Domain Decomposition Approaches for Accelerating Contour Integration Eigenvalue Solvers for Symmetric Eigenvalue Problems

Vassilis Kalantzis

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

SIAM CSE, 03-03-2017

UNIVERSITY OF MINNESOTA

Supercomputing Institute

Acknowledgments

- Joint work with **J. Kestyn and E. Polizzi (UMass), and Y. Saad (UMN)**.
- We are grateful 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

The framework of contour integration eigenvalue solvers

Let Γ be a smooth, counter-clockwise oriented curve. If $\lambda_1, \ldots, \lambda_r \in \Gamma$, $X = [x^{(1)}, \ldots, x^{(r)}]$,

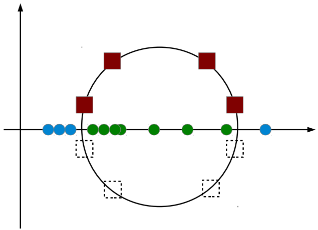
- $\mathcal{P} = \dfrac{-1}{2i\pi} \displaystyle\int_\Gamma (A - \zeta I)^{-1} d\zeta = XX^T$.

In practice, $\mathcal{P} \approx \tilde{\mathcal{P}}$ with

- $\tilde{\mathcal{P}} = \sum_{j=1}^{2N_c} \omega_j (A - \zeta_j I)^{-1}$.
- $\rho(\zeta) = \sum_{j=1}^{2N_c} \dfrac{\omega_j}{\zeta - \zeta_j}$.

Different choices for the projection method:

- FEAST (Subspace iteration).
- SS & CI-RR (Complex moments and RR).



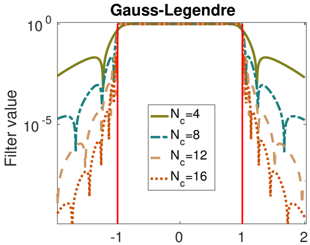


Figure: x-axis:ζ. y-axis:$\rho(\zeta)$.

Considerations and contributions

Solution of linear systems with $A - \zeta I$

- Main computational kernel of contour integration approaches.
- Considerations: direct vs iterative solvers, scalability...
- We consider both direct and iterative solvers from a domain decomposition viewpoint.

Integrating the matrix resolvent fully/partially

- Typical contour integration approaches integrate the entire matrix resolvent.
- We consider exploiting a domain decomposition framework and integrate the matrix resolvent only partially.
- Tradeoff of accuracy/runtime.

Considerations and contributions

Solution of linear systems with $A - \zeta I$

- Main computational kernel of contour integration approaches.
- Considerations: direct vs iterative solvers, scalability...
- We consider both direct and iterative solvers from a domain decomposition viewpoint.

Integrating the matrix resolvent fully/partially

- Typical contour integration approaches integrate the entire matrix resolvent.
- We consider exploiting a domain decomposition framework and integrate the matrix resolvent only partially.
- Tradeoff of accuracy/runtime.

We focus on the "FEAST" framework.

Contents

Reordering equations/unknowns using p subdomains

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

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

Notation: write as

$$A = \begin{pmatrix} B & E \\ E^T & C \end{pmatrix}.$$

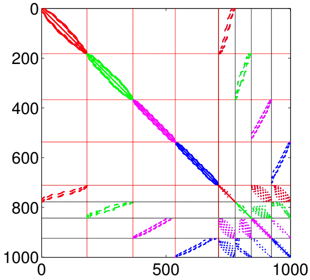


Figure: An example ($p = 4$). Different colors $\rightarrow$ different subdomains.

Spectral projectors and DD (I)

Contour integral spectral projector

- Let Γ be a smooth, counter-clockwise oriented curve such that $\{\lambda_1, \ldots, \lambda_r\}$ lie inside Γ and $\{\lambda_{r+1}, \ldots, \lambda_n\} \notin \Gamma$.
- Then,

$$\mathcal{P} = \frac{-1}{2i\pi} \int_\Gamma (A - \zeta I)^{-1} d\zeta = XX^T, \quad \text{with } X = [x^{(1)}, \ldots, x^{(r)}].$$

Spectral projectors and DD (I)

Contour integral spectral projector

- Let Γ be a smooth, counter-clockwise oriented curve such that $\{\lambda_1,\ldots,\lambda_r\}$ lie inside Γ and $\{\lambda_{r+1},\ldots,\lambda_n\} \notin \Gamma$.
- Then,

$$\mathcal{P} = \frac{-1}{2i\pi}\int_\Gamma (A-\zeta I)^{-1} d\zeta = XX^T, \quad \text{with } X = [x^{(1)},\ldots,x^{(r)}].$$

$(A-\zeta I)^{-1}$ in a domain decomposition framework

$$(A-\zeta I)^{-1} = \begin{pmatrix} (B-\zeta I)^{-1} + F(\zeta)S(\zeta)^{-1}F(\zeta)^T & -F(\zeta)S(\zeta)^{-1} \\ -S(\zeta)^{-1}F(\zeta)^T & S(\zeta)^{-1} \end{pmatrix},$$

where

$$F(\zeta) = (B-\zeta I)^{-1}E, \qquad S(\zeta) = C - \zeta I - E^T(B-\zeta I)^{-1}E.$$

Spectral projectors and DD (II)

Recall,

$$(A-\zeta I)^{-1} = \begin{pmatrix} (B-\zeta I)^{-1} + F(\zeta)S(\zeta)^{-1}F(\zeta)^T & -F(\zeta)S(\zeta)^{-1} \\ -S(\zeta)^{-1}F(\zeta)^T & S(\zeta)^{-1} \end{pmatrix}.$$

Then,

$$\mathcal{P} = \frac{-1}{2i\pi}\int_\Gamma (A-\zeta I)^{-1} d\zeta \equiv \begin{pmatrix} \mathcal{H} & -\mathcal{W} \\ -\mathcal{W}^T & \mathcal{G} \end{pmatrix},$$

Spectral projectors and DD (II)

Recall,

$$(A-\zeta I)^{-1} = \begin{pmatrix} (B-\zeta I)^{-1} + F(\zeta)S(\zeta)^{-1}F(\zeta)^T & -F(\zeta)S(\zeta)^{-1} \\ -S(\zeta)^{-1}F(\zeta)^T & S(\zeta)^{-1} \end{pmatrix}.$$

Then,

$$\mathcal{P} = \frac{-1}{2i\pi}\int_\Gamma (A-\zeta I)^{-1} d\zeta \equiv \begin{pmatrix} \mathcal{H} & -\mathcal{W} \\ -\mathcal{W}^T & \mathcal{G} \end{pmatrix},$$

with

$$\begin{cases} \mathcal{H} = \dfrac{-1}{2i\pi}\displaystyle\int_\Gamma [(B-\zeta I)^{-1} + F(\zeta)S(\zeta)^{-1}F(\zeta)^T] d\zeta \\ \mathcal{G} = \dfrac{-1}{2i\pi}\displaystyle\int_\Gamma S(\zeta)^{-1} d\zeta \\ \mathcal{W} = \dfrac{-1}{2i\pi}\displaystyle\int_\Gamma F(\zeta)S(\zeta)^{-1} d\zeta. \end{cases}$$

FEAST+DD $\rightarrow$ DD-FP (I)

Let V be a set of $\hat{r} \geq r$ vectors to multiply $\mathcal{P}$

$$\mathcal{P}\begin{pmatrix} V_u \\ V_s \end{pmatrix} = \begin{pmatrix} \mathcal{H}V_u - \mathcal{W}V_s \\ -\mathcal{W}^T V_u + \mathcal{G}V_s \end{pmatrix} \equiv \begin{pmatrix} Z_u \\ Z_s \end{pmatrix}, \textit{with}$$

$$\begin{cases} Z_u = \dfrac{-1}{2i\pi}\displaystyle\int_\Gamma (B-\zeta I)^{-1} V_u d\zeta - \dfrac{-1}{2i\pi}\displaystyle\int_\Gamma F(\zeta)S(\zeta)^{-1}[V_s - F(\zeta)^T V_u]d\zeta \\ Z_s = \dfrac{-1}{2i\pi}\displaystyle\int_\Gamma S(\zeta)^{-1}[V_s - F(\zeta)^T V_u]d\zeta. \end{cases}$$

FEAST+DD → DD-FP (I)

Let V be a set of $\hat{r} \geq r$ vectors to multiply $\mathcal{P}$

$$\mathcal{P}\begin{pmatrix} V_u \\ V_s \end{pmatrix} = \begin{pmatrix} \mathcal{H}V_u - \mathcal{W}V_s \\ -\mathcal{W}^T V_u + \mathcal{G}V_s \end{pmatrix} \equiv \begin{pmatrix} Z_u \\ Z_s \end{pmatrix}, \textit{with}$$

$$\begin{cases} Z_u = \dfrac{-1}{2i\pi}\displaystyle\int_\Gamma (B-\zeta I)^{-1} V_u d\zeta - \dfrac{-1}{2i\pi}\displaystyle\int_\Gamma F(\zeta)S(\zeta)^{-1}[V_s - F(\zeta)^T V_u]d\zeta \\ Z_s = \dfrac{-1}{2i\pi}\displaystyle\int_\Gamma S(\zeta)^{-1}[V_s - F(\zeta)^T V_u]d\zeta. \end{cases}$$

In practice:

- A quadrature rule will be used.
- If $\zeta_j = \bar{\zeta}_{j+N_c},\ j = 1, \ldots, N_c$,

$$B - \zeta_j I = \overline{B - \zeta_{j+N_c} I}, \quad S(\zeta_j) = \overline{S(\zeta_{j+N_c})}, \quad j = 1, \ldots, N_c.$$

FEAST+DD $\rightarrow$ DD-FP (II)

DD-FP

while not converged **do**
  for $j = 1$ to N_c **do**
    $W_u := (B - \zeta_j I)^{-1} V_u$ (*local*)
    $W_s := V_s - E^T W_u$ (*local*)
    $W_s := S(\zeta_j)^{-1} W_s$; $\tilde{Z}_s := \tilde{Z}_s - \Re\{\omega_j W_s\}$ (*distributed*)
    $W_u := W_u - (B - \zeta_j)^{-1} E W_s$; $\tilde{Z}_u := \tilde{Z}_u - \Re\{\omega_j W_u\}$ (*local*)
  end for
  Perform RR projection on $\tilde{Z} = [\tilde{Z}_u^T, \tilde{Z}_s^T]^T$ and update $V = [V_u^T, V_s^T]^T$.
end while

Practical considerations

- For each ζ_j: Two products with $F(\zeta_j)$ + one solve with $S(\zeta_j)$.
- Only the real part need be retained.
- The procedure can be repeated with an updated V.
- Equivalent to FEAST using a domain decomposition linear system solver.

Partial integration of the resolvent

$$\mathcal{P} = \frac{-1}{2i\pi}\int_\Gamma (A-\zeta I)^{-1} d\zeta = \quad [\mathcal{P}_1, \mathcal{P}_2] \quad \equiv \begin{pmatrix} * & -\mathcal{W} \\ * & \mathcal{G} \end{pmatrix},$$

$$\mathcal{G} = \frac{-1}{2i\pi}\int_\Gamma S(\zeta)^{-1} d\zeta, \qquad -\mathcal{W} = \frac{1}{2i\pi}\int_\Gamma F(\zeta)S(\zeta)^{-1} d\zeta.$$

We can recover the eigenpairs of A from $\mathcal{P}_2$.

Partial integration of the resolvent

$$\mathcal{P} = \frac{-1}{2i\pi}\int_\Gamma (A-\zeta I)^{-1} d\zeta = \quad [\mathcal{P}_1, \mathcal{P}_2] \quad \equiv \begin{pmatrix} * & -\mathcal{W} \\ * & \mathcal{G} \end{pmatrix},$$

$$\mathcal{G} = \frac{-1}{2i\pi}\int_\Gamma S(\zeta)^{-1} d\zeta, \qquad -\mathcal{W} = \frac{1}{2i\pi}\int_\Gamma F(\zeta) S(\zeta)^{-1} d\zeta.$$

We can recover the eigenpairs of A from $\mathcal{P}_2$.

$$\mathcal{P} = XX^T, \; X = \begin{pmatrix} U \\ Y \end{pmatrix} \quad \rightarrow \quad \mathcal{P} = \begin{pmatrix} * & UY^T \\ * & YY^T \end{pmatrix}$$

- Just capture the range of $\mathcal{P}_2 = XY^T$.
- Lanczos on $\mathcal{P}_2\mathcal{P}_2^T \rightarrow$ needs no estimation of r, sequential, doubles the work.
- Alternative: Multiply $\mathcal{P}_2$ by a matrix R with $\hat{r} \geq r$ columns.

Formulating an algorithm (DD-PP)

Let R be a set of $\hat{r} \geq r$ vectors

1. Compute
$$\mathcal{G}R = \frac{-1}{2i\pi} \int_{\Gamma} S(\zeta)^{-1} R d\zeta = Y(Y^T R),$$
$$-\mathcal{W}R = \frac{1}{2i\pi} \int_{\Gamma} F(\zeta) S(\zeta)^{-1} R d\zeta = U(Y^T R).$$
2. Perform a Rayleigh-Ritz projection to extract $(\lambda_1, x^{(1)}), \ldots, (\lambda_r, x^{(r)})$.

Considerations

- Only a part of the spectral projector is needed.
- It is necessary to have $r \leq \mathrm{size}(S(.))$.
- How to compute $\mathcal{G}R$ and $-\mathcal{W}R$ in practice?

DD-PP with numerical integration

Numerical integration of $\mathcal{G}R,\ -\mathcal{W}R$

$$\tilde{\mathcal{G}}R = \sum_{j=1}^{2N_c} \omega_j S(\zeta_j)^{-1}R, \quad -\tilde{\mathcal{W}}R = \sum_{j=1}^{2N_c} \omega_j F(\zeta_j) S(\zeta_j)^{-1}R.$$

for $j = 1$ to N_c **do**

$\quad \cancel{W_u := (B - \zeta_j I)^{-1} V_u}$ $\quad (local)$

$\quad \cancel{W_s := V_s - E^T W_u}$ $\quad (local)$

$\quad W_s := S(\zeta_j)^{-1}R; \quad \tilde{Z}_s := \tilde{Z}_s - \Re\{\omega_j W_s\}$ $\quad (distributed)$

$\quad W_u := \cancel{W_u} - (B - \zeta_j)^{-1} E W_s; \quad \tilde{Z}_u := \tilde{Z}_u - \Re\{\omega_j W_u\}$ $\quad (local)$

end for

Perform RR projection on $\tilde{Z} = [\tilde{Z}_u^T, \tilde{Z}_s^T]^T$.

- Equivalent to DD-FP if we set $V_u = 0$ in the latter.
- For each ζ_j: One product with $F(\zeta_j)$ and one solve with $S(\zeta_j)$.

A comparison of DD-PP and DD-FP

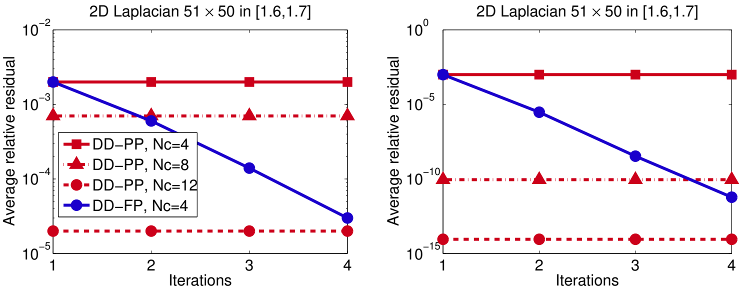


Figure: Left: $\hat{r} = r$. Right: $\hat{r} = 2r$.

Parallel implementation

$$S(\zeta) = \begin{pmatrix} S_1(\zeta) & E_{12} & \dots & E_{1p} \\ E_{21} & S_2(\zeta) & \dots & E_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ E_{p1} & E_{p2} & \dots & S_p(\zeta) \end{pmatrix}.$$

- $S_i(\zeta) = C_i - \zeta I - E_i^T (B_i - \zeta I)^{-1} E_i$.
- $E_{ik} \neq 0$ iff $i,\ k$ are neighboring subdomains.
- 2D discretizations: form and factorize $S(\zeta)$.
- 3D discretizations: Preconditioned GMRES(250).

Solving linear systems with $B - \zeta I$

- B is block-diagonal $\rightarrow$ solves with $B - \zeta I$ are embarrassingly parallel among the subdomains.

The MV product with $S(\zeta)$

$$S(\zeta)v = (C - \zeta I)v - E^T (B - \zeta I)^{-1} Ev$$

can be accomplished as:

1. Compute $E^T (B - \zeta I)^{-1} Ev$.
2. Distribute (exchange) the necessary parts of v and perform $(C - \zeta I)v$.
3. Subtract the vector in 1) from the vector in 2).

Contents

Implementation and computing environment (I)

Hardware

- ITASCA HP Linux cluster at Minnesota Supercomputing Inst.
- 1,091 HP ProLiant BL280c G6 blade servers, each with two-socket, quad-core 2.8 GHz Intel Xeon X5560 “Nehalem EP” (24 GB per node)
- 40-gigabit QDR InfiniBand (IB) interconnect

Software

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

Implementation and computing environment (II)

Comparisons

- We compare three schemes:
  - DD-PP (partial projector)
  - DD-FP (FEAST with DD linear system solver)
  - CI-M (FEAST with MUMPS)

Test matrices

The Laplace eigenvalue problem

$$-\Delta u = \lambda u$$

- Dirichlet boundary conditions.
- Second order centered finite differences.
- $n = n_x n_y n_z$.

2D Laplacians

Table: Avg. time spent on a single quadrature node when approximating the eigenvalues $\lambda_{1001}, \ldots, \lambda_{1200}$ and associated eigenvectors by DD-PP, and speedup over DD-FP.

	$p = 8$		$p = 16$		$p = 32$	
	DD-PP	(x)DD-FP	DD-PP	(x)DD-FP	DD-PP	(x)DD-FP
$n = 500^2$						
$\hat{r} = r + 10$	9.45	**1.45**	6.77	**1.31**	5.25	**1.20**
$\hat{r} = 3r/2 + 10$	13.5	**1.47**	9.65	**1.32**	7.59	**1.19**
$\hat{r} = 2r + 10$	18.1	**1.44**	12.9	**1.32**	10.0	**1.23**
$n = 1000^2$						
$\hat{r} = r + 10$	41.8	**1.51**	25.3	**1.41**	17.9	**1.29**
$\hat{r} = 3r/2 + 10$	59.7	**1.59**	36.0	**1.40**	25.5	**1.30**
$\hat{r} = 2r + 10$	79.1	**1.62**	68.1	**1.47**	34.1	**1.28**
$n = 1500^2$						
$\hat{r} = r + 10$	100.8	**1.41**	65.2	**1.38**	39.9	**1.10**
$\hat{r} = 3r/2 + 10$	144.2	**1.39**	93.1	**1.40**	57.6	**1.12**
$\hat{r} = 2r + 10$	192.7	**1.49**	124.5	**1.44**	76.0	**1.11**

Max/Avg errors (cont. from previous slide)

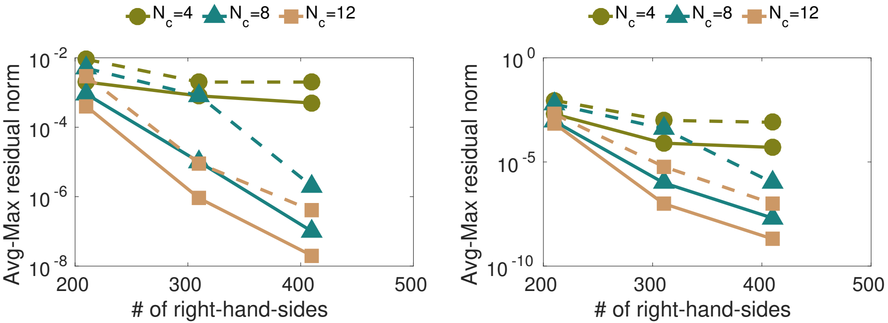


Figure: Maximum (dashed) and average (solid) residual norm.

Impact of DD direct solver for the $n = 1500^2$ Laplacian

Table: Time elapsed to compute eigenvalues $\lambda_{1001}, \ldots, \lambda_{1200}$ and corresponding eigenvectors by the CI-M and DD-FP schemes. "Its" $\rightarrow$ # of outer iter/s.

		$p = 64$		$p = 128$		$p = 256$	
	Its	CI-M	DD-FP	CI-M	DD-FP	CI-M	DD-FP
$N_c = 2$							
$\hat{r} = 3r/2 + 10$	9	3,922.7	2,280.6	2,624.3	1,242.4	1,911.2	859.5
$\hat{r} = 2r + 10$	5	2,863.2	1,764.5	1,877.7	998.5	1,255.5	615.3
$N_c = 4$							
$\hat{r} = 3r/2 + 10$	5	4,181.5	2,357.0	2,815.7	1,280.2	1,874.1	877.5
$\hat{r} = 2r + 10$	4	4,330.3	2,571.4	2,869.5	1,462.9	2,023.2	1,036.2
$N_c = 6$							
$\hat{r} = 3r/2 + 10$	3	3,710.3	2,068.2	2,504.1	1,122.1	1,790.8	766.5
$\hat{r} = 2r + 10$	3	4,774.8	2,798.5	3,177.7	1,595.2	2,743.6	1,125.1

Practical considerations for iterative solvers, $n = 150^3$

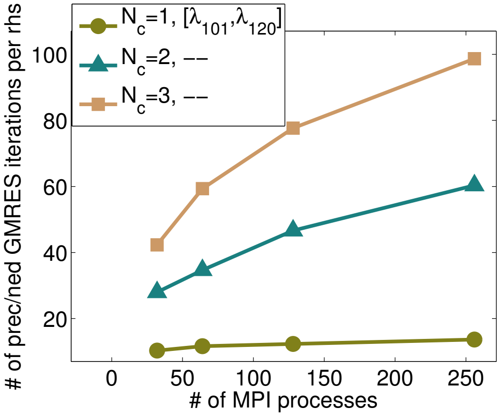


Figure: # of preconditioned GMRES iterations for a random right-hand side. Preconditioner: inexact factorization of $B - \zeta I$ (drop tolerance: 1e-4) and dropping in $S(\zeta)$ (drop tolerance: 1e-2).

A comparison of DD-FP and CI-M for $n = 150^3$

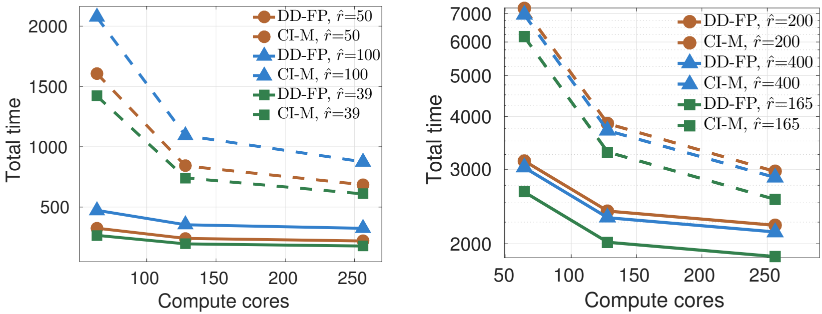


Figure: Only one quadrature node ($N_c = 1$) is used.

Summary and future work

Summary

- We considered domain decomposition in the context of contour integration approaches.
  - As a linear system solver.
  - As a tool to derive new algorithms (partial integration of the resolvent).
- Results performed on distributed computing environments show that these approaches can improve the performance of FEAST.

Considerations

- More work is needed in the direction of iterative solvers in FEAST.
- Additional levels of parallelism can be exploited in a straightforward way.
- Combine the partial projector with Krylov techniques (no need to estimate r a-priori).

About domain decomposition and FEAST

- 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". Preprint, 2016.

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

Analysis of DD-PP

Let

$$(A-\zeta I)^{-1} = \sum_{i=1}^{n} \frac{x^{(i)}(x^{(i)})^T}{\lambda_i - \zeta}, \text{ and let } x^{(i)} = \begin{pmatrix} u^{(i)} \\ y^{(i)} \end{pmatrix}.$$

Then,

$$\sum_{j=1}^{2N_c} \omega_j (A-\zeta_j I)^{-1} = \sum_{i=1}^{n} \rho(\lambda_i) \begin{bmatrix} u^{(i)}(u^{(i)})^T & u^{(i)}(y^{(i)})^T \\ y^{(i)}(u^{(i)})^T & y^{(i)}(y^{(i)})^T \end{bmatrix}.$$

Moreover,

$$\sum_{j=1}^{2N_c} \omega_j (A-\zeta_j I)^{-1} = \begin{pmatrix} * & \sum_{j=1}^{2N_c} \omega_j F(\zeta_j) S(\zeta_j)^{-1} \\ * & \sum_{j=1}^{2N_c} \omega_j S(\zeta_j)^{-1} \end{pmatrix}.$$

Analysis of DD-PP (cont.)

$$\sum_{j=1}^{2N_c} \omega_j S(\zeta_j)^{-1} = \sum_{i=1}^{n} \rho(\lambda_i) y^{(i)} (y^{(i)})^T$$

$$\sum_{j=1}^{2N_c} \omega_j F(\zeta_j) S(\zeta_j)^{-1} = \sum_{i=1}^{n} \rho(\lambda_i) u^{(i)} (y^{(i)})^T.$$

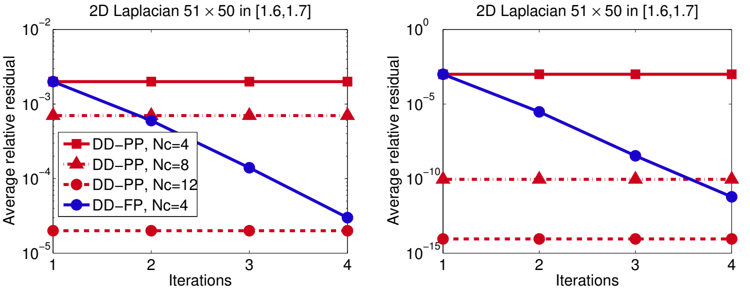


Figure: Left: $\hat{r} = r$. Right: $\hat{r} = 2r$.