BDDC Preconditioned P-Multigrid for High-Order Finite Elements

Jeremy L Thompson

University of Colorado, Boulder

jeremy@jeremylt.org

April 4, 2022

Funding

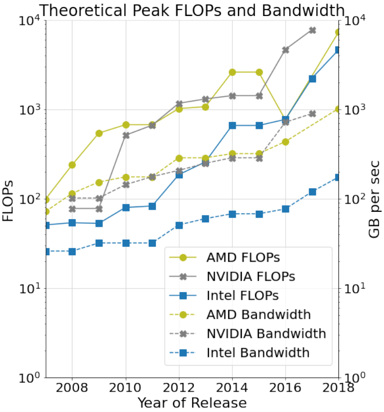
This work is supported by the Department of Energy, National Nuclear Security Administration, Predictive Science Academic Alliance Program (PSAAP) under Award Number DE-NA0003962.

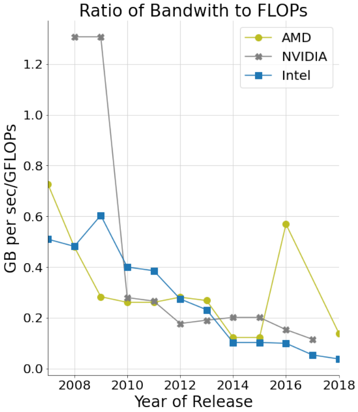
Overview

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 investigate LFA of Balancing Domain Decomposition by Constraints (BDDC) for high-order element subdomains
- We investigate LFA of p-multigrid with a BDDC smoother

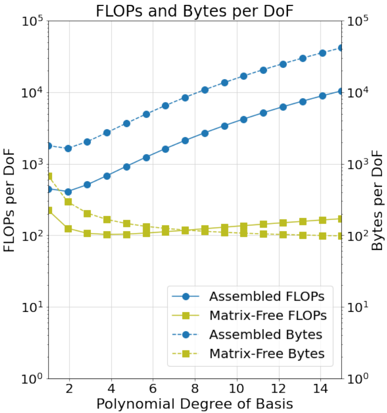
Modern Hardware

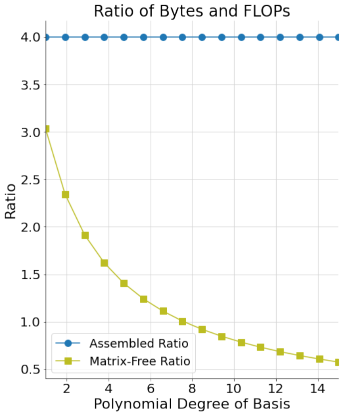




Modern hardware has lower memory bandwidth than FLOPs (https://github.com/karlrupp/cpu-gpu-mic-comparison)

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

For more details - see Rezgar Shakeri's talk Thursday, 13:05 in Session 10B

Matrix-Free Representation

Weak form for an arbitrary second order PDE:

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

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

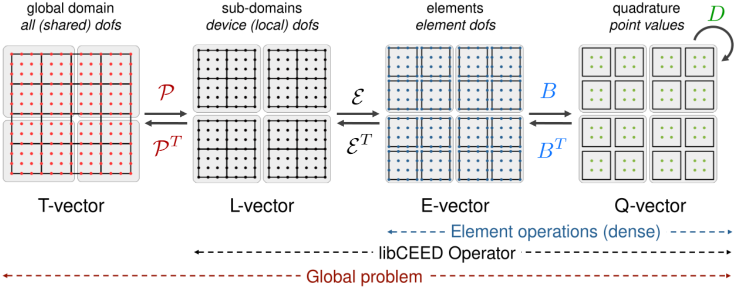
Galerkin form for an arbitrary second order PDE:

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

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

libCEED Representation

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



- $\mathcal{P}$ - parallel element assembly operator
- $\mathcal{E}$ - 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

$$\begin{aligned} L_h &\mathrel{\widehat{=}} [s_\kappa]_h \left(\kappa \in V\right) \\ L_h w_h\left(x\right) &= \sum_{\kappa \in V} s_\kappa w_h\left(x + \kappa h\right) \end{aligned} \tag{3}$$

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:

$$\mathsf{L}_h = \begin{bmatrix} L_h^{1,1} & \cdots & L_h^{1,q} \\ \vdots & \vdots & \vdots \\ L_h^{q,1} & \cdots & L_h^{q,q} \end{bmatrix} \quad \Rightarrow \quad \tilde{\mathsf{L}}_h = \begin{bmatrix} \tilde{L}_h^{1,1} & \cdots & \tilde{L}_h^{1,q} \\ \vdots & \vdots & \vdots \\ \tilde{L}_h^{q,1} & \cdots & \tilde{L}_h^{q,q} \end{bmatrix} \tag{5}$$

LFA of High-Order FEM

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

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

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:

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

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

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

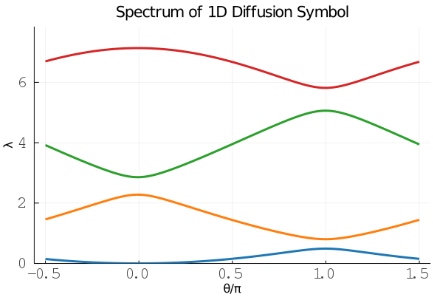
- B - given by tensor H^1 Lagrange basis
- D - given by quadrature weights and product

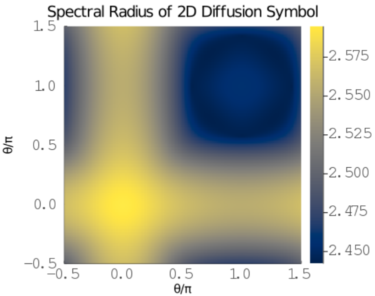
```
# mesh
dim = 1
mesh = Mesh1D(1.0)

# 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

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

Two-Grid Multigrid Error

Multigrid methods target the low frequency error

$$E_{2MG} = S_f \left(I - P_{ctof} A_c^{-1} R_{ftoc} A_f \right) S_f \tag{14}$$

- 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

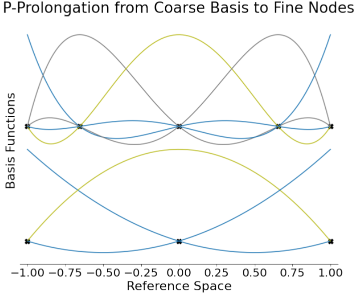
The definition of the symbol follows naturally:

$$\tilde{\mathrm{E}}_{2MG}(\boldsymbol{\theta}) = \tilde{\mathrm{S}}_f(\boldsymbol{\theta}, \omega) \left(\mathrm{I} - \tilde{\mathrm{P}}_{ctof}(\boldsymbol{\theta}) \tilde{\mathrm{A}}_c^{-1}(\boldsymbol{\theta}) \tilde{\mathrm{R}}_{ftoc}(\boldsymbol{\theta}) \tilde{\mathrm{A}}_f(\boldsymbol{\theta}) \right) \tilde{\mathrm{S}}_f(\boldsymbol{\theta}, \omega) \tag{15}$$

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

P-Multigrid Transfer Operators

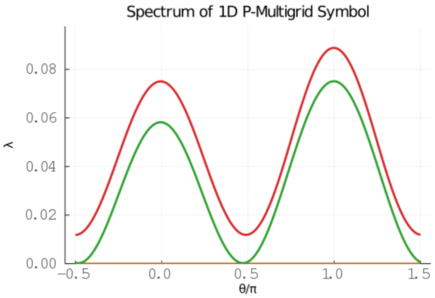
p-multigrid prolongation can be represented as an interpolation from the coarse to fine grid

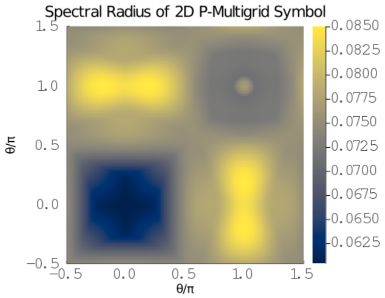


$$\begin{aligned} \mathrm{P}_{\text{ctof}} &= \mathcal{P}_f^T \mathcal{E}_f^T \mathrm{P}^e \mathcal{E}_c \mathcal{P}_c \\ \mathrm{P}^e &= \mathrm{I} \mathrm{D}_{\text{scale}} \mathrm{B}_{\text{ctof}} \end{aligned} \tag{16}$$

D scales for node multiplicity

Example: *P*-Multigrid





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

Significant reduction in spectral radius

Agressive Coarsening

- High-order fine grid is most efficient representation
- Linear coarse grid is easier to solve with traditional methods
- Want to reduce number of intermediate grids
- Typical smoothers do not respond well to agressive coarsening

Experiments: P-Multigrid with Chebyshev

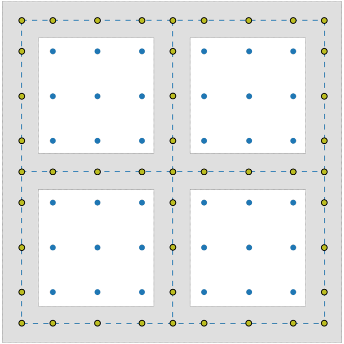
p_{fine} to p_{coarse}	$k=3$			$k=4$		
	LFA	libCEED	its	LFA	libCEED	its
$p=2$ to $p=1$	0.076	0.058	9	0.041	0.033	7
$p=4$ to $p=2$	0.111	0.097	10	0.062	0.050	8
$p=4$ to $p=1$	0.416	0.398	25	0.295	0.276	18
$p=8$ to $p=4$	0.197	0.195	15	0.121	0.110	11
$p=8$ to $p=2$	0.611	0.603	46	0.506	0.469	31
$p=8$ to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

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

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

$$f(x,y,z) = xyz \sin(\pi x) \sin(\pi (1.23 + 0.5y)) \sin(\pi (2.34 + 0.25z)) \quad (17)$$

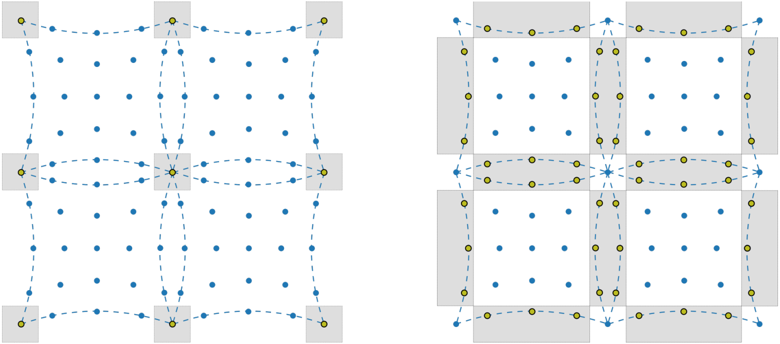
BDDC Overview



High-order single element subdomains

BDDC - non-overlapping domain decomposition method by Dohrmann

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

$$\hat{A}^{-1} = \sum_{e=1}^{N} R_i^{e,T} \hat{A}^{e,-1} R_i^e, \qquad \hat{A}^e = \begin{bmatrix} A_{r,r}^e & \hat{A}_{\Pi,r}^{e,T} \\ \hat{A}_{\Pi,r}^e & \hat{A}_{\Pi,\Pi}^e \end{bmatrix} \tag{18}$$

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

$$\mathsf{R}_1 = \mathrm{diag}\left(\left[\frac{1}{\left|\mathcal{N}\left(x_i\right)\right|}\right]\right) \tag{19}$$

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

$$\begin{aligned} \mathsf{R}_2 &= \mathsf{R}_1 - \mathsf{J}^T \boldsymbol{\mathcal{H}}^T \\ \boldsymbol{\mathcal{H}}^e &= -\mathsf{A}_{\mathsf{I},\mathsf{I}}^{e,-1} \mathsf{A}_{\Gamma,\mathsf{I}}^{e,T} \end{aligned} \tag{20}$$

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

Subassembled Inverse

$$\hat{A}^e = \begin{bmatrix} A^e_{r,r} & \hat{A}^{e,T}_{\Pi,r} \\ \hat{A}^e_{\Pi,r} & \hat{A}^e_{\Pi,\Pi} \end{bmatrix} \qquad \hat{S}_\Pi = A_{\Pi,\Pi} - \hat{A}_{\Pi,r} A^{-1}_{r,r} \hat{A}^T_{\Pi,r} \tag{21}$$

- Subassembled problem inverted with Schur complement
- Coarse grid problem $\hat{S}_\Pi$ is easier to solve with traditional methods
- Dense high-order element interior inverse $A^{-1}_{r,r}$ can be expensive
- Fast diagonalization can provide effiecient approximate solver

Fast Diagonalization

For separable problems of the form

$$\mathrm{A} = a\mathrm{M} + b\mathrm{K} \tag{22}$$

Fast Diagonalization provides fast approximate solver

$$\mathrm{A}^{-1} = \mathrm{S}^{T} \left(a\mathrm{I} + b\boldsymbol{\Lambda} \right)^{-1} \mathrm{S} \tag{23}$$

where

$$\mathrm{S}\mathrm{M}\mathrm{S}^{T} = \mathrm{I}, \;\; \mathrm{S}\mathrm{K}\mathrm{S}^{T} = \boldsymbol{\Lambda} \tag{24}$$

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 injection subdomain operator inverse mitigates expensive setup cost of Dirichlet BDDC

LFA of BDDC

$$\hat{\tilde{A}}^{-1} = \begin{bmatrix} I & -\tilde{A}_{r,r}^{-1} \hat{\tilde{A}}_{\Pi,r}^{T} \\ 0 & I \end{bmatrix} \begin{bmatrix} \tilde{A}_{r,r}^{-1} & 0 \\ 0 & \hat{\tilde{S}}_{\Pi}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -\tilde{A}_{\Pi,r} \hat{\tilde{A}}_{r,r}^{-1} & I \end{bmatrix} \tag{25}$$

$$\tilde{A}_{r,r}^{-1}(\boldsymbol{\theta}) = A_{r,r}^{-1} \odot \left[e^{\imath \left(x_j - x_i \right) \cdot \boldsymbol{\theta} / h} \right], \quad \hat{\tilde{A}}_{r,\Pi}(\boldsymbol{\theta}) = \left(\hat{A}_{r,\Pi} \odot \left[e^{\imath \left(x_j - x_i \right) \cdot \boldsymbol{\theta} / h} \right] \right) Q_{\Pi},$$
$$\hat{\tilde{S}}_{\Pi}^{-1}(\boldsymbol{\theta}) = \left(Q_{\Pi}^{T} \left(\hat{S}_{\Pi} \odot \left[e^{\imath \left(x_j - x_i \right) \cdot \boldsymbol{\theta} / h} \right] \right) Q_{\Pi} \right)^{-1} \tag{26}$$

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 (Brown and He)

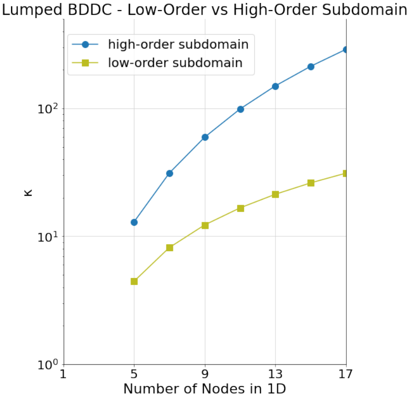
High-Order Experiments

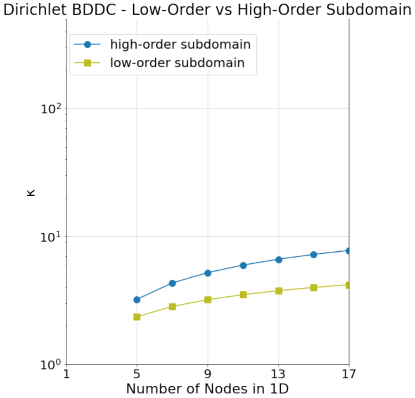
p	Lumped BDDC			Dirichlet BDDC		
	λ_{min}	λ_{max}	κ	λ_{min}	λ_{max}	κ
$p = 2$	1.000	2.800	2.800	1.000	2.042	2.042
$p = 4$	1.000	12.948	12.948	1.000	3.242	3.242
$p = 8$	1.000	59.563	59.563	1.000	5.197	5.197
$p = 16$	1.000	289.678	289.678	1.000	7.761	7.761

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

Single high-order element subdomains less well conditioned

Low vs High-Order BDDC

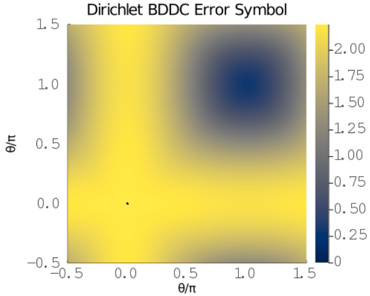




Low-order and high-order subdomain condition number

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

$$\tilde{\mathrm{E}}(\boldsymbol{\theta}, \omega) = \mathrm{I} - \omega \tilde{\mathrm{M}}_2^{-1} \tilde{A}(\boldsymbol{\theta}) \quad (27)$$

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 investigated LFA of Balancing Domain Decomposition by Constraints (BDDC) for high-order element subdomains
- Finally, we investigated LFA of p-multigrid with a BDDC smoother

BDDC Preconditioned P-Multigrid for High-Order Finite Elements

Jeremy L Thompson

University of Colorado, Boulder

jeremy@jeremylt.org

April 4, 2022