Spectral Schur complement techniques for eigenvalue problems

Vassilis Kalantzis
joint work with
Ruipeng Li and Yousef Saad

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

SIAM CSE 2015, 03-14-2015

Contents

Introduction

The symmetric eigenvalue problem

$$Ax = \lambda x$$

where $A \in \mathbb{R}^{n\times n}$, $x \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$. A pair (λ, x) is an *eigenpair* of A.

Focus

1. Find all (λ, x) pairs inside the interval $[\alpha, \beta]$ where $\alpha,\ \beta \in \mathbb{R}$ and $\lambda_1 \leq \alpha,\ \beta \leq \lambda_n$.
2. Given a shift $\zeta \in \mathbb{R}$, find the k (λ, x) pairs closest to ζ.

Roadmap

In this talk

- Propose a Domain Decomposition-type approach.
- Focus on solving the problem along the interface.
- Quadratically convergent Newton scheme.
- No estimation of # eigenvalues inside the interval.
- Parallel implementation.

Contents

Partitioning of the domain (Metis, Scotch,...)

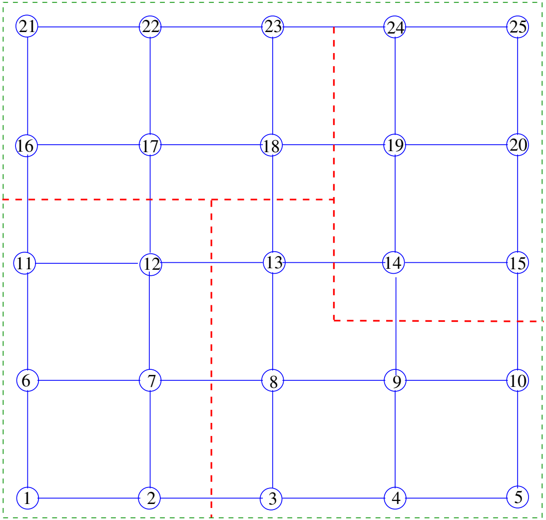
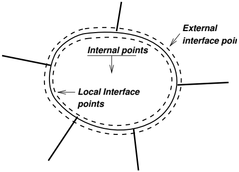
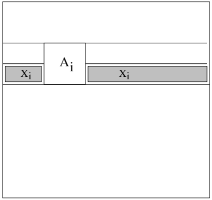


Figure: An edge-separator (vertex-based partitioning)

The local viewpoint – assume p partitions

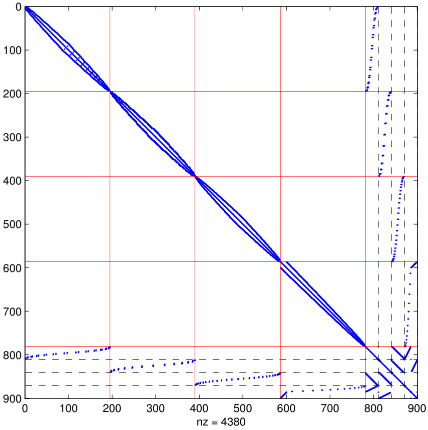




Stack interior variables $u_1, u_2, \ldots, u_p$ into u, then interface variables y,

$$\begin{pmatrix} B_1 & & & & E_1 \\ & B_2 & & & E_2 \\ & & \ddots & & \vdots \\ & & & B_p & E_p \\ E_1^\top & E_2^\top & \cdots & E_p^\top & C \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \\ y \end{pmatrix} = \lambda \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \\ y \end{pmatrix}$$

Notation:



Write as

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

The spectral Schur complement

- Eliminating the u_i's we get

$$\begin{pmatrix} S_1(\lambda) & E_{12} & \cdots & E_{1p} \\ E_{21} & S_2(\lambda) & \cdots & E_{2p} \\ \vdots & & \ddots & \vdots \\ E_{p1} & E_{p2} & \cdots & S_p(\lambda) \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{pmatrix} = 0$$

with $S_i(\lambda) = C_i - \lambda I - E_i^\top (B_i - \lambda I)^{-1} E_i$.

- Interface problem (non-linear):

$$S(\lambda) y(\lambda) = 0.$$

- Top parts can be recovered as $u_i = -(B_i - \lambda I)^{-1} E_i y(\lambda)$.

Contents

Spectral Schur complement revisited

The problem

- Find $\sigma \in \mathbb{R}$ such that

$$\mu(\sigma) = 0,$$

where $\mu(\sigma)$ denotes the smallest ($|.|$) eig of $S(\sigma)$.

Reformulating

- We can treat $\mu(\sigma)$ as a function $\rightarrow$ root-finding problem.
- The function $\mu(\sigma)$ is analytic for any $\sigma \notin \Lambda(B)$ with

$$\frac{d\mu(\sigma)}{d\sigma} = \frac{(S'(\sigma)y(\sigma), y(\sigma))}{(y(\sigma), y(\sigma))} = -1 - \frac{\|(B - \sigma I)^{-1} E y(\sigma)\|_2^2}{\|y(\sigma)\|_2^2}.$$

Basic algorithm - Newton's scheme

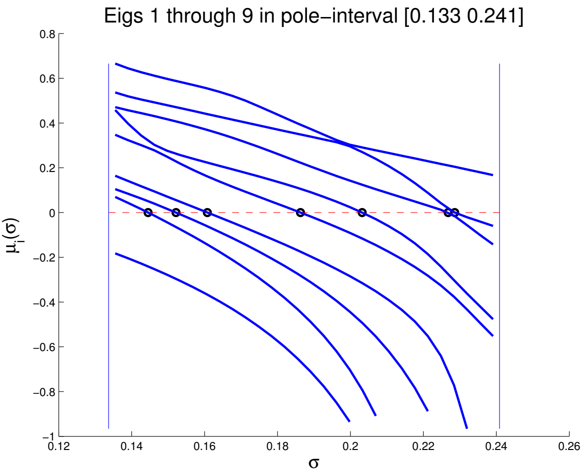
"Chasing" a single eigenvalue

We can formulate a Newton-based algorithm.

Algorithm 3.1

1: Select σ
2: **repeat**
3: Compute $\mu(\sigma) =$ Smallest eigenvalue in modulus of $S(\sigma)$
4: along with associated unit eigenvector $y(\sigma)$
5: Set $\eta := \|(B - \sigma I)^{-1} E y(\sigma)\|_2$
6: Set $\sigma := \sigma + \mu(\sigma)/(1 + \eta^2)$
7: **until** $|\mu(\sigma)| \leq \mathrm{tol}$

Short illustration - eigenvalue branches of $S(\sigma)$ between two poles



Quadratic convergence

The Newton scheme is quadratically convergent. The second derivative is

$$\mu'' = y^\top S'' y + 2y'^\top (S - \mu I) y'$$

and

$$S'' = 2E^\top (B - \sigma I)^{-3} E.$$

Residual of the approximation

It can be shown that

$$\|(A - \sigma I)\hat{x}(\sigma)\| = |\mu(\sigma)|$$

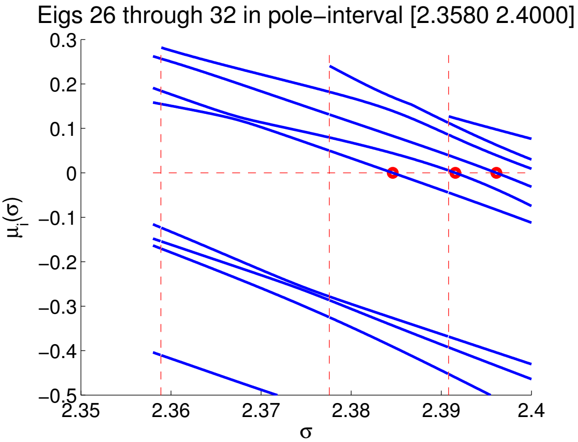
where $\hat{x}(\sigma) = [-(B - \sigma I)^{-1} E y(\sigma); y(\sigma)]$ is the approximate eigenvector.

Connection with RQ

The Newton update also is the Rayleigh Quotient,

$$\sigma = \rho(A, \hat{x}(\sigma)).$$

Eigenvalue branches of $S(\sigma)$ across the poles

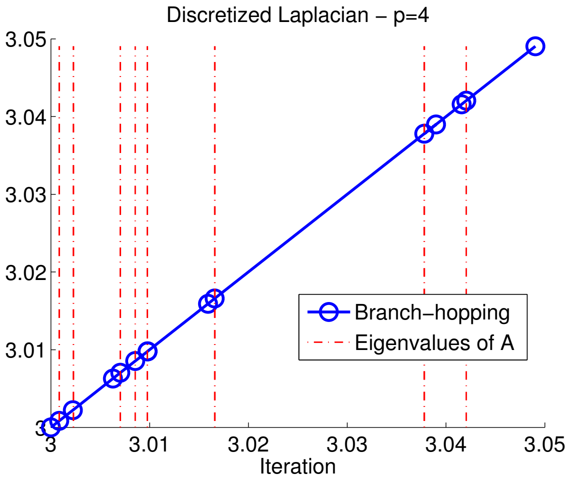


A branch-hopping scheme

Algorithm 3.2 – "Chasing" more than one eigenvalues

Given a, b. Select $\sigma = a$
while $\sigma < b$ **do**
    Compute $S(\sigma)\mu(\sigma) = \mu(\sigma)y(\sigma)$
    if $|\mu(\sigma)| \leq \mathrm{tol}$ **then**
        Obtain $\mu(\sigma)$ = smallest positive eigenvalue of $S(\sigma)$
    end if
    Compute derivative and update σ as in Algorithm 3.1
end while

An example of the Branch-hopping scheme



Effects of p in convergence

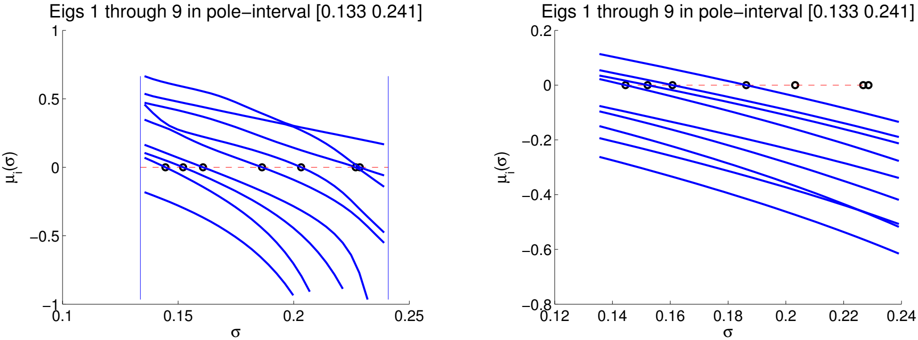


Figure: Eigenvalue branches $\mu_1(\sigma), \ldots, \mu_9(\sigma)$ in $[0.133, 0.241]$ with $n_x = 33,\ n_y = 23$ and $n_z = 1$. Left subfigure $p = 4$, right subfigure $p = 16$.

Implementation aspects

Evaluation of $S(\sigma)\mu(\sigma) = \mu(\sigma)y(\sigma)$

- For any σ we just need one or two eigenvalues of $S(\sigma)$.
- We can use "Inverse-Iteration" type approaches.
- In this talk we use the Lanczos algorithm with partial re/tion.
- Lanczos has the ability to compute inertia of $S(\sigma)$.

Parallel implemenation

- The initial interval can be broken in multiple parts.
- In each subinterval we can compute $\mu(\sigma)$ by using the DD framework.
- Single-level partitioning – One node per sub-domain.
- Implemented in C++ using the PETSc framework.

Contents

Numerical experiments

Some details

- Tests performed on Itasca Linux cluster @ MSI.
- Each node is a two-socket, quad-core 2.8 GHz Intel Xeon X5560 "Nehalem EP" with 24 GB of system memory.
- Interconnection : 40-gigabit QDR InfiniBand (IB).

The model problem

- Tests on 3-D dicretized Laplacians (7pt. st. – FD).
- We use n_x, n_y, n_z to denote the three dimensions.
- tol set to $1 \times e^{-12}$.

Convergence of the proposed scheme

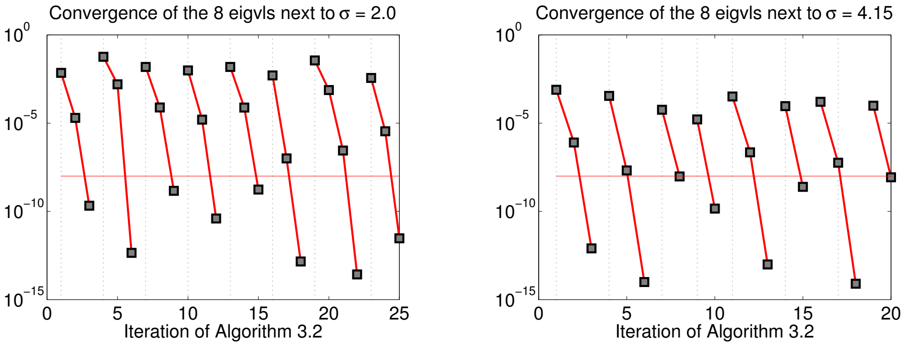


Figure: Rel. res. for a few consecutive eigenvalues. Left subfigure $n_x = 20,\ n_y = 20$ and $n_z = 20$, right subfigure $n_x = 40,\ n_y = 20$ and $n_z = 20$.

Computing eigenvalues inside an interval

		$[\alpha,\beta]$:=[0,0.5]			$[\alpha,\beta]$:=[2,2.2]			$[\alpha,\beta]$:=[4.1,4.2]		
		#Eigvls	It	Avg. Lan	#Eigvls	It	Avg. Lan	#Eigvls	It	Avg. Lan
n = 4000										
# of subdomains (p)	2		41	169		85	210		124	338
	4	15	26	197	39	74	367	46	80	652
	8		32	284		60	551		70	1020
	16		32	255		55	721		70	1480
n = 8000										
# of subdomains (p)	2		60	176		76	337		90	517
	4	35	43	194	81	65	573	117	43	967
	8		35	279		58	778		38	1388
	16		39	281		48	1037		37	1900
n = 16000										
# of subdomains (p)	2		210	166		342	406		424	735
	4	73	170	199	156	292	746	217	314	1502
	8		154	294		273	1194		310	2526
	16		138	360		241	1694		300	3548

Computing $k = 1$ and $k = 5$ eigenvalues closest to ζ

			$\zeta = 0.0$			$\zeta = 0.05$		
	(p, k)	s	Time(s)	It	Lan	Time(s)	It	Lan
$\mathbf{n = 60^3}$								
	(8,1)	22078	19.1	4	65	31.2	3	312
	(8,5)	$\gg$	105.6	14	180	139.2	12	372
	(16,1)	35702	3.9	4	65	12.0	4	474
	(16,5)	$\gg$	29.3	14	228	56.1	12	510
	(32,1)	47200	**1.0**	**4**	**80**	**5.5**	**3**	**534**
	(32,5)	$\gg$	**10.7**	**12**	**480**	**25.7**	**11**	**610**
$\mathbf{n = 70^3}$								
	(8,1)	30077	38.1	4	88	73.2	3	350
	(8,5)	$\gg$	223.0	14	157	352.1	15	374
	(16,1)	49596	12.6	4	115	46.9	4	600
	(16,5)	$\gg$	87.3	15	223	153.8	11	702
	(32,1)	65647	**2.7**	**4**	**135**	**21.7**	**3**	**750**
	(32,5)	$\gg$	**23**	**13**	**282**	**58.7**	**10**	**864**

Computing $k = 1$ and $k = 5$ eigenvalues closest to ζ

	(p, k)	s	$\zeta = 1.0$ Time(s)	$\zeta = 1.0$ It	$\zeta = 1.0$ Lan	$\zeta = 1.5$ Time(s)	$\zeta = 1.5$ It	$\zeta = 1.5$ Lan
$\mathbf{n = 40^3}$								
	(2,1)	3280	**23.3**	**3**	**478**	**29.4**	**2**	**758**
	(2,5)	$\gg$	**117.1**	**15**	**486**	**147.5**	**10**	**781**
	(4,1)	6466	33.2	3	850	65.4	2	1200
	(4,5)	$\gg$	150.1	15	855	331.9	11	1242
	(8,1)	9579	45.3	3	1100	167.7	2	1700
	(8,5)	$\gg$	220.5	15	1112	724.2	10	1731
$\mathbf{n = 50^3}$								
	(2,1)	5100	75.1	3	680	150.2	3	1014
	(2,5)	$\gg$	348.2	15	691	720.1	14	1025
	(4,1)	10148	**50.7**	**3**	**950**	**78.4**	**3**	**1200**
	(4,5)	$\gg$	**235.3**	**13**	**978**	**342.2**	**14**	**1257**
	(8,1)	14795	81.1	3	1200	163.1	3	1600
	(8,5)	$\gg$	402.8	14	1226	723.3	13	1654

Contents

Conclusion

In this talk

- The DD scheme presented focuses solely on the interface problem.
- Eigenvalue branches of the SSC $\rightarrow$ Newton's method.
- Parallelism can be exploited in two different levels.
- Ultimately, k eigenpairs are computed at the cost of one.

Considerations

- Exploit previous information in the form of a subspace.
- Use other than Lanczos method for computing interior eigenvalues.
- Comparisons against state-of-the-art methods.