Solving large scale eigenvalue problems

Lecture 9, April 25, 2018: Lanczos and Arnoldi methods

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

We continue with the Arnoldi algorithm and its 'symmetric cousin', the Lanczos algorithm.

- The Lanczos algorithm and its deficiencies
- Loss of orthogonality
- Limiting the memory consumption of Arnoldi:
  Restarting Lanczos/Arnoldi algorithms

Reminder: the Arnoldi algorithm

- The Arnoldi algorithm constructs orthonormal bases for the Krylov spaces

$$\mathcal{K}^j(\mathbf{x}) = \mathcal{K}^j(\mathbf{x}, A) := \mathcal{R}([\mathbf{x}, A\,\mathbf{x}, \ldots, \mathbf{A}^{j-1}\,\mathbf{x}]) \in \mathbb{R}^{n\times j}, \quad j = 1, 2, \ldots$$

- These bases are nested.
- Let $\{\mathbf{v}_1, \ldots, \mathbf{v}_j\}$ be an orthonormal bases for $\mathcal{K}^j(\mathbf{x}, A)$. We obtain $\mathbf{v}_{j+1}$ by orthogonalizing $A\,\mathbf{v}_j$ against $\{\mathbf{v}_1, \ldots, \mathbf{v}_j\}$:

$$\mathbf{r}_j = A\,\mathbf{v}_j - V_j V_j^* A\,\mathbf{v}_j = A\,\mathbf{v}_j - \sum_{i=1}^{j} \mathbf{v}_i(\mathbf{v}_i^* A\,\mathbf{v}_j),$$

$$\mathbf{v}_{j+1} = \mathbf{r}_j / \|\mathbf{r}_j\|.$$

- This is the Gram–Schmidt orthogonalization procedure.

Arnoldi algorithm

Let $A \in \mathbb{R}^{n \times n}$. This algorithm computes orthonormal basis for $\mathcal{K}^j(\mathbf{x})$.
$\mathbf{v}_1 = \mathbf{x}/\|\mathbf{x}\|_2$;
for $j = 1, \ldots$ **do**
    $\mathbf{r}_j := A\mathbf{v}_j$;
    for $i = 1, \ldots, j$ **do** {Gram-Schmidt orthogonalization}
        $h_{ij} := \mathbf{v}_i^* \mathbf{r}_j, \quad \mathbf{r}_j := \mathbf{r}_j - \mathbf{v}_i h_{ij}$;
    end for
    $h_{j+1,j} := \|\mathbf{r}_j\|$;
    if $h_{j+1,j} = 0$ **then** {Found an invariant subspace}
        return $(\mathbf{v}_1, \ldots, \mathbf{v}_j, H \in \mathbb{R}^{j \times j})$
    end if
    $\mathbf{v}_{j+1} = \mathbf{r}_j / h_{j+1,j}$;
end for

Lanczos algorithm = Arnoldi + symmetry

Let A be symmetric, $A = A^*$. In the Arnoldi algorithm we form

$$\mathbf{r}_j = A\,\mathbf{v}_j - \sum_{i=1}^{j} \mathbf{v}_i(\mathbf{v}_i^* A\,\mathbf{v}_j),$$

$$\mathbf{v}_i^* A\,\mathbf{v}_j = (A\,\mathbf{v}_i)^*\,\mathbf{v}_j$$

$$\begin{aligned} A\,\mathbf{v}_i \in \mathcal{K}^{i+1}(\mathbf{x}) \quad &\Longrightarrow \quad A\,\mathbf{v}_i \perp \mathbf{v}_j \text{ for } i+1<j, \\ &\Longrightarrow \quad \mathbf{v}_i^* A\,\mathbf{v}_j = 0 \text{ for } i+1<j. \end{aligned}$$

Thus,

$$\mathbf{r}_j = A\,\mathbf{v}_j - \mathbf{v}_j(\mathbf{v}_j^* A\,\mathbf{v}_j), -\mathbf{v}_{j-1}(\mathbf{v}_{j-1}^* A\,\mathbf{v}_j) =: A\,\mathbf{v}_j - \mathbf{v}_j\alpha_j - \mathbf{v}_{j-1}\beta_{j-1}.$$

Lanczos algorithm = Arnoldi + symmetry (cont.)

$$\|\mathbf{r}_j\| = \mathbf{v}_{j+1}^*\mathbf{r}_j = \mathbf{v}_{j+1}^*(A\mathbf{v}_j - \alpha_j\mathbf{v}_j - \beta_{j-1}\mathbf{v}_{j-1}) = \mathbf{v}_{j+1}^*A\mathbf{v}_j = \bar{\beta}_j.$$

From this it follows that $\beta_j \in \mathbb{R}$.
Therefore,

$$\beta_j\mathbf{v}_{j+1} = \mathbf{r}_j, \qquad \beta_j = \|\mathbf{r}_j\|.$$

Altogether

$$A\mathbf{v}_j = \beta_{j-1}\mathbf{v}_{j-1} + \alpha_j\mathbf{v}_j + \beta_j\mathbf{v}_{j+1}.$$

Lanczos algorithm = Arnoldi + symmetry (cont.)

Gathering these equations for $j = 1, \ldots, k$ we get

$$AV_k = V_k \underbrace{\begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{k-1} \\ & & & \beta_{k-1} & \alpha_k \end{pmatrix}}_{T_k} + \beta_k[\mathbf{0}, \ldots, \mathbf{0}, \mathbf{v}_{k+1}].$$

$T_k \in \mathbb{R}^{k \times k}$ is *real symmetric*.

The equation above is called Lanczos relation.

Lanczos algorithm

```
Let $A \in \mathbb{R}^{n\times n}$ be symmetric. This algorithm computes an
orthonormal basis $V_m = [\mathbf{v}_1, \dots, \mathbf{v}_m]$ for $\mathcal{K}^m(\mathbf{x})$ where $m$ is the
smallest index such that $\mathcal{K}^m(\mathbf{x}) = \mathcal{K}^{m+1}(\mathbf{x})$, and the matrix $T_m$.
$\mathbf{v} := \mathbf{x}/\|\mathbf{x}\|$;   $V_1 = [\mathbf{v}]$;
$\mathbf{r} := A\mathbf{v}$;
$\alpha_1 := \mathbf{v}^*\mathbf{r}$;   $\mathbf{r} := \mathbf{r} - \alpha_1\mathbf{v}$;
$\beta_1 := \|\mathbf{r}\|$;
for $j = 2, 3, \dots$ do
    $\mathbf{q} = \mathbf{v}$;   $\mathbf{v} := \mathbf{r}/\beta_{j-1}$;   $V_j := [V_{j-1}, \mathbf{v}]$;
    $\mathbf{r} := A\mathbf{v} - \beta_{j-1}\mathbf{q}$;
    $\alpha_j := \mathbf{v}^*\mathbf{r}$;   $\mathbf{r} := \mathbf{r} - \alpha_j\mathbf{v}$;
    $\beta_j := \|\mathbf{r}\|$;
    if $\beta_j = 0$ then
        return $(V \in \mathbb{R}^{n\times j};\ \alpha_1, \dots, \alpha_j;\ \beta_1, \dots, \beta_{j-1})$
    end if
end for
```

Discussion of the Lanczos algorithm

- Lanczos algorithm needs just three vectors to compute T_m.
- The cost of an iteration step j does not depend on the index j.
- The storage requirement depends on j.
- Remark on very large eigenvalue problems.
- From $AV_m = V_m T_m$ and $T_m \mathbf{s}_i^{(m)} = \vartheta_i^{(m)} \mathbf{s}_i^{(m)}$ we have

$$A\mathbf{y}_i^{(m)} = \vartheta_i^{(m)} \mathbf{y}_i^{(m)}, \qquad \mathbf{y}_i^{(m)} = V_m \mathbf{s}_i^{(m)}.$$

- In general m is very large. We do not want go so far.
  When should we stop?

The Lanczos process as an iterative method

- We have seen earlier that eigenvalues at the end of the spectrum are approximated very quickly in Krylov spaces.
- Thus, only a very few iteration steps may be required to get those eigenvalues (and corresponding eigenvectors) within the desired accuracy.
- Can we check this? Can we check if $|\vartheta_i^{(j)} - \lambda_i|$ is small?

Lemma (Eigenvalue inclusion of Krylov–Bogoliubov)

Let $A \in \mathbb{R}^{n\times n}$ be symmetric. Let $\vartheta \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^n$ with $\mathbf{x} \neq \mathbf{0}$ be arbitrary. Set $\tau := \|(A - \vartheta I)\mathbf{x}\|/\|\mathbf{x}\|$. Then there is an eigenvalue of A in the interval $[\vartheta - \tau, \vartheta + \tau]$.

The Lanczos process as an iterative method (cont.)

Apply the Lemma with $\mathbf{x} = \mathbf{y}_i^{(j)} = V_j \mathbf{s}_i^{(j)}$ and $\vartheta = \vartheta_i^{(j)}$, a Ritz pair of the j-th step of the Lanczos algorithm, i.e., $T_j \mathbf{s}_i^{(j)} = \vartheta_i^{(j)} \mathbf{s}_i^{(j)}$.

$$\begin{aligned}
\|A\mathbf{y}_i^{(j)} - \vartheta_i^{(j)} \mathbf{y}_i^{(j)}\| &= \|AV_j \mathbf{s}_i^{(j)} - \vartheta_i^{(j)} V_j \mathbf{s}_i^{(j)}\| \\
&= \|(AV_j - V_j T_j)\mathbf{s}_i^{(j)}\| \quad \text{(Lanczos relation)} \\
&= \|\beta_j \mathbf{v}_{j+1} \mathbf{e}_j^* \mathbf{s}_i^{(j)}\| = |\beta_j||\mathbf{e}_j^* \mathbf{s}_i^{(j)}| = |\beta_j||s_{ji}^{(j)}|.
\end{aligned}$$

$s_{ji}^{(j)}$ is the j-th, i.e., the last element of the eigenvector $\mathbf{s}_j$ of T_j.

Exercise: Sketch an algorithm that computes the eigenvalues of a real symmetric tridiagonl matrix plus the last component of all its eigenvectors. This is the Golub–Welsch algorithm [3].

The Lanczos process as an iterative method (cont.)

Lemma $\implies$ there is an eigenvalue λ of A such that

$$|\lambda - \vartheta_i^{(j)}| \leq \beta_j |s_{ji}^{(j)}|. \tag{1}$$

It is possible to get good eigenvalue approximations even if β_j is not small!

Further, it is also known that

$$\sin \angle(\mathbf{y}_i^{(j)}, \mathbf{z}) \leq \beta_j \frac{|s_{ji}|}{\gamma}, \tag{2}$$

where $\mathbf{z}$ is the eigenvector corresponding to λ in the Lemma and γ is the gap between λ and the next eigenvalue $\neq \lambda$ of A.
γ may be estimated by $|\lambda - \vartheta_k^{(j)}|$, $k \neq i$.

Numerical example

Matrix:

$$A = \text{diag}(0, 1, 2, 3, 4, 100000),$$

Initial vector:

$$\mathbf{x} = (1, 1, 1, 1, 1, 1)^T/\sqrt{6}.$$

- The Lanczos algorithm should stop after $m = n = 6$ iteration steps with the complete Lanczos relation.
- Up to rounding error, we expect that $\beta_6 = 0$ and that the eigenvalues of T_6 are identical with those of A.

Numerical example (cont.)

$\boxed{j = 1}$

$$\alpha_1 = 16668.33333333334, \quad \beta_1 = 37267.05429136513.$$

$\boxed{j = 2}$

$$\alpha_2 = 83333.66652666384, \qquad \beta_2 = 3.464101610531258.$$

The diagonal of the eigenvalue matrix Θ_2 is:

$$\text{diag}(\Theta_2) = (1.999959999195565, 99999.99989999799)^T.$$

The last row of $\beta_2 S_2$ is

$$\beta_2 S_{2,:} = (1.414213562613906, 3.162277655014521).$$

Numerical example (cont.)

The matrix of Ritz vectors $Y_2 = Q_2 S_2$ is

$$\begin{pmatrix} -0.44722 & -2.0000 \cdot 10^{-05} \\ -0.44722 & -9.9998 \cdot 10^{-06} \\ -0.44721 & 4.0002 \cdot 10^{-10} \\ -0.44721 & 1.0001 \cdot 10^{-05} \\ -0.44720 & 2.0001 \cdot 10^{-05} \\ 4.4723 \cdot 10^{-10} & 1.0000 \end{pmatrix}$$

Numerical example (cont.)

$\boxed{j = 3}$

$$\alpha_3 = 2.000112002245340 \qquad \beta_3 = 1.183215957295906.$$

The diagonal of the eigenvalue matrix is

$$\text{diag}(\Theta_3) = (0.5857724375775532, 3.414199561869119, 99999.99999999999)^T$$

The largest eigenvalue has converged already. This is not surprising as $\lambda_2/\lambda_1 = 4 \cdot 10^{-5}$. With simple vector iteration the eigenvalues would converge with the factor $\lambda_2/\lambda_1 = 4 \cdot 10^{-5}$.

The last row of $\beta_3 S_3$ is

$$\beta_3 S_{3,:} = \left(0.83665,\ 0.83667,\ 3.74173 \cdot 10^{-5}\right).$$

Numerical example (cont.)

The matrix of Ritz vectors $Y_3 = Q_3 S_3$ is

$$\begin{pmatrix} 0.76345 & 0.13099 & 2.0000 \cdot 10^{-10} \\ 0.53983 & -0.09263 & -1.0001 \cdot 10^{-10} \\ 0.31622 & -0.31623 & -2.0001 \cdot 10^{-10} \\ 0.09262 & -0.53984 & -1.0000 \cdot 10^{-10} \\ -0.13098 & -0.76344 & 2.0001 \cdot 10^{-10} \\ -1.5864 \cdot 10^{-13} & -1.5851 \cdot 10^{-13} & 1.00000 \end{pmatrix}$$

The largest element (in modulus) of $Y_3^T Y_3$ is $\approx 3 \cdot 10^{-12}$.

The Ritz vectors (and thus the Lanczos vectors $\mathbf{q}_i$) are mutually orthogonal up to rounding error.

Numerical example (cont.)

$\boxed{j = 4}$

$$\alpha_4 = 2.000007428756856 \qquad \beta_4 = 1.014186947306611.$$

The diagonal of the eigenvalue matrix is

$$\operatorname{diag}(\Theta_4) = \begin{pmatrix} 0.1560868732577987 \\ 1.999987898940119 \\ 3.843904656006355 \\ 99999.99999999999 \end{pmatrix}.$$

The last row of $\beta_4 S_4$ is

$$\beta_4 S_{4,:} = \left(0.46017, -0.77785, -0.46018, 3.7949 \cdot 10^{-10}\right).$$

Numerical example (cont.)

The matrix of Ritz vectors $Y_4 = Q_4 S_4$ is

$$\begin{pmatrix} -0.93229 & 0.12299 & 0.03786 & -1.2 \cdot 10^{-15} \\ -0.34487 & -0.49196 & -0.10234 & 2.4 \cdot 10^{-15} \\ 2.7 \cdot 10^{-6} & -0.69693 & 2.7 \cdot 10^{-6} & -3.0 \cdot 10^{-15} \\ 0.10234 & -0.49195 & 0.34488 & -2.4 \cdot 10^{-15} \\ -0.03785 & 0.12299 & 0.93228 & 1.2 \cdot 10^{-15} \\ -8.8 \cdot 10^{-9} & 1.5 \cdot 10^{-8} & 8.8 \cdot 10^{-9} & 1.0000 \end{pmatrix}.$$

We have $\beta_4 s_{4,4} \doteq 4 \cdot 10^{-10}$. According to our previous estimates $(\vartheta_4, \mathbf{y}_4)$, $\mathbf{y}_4 = Y_4 \mathbf{e}_4$ is a very good approximation for an eigenpair of A. This is the case.

$Y_4^T Y_4$ has off-diagonal elements of the order 10^{-8}. They are in the last row/column of $Y_4^T Y_4$. So, all Ritz vectors have a small but not negligible component in the direction of the 'largest' Ritz vector.

Numerical example (cont.)

$j = 5$

$$\alpha_5 = 2.363169101109444 \quad \beta_5 = 190.5668098726485.$$

The diagonal of the eigenvalue matrix is

$$\mathrm{diag}(\Theta_5) = \begin{pmatrix} 0.04749223464478182 \\ 1.413262891598485 \\ 2.894172742223630 \\ 4.008220660846780 \\ 9.999999999999999 \cdot 10^4 \end{pmatrix}.$$

The last row of $\beta_5 S_5$ is

$$\beta_5 S_{5,:} = \left(-43.570, -111.38, 134.09, 63.495, 7.2320 \cdot 10^{-13}\right).$$

Numerical example (cont.)

The matrix of Ritz vectors Y_5 is

$$\begin{pmatrix} -0.98779 & -0.084856 & 0.049886 & 0.017056 & -1.1424 \cdot 10^{-17} \\ -0.14188 & 0.83594 & -0.21957 & -0.065468 & -7.2361 \cdot 10^{-18} \\ 0.063480 & 0.54001 & 0.42660 & 0.089943 & -8.0207 \cdot 10^{-18} \\ -0.010200 & -0.048519 & 0.87582 & -0.043531 & -5.1980 \cdot 10^{-18} \\ -0.0014168 & -0.0055339 & 0.015585 & -0.99269 & -1.6128 \cdot 10^{-17} \\ 4.3570 \cdot 10^{-4} & 0.0011138 & -0.0013409 & -6.3497 \cdot 10^{-4} & 1.0000 \end{pmatrix}$$

Evidently, the last column of Y_5 is an excellent eigenvector approximation. Notice, however, that all Ritz vectors have a relatively large ($\sim 10^{-4}$) last component.

Numerical example (cont.)

This, gives rise to quite large off-diagonal elements of $Y_5^T Y_5 - I_5 =$

$$\begin{pmatrix} 2.220\cdot10^{-16} & -1.587\cdot10^{-16} & -3.430\cdot10^{-12} & -7.890\cdot10^{-9} & -7.780\cdot10^{-4} \\ -1.587\cdot10^{-16} & -1.110\cdot10^{-16} & 1.283\cdot10^{-12} & -1.764\cdot10^{-8} & -1.740\cdot10^{-3} \\ -3.430\cdot10^{-12} & 1.283\cdot10^{-12} & 0 & 5.6800\cdot10^{-17} & -6.027\cdot10^{-8} \\ -7.890\cdot10^{-9} & -1.764\cdot10^{-8} & 5.6800\cdot10^{-17} & -2.220\cdot10^{-16} & 4.187\cdot10^{-16} \\ -7.780\cdot10^{-4} & -1.740\cdot10^{-3} & -6.027\cdot10^{-8} & 4.187\cdot10^{-16} & -1.110\cdot10^{-16} \end{pmatrix}$$

Similarly as with $j = 4$, the first four Ritz vectors satisfy the orthogonality condition very well. But they are not perpendicular to the last Ritz vector.

Numerical example (cont.)

$j = 6$

$$\alpha_6 = 99998.06336906151, \qquad \beta_6 = 396.6622037049789.$$

The diagonal of the eigenvalue matrix is

$$\mathrm{diag}(\Theta_6) = \begin{pmatrix} 0.02483483859326367 \\ 1.273835519171372 \\ 2.726145019098232 \\ 3.975161765440400 \\ 9.999842654044850 \cdot 10^{+4} \\ 1.000000000000000 \cdot 10^{+5} \end{pmatrix}.$$

The eigenvalues are not the exact ones.

There are even two copies of the largest eigenvalue of A!

Numerical example (cont.)

The last row of $\beta_6 S_6$ is

$$\beta_6 S_{6,:} = (-0.20603, 0.49322, 0.49323, 0.20604, 396.66, -8.6152 \cdot 10^{-15})$$

although theory predicts that $\beta_6 = 0$. The sixth entry of $\beta_6 S_6$ is very small, which means that the sixth Ritz value and the corresponding Ritz vector are good approximations to an eigenpair of A.

Numerical example (cont.)

In fact, eigenvalue and eigenvector are accurate to machine precision. $\beta_5 s_{6,5}$ does not predict the fifth column of Y_6 to be a good eigenvector approximation, although the angle between the fifth and sixth column of Y_6 is less than 10^{-3}. The last two columns of Y_6 are

$$\begin{pmatrix} -4.7409 \cdot 10^{-4} & -3.3578 \cdot 10^{-17} \\ 1.8964 \cdot 10^{-3} & -5.3735 \cdot 10^{-17} \\ -2.8447 \cdot 10^{-3} & -7.0931 \cdot 10^{-17} \\ 1.8965 \cdot 10^{-3} & -6.7074 \cdot 10^{-17} \\ -4.7414 \cdot 10^{-4} & -4.9289 \cdot 10^{-17} \\ -0.99999 & 1.0000 \end{pmatrix}.$$

Numerical example (cont.)

As $\beta_6 \neq 0$ one could continue the Lanczos process and compute ever larger tridiagonal matrices. If one proceeds in this way one obtains multiple copies of certain eigenvalues [2]. The corresponding values $\beta_j s_{ji}^{(j)}$ will be tiny. The corresponding Ritz vectors will be 'almost' linearly dependent.

From this numerical example we see that the problem of the Lanczos algorithm consists in the loss of orthogonality among Ritz vectors which is a consequence of the loss of orthogonality among Lanczos vectors, since $Y_j = Q_j S_j$ and S_j is unitary (up to roundoff).

Lanczos algorithm with complete reorthogonalization

To verify this claim, we rerun the Lanczos algorithm with complete reorthogonalization.

This is in fact almost the Arnoldi algorithm.

It can be accomplished by modifying line 9 in the Lanczos algorithm.

11: $\alpha_j := \mathbf{v}^*\mathbf{r}; \quad \mathbf{r} := \mathbf{r} - \alpha_j\, \mathbf{v}; \quad \mathbf{r} := \mathbf{r} - V_j(V_j^*\mathbf{r});$

The cost of the algorithm increases considerably. The j-th step of the algorithm requires now a matrix-vector multiplication and $n(2j + \mathcal{O}(1))$ floating point operations.

Numerical example revisited

With matrix and initial vector as before we get the following numbers.

$\boxed{j = 1}$

$$\alpha_1 = 16668.33333333334, \quad \beta_1 = 37267.05429136513.$$

$\boxed{j = 2}$

$$\alpha_2 = 83333.66652666384, \qquad \beta_2 = 3.464101610531258.$$

The diagonal of the eigenvalue matrix Θ_2 is:

$$\text{diag}(\Theta_2) = (1.999959999195565, 99999.99989999799)^T.$$

Numerical example revisited (cont.)

$j = 3$

$$\alpha_3 = 2.000112002240894 \qquad \beta_3 = 1.183215957295905$$

The diagonal of the eigenvalue matrix is

$$\operatorname{diag}(\Theta_3) = \begin{pmatrix} 0.5857724375677908 \\ 3.414199561859357 \\ 100000.0000000000 \end{pmatrix}.$$

Numerical example revisited (cont.)

$\boxed{j = 4}$ $\quad \alpha_4 = 2.000007428719501, \quad \beta_4 = 1.014185105707661$

$$\operatorname{diag}(\Theta_4) = \begin{pmatrix} 0.1560868732475296 \\ 1.999987898917647 \\ 3.843904655996084 \\ 99999.99999999999 \end{pmatrix}$$

The matrix of Ritz vectors $Y_4 = Q_4 S_4$ is

$$\begin{pmatrix} -0.93229 & 0.12299 & 0.03786 & -1.1767 \cdot 10^{-15} \\ -0.34487 & -0.49196 & -0.10234 & 2.4391 \cdot 10^{-15} \\ 2.7058 \cdot 10^{-6} & -0.69693 & 2.7059 \cdot 10^{-6} & 4.9558 \cdot 10^{-17} \\ 0.10233 & -0.49195 & 0.34488 & -2.3616 \cdot 10^{-15} \\ -0.03786 & 0.12299 & 0.93228 & 1.2391 \cdot 10^{-15} \\ 2.7086 \cdot 10^{-17} & 6.6451 \cdot 10^{-17} & -5.1206 \cdot 10^{-17} & 1.00000 \end{pmatrix}$$

Largest off-diagonal element of $|Y_4^T Y_4|$ is $\sim 2 \cdot 10^{-16}$

Numerical example revisited (cont.)

$\boxed{j = 5}$

$$\alpha_5 = 2.000009143040107 \qquad \beta_5 = 0.7559289460488005$$

$$\operatorname{diag}(\Theta_5) = \begin{pmatrix} 0.02483568754088384 \\ 1.273840384543175 \\ 2.726149884630423 \\ 3.975162614480485 \\ 10000.000000000000 \end{pmatrix}$$

The Ritz vectors are $Y_5 =$

$$\begin{pmatrix} -9.91 \cdot 10^{-01} & -4.62 \cdot 10^{-02} & 2.16 \cdot 10^{-02} & -6.19 \cdot 10^{-03} & -4.41 \cdot 10^{-18} \\ -1.01 \cdot 10^{-01} & 8.61 \cdot 10^{-01} & -1.36 \cdot 10^{-01} & -3.31 \cdot 10^{-02} & 1.12 \cdot 10^{-17} \\ 7.48 \cdot 10^{-02} & 4.87 \cdot 10^{-01} & 4.87 \cdot 10^{-01} & -7.48 \cdot 10^{-02} & -5.89 \cdot 10^{-18} \\ -3.31 \cdot 10^{-02} & -1.36 \cdot 10^{-01} & 8.61 \cdot 10^{-01} & -1.01 \cdot 10^{-01} & 1.07 \cdot 10^{-17} \\ 6.19 \cdot 10^{-03} & 2.16 \cdot 10^{-02} & -4.62 \cdot 10^{-02} & -9.91 \cdot 10^{-01} & 1.13 \cdot 10^{-17} \\ 5.98 \cdot 10^{-18} & 1.58 \cdot 10^{-17} & -3.39 \cdot 10^{-17} & -5.96 \cdot 10^{-17} & 1.000000\ldots \end{pmatrix}$$

Numerical example revisited (cont.)

Largest off-diagonal element of $|Y_5^T Y_5|$ is about 10^{-16}
The last row of $\beta_5 S_5$ is

$$\beta_5 S_{5,:} = \left(-0.20603, -0.49322, 0.49322, 0.20603, 2.8687 \cdot 10^{-15}\right).$$

Numerical example revisited (cont.)

$\boxed{j = 6}$

$$\alpha_6 = 2.000011428799386 \qquad \beta_6 = 4.178550866749342 \cdot 10^{-28}$$

$$\mathrm{diag}(\Theta_6) = \begin{pmatrix} 7.950307079340746 \cdot 10^{-13} \\ 1.000000000000402 \\ 2.000000000000210 \\ 3.000000000000886 \\ 4.000000000001099 \\ 9.999999999999999 \cdot 10^{4} \end{pmatrix}$$

The Ritz vectors are very accurate. Y_6 is almost the identity matrix. The largest off diagonal element of $Y_6^T Y_6$ is about 10^{-16}. Finally,

$$\beta_6 S_{6,:} = \left(5.0 \cdot 10^{-29}, -2.0 \cdot 10^{-28}, 3.0 \cdot 10^{-28}, -2.0 \cdot 10^{-28}, 5.0 \cdot 10^{-29}, 1.2 \cdot 10^{-47}\right).$$

An error analysis of the unmodified Lanczos algorithm

Let quantities $V_j, T_j, \mathbf{r}_j$, etc., be the numerically computed quantities.
Despite the gross deviation from their theoretical counterparts, they deliver fully accurate Ritz value and Ritz vector approximations. Let's write

$$AV_j - V_j T_j = \mathbf{r}_j \mathbf{e}_j^* + F_j \qquad (3)$$

where the F_j accounts for errors due to roundoff. Similarly,

$$I_j - V_j^* V_j = C_j^* + \Delta_j + C_j, \qquad (4)$$

where Δ_j is diagonal and C_j is *strictly* upper triangular.
$C_j^* + \Delta_j + C_j$ indicates deviation of the Lanczos vectors from orthogonality.

An error analysis of the unmodified Lanczos algorithm (cont.)

We assume that computations that we actually perform are accurate.

1. The tridiagonal eigenvalue problem can be solved exactly, i.e.,
$$T_j = S_j \Theta_j S_j^*, \quad S_j^* = S_j^{-1}, \quad \Theta_j = \mathrm{diag}(\vartheta_1, \ldots, \vartheta_j). \tag{5}$$
2. The orthogonality of the Lanczos vectors holds *locally*, i.e.,
$$\mathbf{v}_{i+1}^* \mathbf{v}_i = 0, \quad i = 1, \ldots, j-1, \quad \text{and} \quad \mathbf{r}_j^* \mathbf{v}_i = 0. \tag{6}$$
3. Furthermore,
$$\|\mathbf{v}_i\| = 1. \tag{7}$$

By the above assumption $\Longrightarrow$ $\Delta_j = O$ and $c_{i,i+1}^{(j)} = 0$.

Theorem by Paige

One can prove (see [1, p.266])

Theorem (Paige)

$$\mathbf{y}_i^{(j)*}\mathbf{v}_{j+1} = \frac{g_{ii}^{(j)}}{\beta_j s_{ji}^{(j)}} \tag{8}$$

$$(\vartheta_i^{(j)} - \vartheta_k^{(j)})\mathbf{y}_i^{(j)*}\mathbf{y}_k^{(j)} = g_{ii}^{(j)}\frac{s_{jk}^{(j)}}{s_{ji}^{(j)}} - g_{kk}^{(j)}\frac{s_{ji}^{(j)}}{s_{jk}^{(j)}} - (g_{ik}^{(j)} - g_{ki}^{(j)}). \tag{9}$$

where $|g_{ik}^{(j)}| \approx \varepsilon \, \|A\|$.

Interpretation

- $|\mathbf{y}_i^{(j)*}\mathbf{v}_{j+1}|$ becomes large if $\beta_j|s_{ji}^{(j)}|$ becomes small, $\mathbf{y}_i^{(j)}$ is 'good' (converged) Ritz vector. Each *new* Lanczos vector has a significant component in the direction of 'good' Ritz vectors.
  Convergence $\Longleftrightarrow$ loss of orthogonality.
- $|s_{ji}^{(j)}| \ll |s_{jk}^{(j)}|$: 'good' vs. 'bad' Ritz vectors $\mathbf{y}_i^{(j)}$ and $\mathbf{y}_k^{(j)}$.
  Two small ($\mathcal{O}(\varepsilon)$) quantities counteract each other.
  If $|\vartheta_i - \vartheta_k| = \mathcal{O}(1)$, then $|\mathbf{y}_i^*\mathbf{y}_k| \gg \varepsilon$ and 'bad' Ritz vector has significant component in direction of 'good' Ritz vector.
- $\vartheta_i - \vartheta_k = \mathcal{O}(\varepsilon), s_{ji}^{(j)} = \mathcal{O}(\varepsilon), s_{jk}^{(j)} = \mathcal{O}(\varepsilon) \Rightarrow s_{ji}^{(j)}/s_{jk}^{(j)} = \mathcal{O}(1)$.
  Right hand side of (9) as well as $|\vartheta_i - \vartheta_k|$ is $\mathcal{O}(\varepsilon)$.
  Must have $\mathbf{y}_i^{(j)*}\mathbf{y}_k^{(j)} = \mathcal{O}(1)$, i.e. almost parallel vectors.

Partial reorthogonalization

It is possible to monitor the loss of orthogonality.

It is sufficient to keep Lanczos vectors semi-orthogonal, since

$$W_j = V_j^* V_j = I_j + E, \qquad \|E\| < \sqrt{\varepsilon_M},$$

implies that tridiagonal matrix T_j is the projection of A onto the subspace $\mathcal{R}(V_j)$.

In partial reorthogonalization quantities $\omega_{j,k} \approx \mathbf{v}_k^* \mathbf{v}_j$ are monitored [4].

Reorthogonalization takes place in the j-th Lanczos step if $\max_k(\omega_{j+1,k}) > \sqrt{\varepsilon_M}$. $\mathbf{v}_{j+1}$ is orthogonalized against all vectors $\mathbf{v}_k$ with $\omega_{j+1,k} > {\varepsilon_M}^{3/4}$.

Restarting Arnoldi and Lanczos algorithms

- The number of iteration steps can be very high in Arnoldi/Lanczos algorithms.
- Iteration count depends on properties of the matrix (distribution of its eigenvalues) but also on initial vectors.
- High iteration counts entail a large memory requirement and a high amount of computation (reorthogonalization).
- Restarted Arnoldi/Lanczos algorithms reduce these costs by limiting the dimension of the search space [5].
- Iteration is stopped after a number of steps, dimension of search space is reduced, and finally the Arnoldi / Lanczos iteration is resumed.

The m-step Arnoldi iteration

Let $A \in \mathbb{R}^{n\times n}$. This algorithm executes m steps of the Arnoldi algorithm.

$\mathbf{v}_1 = \mathbf{x}/\|\mathbf{x}\|; \quad \mathbf{z} = A\mathbf{v}_1; \quad \alpha_1 = \mathbf{v}_1^*\mathbf{z};$

$\mathbf{r}_1 = \mathbf{w} - \alpha_1\mathbf{v}_1; \quad V_1 = [\mathbf{v}_1]; \quad H_1 = [\alpha_1];$

for $j = 1, \ldots, m-1$ **do**

$\quad \beta_j := \|\mathbf{r}_j\|; \quad \mathbf{v}_{j+1} = \mathbf{r}_j/\beta_j;$

$\quad V_{j+1} := [V_j, \mathbf{v}_{j+1}]; \quad \hat{H}_j := \begin{bmatrix} H_j \\ \beta_j\mathbf{e}_j^T \end{bmatrix} \in \mathbb{F}^{(j+1)\times j};$

$\quad \mathbf{z} := A\mathbf{v}_j;$

$\quad \mathbf{h} := V_{j+1}^*\mathbf{z}; \quad \mathbf{r}_{j+1} := \mathbf{z} - V_{j+1}\mathbf{h};$

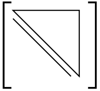
$\quad H_{j+1} := [\hat{H}_j, \mathbf{h}];$

end for

The m-step Arnoldi iteration (cont.)

After execution of m-step Arnoldi algorithm: Arnoldi relation

$$AV_m = V_m H_m + \mathbf{r}_m \mathbf{e}_m^*, \qquad H_m = \tag{10}$$



with

$$\mathbf{r}_m = \beta_m \mathbf{v}_{m+1}, \qquad \|\mathbf{v}_{m+1}\| = 1.$$

If $\beta_m = 0$ then $\mathcal{R}(V_m)$ is invariant under A. This lucky situation implies that $\sigma(H_m) \subset \sigma_m(A)$. Ritz values and Ritz vectors are eigenvalues and eigenvectors of A.

The *m*-step Arnoldi iteration (cont.)

What we can realistically hope for is $\beta_m = \|\mathbf{r}_m\|$ being small:

$$AV_m - \mathbf{r}_m\mathbf{e}_m^* = (A - \mathbf{r}_m\mathbf{v}_m^*)V_m = V_mH_m.$$

Then, $\mathcal{R}(V_m)$ is invariant under a perturbed matrix $A+E$, that differs from A by a perturbation E with $\|E\| = \|\mathbf{r}_m\| = |\beta_m|$.

From general eigenvalue theory we know that in this situation well-conditioned eigenvalues of H_m are good approximations of eigenvalues of A.

In the sequel we investigate how we can find a $\mathbf{q}_1$ such that β_m becomes small?

References

[1] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice Hall, Englewood Cliffs, NJ, 1980. (Republished 1998 by SIAM.).

[2] J. K. Cullum and R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations*, vol. 1: Theory, Birkhäuser, Boston, 1985.

[3] G. H. Golub and J. H. Welsch: *Calculation of Gauss Quadrature Rules*. Math. Comput. 23 (1969), pp. 221–230.

[4] H. Simon, *Analysis of the symmetric Lanczos algorithm with reorthogonalization methods*, Linear Algebra Appl., 61 (1984), pp. 101–132.

References (cont.)

[5] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.