Preconditioning Matrix-Free High-Order Finite Element Operators

Jeremy L Thompson

University of Colorado Boulder

jeremy.thompson@colorado.edu

Apr 13, 2020

Overview

Finite element methods with global sparse matrices are not adequate for exascale solid mechanics problems

High-order matrix-free operators offer superior performance with respect to FLOPs and memory transfer for a matrix-vector product

p-multigrid preconditioners are effective for matrix-free FEM and new to Neo-Hookean hyperelasticity at finite strain

Further research required for matrix-free smoothers, BDDC smoothers, subdomain solvers, and split preconditioners

Overview

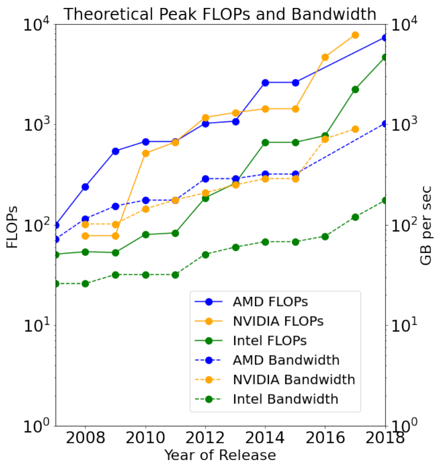
1. Introduction
2. Hardware Constraints
3. Matrix-Free Finite Elements
4. Hyperelasticity
5. Preconditioning
    - p-multigrid
    - Smoothers
    - Subdomain Solvers
    - Split Preconditioning
6. Future Work
7. Questions

Center for Efficient Exascale Discretizations

DoE exascale co-design center

- Design discretization algorithms for exascale hardware that deliver significant performance gain over low order methods
- Collaborate with hardware vendors and software projects for exascale hardware and software stack
- Provide efficient and user-friendly unstructured PDE discretization component for exascale software ecosystem

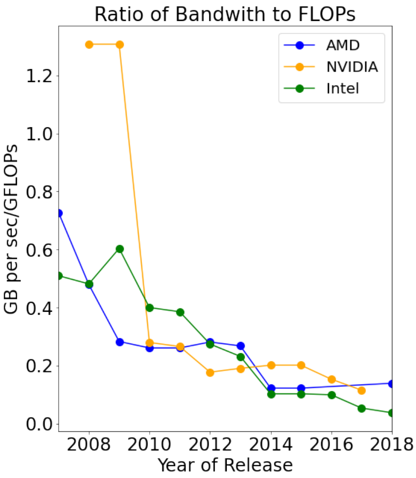
Hardware Limitations



- Memory and network bandwidth improvements lag behind FLOPs
- Trend is consistent across manufacturers
- Similar developments for different systems over 30 years

FLOPs vs Bandwidth

- Memory bound applications can't reach peak FLOPs
- Network communication also a well known scaling issue
- GPU computation also requires host-device communication



Benchmarks

System	HPL TFLOPs	HPCG TFLOPs	% HPL
Summit	148, 600.0	2, 925.75	1.97
Sierra	94, 640.0	1, 795.67	1.90
Trinity	20, 158.7	546.12	2.71
ABCI	19.880.0	508.85	2.56
Piz Daint	21, 230.0	496.98	2.34

- Applications often can't match HPL performance
- Memory and network bandwidth limit application code
- Matrix-free formulations can target this disparity

Finite Element Operators

PDE Weak Form:

find $u \in V$ such that for all $v \in V$

$$\langle v, u \rangle = \int_\Omega v \cdot f_0 \left(u, \nabla u \right) + \nabla v : f_1 \left(u, \nabla u \right) = 0$$

- Weak form of PDEs are linear in the test functions
- PDE need not be linear for this general form
- Boundary integrals introduce similar terms

Galerkin System

Galerkin System:

$$\sum_e \mathcal{E}^T \left[\left(\mathbf{B}^e\right)^T \mathbf{W}^e \Lambda \left(f_0 \left(u^e, \nabla u^e \right) \right) + \sum_{i=0}^{d-1} \left(\mathbf{D}_i^e \right)^T \mathbf{W}^e \Lambda \left(f_1 \left(u^e, \nabla u^e \right) \right) \right] = 0$$

where $u^e = \mathbf{B}^e \mathcal{E}^e u$ and $\nabla u^e = \{ \mathbf{D}_i^e \mathcal{E}^e u \}_{i=0}^{d-1}$

- $\mathcal{E}^e$ - element restriction operator
- $\mathbf{B}^e / \mathbf{D}^e$ - interpolation/derivatives from DoFs to quadrature points
- W^e - element quadrature weights, with geometric deformation
- Λ - pointwise multiplication at quadrature points

Galerkin System

Galerkin System:

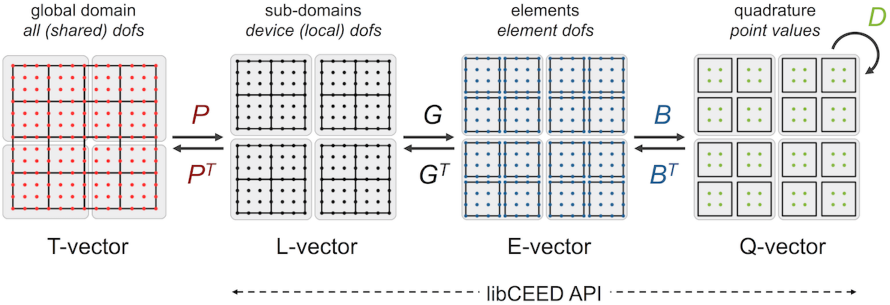
$$\sum_e \mathcal{E}^T \left[\left(\mathbf{B}^e\right)^T \mathbf{W}^e \Lambda \left(f_0\left(u^e, \nabla u^e\right)\right) + \sum_{i=0}^{d-1} \left(\mathbf{D}_i^e\right)^T \mathbf{W}^e \Lambda \left(f_1\left(u^e, \nabla u^e\right)\right) \right] = 0$$

where $u^e = \mathbf{B}^e \mathcal{E}^e u$ and $\nabla u^e = \{\mathbf{D}_i^e \mathcal{E}^e u\}_{i=0}^{d-1}$

- Can express non-linear residual evaluators or linear Jacobians
- Notice no explicit mesh structure or homogeneity assumed
- Face integrals introduce similar terms

Practical Implementation

$$\boxed{A = P^T G^T B^T D B G P}$$



Efficient Implementation

Galerkin System:

$$A = \sum_e \mathcal{E}^T \left[(\mathbf{B}^e)^T \mathbf{W}^e \Lambda \left(f_0 \left(u^e, \nabla u^e \right) \right) + \sum_{i=0}^{d-1} (\mathbf{D}_i^e)^T \mathbf{W}^e \Lambda \left(f_1 \left(u^e, \nabla u^e \right) \right) \right]$$

where $u^e = \mathbf{B}^e \mathcal{E}^e u$ and $\nabla u^e = \{ \mathbf{D}_i^e \mathcal{E}^e u \}_{i=0}^{d-1}$

- Same framework for assembly of sparse matrices for finite elements
- Eschewing assembly allows optimizations and parallelism

Tensor Contractions

3D Tensor Basis Operators:

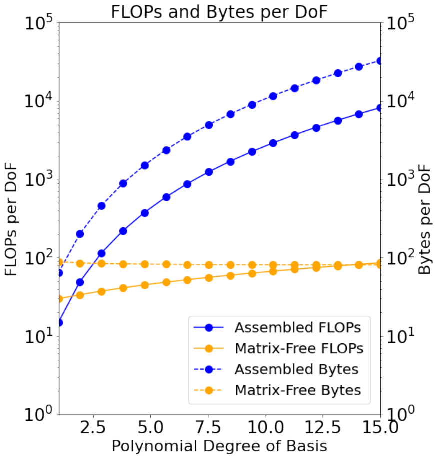
$$\mathbf{B} = B \otimes B \otimes B \qquad \mathbf{D}_0 = D \otimes B \otimes B$$

$$\mathbf{D}_1 = B \otimes D \otimes B \qquad \mathbf{D}_2 = B \otimes B \otimes D$$

$$\mathbf{W} = W \otimes W \otimes W$$

- B, D, W are 1D basis and quadrature weight matrices
- Basis evaluation is computationally expensive
- Tensor product elements allow efficient basis operations

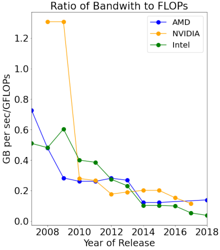
Single Element Performance

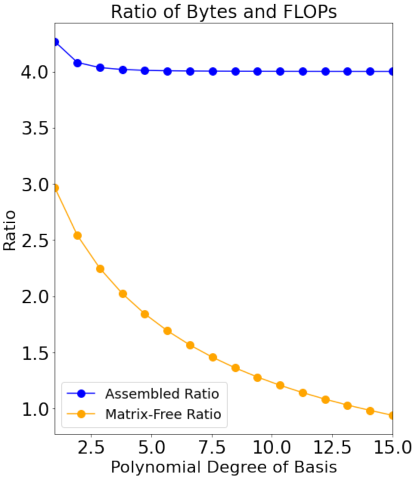


- Test operator: $\left(\nabla^2 - \alpha^2\right) u$
- Assembled:
  FLOPs and memory per DoF scale cubicly
- Matrix-Free:
  FLOPs per DoF scale linearly, memory constant

Single Element Performance

- Matrix-free implementation closer to hardware capabilities
- Performance gets better at higher order





Hyperelasticity

Static balance of linear momentum at finite strain:

$$-\nabla_X \cdot \boldsymbol{P} - \rho_0 \boldsymbol{g} = 0$$

First Piola-Kirchhoff stress tensor:

$$\boldsymbol{P} = \boldsymbol{F}\,\boldsymbol{S} \qquad \boldsymbol{F} = \boldsymbol{I} + \nabla_X u$$

- Hyperelasticity at finite strain near the incompressible regime
- Second Piola-Kirchhoff stress tensor $\boldsymbol{S}$ given by constitutive model
- Standard implementations struggle to scale on modern hardware

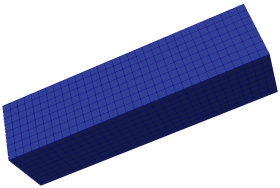
Neo-Hookean Hyperelasticity

Neo-Hookean constitutive model:

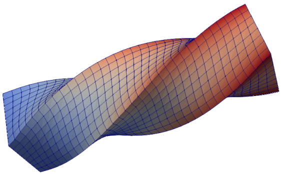
$$\boldsymbol{S} = \lambda \log\left(|\boldsymbol{F}|\right) \boldsymbol{C}^{-1} + 2\mu \boldsymbol{C}^{-1} \boldsymbol{E}$$

- In terms of:
  Right Cauchy-Green tensor $\boldsymbol{C} = \boldsymbol{F}^T \boldsymbol{F}$
  Green-Lagrange strain tensor $\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{C} - \boldsymbol{I}\right)$
- Lamé parameters:
  $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$ and $\mu = \frac{E}{2(1+\nu)}$
- Convergence slow near incompressible limit: $\nu \to 0.5$

Test Case



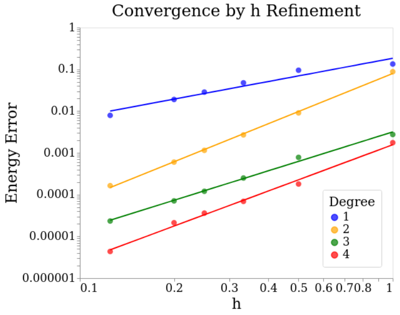
Before Deformation

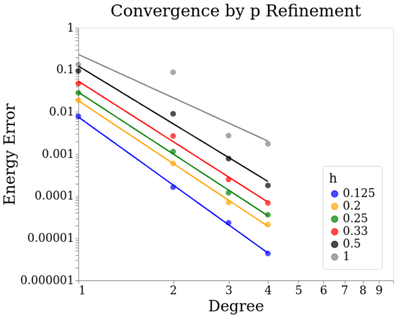


After Deformation

- Deformation of rectangular or cylindrical beam
- 1 radian axial twist while translating opposite end

Self Convergence Study





- Error in strain energy under mesh refinement
- Currently there is an issue with the study for degree 3 and 3

Preconditioners

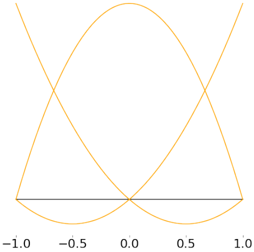
Left Preconditioning:

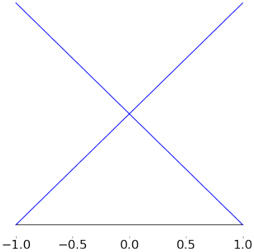
$$\mathbf{A}x = b \quad \rightarrow \quad \mathbf{M}^{-1}\mathbf{A}x = \mathbf{M}^{-1}x$$

where $\mathbf{M}^{-1} \approx \mathbf{A}^{-1}$

- Matrix-free operators require iterative solvers
- Preconditioning is required for efficient implementation
- Conjugate Gradient is a popular but restrictive iterative solver

p-multigrid



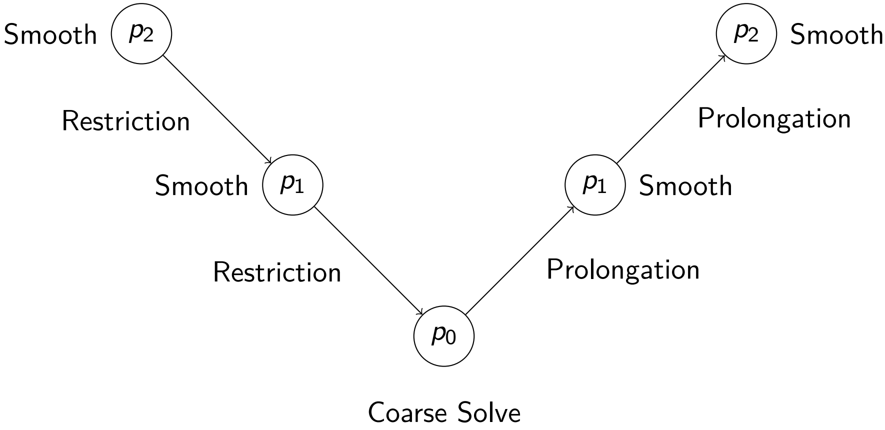


p-multigrid is ideal for matrix-free on unstructured meshes

- Multigrid provides mesh independent convergence
- Algebraic multigrid requires matrix assembly
- *h*-multigrid difficult on unstructured/mixed meshes

V-Cycle

3 level multigrid example



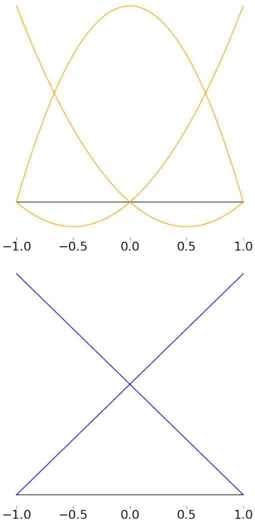
Prolongation and Restriction

p-multigrid Prolongation:

$$\mathbf{P}_{p-1}^{p} = \Lambda\left(m_p^{-1}\right) \mathcal{E}_p{}^T \sum_e \mathbf{B}_{p-1}^{p} \mathcal{E}_{p-1}^{e}$$

where $m_p = \mathcal{E}_p{}^T \mathcal{E}_p 1$

- Matrix-free implementation
- Restriction is the transpose of prolongation $\mathbf{R}_p^{p-1} = \left(\mathbf{P}_{p-1}^{p}\right)^T$



Coarse Solve

Low frequency error correction solve

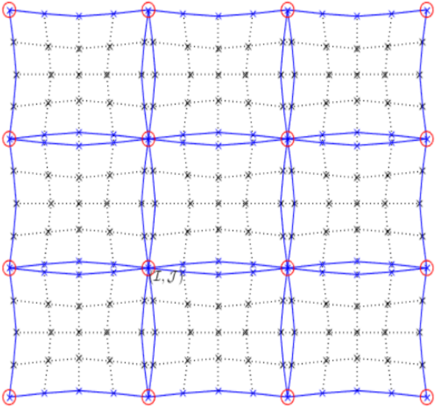
- Coarse solve could be another iterative solver
- Problem size reduced by factor of $\sim p^3/8$
- Algebraic Multigrid attractive as coarse solver
- Balance of accuracy/communication for coarse solve needs tuning

Smoothers

Smoothers required for high frequency errors:

- Jacobi and Chebyshev semi-iterative method
  - Well established but needs efficient diagonal assembly
- Balancing Domain Decomposition by Constraints
  - New technique for multigrid smoother
  - Cheap subdomain solvers for non-separable problems needed

BDDC



- Non-overlapping domain decomposition strategy (Dohrmann 2003)
- Reduced substructure of shared DoFs, similar to FETI-DP

BDDC

BDDC Smoother:

$$\hat{\mathbf{M}}^{-1} = \left(\mathbf{R}_1^T - \mathcal{H}\mathbf{J}_D\right) \hat{\mathbf{A}}^{-1} \left(\mathbf{R}_1 - \mathbf{J}_D^T \mathcal{H}^T\right)$$

- Scaled injection operator: $\mathbf{R}_1$
- Subdomain energy minimizer: $\left(\mathbf{R}_1 - \mathbf{J}_D^T \mathcal{H}\right)$

  where $\mathcal{H}$ is direct sum of $\mathcal{H}^{(i)} = -\left(\mathbf{A}_{II}^{(i)}\right)^{-1} \left(\mathbf{A}_{\Gamma I}^{(i)}\right)^T$

  and $\mathbf{J}_D$ is a map to create a local Dirichlet problem
- Subdomain solver: $\left(\mathbf{A}_{II}^{(i)}\right)^{-1}$, Substructure solver: $\widetilde{\mathbf{A}}^{-1}$

FDM

Fast Diagonalization Method exactly solves separable problems

Simultaneous Diagonalization:

$$M, K \;\rightarrow\; \mathcal{X}^T M \mathcal{X} = I, \quad \mathcal{X}^T K \mathcal{X} = L$$

Screened Poisson Diagonalization:

$$\alpha^2 M + K = \mathcal{X} \left(\alpha^2 I + L \right) \mathcal{X}^T$$

Fast Diagonalization Inverse:

$$\left(\alpha^2 M + K \right)^{-1} = \mathcal{X}^T \left(\alpha^2 I + L \right)^{-1} \mathcal{X}$$

FDM

Tensor Product Elements:

$$\mathbf{M} = M \otimes M \otimes M, \quad \mathbf{K}_0 = K \otimes M \otimes M$$

Screened Poisson Diagonalization:

$$\alpha^2 \mathbf{M} + \sum_{i=0}^{d-1} \mathbf{K}_i = \mathcal{X} \left(\alpha^2 \mathbf{I} + \sum_{i=0}^{d-1} \mathbf{L}_i \right) \mathcal{X}^T$$

Fast Diagonalization Inverse:

$$\left(\alpha^2 \mathbf{M} + \sum_{i=0}^{d-1} \mathbf{K}_i \right)^{-1} = \mathcal{X}^T \left(\alpha^2 \mathbf{I} + \sum_{i=0}^{d-1} \mathbf{L}_i \right)^{-1} \mathcal{X}$$

Separable Approximate Inverses

Separable Approximate Inverse:

$$\mathcal{X}^T \tilde{\lambda}^{-1} \mathcal{X}$$

- FDM can be efficiently applied matrix-free
- FDM cannot handle non-linear PDEs or geometric deformations
- Fisher et al. used separable approximation for geometric non-linearity
- Further approximations may be possible
- Inexact subdomain solvers for BDDC are effective

Incompressible Hyperelasticity

Incompressible hyperelasticity ($\nu = 0.5$) requires pressure field

$$\begin{bmatrix} \mathbf{F} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} g_u \\ g_p \end{bmatrix}$$

- Mixed finite element methods required for stable formulation
- p-multigrid formulation inadequate for mixed formulation
- Current preconditioner can provide ingredients for split preconditioner

Incompressible Hyperelasticity

Split Preconditioning with Schur Compliment:

$$\begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{B}^T & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{F} & 0 \\ 0 & \mathbf{S} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{B} \\ 0 & \mathbf{I} \end{bmatrix}, \quad \mathbf{S} = \mathbf{C} + \mathbf{B}^T \mathbf{F}^{-1} \mathbf{B}$$

- Widely studied by fluid dynamics community
- Preconditioner for displacement **F** can leverage *p*-multigrid

Roadmap

- ☒ Compressible hyperelastic solver
  - ☒ p-multigrid preconditioning
  - ☒ Jacobi and Chebyshev smoothers
  - ☐ BDDC smoother
    - ☐ FDM separable approximate subdomain solvers
- ☐ Incompressible hyperelastic solver
  - ☐ Split preconditioner

Questions?

Advisors : Jed Brown[1] & Daniel Appelö[1]

Collaborators: Arash Mehraban[1], Valeria Barra[1], Oana Marin[2], Tzanio Kolev[3], Jean-Sylvain Camier[3], Veselin Dobrev[3], Yohann Doudouit[3], Tim Warburton[4], David Medina[5], & Thilina Rathnayake[6]

Grant: Exascale Computing Project (17-SC-20-SC)

1: University of Colorado, Boulder
2: Argonne National Laboratory
3: Lawrence Livermore National Laboratory
4: Virginia Polytechnic Institute and State University
5: OCCA
6: University of Illinois, Urbana-Champaign

Preconditioning Matrix-Free High-Order Finite Element Operators

Jeremy L Thompson

University of Colorado Boulder

jeremy.thompson@colorado.edu

Apr 13, 2020