Ratel - Using MI300A APUs with libCEED and PETSc

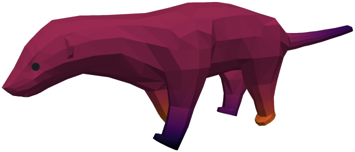
Jeremy L Thompson

University of Colorado Boulder

jeremy@jeremylt.org

11 Feb 2026

Ratel Team



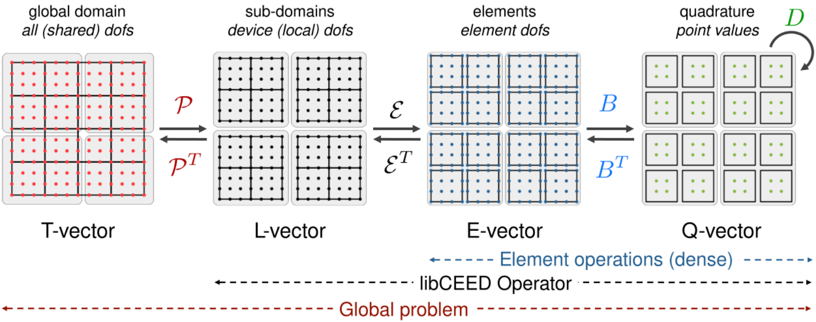
Repository: https://gitlab.com/micromorph/ratel

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.

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 FEM-like operators

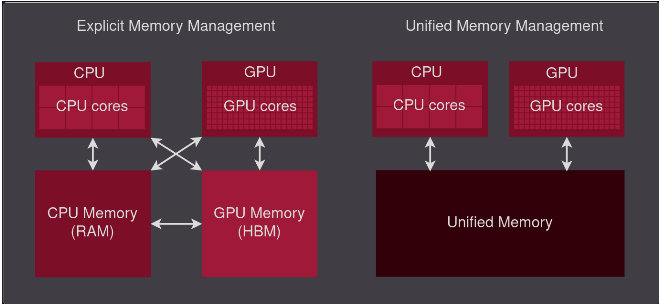
MatCEED

`MatCEED` wraps a linear `CeedOperator`

```c
MatCeedCreate(DM dm_x, DM dm_y, CeedOperator op_mult, CeedOperator op_mult_transpose,
      Mat *mat);
MatCeedSetTime(Mat mat, PetscReal time);

// COO assembly support
MatCeedCreateMatCOO(Mat mat_ceed, Mat *mat_coo);
MatCeedSetPreallocationCOO(Mat mat_ceed, Mat mat_coo);
MatCeedAssembleCOO(Mat mat_ceed, Mat mat_coo);

// Private - Core functionality
MatMult_Ceed(Mat A, Vec X, Vec Y)
MatMultTranspose_Ceed(Mat A, Vec Y, Vec X)

// Private - PC support
MatGetDiagonal_Ceed(Mat A, Vec D);
MatGetDiagonalBlock_Ceed(Mat mat_ceed, Mat *mat_block);
MatInvertBlockDiagonal_Ceed(Mat mat_ceed, const PetscScalar **values);
MatInvertVariableBlockDiagonal_Ceed(Mat mat_ceed, PetscInt num_blocks, const PetscInt
      *block_sizes, PetscScalar *values);
```

`MatCEED` can be used as a standard PETSc `Mat`

MI300As



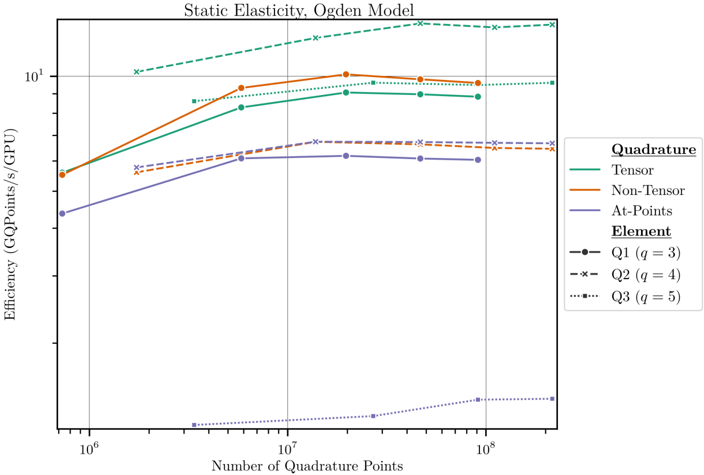
Unified memory on MI300A removes CPU-GPU communication
(See also AI MAX+395)

Mildly slower memory but generally worth the tradeoff

Porting Outcomes

- PETSc and libCEED changes to support APUs
  - Allocate all memory as device memory
  - Treat same pointer as host memory
  - Must forbid mismatched memory operations (`memset`)
- Unified memory improves performance on APU hardware
  - At least 5% speedup with unified memory

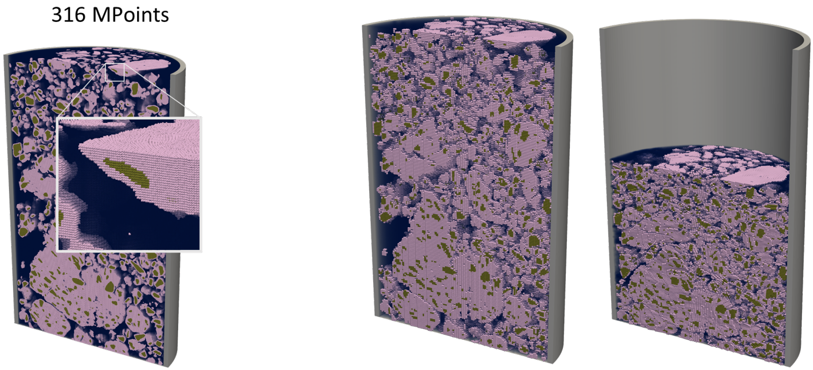
Ogden



Can run MPM models efficiently

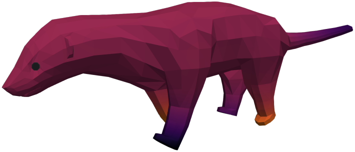
Reduced timestep setup costs (point migration, not shown)

Example - Press Simulation



Compression of mock HE grains (gold) and binder (pink) mixture
(Reset background mesh to computational region on each timestep)
Latest runs use ∼10B material points

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)