Ratel: High Order Solid Mechanics with libCEED and PETSc

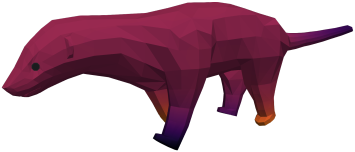
Jeremy L Thompson

University of Colorado Boulder

jeremy@jeremylt.org

15 August 2023

Ratel Team



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

Developers: Zach Atkins, Jed Brown, Leila Ghaffari,
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

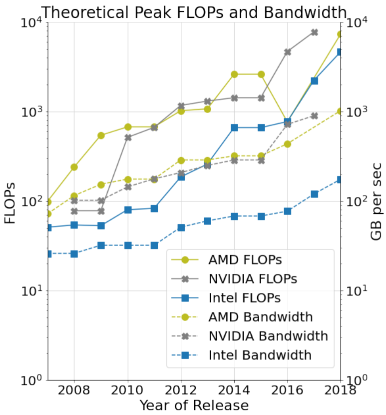
GPU and CPU performance

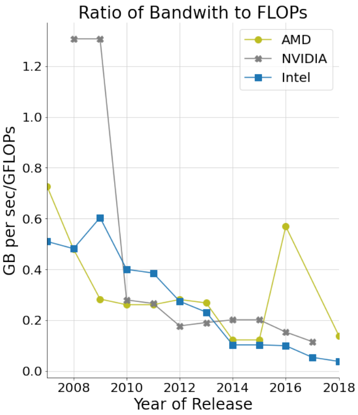
Overview

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

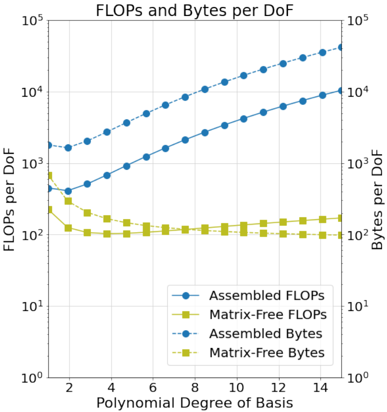
Modern Hardware

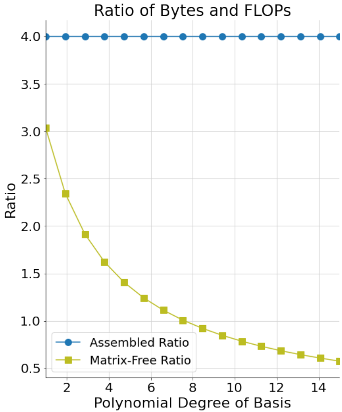




Modern hardware has lower memory bandwidth than FLOPs

Benefits of Matrix-Free



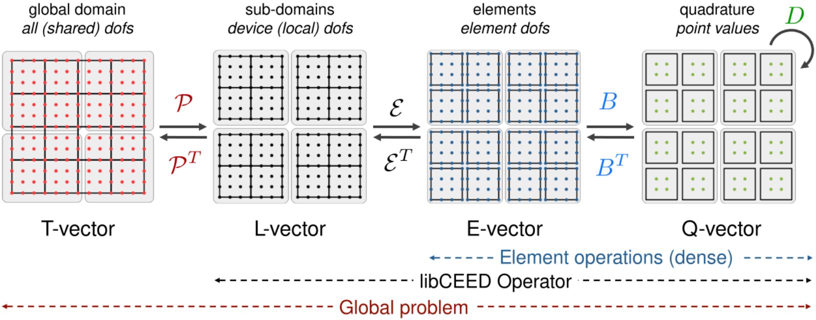


Requirements for matrix-vector product with sparse matrix vs matrix-free for screened Poisson $\nabla^2 u - \alpha^2 u = f$ in 3D

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

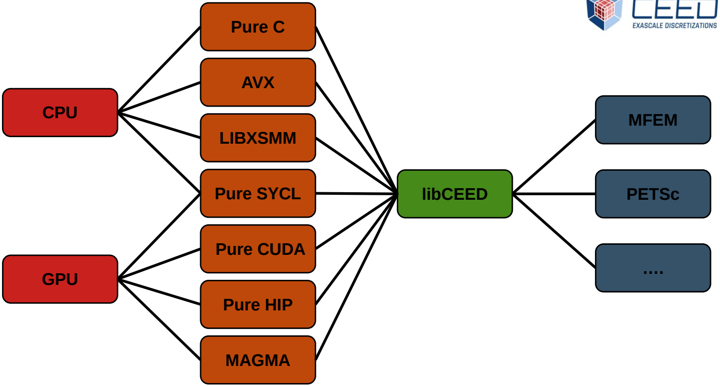
Matrix-Free Operators from libCEED

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



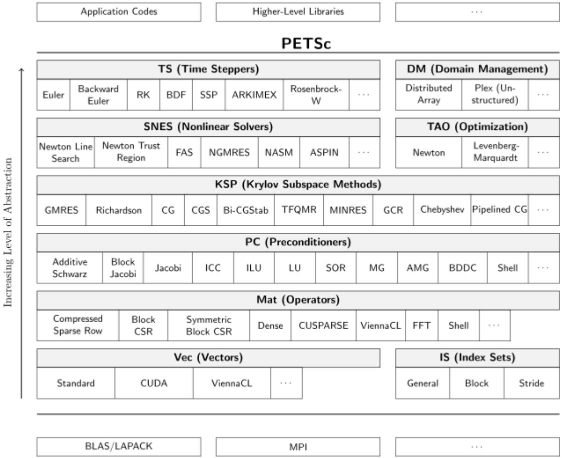
libCEED provides arbitrary order matrix-free operator evaluation

Performance Portability from libCEED



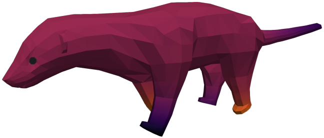
Performance portability with libCEED's matrix-free operators

Extensible Solvers from PETSc



PETSc provides extensible, scalable solvers

Ratel



Ratel supports...

- hyperelastic and plastic material models
- on unstructured meshes with multiple material regions
- with arbitrary order mixed finite elements
- using efficient matrix-free operator evaluation
- with multiple solver and time-stepper options
- and p-multigrid preconditioners using AMG course grid solves
- on both CPU and GPU with runtime selection

Material Models

Ratel supports several material models:

- Linear elasticity
- Neo-Hookean hyperelasticity
- Mooney-Rivlin hyperelasticity
- Odgen hyperelasticity
- Linear plasticity with linear hardening (further work ongoing)
- Linear elasticity with AT2 damage model (in testing)
- CEED benchmark problems
- more in development...

Material Model Specification

```
dm_plex:
  filename: examples/meshes/rod-and-binder.msh
  simplex: 0

material: rod,binder

binder:
  model: elasticity-neo-hookean-initial
  E: 2.
  nu: 0.4
  label_value: 4

rod:
  model: elasticity-mooney-rivlin-initial
  mu_1: 0.5
  mu_2: 0.5
  nu: 0.4
  label_value: 3
```

Additional Material Models

Many material models support several options:

- Initial configuration
- Current configuration
- Automatic differentiation (Enzyme)
- Isochoric formulation
- Mixed FEM (displacement, pressure split, incompressible materials)

Boundary Conditions

Ratel supports various boundary conditions:

- Time varying Dirichlet clamp
- Time varying Dirichlet slip
- Time varying traction
- Pressure loading due to liquid or gas contact
- Nitsche's method solid contact with Coulomb friction

Boundary Condition Specification

```yaml
bc:
  clamp: 5
  clamp_5_translate: -0.2,0,0, 0,0,0, .2,0,0
  clamp_5_rotate: 1,0,0,.2,0, 1,0,0,0,0, 1,0,0,-.2,0,
  clamp_5_times: 0.33, 0.66, 1.0
  clamp_5_interpolation: linear

  platen: 6
  platen_6:
    normal: -1,0,0
    center: 1.0,0.5,0.5
    distance: 0.2
    gamma: 1000
```

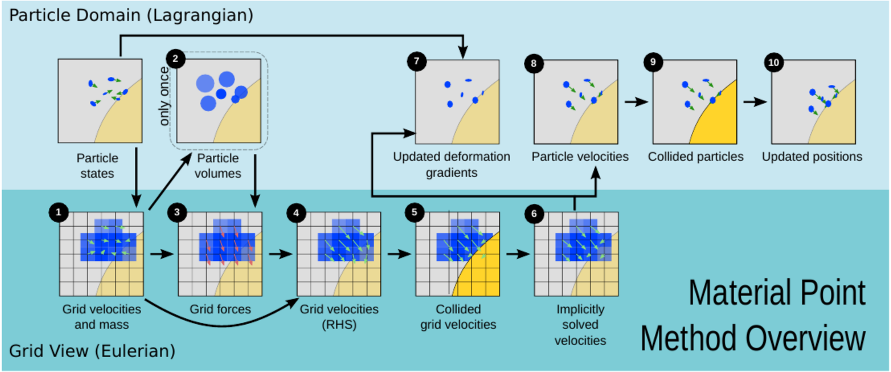
Solver Modes

Three sample applications are provided:

- Static Elasticity
- Quasi-static Elasticity
- Dynamic Elasticity

all using p-multigrid + coarse grid AMG by default

What is MPM?

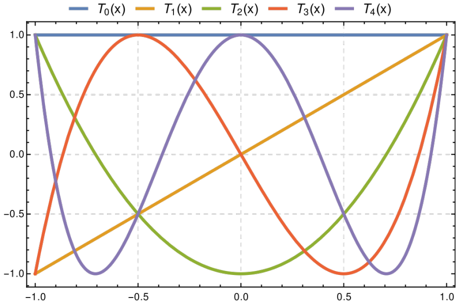


- Continuum based partical method with background mesh for gradients
- Extension of FLIP (which is an extension of PIC)
- Used in rendering for the movie *Frozen*

What does MPM have to do with FEM?

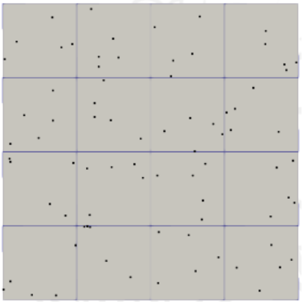
- Problem on background mesh changes when material points move
- Natural fit for matrix-free representation
- Similar reasoning to use matrix-free for adaptive methods
- Ratel FEM infrastructure provides fast background mesh solves

libCEED Basis Evaluation to Points



- Interpolate from primal to dual (quadrature) space
- Fit Chebyshev polynomials to values at quadrature points
- Evaluate Chebyshev polynomials at reference coords of material points
- Transpose the order for projection to mesh from material points

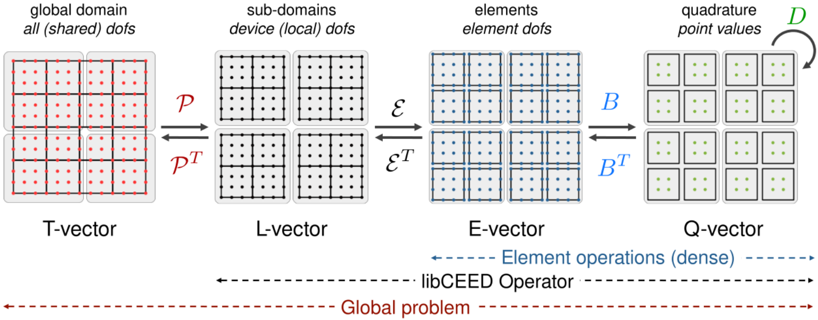
DMSwarm for Material Points



- PETSc DMSwarm manages material points
- PETSc DMPlex manages cells (elements)
- Exposing API for cell reference coordinates of points

Current Work

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



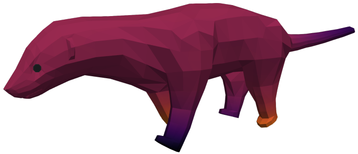
Building the libCEED element restriction operation for material points

Also need CeedOperator API to support evaluation at material points

Future Work

- Continued iMPM development
- Addition of fluid dynamics models
- PCPMG and PCFIELDSPLIT integration
- Upstream PETSc + libCEED integration
- Python and Rust interfaces
- User model interface in C and Rust
- We invite contributors and friendly users

Questions?



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

Ratel Team: Zach Atkins, Jed Brown, Leila Ghaffari, Rezgar Shakeri, Ren Stengel, Jeremy L Thompson

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