Beyond AMLS: domain decomposition with rational filtering

Vassilis Kalantzis

Computer Science and Engineering Department
University of Minnesota - Twin Cities, USA

Argonne National Laboratory, Lemont, IL
11-07-2017

University of Minnesota
Supercomputing Institute

Acknowledgments

- Joint work with **Y. Xi (UMN), and Y. Saad (UMN)**.
- Special thanks to the University of Minnesota Supercomputing Institute for providing us with computational resources to perform our experiments.
- Work supported by NSF and DOE (DE-SC0008877).

Contents

Introduction

Our focus

- We consider the symmetric eigenvalue problem $Ax = \lambda Mx$, where A and M are sparse, and M is SPD.
- We are interested in computing all *nev* eigenvalues-eigenvectors located inside the real interval $[\alpha, \beta]$.
- In this talk: we combine domain decomposition with rational filtering

Contribution of this talk

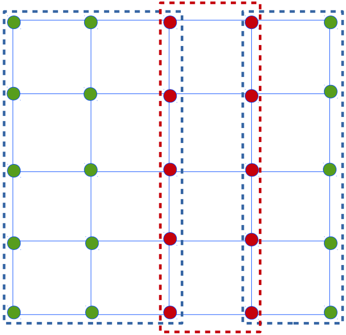
We formulate an algorithm, abbreviated as RF-DDES, that features:

- Reduced orthogonalization costs compared to Krylov projection methods
- Enhanced accuracy compared to existing domain decomposition approaches
- Reduced complex arithmetic

Also: we discuss a parallel (PETSc) implementation of the proposed algorithm

Contents

The main idea behind DD eigenvalue solvers (example for two subdomains)



DD decouples the original eigenvalue problem into two parts:

- The first part considers only interface (red) variables
- The second part considers only interior (green) variables

Reordering equations/unknowns ($p \geq 2$ subdomains)

$$A = \begin{pmatrix} B_1 & & & & E_1 \\ & B_2 & & & E_2 \\ & & \ddots & & \vdots \\ & & & B_p & E_p \\ E_1^T & E_2^T & \cdots & E_p^T & C \end{pmatrix},$$

$$M = \begin{pmatrix} M_B^{(1)} & & & & M_E^{(1)} \\ & M_B^{(2)} & & & M_E^{(2)} \\ & & \ddots & & \vdots \\ & & & M_B^{(p)} & M_E^{(p)} \\ (M_E^{(1)})^T & (M_E^{(2)})^T & \cdots & (M_E^{(p)})^T & M_C \end{pmatrix}.$$

Reordering equations/unknowns ($p \geq 2$ subdomains)

$$A = \begin{pmatrix} B_1 & & & & E_1 \\ & B_2 & & & E_2 \\ & & \ddots & & \vdots \\ & & & B_p & E_p \\ E_1^T & E_2^T & \cdots & E_p^T & C \end{pmatrix},$$

Notation: write as

$$A = \begin{pmatrix} B & E \\ E^T & C \end{pmatrix}, M = \begin{pmatrix} M_B & M_E \\ M_E^T & M_C \end{pmatrix},$$

$$M = \begin{pmatrix} M_B^{(1)} & & & & M_E^{(1)} \\ & M_B^{(2)} & & & M_E^{(2)} \\ & & \ddots & & \vdots \\ & & & M_B^{(p)} & M_E^{(p)} \\ (M_E^{(1)})^T & (M_E^{(2)})^T & \cdots & (M_E^{(p)})^T & M_C \end{pmatrix}.$$

$$x^{(i)} = \begin{pmatrix} u^{(i)} \\ y^{(i)} \end{pmatrix} = \begin{pmatrix} u_1^{(i)} \\ \vdots \\ u_p^{(i)} \\ y_1^{(i)} \\ \vdots \\ y_p^{(i)} \end{pmatrix}.$$

An example of the sparsity pattern of A and M for $p = 4$

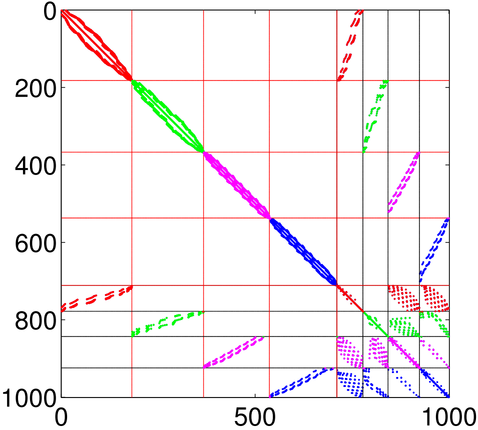


Figure: Different colors → different subdomains

Invariant subspaces from a Schur complement viewpoint

$$(A - \lambda_i M)x^{(i)} = \begin{pmatrix} B - \lambda_i M_B & E - \lambda_i M_E \\ E^T - \lambda_i M_E^T & C - \lambda_i M_C \end{pmatrix} \begin{pmatrix} u^{(i)} \\ y^{(i)} \end{pmatrix} = 0.$$

Invariant subspaces from a Schur complement viewpoint

$$(A - \lambda_i M)x^{(i)} = \begin{pmatrix} B - \lambda_i M_B & E - \lambda_i M_E \\ E^T - \lambda_i M_E^T & C - \lambda_i M_C \end{pmatrix} \begin{pmatrix} u^{(i)} \\ y^{(i)} \end{pmatrix} = 0.$$

A direct computation leads to:

$$S(\lambda_i)y^{(i)} = 0, \qquad u^{(i)} = -(B - \lambda_i M_B)^{-1}(E - \lambda_i M_E)y^{(i)},$$

Invariant subspaces from a Schur complement viewpoint

$$(A - \lambda_i M)x^{(i)} = \begin{pmatrix} B - \lambda_i M_B & E - \lambda_i M_E \\ E^T - \lambda_i M_E^T & C - \lambda_i M_C \end{pmatrix} \begin{pmatrix} u^{(i)} \\ y^{(i)} \end{pmatrix} = 0.$$

A direct computation leads to:

$$S(\lambda_i)y^{(i)} = 0, \qquad u^{(i)} = -(B - \lambda_i M_B)^{-1}(E - \lambda_i M_E)y^{(i)},$$

$$\mathrm{S}(\lambda_i) = C - \lambda_i M_C - (E - \lambda_i M_E)^T(B - \lambda_i M_B)^{-1}(E - \lambda_i M_E).$$

Invariant subspaces from a Schur complement viewpoint

$$(A - \lambda_i M)x^{(i)} = \begin{pmatrix} B - \lambda_i M_B & E - \lambda_i M_E \\ E^T - \lambda_i M_E^T & C - \lambda_i M_C \end{pmatrix} \begin{pmatrix} u^{(i)} \\ y^{(i)} \end{pmatrix} = 0.$$

A direct computation leads to:

$$S(\lambda_i)y^{(i)} = 0, \qquad u^{(i)} = -(B - \lambda_i M_B)^{-1}(E - \lambda_i M_E)y^{(i)},$$

$$\mathrm{S}(\lambda_i) = C - \lambda_i M_C - (E - \lambda_i M_E)^T(B - \lambda_i M_B)^{-1}(E - \lambda_i M_E).$$

To recover the exact eigenpairs $(\lambda_i, x^{(i)})_{i=1,\ldots,nev}$

Perform a Rayleigh-Ritz projection on $\mathcal{Z} = \mathcal{U} \oplus \mathcal{Y}$:

$$\mathcal{Y} = \mathrm{span}\left\{y^{(i)}\right\}_{i=1,\ldots,nev},$$

$$\mathcal{U} = \mathrm{span}\left\{-(B - \lambda_i M_B)^{-1}(E - \lambda_i M_E)y^{(i)}\right\}_{i=1,\ldots,nev}$$

The Automated Multi-Level Substructuring (AMLS) approach

Truncation of the interface eigenvalue problem

- AMLS considers a first-order approximation of $S(\lambda_i),\ i = 1, \ldots, nev$ around a fixed $\sigma \in \mathbb{R}$
- $\mathcal{Y}$ is approximated by $\mathrm{span}\left\{\hat{y}^{(1)}, \ldots, \hat{y}^{(k)}\right\}$, where $\hat{y}^{(1)}, \ldots, \hat{y}^{(k)}$ denote the eigenvectors associated with the k smallest (in modulus) eigenvalues of $(S(\sigma), -S'(\sigma))$.
- Pros: reduced orthogonalization costs
- Cons: only moderate accuracy

Approximation of the solution associated with the interior variables

- Similarly, $\mathcal{U}$ is approximated by $\mathrm{span}\left\{(B - \sigma M_B)^{-1}(E - \sigma M_E)\left[\hat{y}^{(1)}, \ldots, \hat{y}^{(k)}\right]\right\}$
- This step is trivially parallel among the subdomains

Contents

- Let

$$I_{[\alpha,\beta]}(\zeta) = \frac{1}{2\pi i} \int_{\Gamma_{[\alpha,\beta]}} \frac{1}{\nu - \zeta} d\nu.$$

- Let

$$I_{[\alpha,\beta]}(\zeta) = \frac{1}{2\pi i} \int_{\Gamma_{[\alpha,\beta]}} \frac{1}{\nu - \zeta} d\nu.$$

- We approximate -$I_{[\alpha,\beta]}(\zeta)$ by

$$\rho(\zeta) = \sum_{\ell=1}^{2N_c} \frac{\omega_\ell}{\zeta - \zeta_\ell}.$$

- Let

$$I_{[\alpha,\beta]}(\zeta) = \frac{1}{2\pi i} \int_{\Gamma_{[\alpha,\beta]}} \frac{1}{\nu - \zeta} d\nu.$$

- We approximate -$I_{[\alpha,\beta]}(\zeta)$ by

$$\rho(\zeta) = \sum_{\ell=1}^{2N_c} \frac{\omega_\ell}{\zeta - \zeta_\ell}.$$

- Applying the filter to the matrix pencil (A, M) gives:

$$\rho(M^{-1}A) = \sum_{\ell=1}^{2N_c} \omega_\ell (A - \zeta_\ell M)^{-1} M.$$

- Let

$$I_{[\alpha,\beta]}(\zeta) = \frac{1}{2\pi i}\int_{\Gamma_{[\alpha,\beta]}} \frac{1}{\nu-\zeta}\,d\nu.$$

- We approximate $-I_{[\alpha,\beta]}(\zeta)$ by

$$\rho(\zeta) = \sum_{\ell=1}^{2N_c} \frac{\omega_\ell}{\zeta-\zeta_\ell}.$$

- Applying the filter to the matrix pencil (A, M) gives:

$$\rho(M^{-1}A) = \sum_{\ell=1}^{2N_c} \omega_\ell (A-\zeta_\ell M)^{-1} M.$$

- Note that if $(\omega_\ell, \zeta_\ell) = \overline{(\omega_{\ell+N_c}, \zeta_{\ell+N_c})}$:

$$\rho(M^{-1}A) = 2\Re e\left\{\sum_{\ell=1}^{N_c} \omega_\ell (A-\zeta_\ell M)^{-1} M\right\}.$$

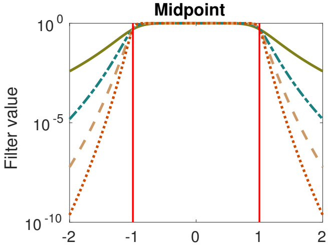


Figure: x-axis:ζ. y-axis:$\rho(\zeta)$.

How to approximate $\mathrm{span}\{y^{(1)}, \ldots, y^{(nev)}\}$ (I)

Let $\zeta \in \mathbb{C}$ and define

$$B_\zeta = B - \zeta M_B, \qquad E_\zeta = E - \zeta M_E,$$
$$C_\zeta = C - \zeta M_C, \qquad S_\zeta = C_\zeta - E_\zeta^T B_\zeta^{-1} E_\zeta.$$

How to approximate $\mathrm{span}\{y^{(1)},\ldots,y^{(nev)}\}$ (I)

Let $\zeta \in \mathbb{C}$ and define

$$B_\zeta = B - \zeta M_B, \qquad E_\zeta = E - \zeta M_E,$$
$$C_\zeta = C - \zeta M_C, \qquad S_\zeta = C_\zeta - E_\zeta^T B_\zeta^{-1} E_\zeta.$$

Then,

$$(A - \zeta M)^{-1} = \begin{pmatrix} B_\zeta^{-1} + B_\zeta^{-1} E_\zeta S_\zeta^{-1} E_\zeta^T B_\zeta^{-1} & -B_\zeta^{-1} E_\zeta S_\zeta^{-1} \\ -S_\zeta^{-1} E_\zeta^T B_\zeta^{-1} & S_\zeta^{-1} \end{pmatrix}.$$

How to approximate $\mathrm{span}\{y^{(1)}, \ldots, y^{(nev)}\}$ (I)

Let $\zeta \in \mathbb{C}$ and define

$$B_\zeta = B - \zeta M_B, \qquad E_\zeta = E - \zeta M_E,$$
$$C_\zeta = C - \zeta M_C, \qquad S_\zeta = C_\zeta - E_\zeta^T B_\zeta^{-1} E_\zeta.$$

Then,

$$(A - \zeta M)^{-1} = \begin{pmatrix} B_\zeta^{-1} + B_\zeta^{-1} E_\zeta S_\zeta^{-1} E_\zeta^T B_\zeta^{-1} & -B_\zeta^{-1} E_\zeta S_\zeta^{-1} \\ -S_\zeta^{-1} E_\zeta^T B_\zeta^{-1} & S_\zeta^{-1} \end{pmatrix}.$$

Recall the partitioning $x^{(i)} = [(u^{(i)})^T, (y^{(i)})^T]^T$:

How to approximate $\mathrm{span}\{y^{(1)}, \ldots, y^{(nev)}\}$ (I)

Let $\zeta \in \mathbb{C}$ and define

$$\begin{aligned} B_\zeta &= B - \zeta M_B, & E_\zeta &= E - \zeta M_E, \\ C_\zeta &= C - \zeta M_C, & S_\zeta &= C_\zeta - E_\zeta^T B_\zeta^{-1} E_\zeta. \end{aligned}$$

Then,

$$(A - \zeta M)^{-1} = \begin{pmatrix} B_\zeta^{-1} + B_\zeta^{-1} E_\zeta S_\zeta^{-1} E_\zeta^T B_\zeta^{-1} & -B_\zeta^{-1} E_\zeta S_\zeta^{-1} \\ -S_\zeta^{-1} E_\zeta^T B_\zeta^{-1} & S_\zeta^{-1} \end{pmatrix}.$$

Recall the partitioning $x^{(i)} = [(u^{(i)})^T, (y^{(i)})^T]^T$:

$$\rho(M^{-1}A) = 2\Re e \left\{ \sum_{\ell=1}^{N_c} \omega_\ell \begin{bmatrix} B_{\zeta_\ell}^{-1} + B_{\zeta_\ell}^{-1} E_{\zeta_\ell} S_{\zeta_\ell}^{-1} E_{\zeta_\ell}^T B_{\zeta_\ell}^{-1} & -B_{\zeta_\ell}^{-1} E_{\zeta_\ell} S_{\zeta_\ell}^{-1} \\ -S_{\zeta_\ell}^{-1} E_{\zeta_\ell}^T B_{\zeta_\ell}^{-1} & S_{\zeta_\ell}^{-1} \end{bmatrix} \right\} M$$

How to approximate $\mathrm{span}\{y^{(1)},\ldots,y^{(nev)}\}$ (I)

Let $\zeta \in \mathbb{C}$ and define

$$\begin{aligned} B_\zeta &= B - \zeta M_B, & E_\zeta &= E - \zeta M_E, \\ C_\zeta &= C - \zeta M_C, & S_\zeta &= C_\zeta - E_\zeta^T B_\zeta^{-1} E_\zeta. \end{aligned}$$

Then,

$$(A - \zeta M)^{-1} = \begin{pmatrix} B_\zeta^{-1} + B_\zeta^{-1} E_\zeta S_\zeta^{-1} E_\zeta^T B_\zeta^{-1} & -B_\zeta^{-1} E_\zeta S_\zeta^{-1} \\ -S_\zeta^{-1} E_\zeta^T B_\zeta^{-1} & S_\zeta^{-1} \end{pmatrix}.$$

Recall the partitioning $x^{(i)} = [(u^{(i)})^T, (y^{(i)})^T]^T$:

$$\begin{aligned} \rho(M^{-1}A) &= 2\Re e \left\{ \sum_{\ell=1}^{N_c} \omega_\ell \begin{bmatrix} B_{\zeta_\ell}^{-1} + B_{\zeta_\ell}^{-1} E_{\zeta_\ell} S_{\zeta_\ell}^{-1} E_{\zeta_\ell}^T B_{\zeta_\ell}^{-1} & -B_{\zeta_\ell}^{-1} E_{\zeta_\ell} S_{\zeta_\ell}^{-1} \\ -S_{\zeta_\ell}^{-1} E_{\zeta_\ell}^T B_{\zeta_\ell}^{-1} & S_{\zeta_\ell}^{-1} \end{bmatrix} \right\} M \\ &= \sum_{i=1}^{n} \rho(\lambda_i) \begin{bmatrix} u^{(i)}(u^{(i)})^T & u^{(i)}(y^{(i)})^T \\ y^{(i)}(u^{(i)})^T & y^{(i)}(y^{(i)})^T \end{bmatrix} M. \end{aligned}$$

How to approximate $\mathrm{span}\{y^{(1)},\ldots,y^{(nev)}\}$ (II)

Equating blocks leads to:

$$2\Re e\left\{\sum_{\ell=1}^{N_c}\omega_\ell S_{\zeta_\ell}^{-1}\right\}=\sum_{i=1}^{n}\rho(\lambda_i)y^{(i)}(y^{(i)})^T.$$

Since $\rho(\lambda_1),\ldots,\rho(\lambda_{nev})\neq 0$:

$$\mathrm{span}\{y^{(1)},\ldots,y^{(nev)}\}\subseteq \mathrm{range}\left(2\Re e\left\{\sum_{\ell=1}^{N_c}\omega_\ell S_{\zeta_\ell}^{-1}\right\}\right).$$

How to approximate $\mathrm{span}\{y^{(1)}, \ldots, y^{(nev)}\}$ (II)

Equating blocks leads to:

$$2\Re e \left\{ \sum_{\ell=1}^{N_c} \omega_\ell S_{\zeta_\ell}^{-1} \right\} = \sum_{i=1}^{n} \rho(\lambda_i) y^{(i)} (y^{(i)})^T.$$

Since $\rho(\lambda_1), \ldots, \rho(\lambda_{nev}) \neq 0$:

$$\mathrm{span}\{y^{(1)}, \ldots, y^{(nev)}\} \subseteq \mathrm{range}\left(2\Re e \left\{ \sum_{\ell=1}^{N_c} \omega_\ell S_{\zeta_\ell}^{-1} \right\} \right).$$

Capture $\mathrm{range}\left(\Re e \left\{ \sum_{\ell=1}^{N_c} \omega_\ell S_{\zeta_\ell}^{-1} \right\}\right)$ by a Krylov projection scheme!

Algorithm 3.1: Krylov restricted to the interface variables

Algorithm

0. *Start with* $q^{(1)} \in \mathbb{R}^s$, *s.t.* $\|q^{(1)}\|_2 = 1$, $q_0 := 0$, $b_1 = 0$, $\mathrm{tol} \in \mathbb{R}$
1. *For* $\mu = 1, 2, \ldots$
2. *Compute* $w = \Re e \left\{ \sum_{\ell=1}^{N_c} \omega_\ell S_{\zeta_\ell}^{-1} q^{(\mu)} \right\} - b_\mu q^{(\mu-1)}$
3. $a_\mu = w^T q^{(\mu)}$
4. *For* $\kappa = 1, \ldots, \mu$
5. $w = w - q^{(\kappa)}(w^T q^{(\kappa)})$
6. *End*
7. $b_{\mu+1} := \|w\|_2$
8. *If* $b_{\mu+1} = 0$
9. *generate a unit-norm* $q^{(\mu+1)}$ *orthogonal to* $q^{(1)}, \ldots, q^{(\mu)}$
10. *Else*
11. $q^{(\mu+1)} = w / b_{\mu+1}$
12 *EndIf*
13. *If the sum of eigenvalue of* T_μ *remains unchanged (up to* tol*) during the last few iterations; BREAK; EndIf*
14. *End*
15. *Return* $Q_\mu = [q^{(1)}, \ldots, q^{(\mu)}]$

How to approximate $\mathrm{span}\{y^{(1)}, \ldots, y^{(nev)}\}$ (III)

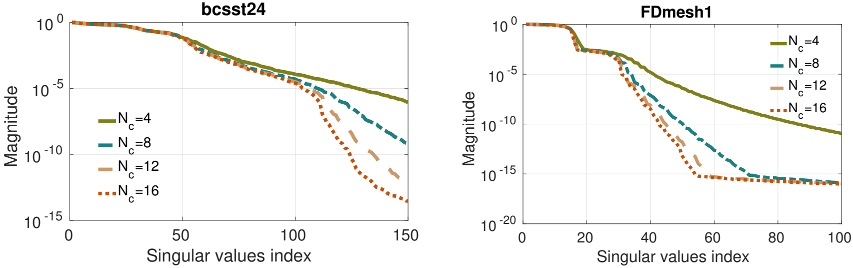


Figure: Leading singular values of $\Re e \left\{ \sum_{\ell=1}^{N_c} \omega_\ell S(\zeta_\ell)^{-1} \right\}$ $([\alpha, \beta] = [\lambda_1, \lambda_{100}])$.

How to approximate $\mathrm{span}\{y^{(1)}, \ldots, y^{(nev)}\}$ (III)

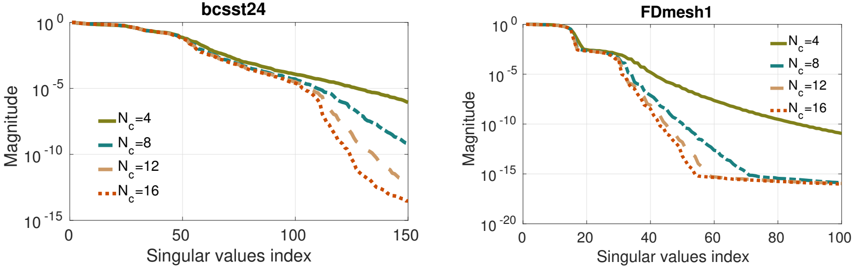


Figure: Leading singular values of $\Re e\left\{\sum_{\ell=1}^{N_c} \omega_\ell S(\zeta_\ell)^{-1}\right\}$ $([\alpha, \beta] = [\lambda_1, \lambda_{100}])$.

- Only vectors of length s (# of interface variables) need be orthonormalized
- Moreover, solve$(A, M, \zeta_\ell) \approx$ solve$(S(\zeta_\ell)) + 2 \times$ solve(B, M_B, ζ_ℓ)

How to approximate $\mathrm{span}\{y^{(1)}, \ldots, y^{(nev)}\}$ (III)

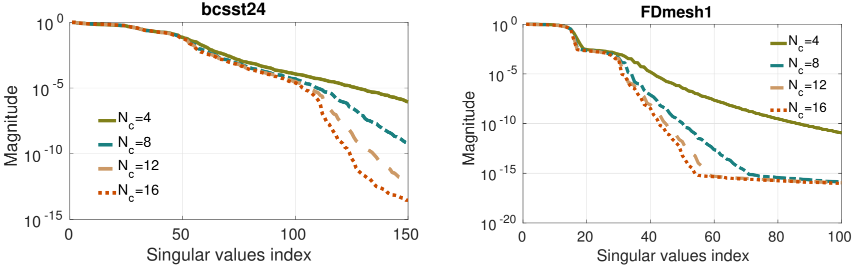


Figure: Leading singular values of $\Re e\left\{\sum_{\ell=1}^{N_c} \omega_\ell S(\zeta_\ell)^{-1}\right\}$ $([\alpha, \beta] = [\lambda_1, \lambda_{100}])$.

- Only vectors of length s (# of interface variables) need be orthonormalized
- Moreover, solve$(A, M, \zeta_\ell) \approx$ solve$(S(\zeta_\ell))$ + 2 $\times$ solve(B, M_B, ζ_ℓ)
- What if $nev > s$, or $\mathrm{rank}[y^{(1)}, \ldots, y^{(nev)}] < nev$?

How to approximate $\mathrm{span}\{u^{(1)}, \ldots, u^{(nev)}\}$ (I)

Standard approach

Compute $u^{(i)} = -(B - \lambda_i M_B)^{-1}(E - \lambda_i M_E)y^{(i)} = -B_{\lambda_i}^{-1} E_{\lambda_i} y^{(i)},\ i = 1, \ldots, nev.$

How to approximate $\mathrm{span}\{u^{(1)}, \ldots, u^{(nev)}\}$ (I)

Standard approach

Compute $u^{(i)} = -(B - \lambda_i M_B)^{-1}(E - \lambda_i M_E)y^{(i)} = -B_{\lambda_i}^{-1} E_{\lambda_i} y^{(i)}, \; i = 1, \ldots, nev.$

- Issue #1: Needs access to both $\lambda_i, \; y^{(i)}$

How to approximate $\mathrm{span}\{u^{(1)}, \ldots, u^{(nev)}\}$ (I)

Standard approach

Compute $u^{(i)} = -(B - \lambda_i M_B)^{-1}(E - \lambda_i M_E)y^{(i)} = -B_{\lambda_i}^{-1} E_{\lambda_i} y^{(i)},\ i = 1, \ldots, nev.$

- Issue #1: Needs access to both $\lambda_i,\ y^{(i)}$
- Issue #2: Impractical for large values of *nev*

The alternative: approximate the action of $B_{\lambda_i}^{-1},\ E_{\lambda_i}$

How to approximate $\mathrm{span}\{u^{(1)}, \ldots, u^{(nev)}\}$ (I)

Standard approach

Compute $u^{(i)} = -(B - \lambda_i M_B)^{-1}(E - \lambda_i M_E)y^{(i)} = -B_{\lambda_i}^{-1} E_{\lambda_i} y^{(i)},\ i = 1, \ldots, nev.$

- Issue #1: Needs access to both $\lambda_i,\ y^{(i)}$
- Issue #2: Impractical for large values of *nev*

The alternative: approximate the action of $B_{\lambda_i}^{-1},\ E_{\lambda_i}$

Assume that $y^{(i)}$ is known:

How to approximate $\mathrm{span}\{u^{(1)}, \ldots, u^{(nev)}\}$ (I)

Standard approach

Compute $u^{(i)} = -(B - \lambda_i M_B)^{-1}(E - \lambda_i M_E)y^{(i)} = -B_{\lambda_i}^{-1} E_{\lambda_i} y^{(i)}, \; i = 1, \ldots, nev.$

- Issue #1: Needs access to both $\lambda_i, \; y^{(i)}$
- Issue #2: Impractical for large values of *nev*

The alternative: approximate the action of $B_{\lambda_i}^{-1}, \; E_{\lambda_i}$

Assume that $y^{(i)}$ is known:

- Let $\sigma \in \mathbb{R}$ and start with a "basic" approximation:

$$\hat{u}^{(i)} = -B_{\sigma}^{-1} E_{\sigma} y^{(i)}$$

How to approximate $\mathrm{span}\{u^{(1)}, \ldots, u^{(nev)}\}$ (I)

Standard approach

Compute $u^{(i)} = -(B - \lambda_i M_B)^{-1}(E - \lambda_i M_E)y^{(i)} = -B_{\lambda_i}^{-1} E_{\lambda_i} y^{(i)},\ i = 1, \ldots, nev.$

- Issue #1: Needs access to both $\lambda_i,\ y^{(i)}$
- Issue #2: Impractical for large values of *nev*

The alternative: approximate the action of $B_{\lambda_i}^{-1},\ E_{\lambda_i}$

Assume that $y^{(i)}$ is known:

- Let $\sigma \in \mathbb{R}$ and start with a "basic" approximation:

$$\hat{u}^{(i)} = -B_{\sigma}^{-1} E_{\sigma} y^{(i)}$$

- The error is of the form:

$$u^{(i)} - \hat{u}^{(i)} = -[B_{\lambda_i}^{-1} - B_{\sigma}^{-1}] E_{\sigma} y^{(i)} + (\lambda_i - \sigma) B_{\lambda_i}^{-1} M_E y^{(i)}.$$

How to approximate $\mathrm{span}\{u^{(1)}, \ldots, u^{(nev)}\}$ (I)

Standard approach

Compute $u^{(i)} = -(B - \lambda_i M_B)^{-1}(E - \lambda_i M_E)y^{(i)} = -B_{\lambda_i}^{-1} E_{\lambda_i} y^{(i)}, \; i = 1, \ldots, nev.$

- Issue #1: Needs access to both $\lambda_i, \; y^{(i)}$
- Issue #2: Impractical for large values of nev

The alternative: approximate the action of $B_{\lambda_i}^{-1}, \; E_{\lambda_i}$

Assume that $y^{(i)}$ is known:

- Let $\sigma \in \mathbb{R}$ and start with a "basic" approximation:

$$\hat{u}^{(i)} = -B_\sigma^{-1} E_\sigma y^{(i)}$$

- The error is of the form:

$$u^{(i)} - \hat{u}^{(i)} = -[B_{\lambda_i}^{-1} - B_\sigma^{-1}] E_\sigma y^{(i)} + (\lambda_i - \sigma) B_{\lambda_i}^{-1} M_E y^{(i)}.$$

- To improve accuracy: extract $\hat{u}^{(i)}$ from a subspace, i.e. $\hat{u}^{(i)} \in \mathcal{U}$

How to approximate $\mathrm{span}\{u^{(1)},\ldots,u^{(nev)}\}$ (II)

Let $(\delta_\ell, v^{(\ell)}),\ \ell=1,\ldots,d$, denote the eigenpairs of (B, M_B).

Higher-order resolvent expansions

- Exploit $\psi \geq 1$ terms of the formula $B_\lambda^{-1} = B_\sigma^{-1}\sum_{\theta=0}\left[(\lambda-\sigma)M_B B_\sigma^{-1}\right]^\theta$:

$$\|u^{(i)} - \hat{u}^{(i)}\|_{M_B} \leq \left\|\sum_{\ell=1}^{\ell=d} \frac{\gamma_\ell(\lambda-\sigma)^{\psi+1} - \epsilon_\ell(\lambda-\sigma)^{\psi}}{(\delta_\ell-\lambda)(\delta_\ell-\sigma)^{\psi}} v^{(\ell)}\right\|_{M_B}$$

How to approximate $\mathrm{span}\{u^{(1)}, \ldots, u^{(nev)}\}$ (II)

Let $(\delta_\ell, v^{(\ell)}),\ \ell = 1, \ldots, d$, denote the eigenpairs of (B, M_B).

Higher-order resolvent expansions

- Exploit $\psi \geq 1$ terms of the formula $B_\lambda^{-1} = B_\sigma^{-1} \sum_{\theta=0} \left[(\lambda - \sigma) M_B B_\sigma^{-1}\right]^\theta$:

$$\|u^{(i)} - \hat{u}^{(i)}\|_{M_B} \leq \left\| \sum_{\ell=1}^{\ell=d} \frac{\gamma_\ell (\lambda - \sigma)^{\psi+1} - \epsilon_\ell (\lambda - \sigma)^\psi}{(\delta_\ell - \lambda)(\delta_\ell - \sigma)^\psi} v^{(\ell)} \right\|_{M_B}$$

Include eigenvectors of (B, M_B) in $\mathcal{U}$

- If we also include the eigenvectors associated with the κ eigenvalues of (B, M_B) lying the closest to σ :

$$\|u^{(i)} - \hat{u}^{(i)}\|_{M_B} \leq \left\| \sum_{\ell=\kappa+1}^{\ell=d} \frac{\gamma_\ell (\lambda - \sigma)^{\psi+1} - \epsilon_\ell (\lambda - \sigma)^\psi}{(\delta_\ell - \lambda)(\delta_\ell - \sigma)^\psi} v^{(\ell)} \right\|_{M_B}$$

The RF-DDES scheme

RF-DDES is a RR approach on a basis of the subspace $\mathcal{Z} = \mathcal{U} \oplus \mathcal{Y}$

- $\mathcal{Y} = \mathrm{range}\{Q\}$, where Q is the Krylov basis formed by applying Lanczos to $\Re e \left\{ \sum_{\ell=1}^{N_c} \omega_\ell S_{\zeta_\ell}^{-1} \right\}$.

The RF-DDES scheme

RF-DDES is a RR approach on a basis of the subspace $\mathcal{Z} = \mathcal{U} \oplus \mathcal{Y}$

- $\mathcal{Y} = \mathrm{range}\{Q\}$, where Q is the Krylov basis formed by applying Lanczos to $\Re e \left\{ \sum_{\ell=1}^{N_c} \omega_\ell S_{\zeta_\ell}^{-1} \right\}$.
- $\mathcal{U} = \mathrm{range}\{\bar{V}, U_1, U_2\}$ where

$$U_1 = -\left[B_\sigma^{-1} E Q, \ldots, (B_\sigma M_B)^{\psi-1} B_\sigma^{-1} E Q \right],$$

$$U_2 = \left[B_\sigma^{-1} M_E Q, \ldots, (B_\sigma M_B)^{\psi-1} B_\sigma^{-1} M_E Q \right],$$

The RF-DDES scheme

RF-DDES is a RR approach on a basis of the subspace $\mathcal{Z} = \mathcal{U} \oplus \mathcal{Y}$

- $\mathcal{Y} = \mathrm{range}\{Q\}$, where Q is the Krylov basis formed by applying Lanczos to $\Re e \left\{ \sum_{\ell=1}^{N_c} \omega_\ell S_{\zeta_\ell}^{-1} \right\}$.
- $\mathcal{U} = \mathrm{range}\{\bar{V}, U_1, U_2\}$ where

$$U_1 = -\left[B_\sigma^{-1} EQ, \ldots, (B_\sigma M_B)^{\psi-1} B_\sigma^{-1} EQ\right],$$

$$U_2 = \left[B_\sigma^{-1} M_E Q, \ldots, (B_\sigma M_B)^{\psi-1} B_\sigma^{-1} M_E Q\right],$$

- $\bar{V}$ includes the eigenvectors associated with the nev_B eigenvalues lying the closest to σ for each $(B_\sigma^{(j)}, M_B^{(j)}),\ j = 1, \ldots, p$

The RF-DDES scheme

RF-DDES is a RR approach on a basis of the subspace $\mathcal{Z} = \mathcal{U} \oplus \mathcal{Y}$

- $\mathcal{Y} = \mathrm{range}\{Q\}$, where Q is the Krylov basis formed by applying Lanczos to $\Re e \left\{ \sum_{\ell=1}^{N_c} \omega_\ell S_{\zeta_\ell}^{-1} \right\}$.
- $\mathcal{U} = \mathrm{range}\{\bar{V}, U_1, U_2\}$ where

$$U_1 = -\left[B_\sigma^{-1} E Q, \ldots, (B_\sigma M_B)^{\psi-1} B_\sigma^{-1} E Q\right],$$

$$U_2 = \left[B_\sigma^{-1} M_E Q, \ldots, (B_\sigma M_B)^{\psi-1} B_\sigma^{-1} M_E Q\right],$$

- $\bar{V}$ includes the eigenvectors associated with the nev_B eigenvalues lying the closest to σ for each $(B_\sigma^{(j)}, M_B^{(j)}),\ j = 1, \ldots, p$
- The subspace $\mathcal{U}$ is formed independently in each one of the p subdomains

The RF-DDES scheme

RF-DDES is a RR approach on a basis of the subspace $\mathcal{Z} = \mathcal{U} \oplus \mathcal{Y}$

- $\mathcal{Y} = \mathrm{range}\{Q\}$, where Q is the Krylov basis formed by applying Lanczos to $\Re e \left\{ \sum_{\ell=1}^{N_c} \omega_\ell S_{\zeta_\ell}^{-1} \right\}$.
- $\mathcal{U} = \mathrm{range}\{\bar{V}, U_1, U_2\}$ where

$$U_1 = -\left[B_\sigma^{-1} E Q, \ldots, (B_\sigma M_B)^{\psi-1} B_\sigma^{-1} E Q \right],$$

$$U_2 = \left[B_\sigma^{-1} M_E Q, \ldots, (B_\sigma M_B)^{\psi-1} B_\sigma^{-1} M_E Q \right],$$

- $\bar{V}$ includes the eigenvectors associated with the nev_B eigenvalues lying the closest to σ for each $(B_\sigma^{(j)}, M_B^{(j)}),\ j = 1, \ldots, p$
- The subspace $\mathcal{U}$ is formed independently in each one of the p subdomains
- Only real arithmetic need be exploited to form $\mathcal{U}$

The RF-DDES scheme

RF-DDES is a RR approach on a basis of the subspace $\mathcal{Z} = \mathcal{U} \oplus \mathcal{Y}$

- $\mathcal{Y} = \mathrm{range}\{Q\}$, where Q is the Krylov basis formed by applying Lanczos to $\Re e \left\{ \sum_{\ell=1}^{N_c} \omega_\ell S_{\zeta_\ell}^{-1} \right\}$.
- $\mathcal{U} = \mathrm{range}\{\bar{V}, U_1, U_2\}$ where

$$U_1 = -\left[B_\sigma^{-1} E Q, \ldots, (B_\sigma M_B)^{\psi-1} B_\sigma^{-1} E Q \right],$$

$$U_2 = \left[B_\sigma^{-1} M_E Q, \ldots, (B_\sigma M_B)^{\psi-1} B_\sigma^{-1} M_E Q \right],$$

- $\bar{V}$ includes the eigenvectors associated with the nev_B eigenvalues lying the closest to σ for each $(B_\sigma^{(j)}, M_B^{(j)}),\ j = 1, \ldots, p$
- The subspace $\mathcal{U}$ is formed independently in each one of the p subdomains
- Only real arithmetic need be exploited to form $\mathcal{U}$
- When ψ resolvent terms are kept, we will write RF-DDES(ψ)

Contents

Implementation and computing environment

Hardware

- Experiments performed at the `mesabi` linux cluster at Minnesota Supercomputing Institute
- 741 two-socket nodes, each socket equipped with an Intel Haswell E5-2680v3 processor and 32 GB of memory

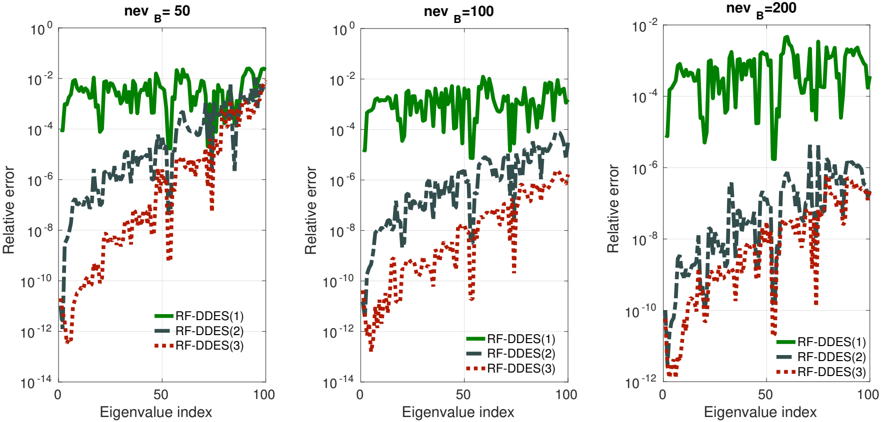
Software

- All methods were implemented in C++ and on top of PETSc (MPI)
- Linked to METIS, PARDISO, MUMPS, and MKL
- Compiled with mpiicpc (-O3)

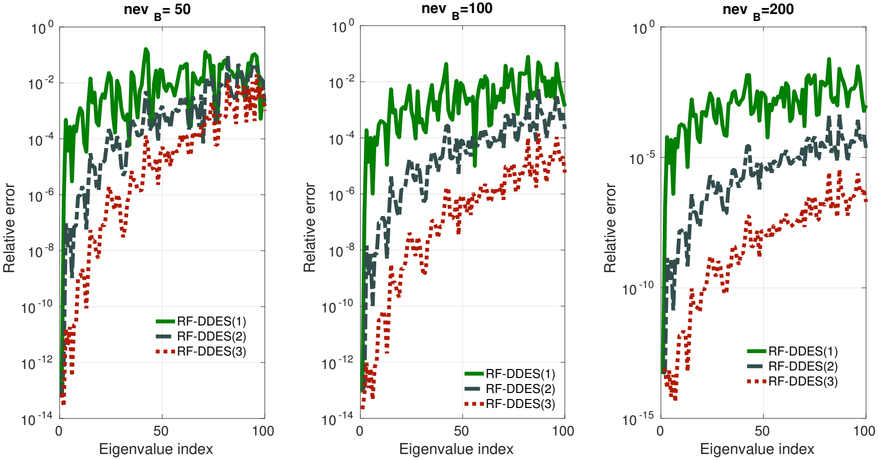
Parameters and details

- Default values: $p = 2$, $N_c = 2$, $nev_B = 100$, and $\sigma = 0$
- All times are listed in seconds
- All experiments are performed in 64-bit arithmetic

Approximation of the $nev = 100$ algebraically smallest eigenvalues of matrix bcsstk39



Approximation of the $nev = 100$ algebraically smallest eigenvalues of pencil qa8fk/qafm



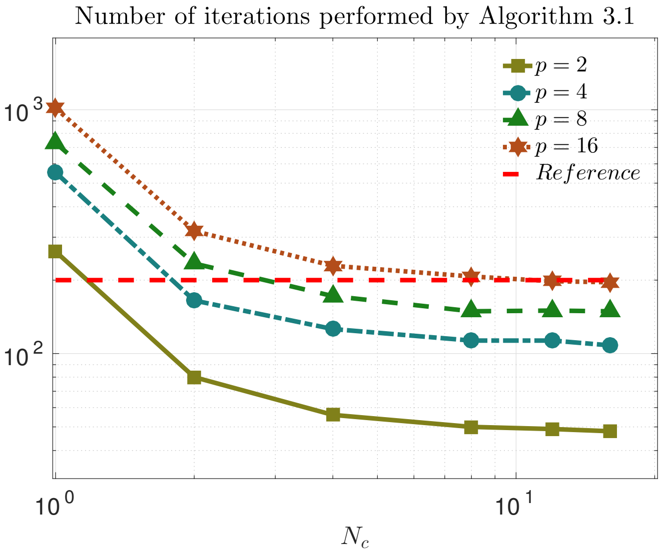


Figure: Matrix: "FDmesh1" (2D Laplacian of size $n = 160 \times 150$). Results are reported for all different combinations of $p = 2,\ 4,\ 8$ and $p = 16$, and $N_c = 1,\ 2,\ 4,\ 8$ and $N_c = 16$. Interval: $[\alpha, \beta] = [\lambda_1, \lambda_{200}]$.

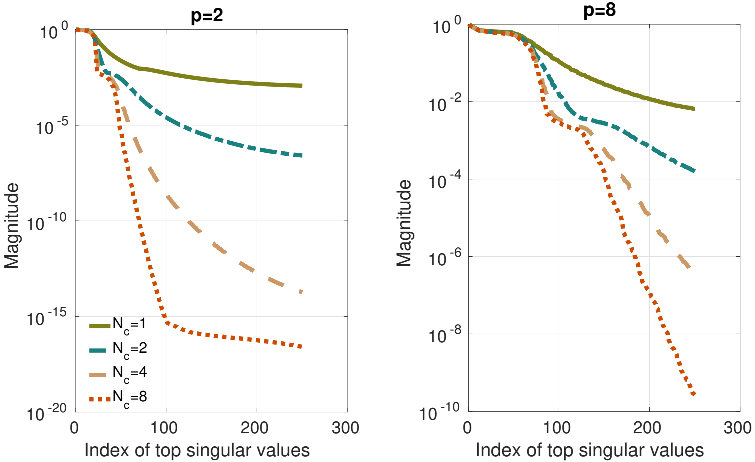


Figure: The leading 250 singular values of $\Re e \left\{ \sum_{\ell=1}^{N_c} \omega_\ell S(\zeta_\ell)^{-1} \right\}$ for matrix "FDmesh1". Left: $p = 2$. Right: $p = 8$.

Rational Filtering Krylov (RF-KRYLOV)

Algorithm

0. *Start with* $q^{(1)} \in \mathbb{R}^n$ *s.t.* $\|q^{(1)}\|_2 = 1$
1. *For* $\mu = 1, 2, \ldots$
2.   *Compute* $w = \Re e \left\{ \sum_{\ell=1}^{N_c} \omega_\ell (A - \zeta_\ell M)^{-1} M q^{(\mu)} \right\}$
3.   *For* $\kappa = 1, \ldots, \mu$
4. $h_{\kappa,\mu} = w^T q^{(\kappa)}$, $w = w - h_{\kappa,\mu} q^{(\kappa)}$
5.   *End*
6. $h_{\mu+1,\mu} = \|w\|_2$
7.   *If* $h_{\mu+1,\mu} \neq 0$
8. $q^{(\mu+1)} = w / h_{\mu+1,\mu}$
9.   *EndIf*
10.   *Check convergence*
11. *End*
12. *Return Ritz values located inside* $[\alpha, \beta]$ *and associated Ritz vectors*

A comparison of RF-KRYLOV and RF-DDES (I)

Table: Wall-clock times of RF-KRYLOV and RF-DDES using $\tau = 2,\ 4,\ 8,\ 16$ and $\tau = 32$ computational cores. RFD(2) and RFD(4) denote RF-DDES with $p = 2$ and $p = 4$ subdomains, respectively.

	$nev = 100$			$nev = 200$			$nev = 300$		
Matrix	RFK	RFD(2)	RFD(4)	RFK	RFD(2)	RFD(4)	RFK	RFD(2)	RFD(4)
shipsec8($\tau = 2$)	114	195	-	195	207	-	279	213	-
($\tau = 4$)	76	129	93	123	133	103	168	139	107
($\tau = 8$)	65	74	56	90	75	62	127	79	68
($\tau = 16$)	40	51	36	66	55	41	92	57	45
($\tau = 32$)	40	36	28	62	41	30	75	43	34
boneS01($\tau = 2$)	94	292	-	194	356	-	260	424	-
($\tau = 4$)	68	182	162	131	230	213	179	277	260
($\tau = 8$)	49	115	113	94	148	152	121	180	187
($\tau = 16$)	44	86	82	80	112	109	93	137	132
($\tau = 32$)	51	66	60	74	86	71	89	105	79

A comparison of RF-KRYLOV and RF-DDES (II)

Table: Wall-clock times of RF-KRYLOV and RF-DDES using $\tau = 2,\ 4,\ 8,\ 16$ and $\tau = 32$ computational cores. RFD(2) and RFD(4) denote RF-DDES with $p = 2$ and $p = 4$ subdomains, respectively.

	$nev = 100$			$nev = 200$			$nev = 300$		
Matrix	RFK	RFD(2)	RFD(4)	RFK	RFD(2)	RFD(4)	RFK	RFD(2)	RFD(4)
FDmesh2($\tau = 2$)	241	85	-	480	99	-	731	116	-
($\tau = 4$)	159	34	63	305	37	78	473	43	85
($\tau = 8$)	126	22	23	228	24	27	358	27	31
($\tau = 16$)	89	16	15	171	17	18	256	20	21
($\tau = 32$)	51	12	12	94	13	14	138	15	20
FDmesh3($\tau = 2$)	1021	446	-	2062	502	-	3328	564	-
($\tau = 4$)	718	201	281	1281	217	338	1844	237	362
($\tau = 8$)	423	119	111	825	132	126	1250	143	141
($\tau = 16$)	355	70	66	684	77	81	1038	88	93
($\tau = 32$)	177	47	49	343	51	58	706	62	82

Table: Number of iterations performed by RF-KRYLOV (denoted as RFK) and RF-DDES (denoted as RFD(p)).

	$nev = 100$			$nev = 200$			$nev = 300$		
	RFK	RFD(2)	RFD(4)	RFK	RFD(2)	RFD(4)	RFK	RFD(2)	RFD(4)
shipsec8	280	170	180	500	180	280	720	190	290
boneS01	240	350	410	480	520	600	620	640	740
FDmesh2	200	100	170	450	130	230	680	160	270
FDmesh3	280	150	230	460	180	290	690	200	380

Table: Maximum relative errors returned by RF-DDES.

	$nev = 100$			$nev = 200$			$nev = 300$		
nev_B	25	50	100	25	50	100	25	50	100
shipsec8	1.4e-3	2.2e-5	2.4e-6	3.4e-3	1.9e-3	1.3e-5	4.2e-3	1.9e-3	5.6e-4
boneS01	5.2e-3	7.1e-4	2.2e-4	3.8e-3	5.9e-4	4.1e-4	3.4e-3	9.1e-4	5.1e-4
FDmesh2	4.0e-5	2.5e-6	1.9e-7	3.5e-4	9.6e-5	2.6e-6	3.2e-4	2.0e-4	2.6e-5
FDmesh3	6.2e-5	8.5e-6	4.3e-6	6.3e-4	1.1e-4	3.1e-5	9.1e-4	5.3e-4	5.3e-5

Amount of time spent on orthonormalization

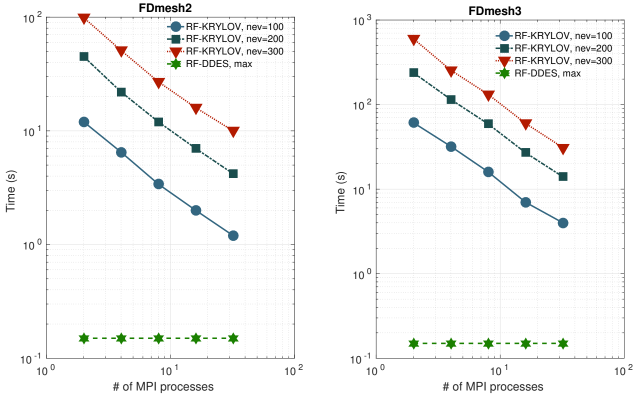


Figure: Left: "FDmesh2" ($n = 250,000$). Right: "FDmesh3" ($n = 1,000,000$).

Runtimes for MPI-only implementation ($nev = 300$)

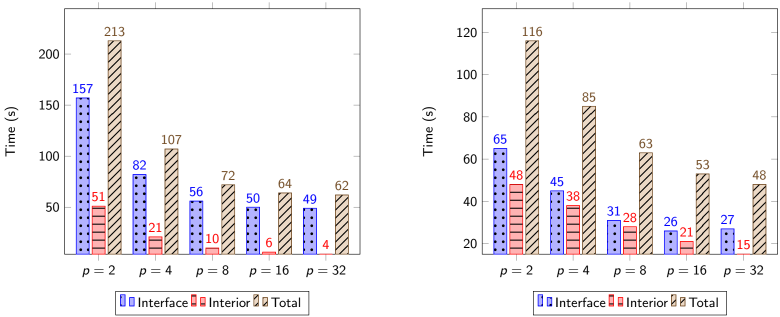


Figure: Left: “shipsec8”. Right: “FDmesh2”.

Conclusion

Summary

The main features of RF-DDES:

- No estimation of *nev* is needed
- Orthogonalization is applied to vectors whose length is equal to the number of interface variables
- The part of the solution associated with the interior variables is computed in real arithmetic
- Ability to exploit a possible low-rank of $y^{(1)}, \ldots, y^{(nev)}$
- Typically, not as accurate as RF-KRYLOV (do we always need high accuracy?)

Considerations

- RF-DDES is well-suited for 2D problems. What about 3D?
- Multi-MPI implementations are possible

Technical reports related to this talk

Main reference:

- V. Kalantzis, Y. Xi, and Y. Saad, "Domain decomposition Krylov rational filtering techniques for symmetric generalized eigenvalue problems".

See also:

- J. Kestyn, V. Kalantzis, E. Polizzi, and Y. Saad, "PFEAST: A High Performance Sparse Eigenvalue Solver Using Distributed-Memory Linear Solvers". *In Proceedings of the ACM/IEEE Supercomputing Conference*, 2016.
- V. Kalantzis, J. Kestyn, E. Polizzi, and Y. Saad, "Domain Decomposition Approaches for Accelerating Contour Integration Eigenvalue Solvers for Symmetric Eigenvalue Problems".

http://www-users.cs.umn.edu/kalantzi/