements
- Compare tensor, non-tensor, and at-points basis evaluation
- MMS w/ partial sum of Weierstrass function, $a = 0.5$, $b = 1.5$, $N = 2$

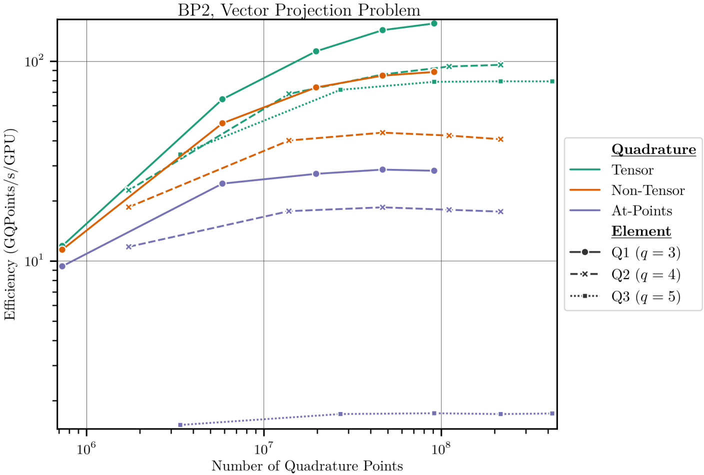
Using 4x AMD Instinct™MI300A Accelerated Processing Units (APUs)

BP1



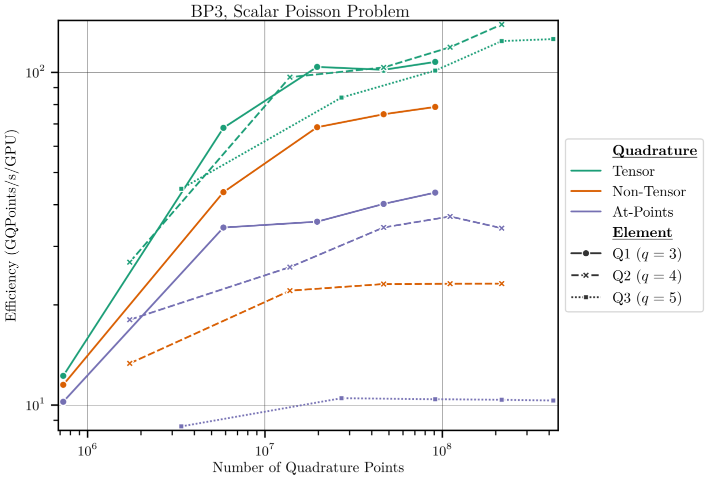
More FLOPs to do leads to lower efficiency

BP2



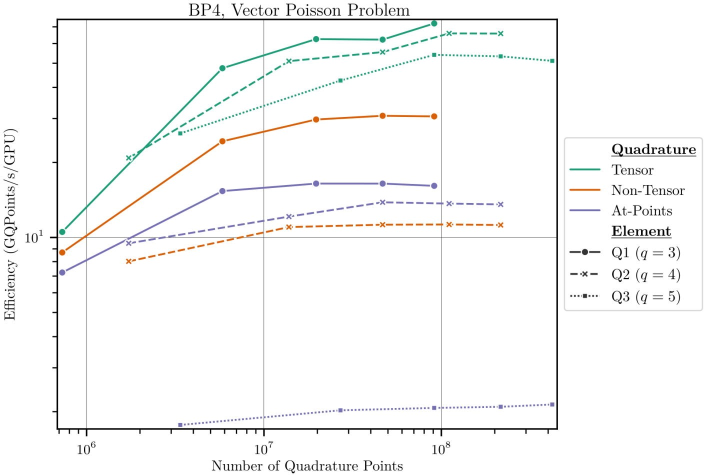
With more components, reach peak efficiency faster

BP3



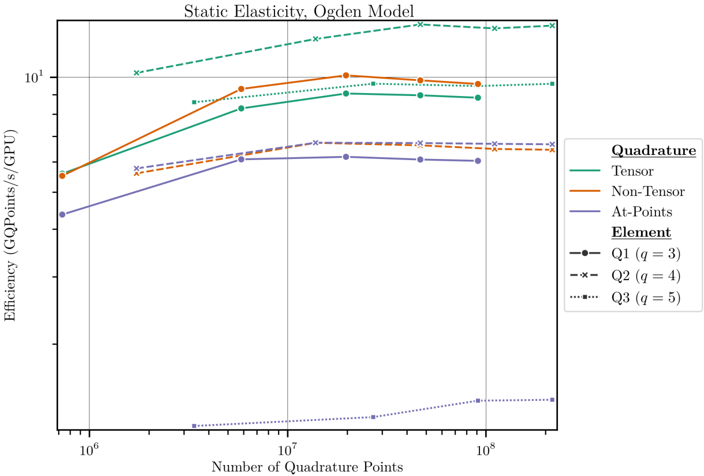
With derivatives, at-points closer to non-tensor

BP4



Closest benchmark to representative workload

Ogden



Basis cost less important with heavier QFunctions

Preconditioning

Practical problems require preconditioning

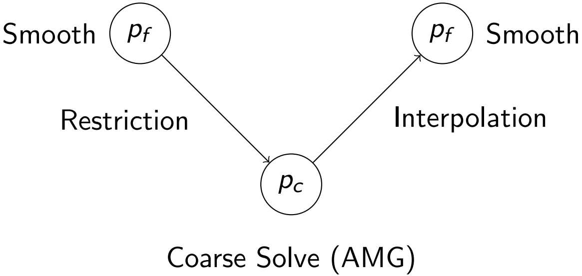
- Problems for MPM tend to be poorly conditioned
- Poor conditioning + expensive Mat-Vec = need preconditioning
- Varying structure between elements makes assembly more difficult

PETSc PCMG

- PCMG - PETSc geometric multigrid preconditioner
- Requires several operators from the user
  - Restriction operator
  - Interpolation operator
  - Smoother
  - Coarse grid solver

Ratel PCpMG

2 level multigrid with PCpMG



Ratel PCpMG

pMG giving promising initial results with GPU impl

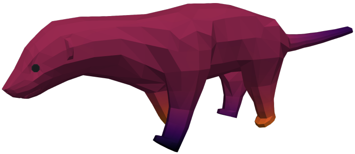
- Finite strain elasticity with damage
- Confined press of grain/binder with "sticky air" voids
- Jacobi iterations tend to double with 2x refinement
- pMG iteration counts robust with refinement

	# MPM Points	Jacobi its	pMG its
Coarse	388,800	900-1000	35-45
Fine	7,372,800	-	25-40

Future Work

- Continued iMPM development
- AtPoints basis and assembly perf tuning
- More models using Automatic Differentiation
- Further contact models development
- Rust QFunctions
- UHyper, UMat integration
- Addition of fluid dynamics models
- Upstream PETSc + libCEED integration
- We invite contributors and friendly users

Questions?



Repository: https://gitlab.com/micromorph/ratel

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

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