Local Fourier Analysis of Domain Decomposition and Multigrid Methods for High-Order Matrix-Free Finite Elements

Jeremy L Thompson

University of Colorado Boulder

jeremy@jeremylt.org

July 8, 2021

Funding

This work is supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering and early testbed platforms, in support of the nation's exascale computing imperative.

Overview

Big Picture

- High-order matrix-free representations of PDEs are better suited to modern hardware than sparse matrices
- High-order matrix-free representations require preconditioned iterative solvers
- Local Fourier Analysis (LFA) provides sharp convergence estimates for these preconditioners
- We develop LFA of p-multigrid and Balancing Domain Decomposition by Constraints (BDDC) on high-order element subdomains
- We investigate LFA of p-multigrid with a BDDC smoother

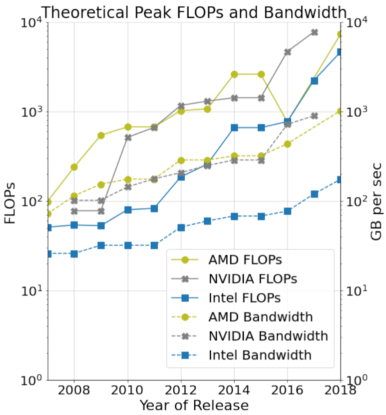
Reproducibility

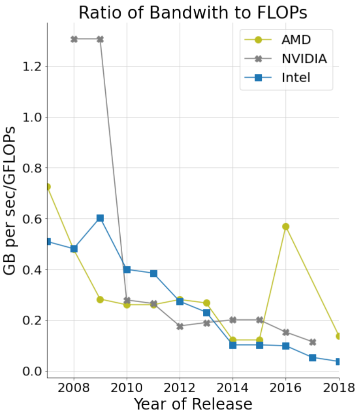
Transparency and reproducibility are the lifeblood of scientific advancement

All software and data used in this dissertation is open source:

- https://www.github.com/jeremylt/LFAToolkit.jl
- https://www.github.com/CEED/libCEED
- https://www.mcs.anl.gov/petsc
- https://github.com/jeremylt/dissertation

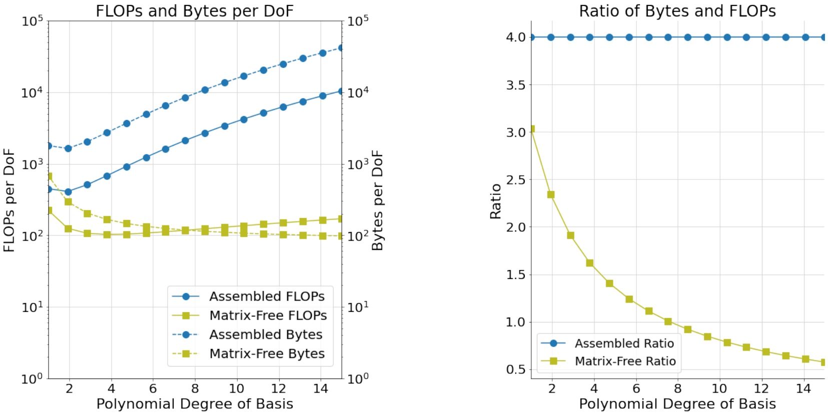
Modern Hardware





Modern hardware has lower memory bandwidth than FLOPs [6]

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 Representation

Weak form for an arbitrary second order PDE [2]:

$$\begin{gathered}\text{find } u \in V \text{ such that for all } v \in V \\ \langle v, u \rangle = \int_\Omega v \cdot f_0 \left(u, \nabla u \right) + \nabla v : f_1 \left(u, \nabla u \right) = 0\end{gathered} \tag{1}$$

where

- $\cdot$ - contraction over fields
- $:$ - contraction over fields and spatial dimensions

Note: pointwise functions f_0 and f_1 don't depend upon the discretization

Matrix-Free Representation

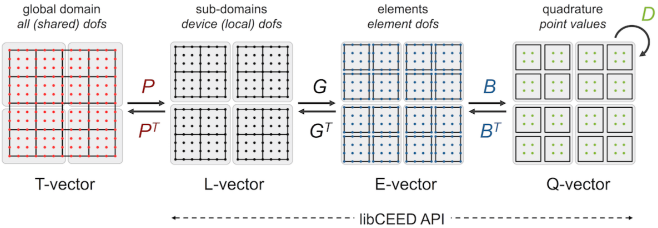
Galerkin form for an arbitrary second order PDE:

$$\sum_e \mathcal{E}^T \left[\left(\mathrm{N}^e\right)^T \mathrm{W}^e \Lambda \left(f_0 \left(u^e, \nabla u^e \right) \right) + \sum_{i=0}^{d-1} \left(\mathrm{D}_i^e\right)^T \mathrm{W}^e \Lambda \left(f_1 \left(u^e, \nabla u^e \right) \right) \right] = 0 \tag{2}$$

- $\mathcal{E}$ - element assembly/restriction operator
- N^e - interpolation to quadrature points
- D_i^e - derivatives at quadrature points
- W^e - quadrature weights
- Λ - pointwise multiplication at quadrature points
- $u^e = \mathrm{N}^e \mathcal{E}^e u$ and $\nabla u^e = \left\{ \mathrm{D}_i^e \mathcal{E}^e u \right\}_{i=0}^{d-1}$

libCEED Representation

$$A = P^T G^T B^T D B G P$$



[1]

- P - parallel element assembly operator
- G - local element assembly operator
- B - basis action operator
- D - weak form and geometry at quadrature points

Preconditioning Required

- Matrix-free representations require iterative solvers
- Iterative solvers are sensitive to conditioning of the operator (among other factors)
- High-order operators are ill-conditioned
- Preconditioners are required for good convergence
- LFA helps us tune these preconditioners

LFA Background

Consider a scalar Toeplitz operator L_h on the infinite 1D grid G_h

$$L_h \mathrel{\hat{=}} [s_\kappa]_h \, (\kappa \in V)$$
$$L_h w_h \left(x \right) = \sum_{\kappa \in V} s_\kappa w_h \left(x + \kappa h \right) \tag{3}$$

where

- $V \subset \mathbb{Z}$ is an index set
- $s_\kappa \in \mathbb{R}$ are constant coefficients
- $w_h \left(x \right)$ is an l^2 function on G_h

LFA Background

Our function can be diagonalized by the standard Fourier modes:

If for all grid functions $\varphi(\theta, x)$

$$L_h \varphi(\theta, x) = \tilde{L}_h(\theta) \varphi(\theta, x) \tag{4}$$

then $\tilde{L}_h(\theta) = \sum_{\kappa \in V} s_\kappa e^{\imath \theta \kappa}$ is the **symbol** of L_h

LFA Background

For a $q \times q$ system of equations, the matrix symbol is given by:

$$\mathrm{L}_h = \begin{bmatrix} L_h^{1,1} & \dots & L_h^{1,q} \\ \vdots & \vdots & \vdots \\ L_h^{q,1} & \dots & L_h^{q,q} \end{bmatrix} \qquad \Rightarrow \qquad \tilde{\mathrm{L}}_h = \begin{bmatrix} \tilde{L}_h^{1,1} & \dots & \tilde{L}_h^{1,q} \\ \vdots & \vdots & \vdots \\ \tilde{L}_h^{q,1} & \dots & \tilde{L}_h^{q,q} \end{bmatrix} \tag{5}$$

LFA of High-Order FEM

For a scalar PDE operator on a single 1D finite element

$$\tilde{A}(\theta) = Q^T \left(A^e \odot \left[e^{\imath \left(x_j - x_i \right) \theta / h} \right] \right) Q \tag{6}$$

where

$$A^e = B^T D B, \qquad Q = \begin{bmatrix} I \\ e_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \end{bmatrix} \tag{7}$$

LFA of High-Order FEM

Natural extension to multiple components and higher dimensions:

$$\tilde{A}\left(\boldsymbol{\theta}\right) = Q^T \left(A^e \odot \left[e^{\imath \left(x_j - x_i \right) \cdot \boldsymbol{\theta} / h} \right] \right) Q \tag{8}$$

Multiple Components:

$$Q_n = I_n \otimes Q \tag{9}$$

Multiple Dimensions:

$$Q_{nd} = Q \otimes Q \otimes \cdots \otimes Q \tag{10}$$

Example: Scalar Poisson

$$\int \nabla v \nabla u = \int fv \tag{11}$$

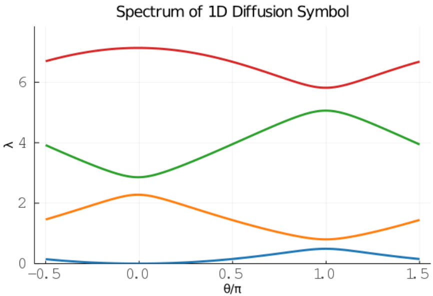
- B - given by tensor H^1 Lagrange basis
- D - given by quadrature weights and product

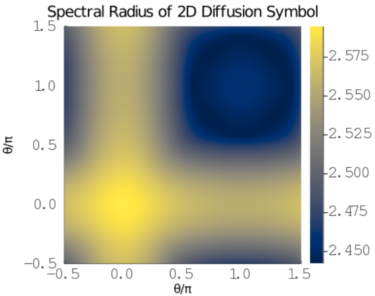
```
# mesh
dim = 1
mesh = Mesh1D(1.0)

# basis
p = 3
ncomp = 1
basis = TensorH1LagrangeBasis(p+1, p+1, ncomp, dim)

# weak form
function diffusionweakform(du::Array{Float64}, w::Array{Float64})
    return dv = du*w[1]
end
```

Example: Scalar Poisson





Scalar Poisson problem on quartic elements

Goal: decrease spectral radius with preconditioners

LFA of High-Order Smoothers

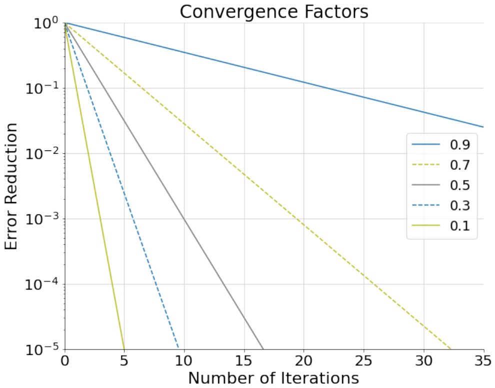
Error propagation operator for smoothers given by

$$\mathsf{S} = \mathsf{I} - \mathsf{M}^{-1} A \tag{12}$$

with a symbol given by

$$\tilde{\mathsf{S}}\left(\boldsymbol{\theta}, \omega\right) = \mathsf{I} - \tilde{\mathsf{M}}^{-1}\left(\boldsymbol{\theta}, \omega\right) \tilde{A}\left(\boldsymbol{\theta}\right) \tag{13}$$

Error Symbol



Iterations required to target error tolerances

Maximum spectral radius of error propagation operator determines convergence rate

Jacobi Smoothing

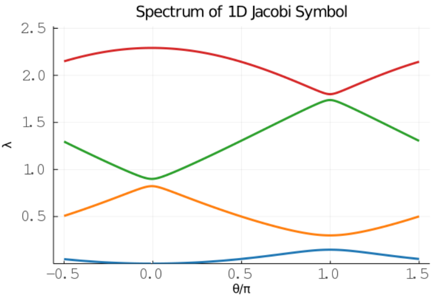
Jacobi smoothing given by

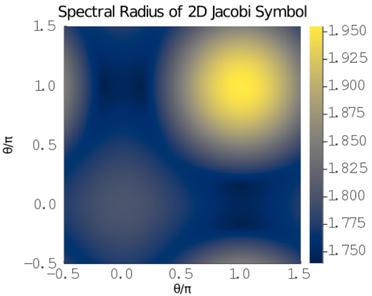
$$\mathrm{M}^{-1} = \omega \ \operatorname{diag}\left(A\right)^{-1} \tag{14}$$

with an error symbol given by

$$\tilde{\mathrm{S}}\left(\boldsymbol{\theta}, \omega\right) = \mathrm{I} - \omega \ \left(\mathrm{Q}^{T} \operatorname{diag}\left(A^{e}\right) \mathrm{Q}\right)^{-1} \tilde{A}\left(\boldsymbol{\theta}\right) \tag{15}$$

Example: Jacobi Smoothing





Jacobi smoothing with $\omega = 1.0$ on quartic elements

Moderate reduction in spectral radius of symbol

Chebyshev Smoother

Error in *kth* order Chebyshev smoothing is given by

$$
\begin{aligned}
\mathsf{E}_0 &= \mathsf{I} \\
\mathsf{E}_1 &= \mathsf{I} - \frac{1}{\alpha} \left(\operatorname{diag} \mathsf{A} \right)^{-1} \mathsf{A} \\
\mathsf{E}_k &= \left(\left(\operatorname{diag} \mathsf{A} \right)^{-1} \mathsf{A} \mathsf{E}_{k-1} - \alpha \mathsf{E}_{k-1} - \beta_{k-2} \mathsf{E}_{k-2} \right) / \gamma_{k-1}
\end{aligned}
\tag{16}
$$

for an operator with a spectrum on the interval $[\alpha - c, \alpha + c]$ where

$$
\begin{aligned}
\beta_0 &= -\frac{c^2}{2\alpha} & \gamma_0 &= -\alpha \\
\beta_k &= \frac{c}{2} \frac{T_k(\eta)}{T_{k+1}(\eta)} = \left(\frac{c}{2} \right)^2 \frac{1}{\gamma_k} & \gamma_k &= \frac{c}{2} \frac{T_{k+1}(\eta)}{T_k(\eta)} = -\left(\alpha + \beta_{k-1} \right).
\end{aligned}
\tag{17}
$$

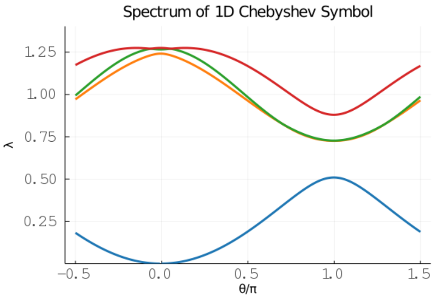
Chebyshev Smoother

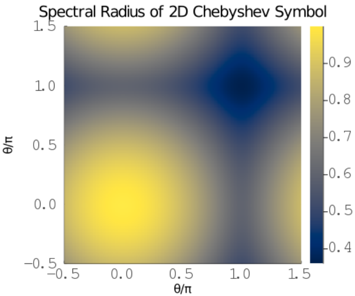
The error symbol of kth order Chebyshev smoother is given by

$$\begin{aligned} \tilde{\mathsf{E}}_0(\boldsymbol{\theta}) &= \mathsf{I} \\ \tilde{\mathsf{E}}_1(\boldsymbol{\theta}) &= \mathsf{I} - \frac{1}{\alpha}\tilde{\mathsf{A}}_J\tilde{\mathsf{A}}(\boldsymbol{\theta}) \\ \tilde{\mathsf{E}}_k(\boldsymbol{\theta}) &= \left(\tilde{\mathsf{A}}_J\tilde{\mathsf{A}}(\boldsymbol{\theta})\tilde{\mathsf{E}}_{k-1}(\boldsymbol{\theta}) - \alpha\tilde{\mathsf{E}}_{k-1}(\boldsymbol{\theta}) - \beta_{k-2}\tilde{\mathsf{E}}_{k-2}(\boldsymbol{\theta})\right)/\gamma_{k-1} \end{aligned} \tag{18}$$

with $\tilde{A}_J$ being the symbol of the Jacobi preconditioner

Example: Chebyshev Smoothing





Third order Chebyshev smoothing quartic elements

Improved reduction in spectral radius of symbol

Two-Grid Multigrid Error

Multigrid methods target the low frequency error

$$E_{2MG} = S_f \left(I - P_{ctof} A_c^{-1} R_{ftoc} A_f \right) S_f \tag{19}$$

- A_f - fine grid operator
- A_c^{-1} - coarse grid solve (low frequency error)
- S_f - fine grid smoother (high frequency error)
- P_{ctof} - coarse to fine grid prolongation operator
- R_{ftoc} - fine to coarse grid restriction operator

Grid transfer operators and coarse representation differentiate *h*-multigrid and *p*-multigrid

Two-Grid Multigrid Error

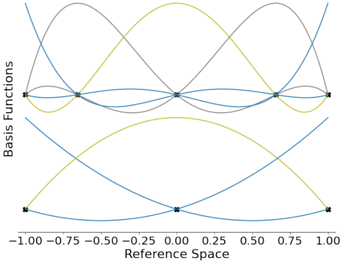
The definition of the symbol follows naturally:

$$\tilde{\mathsf{E}}_{2MG}\left(\boldsymbol{\theta}\right) = \tilde{\mathsf{S}}_f\left(\boldsymbol{\theta}, \omega\right)\left(\mathsf{I} - \tilde{\mathsf{P}}_{\mathrm{ctof}}\left(\boldsymbol{\theta}\right)\tilde{\mathsf{A}}_c^{-1}\left(\boldsymbol{\theta}\right)\tilde{\mathsf{R}}_{\mathrm{ftoc}}\left(\boldsymbol{\theta}\right)\tilde{\mathsf{A}}_f\left(\boldsymbol{\theta}\right)\right)\tilde{\mathsf{S}}_f\left(\boldsymbol{\theta}, \omega\right) \tag{20}$$

- $\tilde{\mathsf{A}}_f$ - fine grid symbol
- $\tilde{\mathsf{A}}_c^{-1}$ - coarse grid symbol inverse (low frequency error)
- $\tilde{\mathsf{S}}_f$ - fine grid smoother symbol (high frequency error)
- $\tilde{\mathsf{P}}_{\mathrm{ctof}}$ - coarse to fine grid prolongation symbol
- $\tilde{\mathsf{R}}_{\mathrm{ftoc}}$ - fine to coarse grid restriction symbol

P-Multigrid Transfer Operators

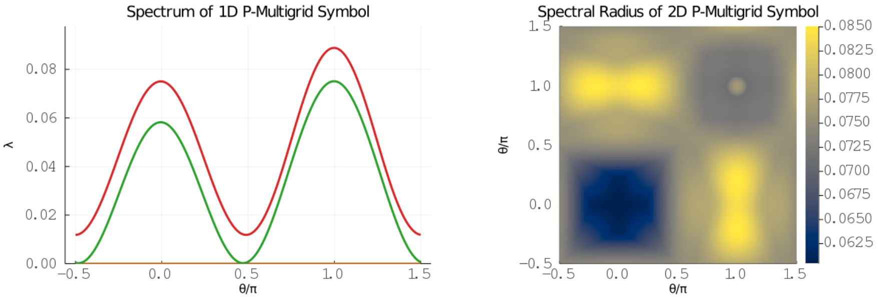
p-multigrid prolongation can be represented as an interpolation from the coarse to fine grid



$$\begin{aligned} \mathrm{P}_{\text{ctof}} &= \mathrm{P}_f^T \mathrm{G}_f^T \mathrm{P}^e \mathrm{G}_c \mathrm{P}_c \\ \mathrm{P}^e &= I\mathrm{D}_{\text{scale}} \mathrm{B}_{\text{ctof}} \end{aligned} \tag{21}$$

D scales for node multiplicity

Example: *P*-Multigrid



p-multigrid with third order Chebyshev on quartic to quadratic elements

Significant reduction in spectral radius

Validation: P-Multigrid

p_{fine} to p_{coarse}	LFA	libCEED
$p = 2$ to $p = 1$	0.312	0.301
$p = 4$ to $p = 2$	1.436	1.402
$p = 4$ to $p = 1$	1.436	1.401
$p = 8$ to $p = 4$	1.989	1.885
$p = 8$ to $p = 2$	1.989	1.874
$p = 8$ to $p = 1$	1.989	1.875

LFA and experimental two-grid convergence factors with Jacobi smoothing for 3D Laplacian with $\omega = 1.0$

3D manufactured solution on the domain $[-3, 3]^3$ with Dirichlet boundaries:

$$f(x, y, z) = xyz \sin(\pi x) \sin(\pi (1.23 + 0.5y)) \sin(\pi (2.34 + 0.25z)) \quad (22)$$

Validation: P-Multigrid

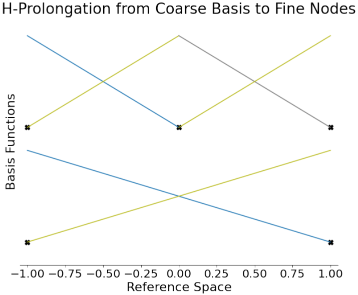
p_{fine} to p_{coarse}		$k = 3$			$k = 4$	
	LFA	libCEED	its	LFA	libCEED	its
$p = 2$ to $p = 1$	0.076	0.058	9	0.041	0.033	7
$p = 4$ to $p = 2$	0.111	0.097	10	0.062	0.050	8
$p = 4$ to $p = 1$	0.416	0.398	25	0.295	0.276	18
$p = 8$ to $p = 4$	0.197	0.195	15	0.121	0.110	11
$p = 8$ to $p = 2$	0.611	0.603	46	0.506	0.469	31
$p = 8$ to $p = 1$	0.871	0.861	154	0.827	0.814	112

LFA and experimental two-grid convergence factors with Chebyshev smoothing for 3D Laplacian

Iterations required to reach 10x reduction in error grows with rapid coarsening

H-Multigrid Transfer Operators

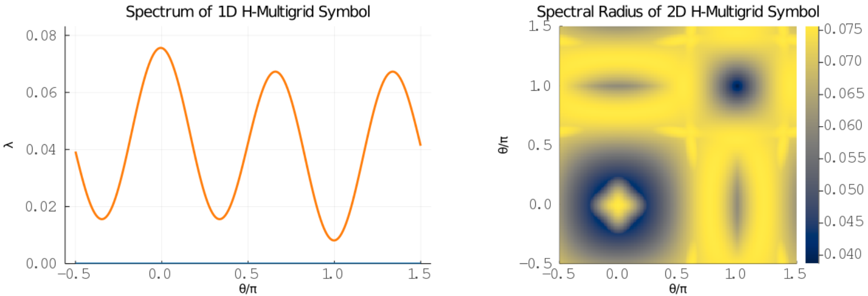
h-multigrid prolongation can be represented as an interpolation from the coarse grid to fine grid macro-elements



$$\begin{aligned} P_{\text{ctof}} &= P_f^T G_f^T P^e G_c P_c \\ P^e &= I D_{\text{scale}} B_{\text{ctof}} \end{aligned} \tag{23}$$

D scales for node multiplicity

Example: *H*-Multigrid



h-multigrid with third order Chebyshev on linear elements

Significant reduction in spectral radius

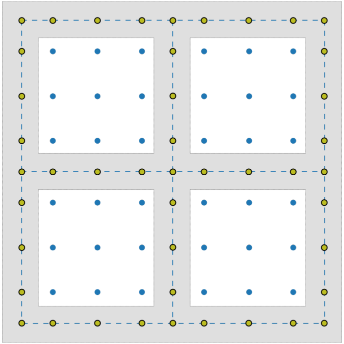
Validation: H-Multigrid

p, d	$\nu = (0,1)$		$\nu = (1,1)$		$\nu = (2,2)$	
	ρ	ω	ρ	ω	ρ	ω
$p = 2, d = 1$	0.821	1.000	0.821	1.000	1.279	1.000
$p = 2, d = 1$	0.526	0.838	0.495	0.838	0.302	0.838
$p = 2, d = 1$	0.291	0.709	0.249	0.709	0.064	0.709
$p = 3, d = 1$	0.491	0.650	0.337	0.650	0.131	0.650
$p = 4, d = 1$	0.608	0.640	0.559	0.640	0.331	0.640
$p = 2, d = 2$	0.452	1.000	0.288	1.000	0.091	1.000

Two-grid convergence factor and Jacobi smoothing parameter for high-order h-multigrid

Results agree with previous work [4]

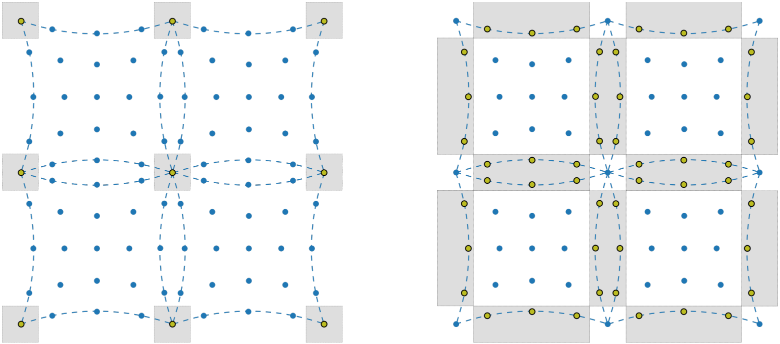
BDDC Overview



High-order single element subdomains

BDDC - non-overlapping domain decomposition method by Dohrmann [3]

Broken Subdomains



Non-overlapping domain decomposition of high-order mesh

Global problem only "partially subassembled" on primal (Π) vertices

Remaining interface nodes replicated across broken interface

Subassembled Problem

$$\hat{A}^{-1} = \sum_{e=1}^{N} R_i^{e,T} \hat{A}^{e,-1} R_i^e, \qquad \hat{A}^e = \begin{bmatrix} A_{r,r}^e & \hat{A}_{\Pi,r}^{e,T} \\ \hat{A}_{\Pi,r}^e & \hat{A}_{\Pi,\Pi}^e \end{bmatrix} \tag{24}$$

Partially subassembled problem is easier to invert

Injection operator R_i maps from global space to broken space and provides different BDDC variants

Injection Operators

$$\mathrm{R}_1 = \mathrm{diag}\left(\left[\frac{1}{|\mathcal{N}(x_i)|}\right]\right) \tag{25}$$

where $|\mathcal{N}(x_i)|$ is node multiplicity across broken spaces

$$\begin{aligned} \mathrm{R}_2 &= \mathrm{R}_1 - \mathrm{J}^T \boldsymbol{\mathcal{H}}^T \\ \boldsymbol{\mathcal{H}}^e &= -\mathrm{A}_{\mathrm{I,I}}^{e,-1} \mathrm{A}_{\Gamma,\mathrm{I}}^{e,T} \end{aligned} \tag{26}$$

where $\boldsymbol{\mathcal{H}}$ is a harmonic extension, J a map over the interfaces

Lumped BDDC with R_1 cheaper to setup but poorer conditioning

Dirichlet BDDC with R_2 equivalent to Dirichlet FETI-DP [5]

Fast Diagonalization

For separable problems of the form

$$\mathsf{A} = a\mathsf{M} + b\mathsf{K} \tag{27}$$

Fast Diagonalization provides fast approximate solver

$$\mathsf{A}^{-1} = \mathsf{S}^T \left(a\mathsf{I} + b\Lambda \right)^{-1} \mathsf{S} \tag{28}$$

where

$$\mathsf{S}\mathsf{M}\mathsf{S}^T = \mathsf{I}, \;\; \mathsf{S}\mathsf{K}\mathsf{S}^T = \Lambda \tag{29}$$

Fast Diagonalization

- Tensor product bases have tensor product diagonalizations
- Convergence impact of approximate solver formulations is ongoing research
- Cheaper to compute Fast Diagonalization solver than invert assembled subdomain matrices
- Reusing diagonalization for both subdomain solvers mitigates expensive setup cost of Dirichlet BDDC

LFA of BDDC

$$\hat{\tilde{A}}^{-1} = \begin{bmatrix} I & -\tilde{A}_{r,r}^{-1} \hat{\tilde{A}}_{\Pi,r}^{T} \\ 0 & I \end{bmatrix} \begin{bmatrix} \tilde{A}_{r,r}^{-1} & 0 \\ 0 & \hat{\tilde{S}}_{\Pi}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -\tilde{A}_{\Pi,r} \hat{\tilde{A}}_{r,r}^{-1} & I \end{bmatrix} \tag{30}$$

$$\tilde{A}_{r,r}^{-1}(\boldsymbol{\theta}) = A_{r,r}^{-1} \odot \left[e^{\imath \left(x_j - x_i \right) \cdot \boldsymbol{\theta} / h} \right], \quad \hat{\tilde{A}}_{r,\Pi}(\boldsymbol{\theta}) = \left(\hat{A}_{r,\Pi} \odot \left[e^{\imath \left(x_j - x_i \right) \cdot \boldsymbol{\theta} / h} \right] \right) Q_{\Pi},$$
$$\hat{\tilde{S}}_{\Pi}^{-1}(\boldsymbol{\theta}) = \left(Q_{\Pi}^{T} \left(\hat{S}_{\Pi} \odot \left[e^{\imath \left(x_j - x_i \right) \cdot \boldsymbol{\theta} / h} \right] \right) Q_{\Pi} \right)^{-1} \tag{31}$$

Only primal modes are localized for subassembled operator symbol

Symbols of injection operators are relatively straightforward

Low-Order Validation

m	Lumped BDDC			Dirichlet BDDC		
	λ_{min}	λ_{max}	κ	λ_{min}	λ_{max}	κ
$m = 4$	1.000	4.444	4.444	1.000	2.351	2.351
$m = 8$	1.000	12.269	12.269	1.000	3.196	3.196
$m = 16$	1.000	31.179	31.179	1.000	4.188	4.188
$m = 32$	1.000	75.761	75.761	1.000	5.335	5.335

Condition numbers and maximal eigenvalues
for low-order macro-elements

Exactly reproduces original work on LFA of low-order subdomains

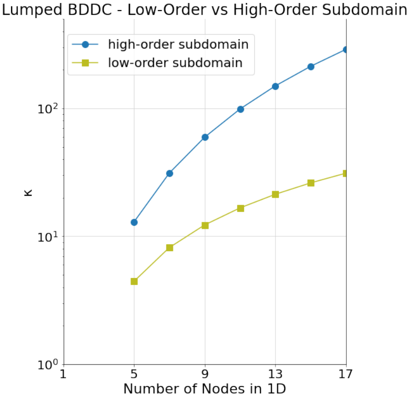
High-Order Experiments

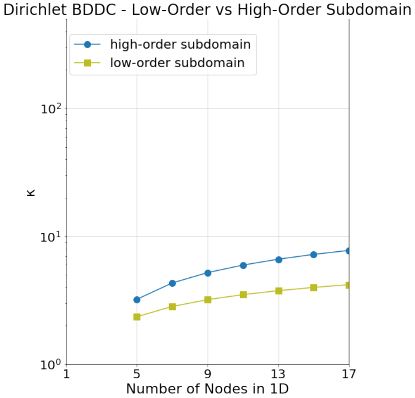
p	Lumped BDDC			Dirichlet BDDC		
	λ_{min}	λ_{max}	κ	λ_{min}	λ_{max}	κ
$p = 2$	1.000	2.800	2.800	1.000	2.042	2.042
$p = 4$	1.000	12.948	12.948	1.000	3.242	3.242
$p = 8$	1.000	59.563	59.563	1.000	5.197	5.197
$p = 16$	1.000	289.678	289.678	1.000	7.761	7.761

Condition numbers and maximal eigenvalues
for single high-order element subdomains

Single high-order element subdomains less well conditioned

Low vs High-Order BDDC

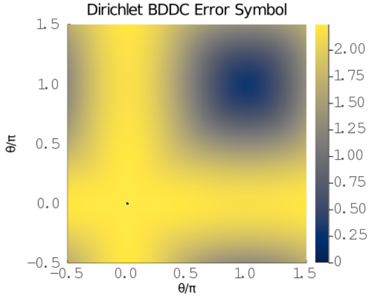




Low-order and high-order subdomain condition number

Dirichlet BDDC important for single high-order element subdomains

BDDC Smoother for P-Multigrid



Symbol of error operator for Dirichlet BDDC of 2D Laplacian for $p = 4$

Dirichlet BDDC smoother still has large spectral radius, so we introduce relaxation parameter

$$\tilde{\mathrm{E}}\left(\boldsymbol{\theta}, \omega\right) = \mathrm{I} - \omega \tilde{\mathrm{M}}_2^{-1} \tilde{A}\left(\boldsymbol{\theta}\right) \quad (32)$$

BDDC Smoother for P-Multigrid

p_{fine} to p_{coarse}	Dirichlet BDDC			Chebyshev	
	ρ	ω_{opt}	its	ρ	its
$p = 2$ to $p = 1$	0.121	0.66	11	0.075	9
$p = 4$ to $p = 2$	0.272	0.48	18	0.085	10
$p = 4$ to $p = 1$	0.281	0.47	19	0.219	16
$p = 8$ to $p = 4$	0.409	0.38	26	0.110	11
$p = 8$ to $p = 1$	0.462	0.32	30	0.795	101
$p = 16$ to $p = 8$	0.504	0.32	34	0.435	28
$p = 16$ to $p = 1$	0.597	0.23	45	0.959	551

Two-grid convergence factor for p-multigrid with BDDC vs cubic Chebyshev smoothing for 2D Laplacian

Weighted Dirichlet BDDC smoother better supports rapid coarsening

BDDC Smoother for P-Multigrid

p_{fine} to p_{coarse}	Dirichlet BDDC			Chebyshev	
	ρ	ω_{opt}	its	ρ	its
$p = 2$ to $p = 1$	0.121	0.66	11	0.252	17
$p = 4$ to $p = 2$	0.272	0.48	18	0.281	19
$p = 4$ to $p = 1$	0.281	0.47	19	0.424	27
$p = 8$ to $p = 4$	0.409	0.38	26	0.278	18
$p = 8$ to $p = 1$	0.462	0.32	30	0.873	170
$p = 16$ to $p = 8$	0.504	0.32	34	0.613	48
$p = 16$ to $p = 1$	0.597	0.23	45	0.975	910

Two-grid convergence factor for p-multigrid with BDDC vs quadratic Chebyshev smoothing for 2D Laplacian

Weighted Dirichlet BDDC smoother better supports rapid coarsening

Summary

- High-order matrix-free representations of PDEs are better suited to modern hardware than sparse matrices
- High-order matrix-free representations require preconditioned iterative solvers
- Local Fourier Analysis (LFA) provides sharp convergence estimates for these preconditioners
- We develop LFA of p-multigrid and Balancing Domain Decomposition by Constraints (BDDC) on high-order element subdomains
- Finally, we investigated LFA of p-multigrid with a BDDC smoother

Future Work

Local Fourier Analysis of

- *hp*-multigrid methods
- BDDC with inexact subdomain solvers
- mixed finite elements or modal bases
- BDDC with enriched primal spaces
- overlapping domain decomposition methods

Local Fourier Analysis of Domain Decomposition and Multigrid Methods for High-Order Matrix-Free Finite Elements

Jeremy L Thompson

University of Colorado Boulder

jeremy@jeremylt.org

July 8, 2021

Ahmad Abdelfattah, Valeria Barra, Natalie Beams, Jed Brown, Jean-Sylvain Camier, Veselin Dobrev, Yohann Dudouit, Leila Ghaffari, Tzanio Kolev, David Medina, Thilina Rathnayake, Jeremy L Thompson, and Stanimire Tomov.
libCEED User Manual, July 2021.

Jed Brown.
Efficient nonlinear solvers for nodal high-order finite elements in 3D.
Journal of Scientific Computing, 45(1-3):48–63, 2010.

Clark R Dohrmann.
A preconditioner for substructuring based on constrained energy minimization.
SIAM Journal on Scientific Computing, 25(1):246–258, 2003.

Yunhui He and Scott MacLachlan.
Two-level fourier analysis of multigrid for higher-order finite-element discretizations of the laplacian.
Numerical Linear Algebra with Applications, 27(3):e2285, 2020.

Jing Li and Olof B Widlund.
On the use of inexact subdomain solvers for BDDC algorithms.
Computer Methods in Applied Mechanics and Engineering, 196(8):1415–1428, 2007.

Karl Rupp.
CPU-GPU-MIC comparision charts, 2020.

Local Fourier Analysis of Domain Decomposition and Multigrid Methods for High-Order Matrix-Free Finite Elements

Jeremy L Thompson

University of Colorado Boulder

jeremy@jeremylt.org

July 8, 2021