Solving large scale eigenvalue problems

Lecture 12, May 16, 2018: Rayleigh quotient minimization

http://people.inf.ethz.ch/arbenz/ewp/

Peter Arbenz
Computer Science Department, ETH Zürich
E-mail: arbenz@inf.ethz.ch

Survey of today's lecture

- Rayleigh quotient minimization
- Method of steepest descent
- Conjugate gradient algorithm
- Preconditioned conjugate gradient algorithm
- Locally optimal PCG (LOPCG)
- Locally optimal block PCG (LOBPCG)

Rayleigh quotient

We consider symmetric/Hermitian eigenvalue problem

$$A\boldsymbol{x} = \lambda M\boldsymbol{x}, \qquad A = A^*, \quad M = M^* > 0.$$

The Rayleigh quotient is defined as

$$\rho(\boldsymbol{x}) = \frac{\boldsymbol{x}^* A\boldsymbol{x}}{\boldsymbol{x}^* M\boldsymbol{x}}.$$

We want to exploit that

$$\lambda_1 = \min_{\boldsymbol{x}\neq\boldsymbol{0}} \rho(\boldsymbol{x}) \tag{1}$$

and

$$\lambda_k = \min_{S_k\subset\mathbb{R}^n} \max_{\substack{\boldsymbol{x}\neq\boldsymbol{0}\\ \boldsymbol{x}\in S_k}} \rho(\boldsymbol{x})$$

where S_k is a subspace of dimension k.

Rayleigh quotient minimization

Want to construct sequence $\{\boldsymbol{x}_k\}_{k=0,1,\ldots}$ such that

$$\rho(\boldsymbol{x}_{k+1}) < \rho(\boldsymbol{x}_k) \quad \text{for all } k.$$

The hope is that the sequence $\{\rho(\boldsymbol{x}_k)\}$ converges to λ_1 and by consequence the vector sequence $\{\boldsymbol{x}_k\}$ towards the corresponding eigenvector.
Procedure: For any given $\boldsymbol{x}_k$ we choose a **search direction** $\boldsymbol{p}_k$ s.t.

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \delta_k \boldsymbol{p}_k.$$

Parameter δ_k determined s.t. Rayleigh quotient of $\boldsymbol{x}_{k+1}$ is minimal:

$$\rho(\boldsymbol{x}_{k+1}) = \min_{\delta} \rho(\boldsymbol{x}_k + \delta \boldsymbol{p}_k).$$

Rayleigh quotient minimization (cont.)

$$\rho(\boldsymbol{x}_k + \delta \boldsymbol{p}_k) = \frac{\boldsymbol{x}_k^* A \boldsymbol{x}_k + 2\delta \boldsymbol{x}_k^* A \boldsymbol{p}_k + \delta^2 \boldsymbol{p}_k^* A \boldsymbol{p}_k}{\boldsymbol{x}_k^* M \boldsymbol{x}_k + 2\delta \boldsymbol{x}_k^* M \boldsymbol{p}_k + \delta^2 \boldsymbol{p}_k^* M \boldsymbol{p}_k}$$

$$= \frac{\begin{pmatrix} 1 \\ \delta \end{pmatrix}^* \begin{bmatrix} \boldsymbol{x}_k^* A \boldsymbol{x}_k & \boldsymbol{x}_k^* A \boldsymbol{p}_k \\ \boldsymbol{p}_k^* A \boldsymbol{x}_k & \boldsymbol{p}_k^* A \boldsymbol{p}_k \end{bmatrix} \begin{pmatrix} 1 \\ \delta \end{pmatrix}}{\begin{pmatrix} 1 \\ \delta \end{pmatrix}^* \begin{bmatrix} \boldsymbol{x}_k^* M \boldsymbol{x}_k & \boldsymbol{x}_k^* M \boldsymbol{p}_k \\ \boldsymbol{p}_k^* M \boldsymbol{x}_k & \boldsymbol{p}_k^* M \boldsymbol{p}_k \end{bmatrix} \begin{pmatrix} 1 \\ \delta \end{pmatrix}}.$$

This is Rayleigh quotient associated with eigenvalue problem

$$\begin{bmatrix} \boldsymbol{x}_k^* A \boldsymbol{x}_k & \boldsymbol{x}_k^* A \boldsymbol{p}_k \\ \boldsymbol{p}_k^* A \boldsymbol{x}_k & \boldsymbol{p}_k^* A \boldsymbol{p}_k \end{bmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \lambda \begin{bmatrix} \boldsymbol{x}_k^* M \boldsymbol{x}_k & \boldsymbol{x}_k^* M \boldsymbol{p}_k \\ \boldsymbol{p}_k^* M \boldsymbol{x}_k & \boldsymbol{p}_k^* M \boldsymbol{p}_k \end{bmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \quad (2)$$

Smaller of the two eigenvalues of (2) is the searched value $\rho_{k+1} := \rho(\boldsymbol{x}_{k+1})$ that minimizes the Rayleigh quotient.

Rayleigh quotient minimization (cont.)

We normalize corresponding eigenvector such that its first component equals one. (Is this always possible?)
Second component of eigenvector is $\delta = \delta_k$.
Second line of (2) gives

$$\boldsymbol{p}_k^* A(\boldsymbol{x}_k + \delta_k \boldsymbol{p}_k) = \rho_{k+1} \boldsymbol{p}_k^* M(\boldsymbol{x}_k + \delta_k \boldsymbol{p}_k)$$

or

$$\boldsymbol{p}_k^*(A - \rho_{k+1} M)(\boldsymbol{x}_k + \delta_k \boldsymbol{p}_k) = \boldsymbol{p}_k^* \boldsymbol{r}_{k+1} = 0. \qquad (3)$$

'Next' residual $\boldsymbol{r}_{k+1}$ is orthogonal to actual search direction $\boldsymbol{p}_k$.

How shall we choose the search directions $\boldsymbol{p}_k$?

Detour: steepest descent method for linear systems

We consider linear systems

$$Ax = b, \tag{4}$$

where A is SPD (or HPD). We define the functional

$$\varphi(x) \equiv \frac{1}{2}x^* Ax - x^* b + \frac{1}{2}b^* A^{-1} b = \frac{1}{2}(Ax - b)^* A^{-1}(Ax - b).$$

φ is minimized (actually zero) at the solution x_* of (4). The negative gradient of φ is

$$-\nabla\varphi(x) = b - Ax =: r(x). \tag{5}$$

This is the direction in which φ decreases the most. Clearly, $\nabla\varphi(x) \neq 0 \iff x \neq x_*$.

Steepest descent method for eigenvalue problem

We choose $\boldsymbol{p}_k$ to be the negative gradient of the Rayleigh quotient

$$\boldsymbol{p}_k = -\boldsymbol{g}_k = -\nabla\rho(\boldsymbol{x}_k) = -\frac{2}{\boldsymbol{x}_k^* M\boldsymbol{x}_k}(A\boldsymbol{x}_k - \rho(\boldsymbol{x}_k)M\boldsymbol{x}_k).$$

Since we only care about directions we can equivalently set

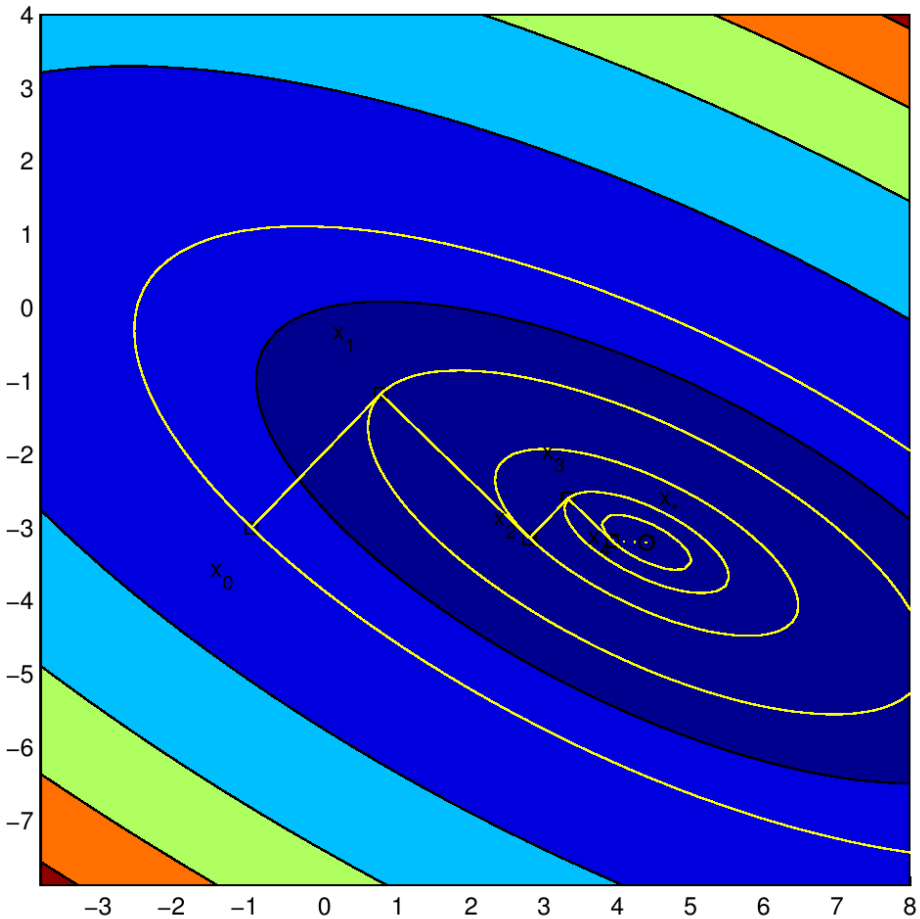
$$\boldsymbol{p}_k = \boldsymbol{r}_k = A\boldsymbol{x}_k - \rho_k M\boldsymbol{x}_k, \qquad \rho_k = \rho(\boldsymbol{x}_k).$$

With this choice of search direction we have from (3)

$$\boldsymbol{r}_k^* \boldsymbol{r}_{k+1} = 0. \tag{6}$$

The method of steepest descent often converges slowly, as for linear systems. This happens if the spectrum is very much spread out, i.e., if the condition number of A relative to M is big.

Slow convergence of steepest descent method



Picture: M. Gutknecht

Detour: Conjugate gradient algorithm for linear systems

As with linear systems of equations a remedy against the slow convergence of steepest descent are conjugate search directions. In the cg algorithm, we define search directions as

$$\boldsymbol{p}_k = -\boldsymbol{g}_k + \beta_k \boldsymbol{p}_{k-1}, \qquad k > 0. \tag{7}$$

where coefficient β_k is determined s.t. $\boldsymbol{p}_k$ and $\boldsymbol{p}_{k-1}$ are conjugate:

$$\boldsymbol{p}_k^* A \boldsymbol{p}_{k-1} = -\boldsymbol{g}_k^* A \boldsymbol{p}_{k-1} + \beta_k \boldsymbol{p}_{k-1}^* A \boldsymbol{p}_{k-1} = 0,$$

So,

$$\beta_k = \frac{\boldsymbol{g}_k^* A \boldsymbol{p}_{k-1}}{\boldsymbol{p}_{k-1}^* A \boldsymbol{p}_{k-1}} = \cdots = \frac{\boldsymbol{g}_k^* \boldsymbol{g}_k}{\boldsymbol{g}_{k-1}^* \boldsymbol{g}_{k-1}}. \tag{8}$$

One can show that $\boldsymbol{p}_k^* A \boldsymbol{p}_j = \boldsymbol{g}_k^* \boldsymbol{g}_j = 0$ for $j < k$.

The conjugate gradient algorithm

The conjugate gradient algorithm can be adapted to eigenvalue problems.

The idea is straightforward: consecutive search directions must satisfy $\boldsymbol{p}_k^* A \boldsymbol{p}_{k-1} = 0$.

The crucial difference to linear systems stems from the fact, that the functional that is to be minimized, i.e., the Rayleigh quotient, is not quadratic anymore. (E.g., there is no finite termination property.)

The gradient of $\rho(\boldsymbol{x})$ is

$$\boldsymbol{g} = \nabla\rho(\boldsymbol{x}_k) = \frac{2}{\boldsymbol{x}^* M \boldsymbol{x}}(A\boldsymbol{x} - \rho(\boldsymbol{x})M\boldsymbol{x}).$$

The conjugate gradient algorithm (cont.)

In the case of eigenvalue problems the different expressions for β_k in (7)–(8) are not equivalent anymore. We choose

$$\begin{cases} \boldsymbol{p}_0 = -\boldsymbol{g}_0, & k = 0, \\ \boldsymbol{p}_k = -\boldsymbol{g}_k + \dfrac{\boldsymbol{g}_k^* M \boldsymbol{g}_k}{\boldsymbol{g}_{k-1}^* M \boldsymbol{g}_{k-1}} \boldsymbol{p}_{k-1}, & k > 0, \end{cases} \tag{9}$$

which is the best choice according to Feng and Owen [1].

The above formulae is for the generalized eigenvalue problem $A\boldsymbol{x} = \lambda M \boldsymbol{x}$.

The Rayleigh quotient algorithm

1: Let $\boldsymbol{x}_0$ be a unit vector, $\|\boldsymbol{x}_0\|_M = 1$.
2: $\boldsymbol{v}_0 := A\boldsymbol{x}_0, \quad \boldsymbol{u}_0 := M\boldsymbol{x}_0, \quad \rho_0 := \dfrac{\boldsymbol{v}_0^* \boldsymbol{x}_0}{\boldsymbol{u}_0^* \boldsymbol{x}_0}, \quad \boldsymbol{g}_0 := 2(\boldsymbol{v}_0 - \rho_0 \boldsymbol{u}_0)$
3: **while** $\|\boldsymbol{g}_k\| > tol$ **do**
4: **if** $k = 1$ **then**
5: $\boldsymbol{p}_k := -\boldsymbol{g}_{k-1}$;
6: **else**
7: $\boldsymbol{p}_k := -\boldsymbol{g}_{k-1} + \dfrac{\boldsymbol{g}_{k-1}^* M \boldsymbol{g}_{k-1}}{\boldsymbol{g}_{k-2}^* M \boldsymbol{g}_{k-2}} \boldsymbol{p}_{k-1}$;
8: **end if**
9: Determine smallest Ritz value and associated Ritz vector $\boldsymbol{x}_k$ of (A, M) in $\mathcal{R}([\boldsymbol{x}_{k-1}, \boldsymbol{p}_k])$
10: $\boldsymbol{v}_k := A\boldsymbol{x}_k, \quad \boldsymbol{u}_k := M\boldsymbol{x}_k$
11: $\rho_k := \boldsymbol{x}_k^* \boldsymbol{v}_k / \boldsymbol{x}_k^* \boldsymbol{u}_k$
12: $\boldsymbol{g}_k := 2(\boldsymbol{v}_k - \rho_k \boldsymbol{u}_k)$
13: **end while**

Convergence

Construction of algorithm guarantees that $\rho(\boldsymbol{x}_{k+1}) < \rho(\boldsymbol{x}_k)$ unless $\boldsymbol{r}_k = \boldsymbol{0}$, in which case $\boldsymbol{x}_k$ is the searched eigenvector.

In general, i.e., if the initial vector $\boldsymbol{x}_0$ has a nonvanishing component in the direction of the 'smallest' eigenvector $\boldsymbol{u}_1$, convergence is toward the smallest eigenvalue λ_1.

Let

$$\boldsymbol{x}_k = \cos\vartheta_k \boldsymbol{u}_1 + \sin\vartheta_k \boldsymbol{z}_k =: \cos\vartheta_k \boldsymbol{u}_1 + \boldsymbol{w}_k, \tag{10}$$

where $\|\boldsymbol{x}_k\|_M = \|\boldsymbol{u}_1\|_M = \|\boldsymbol{z}_k\|_M = 1$ and $\boldsymbol{u}_1^* M \boldsymbol{z}_k = 0$. Then we have

$$\begin{aligned}\rho(\boldsymbol{x}_k) &= \cos^2\vartheta_k \lambda_1 + 2\cos\vartheta_k \sin\vartheta_k \boldsymbol{u}_1^* A \boldsymbol{z}_k + \sin^2\vartheta_k \boldsymbol{z}_k^* A \boldsymbol{z}_k \\ &= \lambda_1(1 - \sin^2\vartheta_k) + \sin^2\vartheta_k \rho(\boldsymbol{z}_k),\end{aligned}$$

Convergence (cont.)

Thus,

$$\rho(\boldsymbol{x}_k) - \lambda_1 = \sin^2 \vartheta_k \, (\rho(\boldsymbol{z}_k) - \lambda_1) \leq (\lambda_n - \lambda_1) \sin^2 \vartheta_k .$$

As seen earlier, in symmetric eigenvalue problems, the eigenvalues are much more accurate than the eigenvectors.

Let us suppose that eigenvalue has converged, $\rho(\boldsymbol{x}_k) = \rho_k \cong \lambda_1$, but the eigenvector is not yet as accurate as desired. Then,

$$\begin{aligned} \boldsymbol{r}_k = (A - \rho_k M)\boldsymbol{x}_k \cong (A - \lambda_1 M)\boldsymbol{x}_k &= \sum_{j=1}^{n} (\lambda_j - \lambda_1) M \boldsymbol{u}_j \, \boldsymbol{u}_j^* M \boldsymbol{x}_k \\ &= \sum_{j=2}^{n} (\lambda_j - \lambda_1) M \boldsymbol{u}_j \, \boldsymbol{u}_j^* M \boldsymbol{x}_k , \end{aligned}$$

Convergence (cont.)

Therefore, $\boldsymbol{u}_1^* \boldsymbol{r}_k = 0$. From (10) we have $\boldsymbol{w}_k = \sin\vartheta_k \boldsymbol{z}_k \perp_M \boldsymbol{u}_1$. Thus,

$$\begin{cases} (A - \lambda_1 M)\boldsymbol{w}_k = (A - \lambda_1 M)\boldsymbol{x}_k = \boldsymbol{r}_k \perp \boldsymbol{u}_1, \\ \boldsymbol{w}_k^* M \boldsymbol{u}_1 = 0. \end{cases}$$

If λ_1 is a simple eigenvalue of the pencil $(A; M)$ then $A - \lambda_1 M$ is a bijective mapping of $\mathcal{R}(\boldsymbol{u}_1)^{\perp_M}$ onto $\mathcal{R}(\boldsymbol{u}_1)^{\perp}$.

If $\boldsymbol{r}_k \in \mathcal{R}(\boldsymbol{u}_1)^{\perp}$ then the equation

$$(A - \lambda_1 M)\boldsymbol{w}_k = \boldsymbol{r}_k, \qquad \boldsymbol{w}_k^* M \boldsymbol{u}_1 = 0, \tag{11}$$

has a *unique* solution $\boldsymbol{w}_k$ in $\mathcal{R}(\boldsymbol{u}_1)^{\perp_M}$.

Convergence (cont.)

Close to convergence, Rayleigh quotient minimization does nothing but solve equation (11). i.e., CG algorithm is applied to solve (11).

Convergence of RQMIN is determined by the condition number of $A - \lambda_1 M$ (as a mapping of $\mathcal{R}(\boldsymbol{u}_1)^{\perp_M}$ onto $\mathcal{R}(\boldsymbol{u}_1)^{\perp}$):

$$\kappa_0 = \mathcal{K}(A - \lambda_1 M)\Big|_{\mathcal{R}(\boldsymbol{u}_1)^{\perp_M}} = \frac{\lambda_n - \lambda_1}{\lambda_2 - \lambda_1},$$

High condition number if $|\lambda_1 - \lambda_2| \ll |\lambda_1 - \lambda_n|$.

Rate of convergence:

$$\frac{\sqrt{\kappa_0} - 1}{\sqrt{\kappa_0} + 1}.$$

Preconditioning

We try to turn

$$A\boldsymbol{x} = \lambda M\boldsymbol{x}$$

into

$$\tilde{A}\tilde{\boldsymbol{x}} = \tilde{\lambda}\tilde{M}\tilde{\boldsymbol{x}},$$

such that

$$\kappa(\tilde{A} - \tilde{\lambda}_1\tilde{M})\Big|_{\mathcal{R}(\tilde{\boldsymbol{u}}_1)^{\perp_{\tilde{M}}}} \ll \kappa(A - \lambda_1 M)\Big|_{\mathcal{R}(\boldsymbol{u}_1)^{\perp_M}}.$$

Change of variables: $\boldsymbol{y} = C\boldsymbol{x}$ with C nonsingular

$$\rho(\boldsymbol{x}) = \frac{\boldsymbol{x}^* A\boldsymbol{x}}{\boldsymbol{x}^* M\boldsymbol{x}} = \frac{\boldsymbol{y}^* C^{-*} A C^{-1}\boldsymbol{y}}{\boldsymbol{y}^* C^{-*} M C^{-1}\boldsymbol{y}} = \frac{\boldsymbol{y}^* \tilde{A}\boldsymbol{y}}{\tilde{\boldsymbol{y}}^* \tilde{M}\boldsymbol{y}} = \tilde{\rho}(\boldsymbol{y})$$

Preconditioning (cont.)

Thus,

$$\tilde{A} - \lambda_1 \tilde{M} = C^{-*}(A - \lambda_1 M)C^{-1},$$

or, after a similarity transformation,

$$C^{-1}(\tilde{A} - \lambda_1 \tilde{M})C = (C^* C)^{-1}(A - \lambda_1 M).$$

How should we choose C to satisfy (18)?

Let us tentatively set $C^* C = A$. Then

$$(C^* C)^{-1}(A - \lambda_1 M)\boldsymbol{u}_j = (I - \lambda_1 A^{-1} M)\boldsymbol{u}_j = \left(1 - \frac{\lambda_1}{\lambda_j}\right)\boldsymbol{u}_j.$$

Note that

$$0 \leq 1 - \frac{\lambda_1}{\lambda_j} < 1.$$

Preconditioning (cont.)

The 'true' condition number of the modified problem is

$$\kappa_1 := \kappa\left(A^{-1}(A-\lambda_1 M)\big|_{\mathcal{R}(\boldsymbol{u}_1)^{\perp_M}}\right) = \frac{1-\frac{\lambda_1}{\lambda_n}}{1-\frac{\lambda_1}{\lambda_2}} = \frac{\lambda_2}{\lambda_n}\frac{\lambda_n-\lambda_1}{\lambda_2-\lambda_1} = \frac{\lambda_2}{\lambda_n}\kappa_0.$$

If $\lambda_2 \ll \lambda_n$ then condition number is *much* reduced. Further,

$$\kappa_1 = \frac{1-\lambda_1/\lambda_n}{1-\lambda_1/\lambda_2} \lesssim \frac{1}{1-\lambda_1/\lambda_2}.$$

In FE applications, κ_1 does not dependent on mesh-width h.

Conclusion: choose C such that $C^*C \cong A$, e.g. IC(0).

Preconditioning (cont.)

Transformation $\boldsymbol{x} \longrightarrow \boldsymbol{y} = C\boldsymbol{x}$ need not be made explicitly.

In the code of page 12 we modify the computation of the gradient $\boldsymbol{g}_k$.

Statement 12 becomes

$$\boldsymbol{g}_k = 2(C^*C)^{-1}(\boldsymbol{v}_k - \rho_k \boldsymbol{u}_k)$$

Then preconditioner need not be an (incomplete) factorization of A.

Locally optimal PCG (LOPCG)

Parameters δ_k and α_k in RQMIN and (P)CG:

$$\rho(\boldsymbol{x}_{k+1}) = \rho(\boldsymbol{x}_k + \delta_k \boldsymbol{p}_k), \quad \boldsymbol{p}_k = -\boldsymbol{g}_k + \alpha_k \boldsymbol{p}_{k-1}$$

The parameters are determined such that $\rho(\boldsymbol{x}_{k+1})$ is minimized and consecutive search directions are conjugate.

Knyazev [2]: optimize both parameters, α_k and δ_k, at once

$$\rho(\boldsymbol{x}_{k+1}) = \min_{\delta,\gamma} \rho(\boldsymbol{x}_k - \delta \boldsymbol{g}_k + \gamma \boldsymbol{p}_{k-1}) \tag{12}$$

Results in potentially smaller values for the Rayleigh quotient, as

$$\min_{\delta,\gamma} \rho\big(\boldsymbol{x}_k - \delta \boldsymbol{g}_k + \gamma \boldsymbol{p}_{k-1}\big) \leq \min_{\delta} \big(\boldsymbol{x}_k - \delta(\boldsymbol{g}_k - \alpha_k \boldsymbol{p}_k)\big).$$

Procedure is "locally optimal".

Locally optimal PCG (LOPCG) (cont.)

$\rho(\boldsymbol{x}_{k+1})$ in (12) is minimal eigenvalue of 3×3 eigenvalue problem

$$\begin{bmatrix} \boldsymbol{x}_k^* \\ -\boldsymbol{g}_k^* \\ \boldsymbol{p}_{k-1}^* \end{bmatrix} A[\boldsymbol{x}_k, -\boldsymbol{g}_k, \boldsymbol{p}_{k-1}] \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \lambda \begin{bmatrix} \boldsymbol{x}_k^* \\ -\boldsymbol{g}_k^* \\ \boldsymbol{p}_{k-1}^* \end{bmatrix} M[\boldsymbol{x}_k, -\boldsymbol{g}_k, \boldsymbol{p}_{k-1}] \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$$

We normalize eigenvector such that first component is 1:

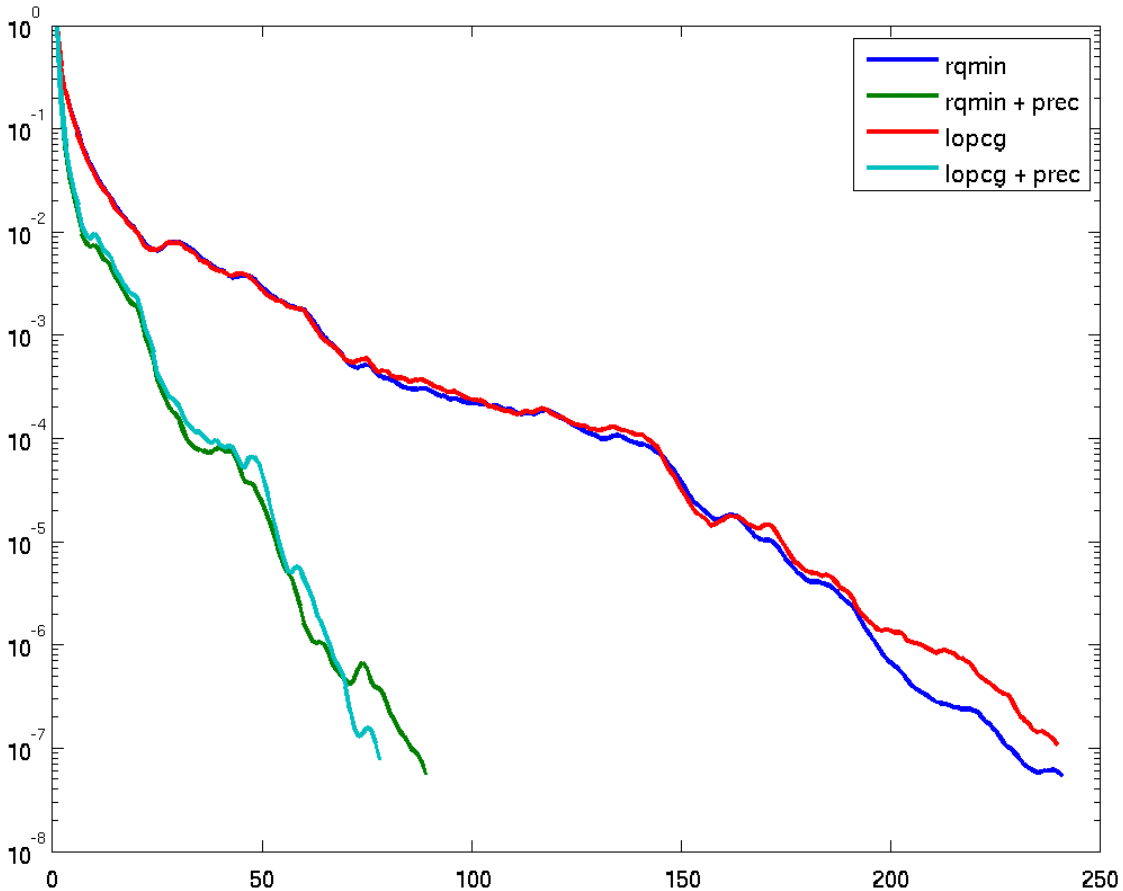
$$[1, \delta_k, \gamma_k] := [1, \beta/\alpha, \gamma/\alpha].$$

Then

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \delta_k \boldsymbol{g}_k + \gamma_k \boldsymbol{p}_{k-1} = \boldsymbol{x}_k + \delta_k \underbrace{\left(-\boldsymbol{g}_k + (\gamma_k/\delta_k)\boldsymbol{p}_{k-1}\right)}_{=:\boldsymbol{p}_k} = \boldsymbol{x}_k + \delta_k \boldsymbol{p}_k.$$

RQ minimization from $\boldsymbol{x}_k$ along $\boldsymbol{p}_k = -\boldsymbol{g}_k + (\gamma_k/\delta_k)\boldsymbol{p}_{k-1}$.

Test: car cross section: 1st eigenvalue



Block versions

Above procedures converge *very* slowly if eigenvalues are clustered.

Hence, these methods should be applied only in blocked form.

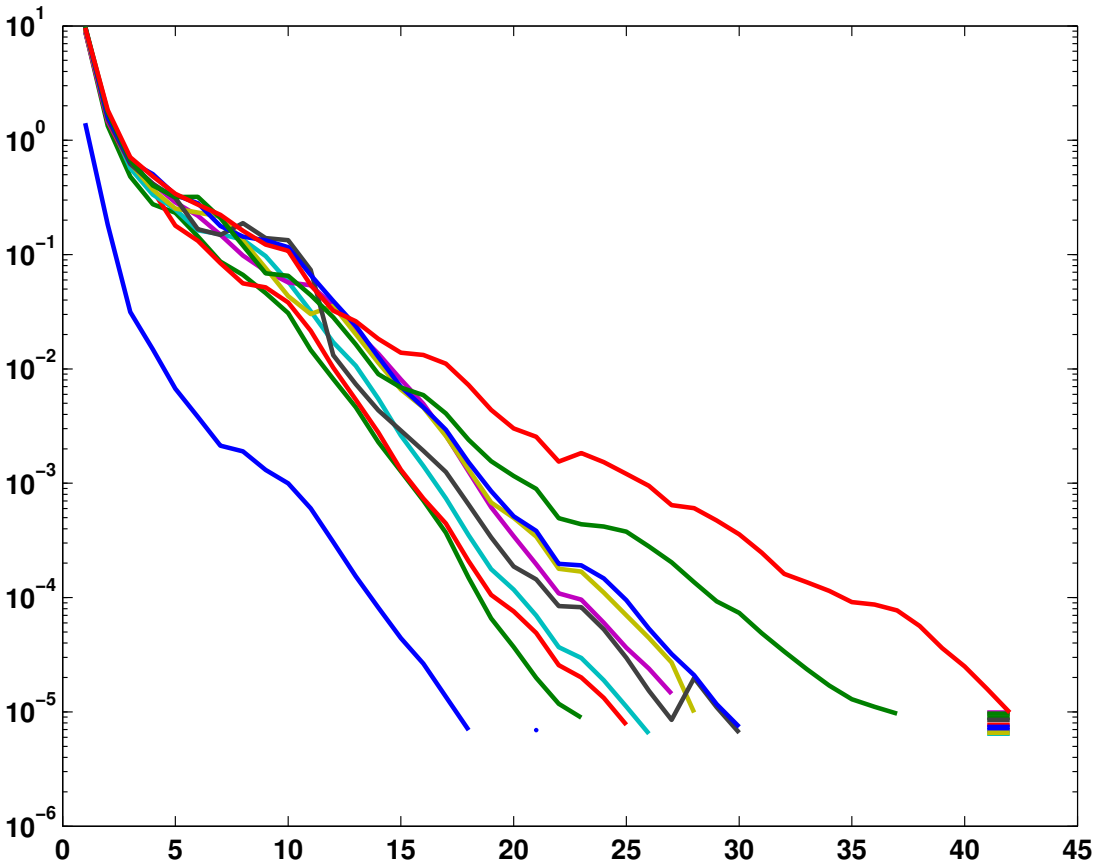
BRQMIN: Rayleigh quotient is minimized in $2q$-dimensional subspace generated by the eigenvector approximations X_k and search directions

$$P_k = -H_k + P_{k-1} B_k.$$

H_k: preconditioned residuals
B_k chosen such that the *block* of search directions is conjugate.

LOBPCG: Similar as BRQMIN, but search space is $3q$ dimensional: $\mathcal{R}([X_k, H_k, P_{k-1}])$.

Block versions (cont.)



430 system solves are needed to get 10 eigenpairs (283 with locking).

Trace minimization

Theorem

(Trace theorem for the generalized eigenvalue problem) *Let $A = A$ and M be as in* (3)*. Then,*

$$\lambda_1 + \lambda_2 + \cdots + \lambda_p = \min_{X \in \mathbb{F}^{n \times p},\ X^* M X = I_p} \operatorname{trace}(X^* A X) \qquad (13)$$

where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of problem (3)*. Equality holds in* (13) *if and only if the columns of the matrix X that achieves the minimum span the eigenspace corresponding to the smallest p eigenvalues.*

Let's try to use the theorem to derive an algorithm.

Trace minimization (cont.)

Sameh and coworkers [3] suggested the tracemin algorithm that follows the lines of Rayleigh quotient minimization.

Let $X_k \in \mathbb{F}^{n\times p}$ with $X_k^* M X_k = I_p$ and

$$X_k^* A X_k = \Sigma_k = \text{diag}(\sigma_1^{(k)}, \ldots, \sigma_p^{(k)}).$$

Want to construct the next iterate X_{k+1} by setting

$$X_{k+1} = (X_k - \Delta_k) S_k$$

S_k needed to enforce orthogonality of X_{k+1}.
We choose the correction Δ_k to be orthogonal to X_k,

$$\Delta_k^* M X_k = 0. \tag{14}$$

Trace minimization (cont.)

Want to minimize

$$\begin{aligned}\operatorname{trace}((X_k - \Delta_k)^* A(X_k - \Delta_k)) &= \sum_{i=1}^{p} \boldsymbol{e}_i^* (X_k - \Delta_k)^* A(X_k - \Delta_k)\boldsymbol{e}_i \\ &= \sum_{i=1}^{p} (\boldsymbol{x}_i - \boldsymbol{d}_i)^* A(\boldsymbol{x}_i - \boldsymbol{d}_i)\end{aligned}$$

with $\boldsymbol{x}_i = X_k \boldsymbol{e}_i$ and $\boldsymbol{d}_i = \Delta_k \boldsymbol{e}_i$.

These are p individual minimization problems:

Minimize $\quad (\boldsymbol{x}_i - \boldsymbol{d}_i)^* A(\boldsymbol{x}_i - \boldsymbol{d}_i) \quad$ subject to $X_k^* M \boldsymbol{d}_i = \boldsymbol{0}, \quad i = 1, \ldots, p.$

To solve this eq. we define the functional

$$f(\boldsymbol{d}, \boldsymbol{l}) := (\boldsymbol{x}_i - \boldsymbol{d})^* A(\boldsymbol{x}_i - \boldsymbol{d}) + \boldsymbol{l}^* X_k^* M \boldsymbol{d}.$$

Trace minimization (cont.)

The method of Lagrange multipliers leads to

$$\begin{bmatrix} A & MX_k \\ X_k^* M & O \end{bmatrix} \begin{pmatrix} \boldsymbol{d} \\ \boldsymbol{l} \end{pmatrix} = \begin{pmatrix} A\boldsymbol{x}_i \\ 0 \end{pmatrix}, \qquad 1 \le i \le p.$$

Collecting all p equations in one yields

$$\begin{bmatrix} A & MX_k \\ X_k^* M & O \end{bmatrix} \begin{pmatrix} \Delta_k \\ L \end{pmatrix} = \begin{pmatrix} AX_k \\ O \end{pmatrix}. \tag{15}$$

By Gaussian elimination we obtain

$$L = (X_k^* M A^{-1} M X_k)^{-1}.$$

Multiplying the first equation in (15) by A^{-1} we get

$$\Delta_k + A^{-1} M X_k L = X_k,$$

Trace minimization (cont.)

So,

$$Z_{k+1} \equiv X_k - \Delta_k = A^{-1}MX_kL = A^{-1}MX_k(X_k^*MA^{-1}MX_k)^{-1}.$$

such that, one step of the above trace minimization algorithm amounts to one step of subspace iteration with shift $\sigma = 0$.

This implies convergence. Bit requires factorization of A.

Rewrite saddle point problem (15) as a simple linear problem, which can be solved iteratively.

Trace minimization (cont.)

Let P be the orthogonal projection onto $\mathcal{R}(MX_k)^{\perp}$,

$$P = I - MX_k(X_k^* M^2 X_k)^{-1} X_k^* M.$$

Then the linear systems of equations (15) and

$$PAP\Delta_k = PAX_k, \qquad X_k^* M \Delta_k = 0, \tag{16}$$

are *equivalent*, i.e., they have the same solution Δ_k.

PAP is positive semidefinite. Can use modification of PCG or MINRES to solve (16).

Tricks of the trade

- *Simple shifts.* Choose a shift $\sigma_1 \leq \lambda_1$ until the first eigenpair is found. Then proceed with the shift $\sigma_2 \leq \lambda_2$ and *lock* the first eigenvector. In this way PCG can be used to solve the linear systems as before.
- *Multiple dynamic shifts.* Each linear system

$$P(A - \sigma_i^{(k)} M) P \boldsymbol{d}_i^{(k)} = P \boldsymbol{r}_i, \qquad \boldsymbol{d}_i^{(k)} \perp_M X_k$$

  is solved with an individual shift. The shift is 'turned on' close to convergence. Systems indefinite $\Rightarrow$ PCG has to be adapted.
- *Preconditioning.* Systems above can be preconditioned, e.g., by a matrix of the form $M = CC^*$ where $CC^* \approx A$ is an incomplete Cholesky factorization.

The Tracemin algorithm

1: Choose matrix $V_1 \in \mathbb{R}^{n\times q}$ with $V_1^T M V_1 = I_q$, $q \geq p$.
2: **for** $k = 1, 2, \ldots$ until convergence **do**
3: Compute $W_k = AV_k$ and $H_k := V_k^* W_k$.
4: Compute spectral decomposition $H_k = U_k \Theta_k U_k^*$, with $\Theta_k = \mathrm{diag}(\vartheta_1^{(k)}, \ldots, \vartheta_q^{(k)})$, $\quad \vartheta_1^{(k)} \leq \ldots \leq \vartheta_q^{(k)}$.
5: Compute Ritz vectors $X_k = V_k U_k$ and residuals $R_k = W_k U_k - M X_k \Theta_k$
6: For $i = 1, \ldots, q$ solve *approximatively*

$$P(A - \sigma_i^{(k)} M) P \boldsymbol{d}_i^{(k)} = P\boldsymbol{r}_i, \qquad \boldsymbol{d}_i^{(k)} \perp_M X_k$$

by some modified PCG solver.
7: Compute $V_{k+1} = (X_k - \Delta_k) S_k$, $\Delta_k = [\boldsymbol{d}_1^{(k)}, \ldots, \boldsymbol{d}_q^{(k)}]$, by a M-orthogonal modified Gram-Schmidt procedure.
8: **end for**

Numerical experiment (from [3])

Problem	Size	Max # inner its	Block Jacobi–Davidson			Davidson-type Tracemin		
			#its	A mults	time[sec]	#its	A mults	time[s
BCSST08	1074	40	34	3954	4.7	10	759	0.8
BCSST09	1083	40	15	1951	2.2	15	1947	2.2
BCSST11	1473	100	90	30990	40.5	54	20166	22.4
BCSST21	3600	100	40	10712	35.1	39	11220	36.2
BCSST26	1922	100	60	21915	32.2	39	14102	19.6

Table 1: Numerical results for problems from the Harwell–Boeing collection with four processors.
IC(0) of A was used as preconditioner.
Davidson-type trace minimization algorithm with multiple dynamic shifts works better than block Jacobi–Davidson for three out of five problems.

References

[1] Y. T. Feng and D. R. J. Owen. *Conjugate gradient methods for solving the smallest eigenpair of large symmetric eigenvalue problems*, Internat. J. Numer. Methods Eng., 39 (1996), pp. 2209–2229.

[2] A. V. Knyazev. *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.

[3] A. Sameh and Z. Tong. *The trace minimization method for the symmetric generalized eigenvalue problem*, J. Comput. Appl. Math., 123 (2000), pp. 155–175.

[4] A. Klinvex, F. Saied, and A. Sameh. *Parallel implementations of the trace minimization scheme TraceMIN for the sparse symmetric eigenvalue problem*, Comput. Math. Appl., 65(3):460–468, 2013.


THE END!