Solving large scale eigenvalue problems

Lecture 11, May 9, 2018: Jacobi–Davidson algorithms

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

- Jacobi–Davidson algorithms
- Basic ideas
  - Davidson's subspace expansion
  - Jacobi's orthogonal component correction
- Jacobi–Davidson algorithms
  - Correction equation
  - JDQR, JDSYM, JDQZ
  - Nonlinear Jacobi–Davidson

Jacobi–Davidson algorithm

- The Lanczos and Arnoldi methods are effective to compute extremal eigenvalues.
- Lanczos and Arnoldi methods combined with shift-and-invert spectral transformation are efficient to compute interior eigenvalues close to shift σ. Linear systems of the form

$$(A - \sigma I)\mathbf{x} = \mathbf{y}, \qquad \text{or} \qquad (A - \sigma M)\mathbf{x} = \mathbf{y},$$

  need to be solved in each iteration step.
- Systems have to be solved accurately. Otherwise the Lanczos/ Arnoldi relation does not hold anymore. In most cases the matrix $A - \sigma I$ (or $A - \sigma M$) is LU or Cholesky factored.
- The Jacobi–Davidson (JD) algorithm is particularly attractive if this factorization is not feasible.

The Davidson algorithm

Let $\mathbf{v}_1, \ldots, \mathbf{v}_m$ be a set of *orthonormal* vectors, spanning the *search space* $\mathcal{R}(V_m)$ with $V_m = [\mathbf{v}_1, \ldots, \mathbf{v}_m]$.

Galerkin approach: we look for m-vector $\mathbf{s}$ for which *Galerkin condition* holds,

$$AV_m\mathbf{s} - \vartheta V_m\mathbf{s} \perp \mathbf{v}_1, \ldots, \mathbf{v}_m.$$

This leads to (small) eigenvalue problem

$$V_m^* A V_m \mathbf{s} = \vartheta V_m^* V_m \mathbf{s} = \vartheta \mathbf{s}$$

with solutions $(\vartheta_j^{(m)}, \mathbf{s}_j^{(m)})$, $j = 1, \ldots, m$.

In the sequel we omit superscript m.

The Davidson algorithm (cont.)

Consider Ritz pair $(\vartheta_j, \mathbf{u}_j = V_m \mathbf{s}_j)$ and residual $\mathbf{r}_j = A\mathbf{u}_j - \vartheta_j \mathbf{u}_j$.

Can we improve $(\vartheta_j, \mathbf{u}_j)$ if $\|\mathbf{r}_j\|$ is still large?

We try to find a better approximate eigenpair by *expanding* the search space.

Davidson [1] suggested to compute vector $\mathbf{t}$ from

$$(D_A - \vartheta_j I)\mathbf{t} = \mathbf{r}_j, \qquad D_A = \mathrm{diag}(A).$$

The vector $\mathbf{t}$ is then orthogonalized against $\mathbf{v}_1, \ldots, \mathbf{v}_m$. The resulting vector, after normalization, is chosen as $\mathbf{v}_{m+1}$ by which $\mathcal{R}(V_m)$ is expanded, i.e., $V_{m+1} = [\mathbf{v}_1, \ldots, \mathbf{v}_m, \mathbf{v}_{m+1}]$.

The Davidson algorithm (cont.)

- Method turned out to be successful in finding dominant eigenvalues of (strongly) diagonally dominant matrices.
- Matrix $D_A - \vartheta_j I$ has therefore often been viewed as a *preconditioner* for the matrix $A - \vartheta_j I$.
- A number of investigations were made with more sophisticated preconditioners $M - \vartheta_j I$.
- They lead to the conclusion that $M - \vartheta_j I$ should not be 'too close' to $A - \vartheta_j I$ which contradicts the notion of a preconditioner as being an easily invertible (factorizable) approximation of $A - \vartheta_j I$.

The Jacobi orthogonal component correction

- Jacobi (1846) introduced Jacobi rotations to solve the symmetric eigenvalue problem. He also presented an approach to improve an approximate eigenpair with an iterative procedure.
- Sleijpen and van der Vorst [2] used Jacobi's idea to improve Davidson's subspace expansion.
- Let $\mathbf{u}_j$ be approximation to eigenvector $\mathbf{x}$ with $A\mathbf{x} = \lambda\mathbf{x}$.
- Jacobi proposed to *correct* $\mathbf{u}_j$ by a vector $\mathbf{t}$, $\mathbf{u}_j \perp \mathbf{t}$, such that

$$A(\mathbf{u}_j + \mathbf{t}) = \lambda(\mathbf{u}_j + \mathbf{t}), \qquad \mathbf{u}_j \perp \mathbf{t}. \tag{1}$$

Sleijpen & van der Vorst called this the Jacobi orthogonal component correction (JOCC).

The Jacobi orthogonal component correction (cont.)

As $\mathbf{t} \perp \mathbf{u}_j$ we may split equation (1) in the part parallel to $\mathbf{u}_j$ and in the part orthogonal to $\mathbf{u}_j$.

Let $\|\mathbf{u}_j\| = 1$. (Then $\mathbf{u}_j\mathbf{u}_j^*$ an orthogonal projector.)

Then part parallel to $\mathbf{u}_j$ is

$$\mathbf{u}_j\mathbf{u}_j{}^*A(\mathbf{u}_j + \mathbf{t}) = \lambda\mathbf{u}_j\mathbf{u}_j{}^*(\mathbf{u}_j + \mathbf{t})$$

which simplifies to the scalar equation

$$\vartheta_j + \mathbf{u}_j{}^*A\mathbf{t} = \lambda.$$

Here, $\vartheta_j = \rho(\mathbf{u}_j)$ is the Rayleigh quotient of $\mathbf{u}_j$.

The Jacobi orthogonal component correction (cont.)

The part orthogonal to $\mathbf{u}_j$ is

$$(I - \mathbf{u}_j \mathbf{u}_j^*) A (\mathbf{u}_j + \mathbf{t}) = \lambda (I - \mathbf{u}_j \mathbf{u}_j^*)(\mathbf{u}_j + \mathbf{t})$$

which is equivalent to (move $\mathbf{t}$ to left, $\mathbf{u}$ to right.)

$$\begin{aligned}(I - \mathbf{u}_j \mathbf{u}_j^*)(A - \lambda I)\mathbf{t} &= (I - \mathbf{u}_j \mathbf{u}_j^*)(-A\mathbf{u}_j + \lambda \mathbf{u}_j) \\ &= -(I - \mathbf{u}_j \mathbf{u}_j^*) A \mathbf{u}_j = -(A - \vartheta_j I)\mathbf{u}_j =: -\mathbf{r}_j.\end{aligned}$$

As $(I - \mathbf{u}_j \mathbf{u}_j^*)\mathbf{t} = \mathbf{t}$ we can 'symmetrize' this equation:

$$(I - \mathbf{u}_j \mathbf{u}_j^*)(A - \lambda I)(I - \mathbf{u}_j \mathbf{u}_j^*)\mathbf{t} = -\mathbf{r}_j.$$

If A is symmetric then the matrix above is symmetric indeed.

The Jacobi–Davidson correction equation

Unfortunately, we do not know λ!

So, we replace λ by ϑ_j to get Jacobi–Davidson correction equation

$$\boxed{(I - \mathbf{u}_j\mathbf{u}_j^*)(A - \vartheta_j I)(I - \mathbf{u}_j\mathbf{u}_j^*)\mathbf{t} = -\mathbf{r}_j = -(A - \vartheta_j I)\mathbf{u}_j, \qquad \mathbf{t} \perp \mathbf{u}_j.}$$

As $\mathbf{r}_j \perp \mathbf{u}_j$ (in fact $\mathbf{r}_j \perp \mathcal{V}_m$) this equation is consistent if $A - \vartheta_j I$ is nonsingular.

The correction equation is, in general, solved iteratively by the GMRES or the MINRES algorithm. Often, only little accuracy in the solution is required.

$\mathcal{V}_m$ isn't a Krylov space!

The Jacobi–Davidson correction equation (cont.)

Once $\mathbf{t}$ is (approximately) known we set

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{t}.$$

and

$$\vartheta_{j+1} = \vartheta_j + \mathbf{u}_j{}^* A\mathbf{t}.$$

If A is symmetric we may set

$$\vartheta_{j+1} = \rho(\mathbf{u}_{j+1}).$$

Solving the Jacobi–Davidson correction equation

The matrix $(I - \mathbf{u}_j\mathbf{u}_j^*)(A - \vartheta_j I)(I - \mathbf{u}_j\mathbf{u}_j^*)$ in the correction equation is evidently singular.

$$(I - \mathbf{u}_j\mathbf{u}_j^*)(A - \vartheta_j I)(I - \mathbf{u}_j\mathbf{u}_j^*)\mathbf{t} = -\mathbf{r}_j, \qquad \mathbf{t} \perp \mathbf{u}_j.$$

If $\vartheta_j \neq \sigma(A)$ then $A - \vartheta_j I$ is nonsingular. Since $\mathbf{r}_j^*\mathbf{u}_j = 0$,

$$\mathbf{r}_j \in \mathcal{R}((I - \mathbf{u}_j\mathbf{u}_j^*)(A - \vartheta_j I)(I - \mathbf{u}_j\mathbf{u}_j^*)).$$

So, there is a solution $\mathbf{t}$. (In fact many.) Uniqueness is obtained through constraint $\mathbf{t}^*\mathbf{u}_j = 0$.

How can we solve the correction equation?

Solving the Jacobi–Davidson correction equation (cont.)

How can we solve the correction equation?

$$(I - \mathbf{u}_j\mathbf{u}_j^*)(A - \vartheta_j I)\mathbf{t} = (A - \vartheta_j I)\mathbf{t} - \alpha\mathbf{u}_j = -\mathbf{r}_j$$

with the scalar $\alpha = \mathbf{u}_j^*(A - \vartheta_j I)\mathbf{t}$. Assuming that $\vartheta_j \neq \sigma(A)$ we get

$$\mathbf{t} = \alpha(A - \vartheta_j I)^{-1}\mathbf{u}_j - (A - \vartheta_j I)^{-1}\mathbf{r}_j.$$

The constraint $\mathbf{u}_j^*\mathbf{t} = 0$ allows us to determine the free variable α,

$$0 = \alpha\mathbf{u}_j^*(A - \vartheta_j I)^{-1}\mathbf{u}_j - \mathbf{u}_j^*(A - \vartheta_j I)^{-1}\mathbf{r}_j,$$

whence

$$\alpha = \frac{\mathbf{u}_j^*(A - \vartheta_j I)^{-1}\mathbf{r}_j}{\mathbf{u}_j^*(A - \vartheta_j I)^{-1}\mathbf{u}_j}.$$

Solving the Jacobi–Davidson correction equation (cont.)

So, the next approximate is (up to normalization)

$$\begin{aligned}\mathbf{u}_{j+1} &= \mathbf{u}_j + \mathbf{t} \\ &= \mathbf{u}_j + \alpha(A - \vartheta_j I)^{-1}\mathbf{u}_j - \underbrace{(A - \vartheta_j I)^{-1}\mathbf{r}_j}_{\mathbf{u}_j} = \alpha(A - \vartheta_j I)^{-1}\mathbf{u}_j\end{aligned}$$

which is a step of Rayleigh quotient iteration! This implies a fast convergence rate of this algorithm:

- quadratic for general matrices and
- cubic in the Hermitian case.

Iterative solution of the correction equation

In general the correction equation

$$\tilde{A}\,\mathbf{t} := (I - \mathbf{u}_j\mathbf{u}_j^*)(A - \vartheta_j I)(I - \mathbf{u}_j\mathbf{u}_j^*)\mathbf{t} = -\mathbf{r}_j,$$
$$(I - \mathbf{u}_j\mathbf{u}_j^*)\mathbf{t} = \mathbf{t}$$

is solved iteratively with a Krylov space solver like GMRES or MINRES.

JD algorithm to compute eigenvalue closest to τ

1: Let $A \in \mathbb{R}^{n\times n}$. Let $\mathbf{t}$ be initial vector. Set $V_0 = []$, $m = 1$.
2: **loop**
3: $\quad \mathbf{t} = (I - V_{m-1}V_{m-1}^*)\mathbf{t}$
4: $\quad \mathbf{v}_m := \mathbf{t}/\|\mathbf{t}\|; \quad V_m := [V_{m-1}, \mathbf{v}_m];$
5: $\quad M = V_m^* A V_m$
6: $\quad$ Rayleigh–Ritz step: Compute eigenpair $(\vartheta, \mathbf{s})$ of M closest to τ: $M\mathbf{s} = \vartheta\mathbf{s}$; $\|\mathbf{s}\| = 1$;
7: $\quad \mathbf{u} := V_m\mathbf{s}; \quad \mathbf{r} := A\mathbf{u} - \vartheta\mathbf{u};$
8: $\quad$ **if** $\|\mathbf{r}\| <$ tol **then**
9: $\quad\quad$ **return** $(\tilde{\lambda} = \vartheta,\ \tilde{\mathbf{x}} = \mathbf{u})$
10: $\quad$ **end if**
11: $\quad$ (Approximatively) solve the correction equation for $\mathbf{t}$,
$$(I - \mathbf{u}\mathbf{u}^*)(A - \vartheta_j I)(I - \mathbf{u}\mathbf{u}^*)\mathbf{t} = -\mathbf{r}, \quad \mathbf{t} \perp \mathbf{u};$$
12: **end loop**

Detour on Krylov space solvers

Let $\mathbf{x}_0$ (e.g. $\mathbf{x}_0 = \mathbf{0}$) be an initial guess for the solution of the linear system of equations

$$A\mathbf{x} = \mathbf{b}.$$

Krylov space methods like the conjugate gradient (CG) method or the generalized minimal residual (GMRES) method search for solutions in Krylov spaces generated with the initial residual

$$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}.$$

The GMRES (Generalized Minimal Residual) algorithm computes $\mathbf{x}_m \in \mathbf{x}_0 + \mathcal{K}_m(A, \mathbf{r}_0)$ that leads to the smallest residual exploiting the *Arnoldi relation*.
The MINRES algorithm does the same for symmetric matrices using the *Lanczos relation*.

Detour on Krylov space solvers (cont.)

Goal: Cheaply minimize

$$\|\mathbf{r}_m\|_2 = \|\mathbf{b} - A\mathbf{x}_m\|_2, \qquad \mathbf{x}_m \in \mathbf{x}_0 + \mathcal{K}_m(A, \mathbf{r}_0),$$

using the Arnoldi relation

$$AV_m = V_m H_m + \mathbf{w}_m \mathbf{e}_m^T = V_{m+1} \bar{H}_m$$

and $\mathcal{R}(V_m) = \mathcal{K}_m(A, \mathbf{r}_0)$.

Detour on Krylov space solvers (cont.)

Arnoldi basis expansion $\mathbf{x}_m = \mathbf{x}_0 + V_m \mathbf{y}$ gives with $\beta := \|\mathbf{r}_0\|_2$:

$$\begin{aligned}
\min \|\mathbf{r}_m\|_2 &= \min \|\mathbf{b} - A(\mathbf{x}_0 + V_m \mathbf{y})\|_2 \\
&= \min \|\mathbf{r}_0 - AV_m \mathbf{y}\|_2 \\
&= \min \|\mathbf{r}_0 - V_{m+1} \bar{H}_m \mathbf{y}\|_2 \\
&= \min \| V_{m+1} \left(\beta \mathbf{e}_1 - \bar{H}_m \mathbf{y} \right) \|_2 \\
&= \min \|\beta \mathbf{e}_1 - \bar{H}_m \mathbf{y}\|_2.
\end{aligned}$$

- Hessenberg least squares problem of dimension $(m+1) \times m$.
- QR factorization of $\bar{H}_m$ can be cheaply computed using Givens rotations.

Detour on Krylov space solvers (cont.)

Convergence is good if

- A has just a few eigenvalues (at least eigenvalue clusters), or
- A is well-conditioned ($A \approx I$)

In linear systems (in contrast to eigenvalue problems) we can modify the original equation using a preconditioner K

$$K^{-1}A\mathbf{x} = K^{-1}\mathbf{b}.$$

K is chosen such that

- one of the above goals is achieved (approximately) for $K^{-1}A$ and
- it is easy (cheap) to solve linear system $K\mathbf{x} = \mathbf{b}$.

Iterative solution of the correction equation

In general the correction equation

$$\tilde{A}\,\mathbf{t} := (I - \mathbf{u}_j\mathbf{u}_j^*)(A - \vartheta_j I)(I - \mathbf{u}_j\mathbf{u}_j^*)\mathbf{t} = -\mathbf{r}_j, \qquad \mathbf{t} \perp \mathbf{u}_j, \tag{2}$$

is solved iteratively with a Krylov space solver like GMRES or MINRES.

For decent performance a preconditioner is needed. We set [2]

$$\tilde{K} = (I - \mathbf{u}_j\mathbf{u}_j^*)K(I - \mathbf{u}_j\mathbf{u}_j^*), \qquad K \approx A - \vartheta_j I. \tag{3}$$

K is a preconditioner for $A \approx A - \vartheta_j I$. We assume that K is (easily) invertible, i.e., that it is computationaly much cheaper to solve a system of equation with K than with A.

Iterative solution of the correction equation (cont.)

The first residual in the Arnoldi/Lanczos procedure is

$$\mathbf{v}_1 = -\mathbf{r}_j - \tilde{A}\mathbf{t}_0.$$

Clearly, $\mathbf{v}_1 \perp \mathbf{u}_j$ since $\mathbf{r}_j \perp \mathbf{u}_j$.
The preconditioned residual is obtained by solving

$$\tilde{K}\mathbf{z}_1 = \mathbf{v}_1, \qquad \mathbf{z}_1 \perp \mathbf{u}_j.$$

$\mathbf{z}_1$ is (up to normalization) the first Arnoldi/Lanczos basis vector.
Clearly, $\mathbf{z}_1 \perp \mathbf{u}_j$.

Iterative solution of the correction equation (cont.)

The solution of

$$\tilde{K}\mathbf{z}_1 = (I - \mathbf{u}_j\mathbf{u}_j^*)K\mathbf{z}_1 = \mathbf{v}_1, \qquad \mathbf{z}_1 \perp \mathbf{u}_j,$$

is obtained similarly as earlier:

$$K\mathbf{z}_1 = \mathbf{v}_1 + \beta\mathbf{u}_j$$

The scalar factor β is determined such that $\mathbf{z}_1^*\mathbf{u}_j = 0$.

The further basis vectors $\mathbf{z}_2, \mathbf{z}_3, \ldots,$ are obtained in the same fashion.

In particular, $\mathbf{z}_k^*\mathbf{u}_j = 0,\ k = 2, 3, \ldots$

Restarts

- The dimension m of the search space can get large.
- To limit memory consumption, we bound m: $m \leq m_{\max}$.
- If $m = m_{\max}$ we restart: $V_m = V_{m_{\max}}$ is replaced by the, say, q Ritz vectors corresponding to the Ritz values closest to τ.
- The restart is easy because we do not need to respect the Krylov space structure.

Computing multiple eigenvalues

Let $\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_k$ be already computed eigenvectors or Schur vectors with $\tilde{\mathbf{x}}_i^* \tilde{\mathbf{x}}_j = \delta_{ij}$, $1 \leq i, j \leq k$. Then

$$AQ_k = Q_k T_k, \qquad Q_k = [\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_k].$$

is a partial Schur decomposition of A.

We want to *extend* the partial Schur decomposition by one vector employing the Jacobi–Davidson algorithm.

Since Schur vectors are mutually orthogonal we apply the JD algorithm in $\mathcal{R}(Q_k)^\perp$, i.e., we apply JD algorithm to matrix

$$(I - Q_k Q_k^*) A (I - Q_k Q_k^*), \qquad Q_k = [\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_k].$$

Computing multiple eigenvalues (cont.)

The correction equation gets the form

$$(I - \tilde{Q}_k \tilde{Q}_k^*)(A - \vartheta_j I)(I - \tilde{Q}_k \tilde{Q}_k^*)\mathbf{t} = -\mathbf{r}_j, \qquad \tilde{Q}_k^* \mathbf{t} = \mathbf{0}.$$

with $\tilde{Q}_k = [\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_k, \mathbf{u}_j]$.
If the iteration has converged to vector $\tilde{\mathbf{x}}_{k+1}$ we can extend the partial Schur decomposition. Setting

$$Q_{k+1} := [Q_k, \tilde{\mathbf{x}}_{k+1}],$$

we get

$$AQ_{k+1} = Q_{k+1} T_{k+1} \tag{4}$$

with

$$T_{k+1} = \begin{bmatrix} T_k & Q_k^* A\tilde{\mathbf{x}}_{k+1} \\ 0 & \tilde{\mathbf{x}}_{k+1}^* A\tilde{\mathbf{x}}_{k+1} \end{bmatrix}.$$

Spectral shifts

- In correction equation and implicitly in preconditioner a spectral shift ϑ_j appears.
- It is not wise to always choose Rayleigh quotient $\rho(\tilde{\mathbf{q}})$ as shift. In first few iteration steps, Rayleigh quotient may be far away from the (desired) eigenvalue, and even may direct the JD iteration to an unwanted solution.
- One proceeds similarly as in Rayleigh quotient iteration: Initially, the shift is held fixed, usually equal to the *target value*. As soon as the norm of the residual is small enough, the Rayleigh quotient of actual approximate is chosen as spectral shift in the correction equation.

Spectral shifts (cont.)

- For efficiency reasons, the spectral shift in the preconditioner K is always fixed. In this way it has to be computed just once. Notice that $\tilde{K}$ is changing with each correction equation.
- As long as the shift is held fixed Jacobi–Davidson is actually performing a shift-and-invert Arnoldi iteration.

JDQR algo to compute the p eigenvalues closest to τ

1: Choose $\mathbf{v}_1$ with $\|\mathbf{v}_1\| = 1$. $Q_0 := []$; $k := 0$. (# conv. ev's)
2: $H_1 := \mathbf{v}_1^* A \mathbf{v}_1$; $V_1 := [\mathbf{v}_1]$; $j := 1$. (dim. search space)
3: $\tilde{\mathbf{q}} := \mathbf{v}_1$; $\tilde{\vartheta} := \tilde{\mathbf{q}}^* A \tilde{\mathbf{q}}$; $\mathbf{r} := A\tilde{\mathbf{q}} - \tilde{\vartheta}\tilde{\mathbf{q}}$.
4: **while** $k < p$ **do** {Compute Schur vectors one after the other}
5: Approximatively solve correction equation for $\mathbf{t}$

$$(I - \tilde{Q}_k \tilde{Q}_k^*)(A - \tilde{\vartheta} I)(I - \tilde{Q}_k \tilde{Q}_k^*)\mathbf{t} = -\mathbf{r}_j, \qquad \tilde{Q}_k^* \mathbf{t} = \mathbf{0}.$$

with $\tilde{Q}_k = [Q_k, \tilde{\mathbf{q}}]$.
6: $\mathbf{v}_j := (I - V_{j-1} V_{j-1}^*)\mathbf{t}$.
7: $\mathbf{v}_j := \mathbf{v}_j / \|\mathbf{v}_j\|$; $V_j := [V_{j-1}, \mathbf{v}_j]$.
8: $H_j := V_j^* A V_j$.
9: Compute Schur decomposition of $H_j =: S_j R_j S_j$ with the eigenvalues $r_{ii}^{(j)}$ sorted according to their distance to τ.

JDQR algo to compute the p eigenvalues closest to τ (cont.)

10: {Test for convergence}
11: $\tilde{\vartheta} = \lambda_1^{(j)}; \quad \tilde{\mathbf{q}} = V_j \mathbf{s}_1; \quad \mathbf{r} = A\,\tilde{\mathbf{q}} - \tilde{\vartheta}\tilde{\mathbf{q}}$
12: **if** $\|\mathbf{r}\| < \varepsilon$ **then**
13: $Q_{k+1} = [Q_k, \tilde{\mathbf{q}}]; \quad k := k+1;$
14: **end if**
15: {Restart}
16: **if** $j = j_{\max}$ **then**
17: $V_{j_{\min}} := V_j[\mathbf{s}_1, \ldots, \mathbf{s}_{\min}]; \quad T_{j_{\min}} := T_j(1 : j_{\min}, 1 : j_{\min});$
18: $H_{j_{\min}} := T_{j_{\min}}; \quad S_{j_{\min}} := I_{j_{\min}}; \quad J := j_{\min}$
19: **end if**
20: **end while**

Numerical experiment

We compute the 5 smallest eigenvalues and associated eigenvectors of the accustic behavior in the interior of a car. The problem size is $n = 1095$. The dimension of the search space varies between 10 and 20. The target is $\tau = 0$.

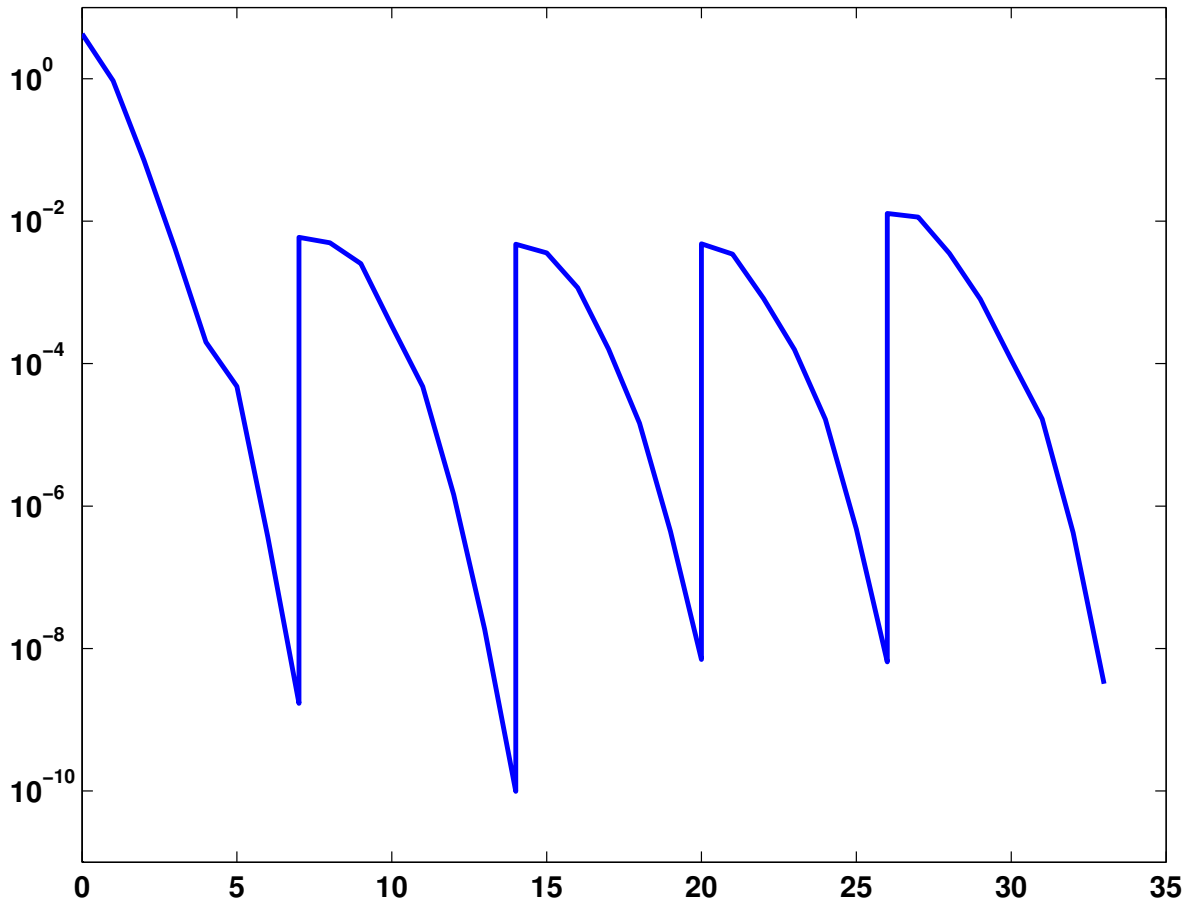
An eigenpair is considered converged if the residual norm $\|A\tilde{\mathbf{q}} - \tilde{\lambda}M\tilde{\mathbf{q}}\| < 10^{-8}\|\tilde{\mathbf{q}}\|$. The shift is changed from fixed $\vartheta_j = \sigma$ to $\vartheta_j = \rho(\tilde{q})$ when $\|A\tilde{\mathbf{q}} - \tilde{\lambda}M\tilde{\mathbf{q}}\| < 10^{-4}\|\tilde{\mathbf{q}}\|$.

The correction equation is solved with MINRES. The preconditioner is $K = \mathrm{diag}(A - \tau M)$. The linear system solver stops iterating if the residual is below

$$\|\tilde{\mathbf{r}}_i\| < \gamma^{-j}\|\tilde{\mathbf{r}}_0\|, \qquad \gamma = 2.$$

where j is the JD iteration number.

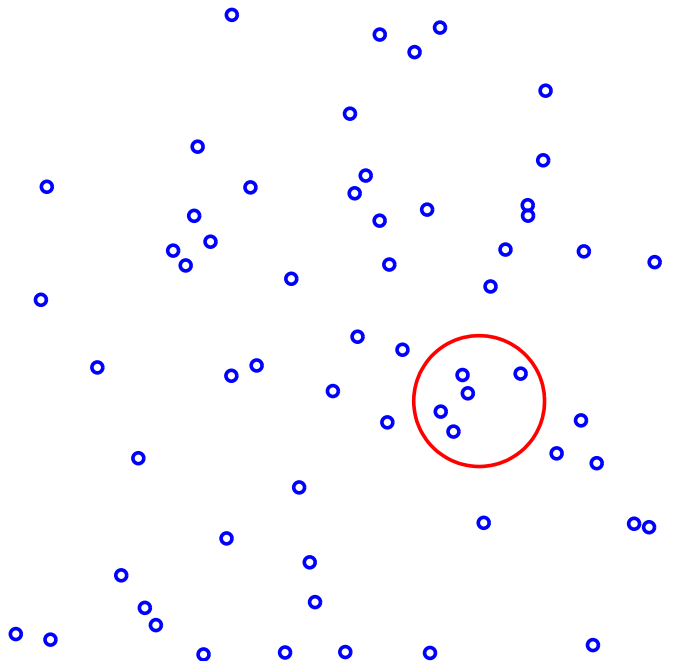
Numerical experiment (cont.)



Typical Jacobi–Davidson convergence history

The Jacobi–Davidson algorithm for interior eigenvalues

Interior eigenvalues are eigenvalues that do not lie at the 'border' of the convex hull of the spectrum.



View of a spectrum $\sigma(A)$ in the complex plane. The eigenvalues in the red circle are to be computed

The Jacobi–Davidson algorithm for interior eigenvalues (cont.)

Success of Jacobi–Davidson algorithm depends heavily on quality of actual Ritz pair $(\tilde{\vartheta}_j, \tilde{\mathbf{q}})$.
However, the Rayleigh–Ritz procedure can lead to problems if it is applied to *interior* eigenvalues.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{0.5} \\ 0 & \sqrt{0.5} \end{bmatrix}, \quad U^* A U = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad U^* U = I_2.$$

Any linear combination of columns of U is a Ritz vector corresponding to the Ritz value 0, e.g.,

$$U \begin{pmatrix} \sqrt{0.5} \\ \sqrt{0.5} \end{pmatrix} = \begin{pmatrix} \sqrt{0.5} \\ 0.5 \\ 0.5 \end{pmatrix}.$$

The Jacobi–Davidson algorithm for interior eigenvalues (cont.)

Although basis contains the correct eigenvector associated with the eigenvalue 0, the Rayleigh–Ritz procedure fails to find it and, instead, returns a very bad eigenvector approximation.

Contrieved example?

A MATLAB experiment with the same A but with a randomly perturbed U

```
U1=U+1e-4*rand(size(U))
```

shows analogous result:

We get a reasonable approximation ϑ for the eigenvalue 0. However, the large norm of the residual indicates that the Ritz vector is a bad approximation of the eigenvector.

Harmonic Ritz values and vectors

We need an idea how to get at better approximations for the eigenvector.

Shift-and-invert Arnoldi: basic operator is $A - \sigma I$ with some shift σ.

The algorithm finds the largest eigenvalues of $(A - \sigma I)^{-1}$, i.e., the eigenvalues of A closest to the shift.

However, factorization of $A - \sigma I$ is infeasible.

Clever way out: apply Ritz–Galerkin procedure with the matrix $(A - \sigma I)^{-1}$ and some cleverly chosen subspace $\mathcal{R}(V) \subset \mathbb{R}^n$.
Small eigenvalue problem

$$V^*(A - \sigma I)^{-1} V\mathbf{s} = \mu V^* V\mathbf{s}. \tag{5}$$

Harmonic Ritz values and vectors (cont.)

Largest eigenvalues of (5) approximate largest Ritz values μ_j.

$$\mu_j \approx \frac{1}{\lambda_j - \sigma} \Longleftrightarrow \lambda_j \approx \sigma + \frac{1}{\mu_j},$$

where λ_j is an eigenvalue of A close to the shift σ.

The trick is in the choice of V. We set $V := (A - \sigma I)U$. Then (5) becomes

$$U^*(A - \sigma I)^*(A - \sigma I)U\mathbf{s} = \tau U^*(A - \sigma I)^* U\mathbf{s}, \quad \tau = 1/\mu.$$

or

$$V^* V\mathbf{s} = \tau V^* U\mathbf{s}. \tag{6}$$

Harmonic Ritz values and vectors (cont.)

Definition

Let $(\tau, \mathbf{s})$ be an eigenpair of (6). Then the pair $(\sigma + \tau, U\mathbf{s})$ is called a **harmonic Ritz pair** of A with shift σ.

- In practice, we are interested only in the harmonic Ritz pair corresponding to the smallest harmonic Ritz values.
- In correction equation of the JD algorithm the harmonic Ritz vector is used as the latest eigenvector approximation and the harmonic Ritz value as the shift.
- In the symmetric case the harmonic Ritz value is replaced by the Rayleigh quotient of the harmonic Ritz vector $\mathbf{x}$, as

$$\|A\mathbf{x} - \rho(\mathbf{x})\mathbf{x}\| \leq \|A\mathbf{x} - \mu\mathbf{x}\|, \quad \text{for all } \mu.$$

Numerical example revisited

```
V = (A-theta*eye(3))*U1;
[v,l] = eig(V'*V, V'*U1)
theta + l(1,1)    %  Harmonic Ritz value
x = U1*v(:,1)     % Harmonic Ritz vector
x'*A*x
```

Result: The procedure is able to extract a very good vector as eigenvector approximation.

In the algorithm we only have to modify extraction phase (step 9).

- Compute smallest eigenvalue τ of $(AV_j - \sigma V_j)^*(AV_j - \sigma V_j)\mathbf{s} = \tau(AV_j - \sigma V_j)^* V_j\mathbf{s}$.
- Set $\tilde{\mathbf{q}} = V_j\tilde{\mathbf{s}}$ and $\tilde{\vartheta} = \sigma + \tau$ or $\tilde{\vartheta} = \tilde{\mathbf{q}}^* A\tilde{\mathbf{q}}/\tilde{\mathbf{q}}^*\tilde{\mathbf{q}}$.

Refined Ritz vectors

Alternative to harmonic Ritz vectors are refined Ritz vectors [3].

Definition

Let μ be a Ritz value of A restricted to U. A solution of the minimization problem

$$\min_{\hat{\mathbf{x}}\in U, \|\hat{\mathbf{x}}\|=1} \|A\hat{\mathbf{x}} - \mu\hat{\mathbf{x}}\| = \min_{\|\mathbf{s}\|=1} \|(A-\mu I)U\mathbf{s}\| \tag{7}$$

is called a refined Ritz vector.

The problem is solved by the 'smallest' right singular vector of $(A-\mu I)U$ or, equivalently, the 'smallest' eigenvector of $U^*(A-\mu I)^*(A-\mu I)U$.

Numerical example continued

```
[u,s,v]=svd((A - 0*eye(3))*U)
U*v(:,2)
[u,s,v]=svd((A - L(1,1)*eye(3))*U1)
U1*v(:,2)
```

Again, in the algorithm we only have to modify the extraction phase (step 9).

- Compute the Ritzpair $(\tilde{\vartheta}, \tilde{\mathbf{q}})$ of A closest to the target value.
- Compute the 'smallest' singular vector $\tilde{\mathbf{s}}$ of $AV_j - \tilde{\vartheta} V_j$. Replace $\tilde{\mathbf{q}}$ by $V_j \tilde{\mathbf{s}}$.

References

[1] E. R. Davidson, *The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices*, J. Comp. Phys., 17 (1975), pp. 87–94.

[2] G. L. G. Sleijpen and H. A. van der Vorst, *A Jacobi–Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425.

[3] G. W. Stewart, *Matrix Algorithms II: Eigensystems*, SIAM, Philadelphia, PA, 2001.