SAND2024-09364C

Sandia National Laboratories

Mesh Optimization and Basis Compression for Extreme-Scale Solution of Partial Differential Equations

Graham Harper, Denis Ridzal, Tim Wildey
Center for Computing Research, Sandia National Laboratories

WCCM 2024

July 22, 2024

LDRD
Laboratory Directed Research and Development

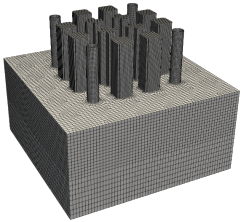
U.S. DEPARTMENT OF ENERGY | NNSA

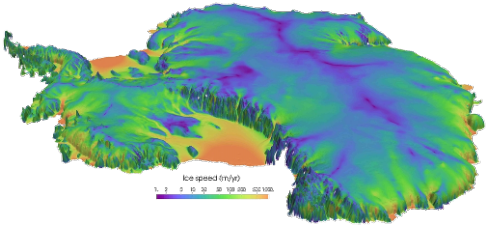
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2024-09364C

Motivation

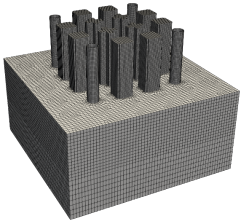
- Modern computing technologies (accelerators, edge, etc.) have **less memory bandwidth**.
- Supercomputing post-Moore's Law cannot rely on flops alone.
- Computations with less data can **save compute time and reduce memory overhead**.
- Mathematical structure in finite element methods (FEMs) can be exploited.

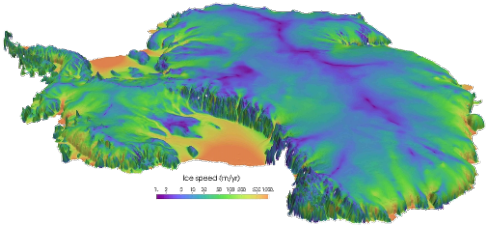




Motivation

- Modern computing technologies (accelerators, edge, etc.) have **less memory bandwidth**.
- Supercomputing post-Moore's Law cannot rely on flops alone.
- Computations with less data can **save compute time and reduce memory overhead**.
- Mathematical structure in finite element methods (FEMs) can be exploited.



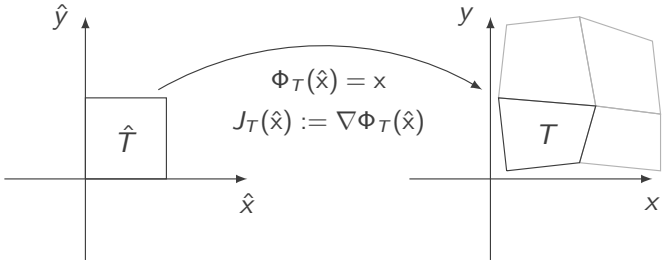


Two main goals of this talk:

- Detection and exploitation of structure in FEM computations
- Enhancement of structure for FEM computations

Reference to Physical Mapping

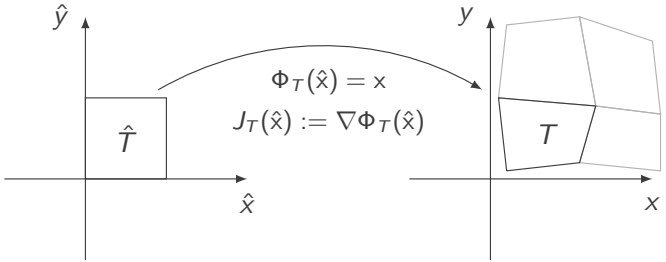
The most important tool for evaluating FEM integrals is the **reference to physical mapping**.



The reference to physical mapping Φ_T for a mesh cell T and reference cell $\hat{T}$, which relates a function $\hat{f}(\hat{x})$ with $f(x)$ by $f(x) = \hat{f}(\Phi_T^{-1}(x))$

Reference to Physical Mapping

The most important tool for evaluating FEM integrals is the **reference to physical mapping**.



The reference to physical mapping Φ_T for a mesh cell T and reference cell $\hat{T}$, which relates a function $\hat{f}(\hat{x})$ with $f(x)$ by $f(x) = \hat{f}(\Phi_T^{-1}(x))$

Oftentimes, such FEM assembly calculations scale like

$$\mathcal{O}(N(p+1)^d q^d d) \tag{1}$$

Our target is the term in red.

Finite Element Operator Assembly

Main computation: **integrals**
From calculus,

$$\int_T f(\mathrm{x})\, d\mathrm{x} = \int_{\hat{T}} \hat{f}(\hat{\mathrm{x}}) \det(J_T(\hat{\mathrm{x}}))\, d\hat{\mathrm{x}}. \tag{2}$$

Finite Element Operator Assembly

Main computation: **integrals**
From calculus,

$$\int_T f(\mathrm{x})\, d\mathrm{x} = \int_{\hat{T}} \hat{f}(\hat{\mathrm{x}}) \det(J_T(\hat{\mathrm{x}}))\, d\hat{\mathrm{x}}. \tag{2}$$

From differential geometry,

$$\int_T \nabla f(\mathrm{x})\, d\mathrm{x} = \int_{\hat{T}} J_T^{-T}(\hat{\mathrm{x}}) \nabla \hat{f}(\hat{\mathrm{x}}) \det(J_T(\hat{\mathrm{x}}))\, d\hat{\mathrm{x}}. \tag{3}$$

Finite Element Operator Assembly

Main computation: **integrals**
From calculus,

$$\int_T f(\mathrm{x})\, d\mathrm{x} = \int_{\hat{T}} \hat{f}(\hat{\mathrm{x}}) \det(J_T(\hat{\mathrm{x}}))\, d\hat{\mathrm{x}}. \tag{2}$$

From differential geometry,

$$\int_T \nabla f(\mathrm{x})\, d\mathrm{x} = \int_{\hat{T}} J_T^{-T}(\hat{\mathrm{x}}) \nabla \hat{f}(\hat{\mathrm{x}}) \det(J_T(\hat{\mathrm{x}}))\, d\hat{\mathrm{x}}. \tag{3}$$

Unified by the **de Rham complex**,

$$\begin{array}{rccccccc} \text{space:} & HGRAD & \xrightarrow{\nabla} & HCURL & \xrightarrow{\nabla\times} & HDIV & \xrightarrow{\nabla\cdot} & HVOL \\ & H^1(\Omega) & & H(curl,\Omega) & & H(div,\Omega) & & L^2(\Omega) \\ \text{pullback:} & I & & J^{-T} & & J\det(J)^{-1} & & \det(J)^{-1} \end{array}$$

Finite Element Operator Assembly

Main computation: **integrals**
From calculus,

$$\int_T f(\mathrm{x})\, d\mathrm{x} = \int_{\hat{T}} \hat{f}(\hat{\mathrm{x}}) \det(J_T(\hat{\mathrm{x}}))\, d\hat{\mathrm{x}}. \tag{2}$$

From differential geometry,

$$\int_T \nabla f(\mathrm{x})\, d\mathrm{x} = \int_{\hat{T}} J_T^{-T}(\hat{\mathrm{x}}) \nabla \hat{f}(\hat{\mathrm{x}}) \det(J_T(\hat{\mathrm{x}}))\, d\hat{\mathrm{x}}. \tag{3}$$

Unified by the **de Rham complex**,

space:	$HGRAD$	$\xrightarrow{\nabla}$	$HCURL$	$\xrightarrow{\nabla\times}$	$HDIV$	$\xrightarrow{\nabla\cdot}$	$HVOL$
	$H^1(\Omega)$		$H(curl, \Omega)$		$H(div, \Omega)$		$L^2(\Omega)$
pullback:	I		J^{-T}		$J\det(J)^{-1}$		$\det(J)^{-1}$

Most importantly, $J_T(\hat{\mathrm{x}})$ **is invariant under mesh cell translations**.

Finite Element Basis Compression Lemma

Lemma

If two mesh cells T_1 and T_2 are translations of each other, FEM operator assembly satisfies

$$\int_{T_1} \nabla\phi_i(\mathrm{x}) \cdot \nabla\phi_j(\mathrm{x})\, d\mathrm{x} = \int_{T_2} \nabla\phi_i(\mathrm{x}) \cdot \nabla\phi_j(\mathrm{x})\, d\mathrm{x} \tag{4}$$

Finite Element Basis Compression Lemma

Lemma

If two mesh cells T_1 and T_2 are translations of each other, FEM operator assembly satisfies

$$\int_{T_1} \nabla\phi_i(\mathrm{x}) \cdot \nabla\phi_j(\mathrm{x})\, d\mathrm{x} = \int_{T_2} \nabla\phi_i(\mathrm{x}) \cdot \nabla\phi_j(\mathrm{x})\, d\mathrm{x} \tag{4}$$

Proof.

As T_1 and T_2 are translations of each other, we have $J_{T_1} \equiv J_{T_2}$ and

$$\begin{aligned}
\int_{T_1} \nabla\phi_i(\mathrm{x}) \cdot \nabla\phi_j(\mathrm{x})\, d\mathrm{x} &= \int_{\hat{T}} J_{T_1}^{-T}(\hat{\mathrm{x}}) \nabla\hat{\phi}_i(\hat{\mathrm{x}}) \cdot J_{T_1}^{-T}(\hat{\mathrm{x}}) \nabla\hat{\phi}_j(\hat{\mathrm{x}}) \det(J_{T_1}(\hat{\mathrm{x}}))\, d\hat{\mathrm{x}} \\
&= \int_{\hat{T}} J_{T_2}^{-T}(\hat{\mathrm{x}}) \nabla\hat{\phi}_i(\hat{\mathrm{x}}) \cdot J_{T_2}^{-T}(\hat{\mathrm{x}}) \nabla\hat{\phi}_j(\hat{\mathrm{x}}) \det(J_{T_2}(\hat{\mathrm{x}}))\, d\hat{\mathrm{x}} \\
&= \int_{T_2} \nabla\phi_i(\mathrm{x}) \cdot \nabla\phi_j(\mathrm{x})\, d\mathrm{x}
\end{aligned}$$

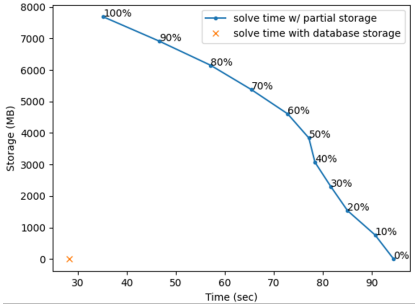
□

Finite Element Basis Compression

- Extension to other operators is a trivial consequence of the de Rham Complex.
- Cell similarity data may be stored with a mesh, costing 1 integer type per cell
- Index basis data by `basis(i,p,q,d) := basis_ref(lookup(i),p,q,d)`

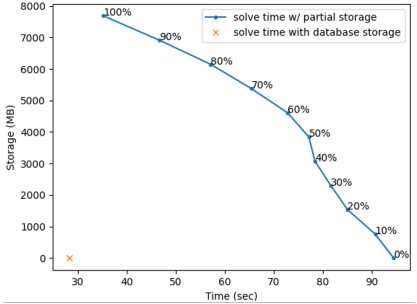
Finite Element Basis Compression

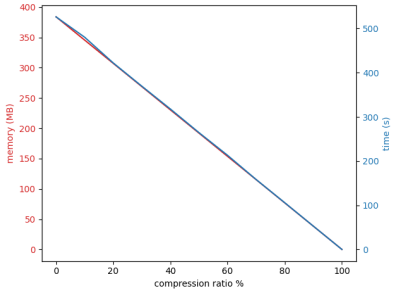
- Extension to other operators is a trivial consequence of the de Rham Complex.
- Cell similarity data may be stored with a mesh, costing 1 integer type per cell
- Index basis data by `basis(i,p,q,d) := basis_ref(lookup(i),p,q,d)`



Finite Element Basis Compression

- Extension to other operators is a trivial consequence of the de Rham Complex.
- Cell similarity data may be stored with a mesh, costing 1 integer type per cell
- Index basis data by `basis(i,p,q,d) := basis_ref(lookup(i),p,q,d)`



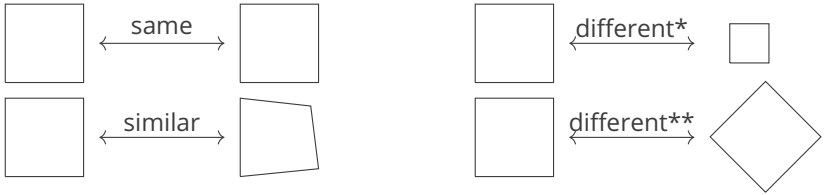


(left) Total time and basis storage costs for partial basis storage on (right) Basis evaluation timings and basis storage costs for compression of one million elements $p = 1, q = 2$ on a 3.00GHz Intel Xeon Gold 6248R.

Finite Element Basis Compression

Let $\varepsilon > 0$ be a compression tolerance. Compress functions evaluating

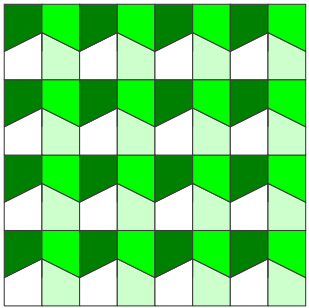
$$d(J_i, J_j) = \int_{\hat{T}} \frac{\|J_i - J_j\|_F}{\|J_i\|_F} \, d\hat{T}. \tag{5}$$



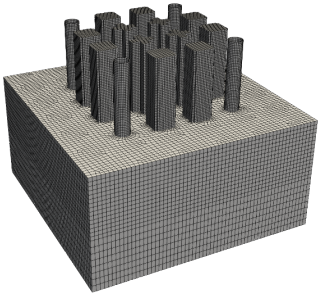
*scalings can be accounted for by a diagonal transform in the de Rham complex, but are beyond current scope

**rotations can be accounted for by a unitary transform in the de Rham complex, but are beyond current scope

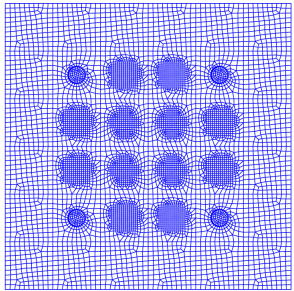
Finite Element Basis Compression



93.75% compression



99.9% compression

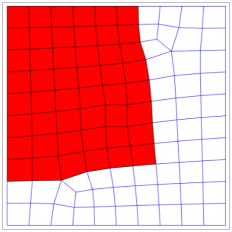


62.52% compression

*Compression is performed on a per-process basis.

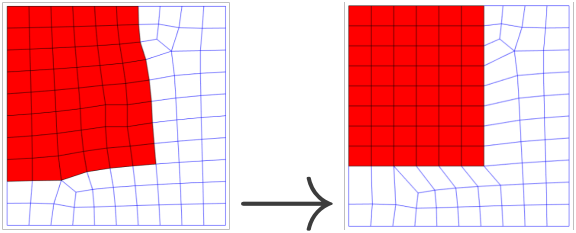
R-Adaptive Mesh Optimization

- R-adaptive mesh optimization repositions mesh nodes without touching topology.
- Goal: Enhance the compressible structure in the mesh by optimizing node locations.
- For simplicity, focus on quadrilateral meshes



R-Adaptive Mesh Optimization

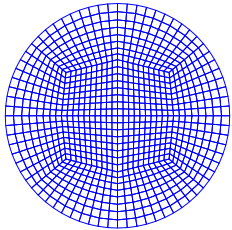
- R-adaptive mesh optimization repositions mesh nodes without touching topology.
- Goal: Enhance the compressible structure in the mesh by optimizing node locations.
- For simplicity, focus on quadrilateral meshes



Example of an r-adaptive mesh modification to enhance basis compression. (left) starting mesh (right) adapted mesh with structured region

Mesh Notation

- $\mathcal{T}_h$ mesh
- N cells, N_p nodes
- $\hat{p}_i = (x_i, y_i)$, $i = 1, \ldots, N_p$ node coordinates
- $\hat{p}_{i,j} = (x_{i,j}, y_{i,j})$, $i = 1, \ldots, N$, $j = 1, 2, 3, 4$ is the jth node of cell i
- $J_{T_i}(\hat{\mathrm{x}})$, $i = 1, \ldots, N$ Jacobian of cell i



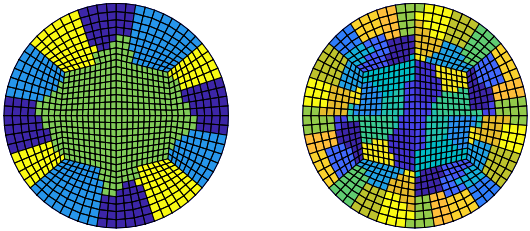
- Split T_i into two triangles using nodes 1,2,4 and nodes 3,4,2
- Define $K_i(\mathrm{p})$, $i = 1, \ldots, N$ as the combination of the Jacobians from the subtriangles

$$K_i(\mathrm{p}) = [x_{i,2} - x_{i,1}, x_{i,4} - x_{i,1}, x_{i,4} - x_{i,3}, x_{i,2} - x_{i,3}, y_{i,2} - y_{i,1}, y_{i,4} - y_{i,1}, y_{i,4} - y_{i,3}, y_{i,2} - y_{i,3}] \tag{6}$$

Mesh Structure Detection

What defines an “ideal” or “target” shape?

- Perform structure detection on 8-dimensional $K_i(\mathrm{p})$ dataset
- Utilize k-means algorithm and related variants
- Set parameters to minimize cluster variance and “overlap”
- Cluster center for cell i becomes target μ_i



The circle mesh clustered using k-means with 4 and 20 clusters based on cell shapes.

Optimization Objective

We want to reduce

$$\|\vec{K}(\mathrm{p}) - \vec{\mu}\|_{\ell_0}^2.$$

Optimization Objective

We want to reduce

$$\|\vec{K}(\mathrm{p}) - \vec{\mu}\|_{\ell_0}^2.$$

Therefore, we consider reducing

$$\|\vec{K}(\mathrm{p}) - \vec{\mu}\|_{\ell_1}^2. \tag{7}$$

Optimization Objective

We want to reduce

$$\|\vec{K}(\mathrm{p}) - \vec{\mu}\|_{\ell_0}^2.$$

Therefore, we consider reducing

$$\|\vec{K}(\mathrm{p}) - \vec{\mu}\|_{\ell_1}^2. \tag{7}$$

Replacing with elastic variables u, v, adding ℓ_2 regularization, we obtain

$$\frac{a}{2}\|(u, v)\|_{\ell_2}^2 + \sum_{i=1}^{N}\sum_{j=1}^{8} u_{i,j} + v_{i,j} \tag{8}$$

subject to

$$K_i(\mathrm{p}) - \mu_i = u_i - v_i, \quad i = 1, \ldots, N, \tag{9}$$

$$u_i \geq 0, \quad i = 1, \ldots, N, \tag{10}$$

$$v_i \geq 0, \quad i = 1, \ldots, N. \tag{11}$$

Volume and Feature Preservation Constraints

To protect mesh quality, bound cell volumes

$$v_{i,min} \leq v_i(\mathrm{p}) \leq v_{i,max}, \quad i = 1, \ldots, N,$$

where

$$v_i(\mathrm{p}) = \frac{1}{2} \sum_{j=1}^{4} x_{i,j+1} y_{i,j} - x_{i,j} y_{i,j+1}, \quad i = 1, \ldots, N$$

Volume and Feature Preservation Constraints

To protect mesh quality, bound cell volumes

$$v_{i,min} \leq v_i(\mathrm{p}) \leq v_{i,max}, \quad i = 1, \ldots, N,$$

where

$$v_i(\mathrm{p}) = \frac{1}{2} \sum_{j=1}^{4} x_{i,j+1} y_{i,j} - x_{i,j} y_{i,j+1}, \quad i = 1, \ldots, N$$

Replace cell volumes by slack variables

$$v_i(\mathrm{p}) - s_i = 0, \quad i = 1, \ldots, N \tag{12}$$

$$v_{i,min} \leq s_i \leq v_{i,max}, \quad i = 1, \ldots, N \tag{13}$$

Volume and Feature Preservation Constraints

To protect mesh quality, bound cell volumes

$$v_{i,min} \leq v_i(\mathrm{p}) \leq v_{i,max}, \quad i = 1, \ldots, N,$$

where

$$v_i(\mathrm{p}) = \frac{1}{2} \sum_{j=1}^{4} x_{i,j+1} y_{i,j} - x_{i,j} y_{i,j+1}, \quad i = 1, \ldots, N$$

Replace cell volumes by slack variables

$$v_i(\mathrm{p}) - s_i = 0, \quad i = 1, \ldots, N \tag{12}$$

$$v_{i,min} \leq s_i \leq v_{i,max}, \quad i = 1, \ldots, N \tag{13}$$

Constrain all nodes on boundaries and features to preserve.

$$p_{k_i} = \hat{p}_{k_i}, \quad i = 1, \ldots, n_k \tag{14}$$

Problem Formulation

Seek (u, v, p, s) satisfying

$$\min_{u,v,p,s} \frac{a}{2}\|(u,v)\|_{\ell_2}^2 + \sum_{i=1}^{N}\sum_{j=1}^{8} u_{i,j} + v_{i,j}, \tag{15}$$

subject to

$$K_i(\mathrm{p}) - \mu_i = u_i - v_i, \quad i = 1, \ldots, N, \tag{16}$$
$$v_i(\mathrm{p}) - s_i = 0, \quad i = 1, \ldots, N, \tag{17}$$
$$u_i \geq 0, \quad i = 1, \ldots, N, \tag{18}$$
$$v_i \geq 0, \quad i = 1, \ldots, N, \tag{19}$$
$$v_{i,min} \leq s_i \leq v_{i,max}, \quad i = 1, \ldots, N. \tag{20}$$

Problem Formulation

Seek (u, v, p, s) satisfying

$$\min_{u,v,p,s} \frac{a}{2}\|(u,v)\|_{\ell_2}^2 + \sum_{i=1}^{N}\sum_{j=1}^{8} u_{i,j} + v_{i,j}, \tag{15}$$

subject to

$$K_i(\mathrm{p}) - \mu_i = u_i - v_i, \quad i = 1, \ldots, N, \tag{16}$$
$$v_i(\mathrm{p}) - s_i = 0, \quad i = 1, \ldots, N, \tag{17}$$
$$u_i \geq 0, \quad i = 1, \ldots, N, \tag{18}$$
$$v_i \geq 0, \quad i = 1, \ldots, N, \tag{19}$$
$$v_{i,min} \leq s_i \leq v_{i,max}, \quad i = 1, \ldots, N. \tag{20}$$

Lemma

The starting mesh configuration lies in the feasible region, so it is nonempty!

Optimization Approach

- Utilize the **Augmented Lagrangian Sequential Quadratic Programming (ALESQP)** method (Antil, Kouri, Ridzal, 2023).
- $\mathcal{L}(x, \lambda) = f(x) + \lambda^T c(x)$
- Penalize inequality constraints, use Lagrange multipliers for equality constraints
- Requires augmented Lagrangian Hessian solves and augmented system solves

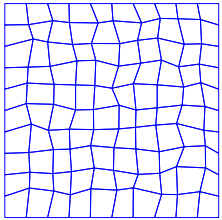
$$\begin{bmatrix} I & \nabla c(x)^T \\ \nabla c(x) & 0 \end{bmatrix} \tag{21}$$

General parameters:

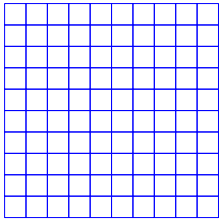
- Volume shrinkage/growth capped at 40%
- Optimization tolerance 1e-7
- L2 regularization constant $a = 1$

Example 1 (Perturbed uniform mesh)

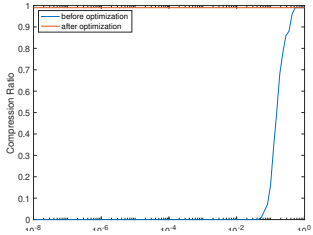
1 cluster, 1 optimization



Before



After

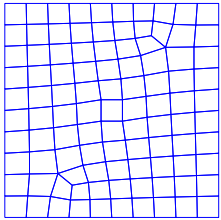


Compression

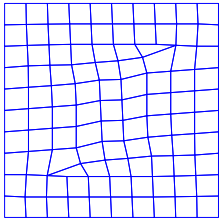
Size	AL Its	SQP Its	CG Its	$\|c(x)\|$	$\|\nabla L(x,\lambda)\|$	feas
10×10	13	34	147	4.32e-16	3.99e-08	4.11e-10
20×20	16	37	221	9.10e-16	2.99e-08	1.28e-09
40×40	500	54	447	1.95e-15	1.81e-06	6.06e-11

Example 2 (99 cell mesh)

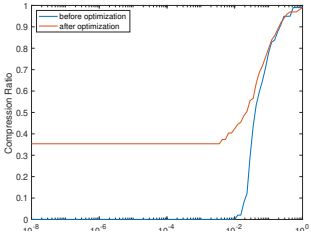
4 clusters, 2 optimizations



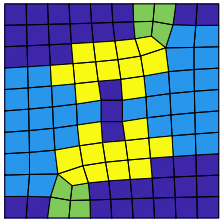
Before



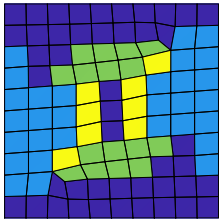
After



Compression



Clusters (first pass)

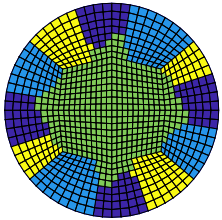


Clusters (second pass)

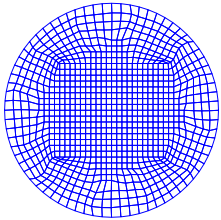
$\|c(x)\|$ =9.05e-16
$\|\nabla L(x,\lambda)\|$ =6.94e-7
feas=3.8e-9

Example 3 (Circle Mesh)

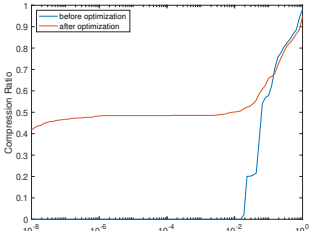
4 clusters, 1 optimization



Before



After

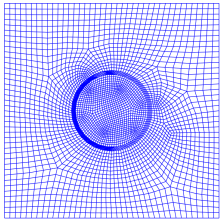


Compression

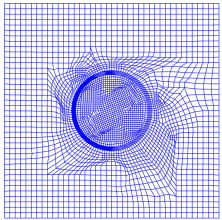
$\|c(x)\|$ =9.05e-16
$\|\nabla L(x,\lambda)\|$ =6.94e-7
feas=3.8e-9

Example 4 (Electromagnetics Straw)

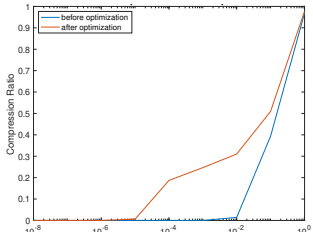
10 clusters, 2 optimizations, preserve annular “straw”



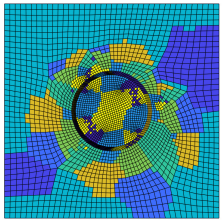
Before



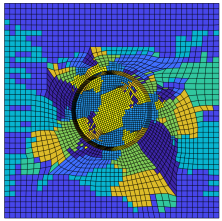
After



Compression



Clusters (first pass)



Clusters (second pass)

$\|c(x)\|$ =1.43e-10
$\|\nabla \mathcal{L}(x,\lambda)\|$ =1.74e-1
feas=3.71e-3

Conclusions

- Finite element basis compression allows for significant speedups
- Mesh optimization allows for even more speedups
- Approach is still not fully “automatic”
- Future studies on scalings, rotations, and discretization error

Conclusions

- Finite element basis compression allows for significant speedups
- Mesh optimization allows for even more speedups
- Approach is still not fully “automatic”
- Future studies on scalings, rotations, and discretization error

Thank you!

Bonus Slide

Why are the Jacobians equal for two cells which are translations?
Derivatives of constants are zero.

$$\Phi_1(\hat{x}) = \begin{bmatrix} a_1 + b_1\hat{x} + c_1\hat{y} + d_1\hat{x}\hat{y} \\ a_2 + b_2\hat{x} + c_2\hat{y} + d_2\hat{x}\hat{y} \end{bmatrix}$$

$$\Phi_2(\hat{x}) = \begin{bmatrix} u + a_1 + b_1\hat{x} + c_1\hat{y} + d_1\hat{x}\hat{y} \\ v + a_2 + b_2\hat{x} + c_2\hat{y} + d_2\hat{x}\hat{y} \end{bmatrix}$$

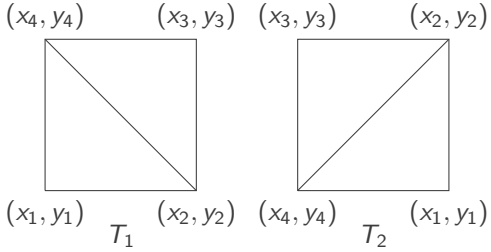
$$\nabla\Phi_1 = \nabla\Phi_2$$

For quadrilaterals, it's easy to see

$$\Phi(\hat{x}) = \begin{bmatrix} x_1 + (x_2 - x_1)\hat{x} + (x_4 - x_1)\hat{y} + (x_1 - x_2 + x_3 - x_4)\hat{x}\hat{y} \\ y_1 + (y_2 - y_1)\hat{x} + (y_4 - y_1)\hat{y} + (y_1 - y_2 + y_3 - y_4)\hat{x}\hat{y} \end{bmatrix}$$

Bonus Slide

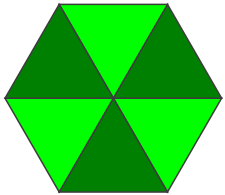
What about local cell orientations?
Local cell orientations are important. To keep things simple, we use a “lower-left” cell orientation scheme; however, it should only impact compression by a factor of 4.



Squares with differing local node orderings

Bonus Slide

Why not use triangles?
Pros: triangles have constant Jacobians, require less data.
Cons: triangle meshes tend to have more variance in vertex degree, are less amenable to synergistic techniques like sum factorization, and a triangular plane tiling does not place similar shapes near each other.



Bonus Slide

Visualization of matches