An Introduction to Arnoldi's Method and Implicit Restarts

Melina Freitag

Department of Mathematical Sciences
University of Bath

Numerical Analysis Seminar
22nd April 2005

1. Outline
2. Motivation
3. Arnoldi's method
   - The Arnoldi process
   - Matrix representations of the Arnoldi process
   - Example
   - Convergence theory
4. Implicitly Restarted Arnoldi (IRA)
5. Lanczos' method

Problem

- Find a few eigenvalues and eigenvectors of $A \in \mathbb{C}^{n,n}$:

$$Av = \lambda v.$$

- let the eigenvalues be

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$$

- associated eigenvectors $v_1, v_2, \ldots, v_n$.

Problem

- Find a few eigenvalues and eigenvectors of $A \in \mathbb{C}^{n,n}$:

$$Av = \lambda v.$$

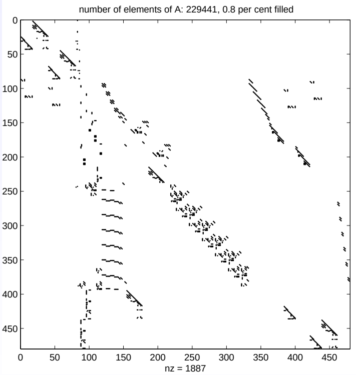
- let the eigenvalues be

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$$

- associated eigenvectors $v_1, v_2, \dots, v_n$.

Large sparse matrices

- Most large matrices that occur in applications are sparse
- Example: MATLAB test matrix `west0479`



Fill-in

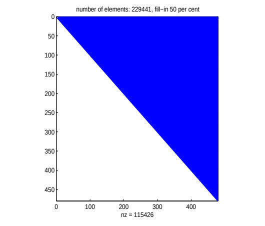


Figure: Hessenberg Reduction

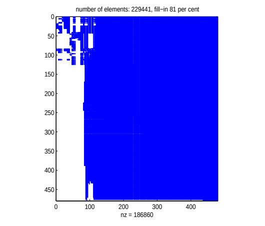


Figure: One step of QR algorithm

Eigenvalue algorithms for large sparse matrices

- Need methods that do not alter the matrix!
- Recall: Power method (Rayleigh quotient iteration): pick a vector q and form the sequence

$$q, Aq, A^2q, A^3q, \dots$$

  Then $\text{span}\{A^m q\} \rightarrow \text{span}\{v_1\}$, v_1 dominant eigenvector
- Simultaneous iteration: choose subspace $\mathcal{S}$ with $\mathcal{S} = \text{span}\{q_1^{(0)}, \dots, q_k^0\}$ and form the sequence

$$\mathcal{S}, A\mathcal{S}, A^2\mathcal{S}, A^3\mathcal{S}, \dots$$

  Then $\text{span}\{A^m\mathcal{S}\} \rightarrow \text{span}\{v_1, \dots, v_k\}$ dominant invariant subspace
- Arnoldi's method, Lanczos method

Eigenvalue algorithms for large sparse matrices

- Need methods that do not alter the matrix!
- Recall: Power method (Rayleigh quotient iteration): pick a vector q and form the sequence

  $$q, Aq, A^2q, A^3q, \dots$$

  Then $\operatorname{span}\{A^m q\} \longrightarrow \operatorname{span}\{v_1\}$, v_1 dominant eigenvector
- Simultaneous iteration: choose subspace $\mathcal{S}$ with $\mathcal{S} = \operatorname{span}\{q_1^{(0)}, \dots, q_k^{0}\}$ and form the sequence

  $$\mathcal{S}, A\mathcal{S}, A^2\mathcal{S}, A^3\mathcal{S}, \dots$$

  Then $\operatorname{span}\{A^m \mathcal{S}\} \longrightarrow \operatorname{span}\{v_1, \dots, v_k\}$ dominant invariant subspace
- Arnoldi's method, Lanczos method

Eigenvalue algorithms for large sparse matrices

- Need methods that do not alter the matrix!
- Recall: Power method (Rayleigh quotient iteration): pick a vector q and form the sequence

  $$q, Aq, A^2q, A^3q, \dots$$

  Then $\operatorname{span}\{A^m q\} \longrightarrow \operatorname{span}\{v_1\}$, v_1 dominant eigenvector
- Simultaneous iteration: choose subspace $\mathcal{S}$ with $\mathcal{S} = \operatorname{span}\{q_1^{(0)}, \dots, q_k^0\}$ and form the sequence

  $$\mathcal{S}, A\mathcal{S}, A^2\mathcal{S}, A^3\mathcal{S}, \dots$$

  Then $\operatorname{span}\{A^m\mathcal{S}\} \longrightarrow \operatorname{span}\{v_1, \dots, v_k\}$ dominant invariant subspace
- Arnoldi's method, Lanczos method

Eigenvalue algorithms for large sparse matrices

- Need methods that do not alter the matrix!
- Recall: Power method (Rayleigh quotient iteration): pick a vector q and form the sequence

  $$q, Aq, A^2q, A^3q, \dots$$

  Then $\text{span}\{A^m q\} \longrightarrow \text{span}\{v_1\}$, v_1 dominant eigenvector
- Simultaneous iteration: choose subspace $\mathcal{S}$ with $\mathcal{S} = \text{span}\{q_1^{(0)}, \dots, q_k^0\}$ and form the sequence

  $$\mathcal{S}, A\mathcal{S}, A^2\mathcal{S}, A^3\mathcal{S}, \dots$$

  Then $\text{span}\{A^m\mathcal{S}\} \longrightarrow \text{span}\{v_1, \dots, v_k\}$ dominant invariant subspace
- Arnoldi's method, Lanczos method

Motivation

- recall - power method with initial vector q computes $q, Aq, \ldots, A^k q$
- but: information from the past is lost, only $A^k q$ at step k
- idea of Arnoldi: retain past information: after k steps we have $k+1$ vectors $q, Aq, \ldots, A^k q$
- look for good eigenvector approximations in the $k+1$-dimensional subspace spanned by these vectors

Motivation

- recall - power method with initial vector q computes $q, Aq, \dots, A^k q$
- but: information from the past is lost, only $A^k q$ at step k
- idea of Arnoldi: retain past information: after k steps we have $k+1$ vectors $q, Aq, \dots, A^k q$
- look for good eigenvector approximations in the $k+1$-dimensional subspace spanned by these vectors

The Arnoldi process I

- $q, Aq, \ldots, A^k q$ usually ill-conditioned
- therefore: replace these vectors by orthogonal vectors $q_1, \ldots, q_{k+1}$ that span the same subspace
- Gram-Schmidt process with slight modification
- theory: orthogonalise $A^k q$ against $q_1, \ldots, q_k$
- praxis: orthogonalise Aq_k against $q_1, \ldots, q_k$
- produces exactly the same sequence of vectors as the Gram-Schmidt process applied to $q, Aq, \ldots, A^k q$

The Arnoldi process I

- $q, Aq, \ldots, A^k q$ usually ill-conditioned
- therefore: replace these vectors by orthogonal vectors $q_1, \ldots, q_{k+1}$ that span the same subspace
- Gram-Schmidt process with slight modification
- theory: orthogonalise $A^k q$ against $q_1, \ldots, q_k$
- praxis: orthogonalise Aq_k against $q_1, \ldots, q_k$
- produces exactly the same sequence of vectors as the Gram-Schmidt process applied to $q, Aq, \ldots, A^k q$

The Arnoldi process I

- $q, Aq, \ldots, A^k q$ usually ill-conditioned
- therefore: replace these vectors by orthogonal vectors $q_1, \ldots, q_{k+1}$ that span the same subspace
- Gram-Schmidt process with slight modification
- theory: orthogonalise $A^k q$ against $q_1, \ldots, q_k$
- praxis: orthogonalise Aq_k against $q_1, \ldots, q_k$
- produces exactly the same sequence of vectors as the Gram-Schmidt process applied to $q, Aq, \ldots, A^k q$

The Arnoldi process I

- $q, Aq, \ldots, A^k q$ usually ill-conditioned
- therefore: replace these vectors by orthogonal vectors $q_1, \ldots, q_{k+1}$ that span the same subspace
- Gram-Schmidt process with slight modification
- theory: orthogonalise $A^k q$ against $q_1, \ldots, q_k$
- praxis: orthogonalise Aq_k against $q_1, \ldots, q_k$
- produces exactly the same sequence of vectors as the Gram-Schmidt process applied to $q, Aq, \ldots, A^k q$

The Arnoldi process I

- $q, Aq, \dots, A^k q$ usually ill-conditioned
- therefore: replace these vectors by orthogonal vectors $q_1, \dots, q_{k+1}$ that span the same subspace
- Gram-Schmidt process with slight modification
- theory: orthogonalise $A^k q$ against $q_1, \dots, q_k$
- praxis: orthogonalise Aq_k against $q_1, \dots, q_k$
- produces exactly the same sequence of vectors as the Gram-Schmidt process applied to $q, Aq, \dots, A^k q$

The Arnoldi process II

Normalisation

$$q_1 = \frac{q}{\|q\|_2}$$

On subsequent steps $k = 1, 2, \dots$ take

$$\tilde{q}_{k+1} = Aq_k - \sum_{j=1}^{k} q_j h_{jk}$$

where h_{jk} is the Gram-Schmidt coefficient $h_{jk} =< Aq_k, q_j >$.
Normalisation

$$q_{k+1} = \frac{\tilde{q}_{k+1}}{h_{k+1,k}} \quad \text{where} \quad h_{k+1,k} = \|\tilde{q}_{k+1}\|_2$$

The Arnoldi algorithm

```
$q_1 = \frac{q}{\|q\|_2}$
**for** $k = 1$ to $m - 1$ **do**
    $q_{k+1} \leftarrow Aq_k$
    **for** $j = 1$ to $k$ **do**
        $h_{jk} \leftarrow q_j^* q_{k+1}$
        $q_{k+1} \leftarrow q_{k+1} - q_j h_{jk}$
    **end for**
    reorthogonalise
    $h_{k+1,k} \leftarrow \|q_{k+1}\|_2$
    **if** $h_{k+1,k} = 0$ **then**
        $\text{span}\{q_1, \dots, q_k\}$ is invariant under $A$
    **end if**
    $q_{k+1} = \frac{q_{k+1}}{h_{k+1,k}}$
**end for**
```

Krylov subspaces

Definition

For any j the space $\text{span}\{q, Aq, \ldots, A^{j-1}q\}$ is called the jth Krylov subspace associated with A and q and is denoted by $\mathcal{K}_j(A, q)$.

Theorem

Suppose $q, Aq, \ldots, A^{m-1}q$ are linearly independent and $q_1, \ldots, q_m$ are generated by the Arnoldi process. Then

1. $\text{span}\{q_1, \ldots, q_k\} = \mathcal{K}_k(A, q)$ *for* $k = 1, \ldots, m$.
2. *For* $k = 1, \ldots, m-1$, $h_{k+1,k} > 0$.
3. $h_{m+1,m} = 0$ *iff* $q, Aq, \ldots, A^m q$ *are linearly dependent which holds in turn iff the Krylov subspace* $\mathcal{K}_m(A, q)$ *is invariant under* A.

Krylov subspaces

Definition

For any j the space $\text{span}\{q, Aq, \dots, A^{j-1}q\}$ is called the jth Krylov subspace associated with A and q and is denoted by $\mathcal{K}_j(A, q)$.

Theorem

Suppose $q, Aq, \dots, A^{m-1}q$ are linearly independent and $q_1, \dots, q_m$ are generated by the Arnoldi process. Then

1. $\text{span}\{q_1, \dots, q_k\} = \mathcal{K}_k(A, q)$ *for* $k = 1, \dots, m$.
2. *For* $k = 1, \dots, m-1$, $h_{k+1,k} > 0$.
3. $h_{m+1,m} = 0$ *iff* $q, Aq, \dots, A^m q$ *are linearly dependent which holds in turn iff the Krylov subspace* $\mathcal{K}_m(A, q)$ *is invariant under* A.

Krylov subspaces

Definition

For any j the space $\text{span}\{q, Aq, \ldots, A^{j-1}q\}$ is called the jth Krylov subspace associated with A and q and is denoted by $\mathcal{K}_j(A, q)$.

Theorem

Suppose $q, Aq, \ldots, A^{m-1}q$ are linearly independent and $q_1, \ldots, q_m$ are generated by the Arnoldi process. Then

1. *$\text{span}\{q_1, \ldots, q_k\} = \mathcal{K}_k(A, q)$ for $k = 1, \ldots, m$.*
2. *For $k = 1, \ldots, m-1$, $h_{k+1,k} > 0$.*
3. *$h_{m+1,m} = 0$ iff $q, Aq, \ldots, A^m q$ are linearly dependent which holds in turn iff the Krylov subspace $\mathcal{K}_m(A, q)$ is invariant under A.*

Krylov subspaces

Definition

For any j the space $\text{span}\{q, Aq, \ldots, A^{j-1}q\}$ is called the jth Krylov subspace associated with A and q and is denoted by $\mathcal{K}_j(A, q)$.

Theorem

Suppose $q, Aq, \ldots, A^{m-1}q$ are linearly independent and $q_1, \ldots, q_m$ are generated by the Arnoldi process. Then

1. *$\text{span}\{q_1, \ldots, q_k\} = \mathcal{K}_k(A, q)$ for $k = 1, \ldots, m$.*
2. *For $k = 1, \ldots, m-1$, $h_{k+1,k} > 0$.*
3. *$h_{m+1,m} = 0$ iff $q, Aq, \ldots, A^m q$ are linearly dependent which holds in turn iff the Krylov subspace $\mathcal{K}_m(A, q)$ is invariant under A.*

Krylov subspaces

Definition

For any j the space $\text{span}\{q, Aq, \ldots, A^{j-1}q\}$ is called the jth Krylov subspace associated with A and q and is denoted by $\mathcal{K}_j(A, q)$.

Theorem

Suppose $q, Aq, \ldots, A^{m-1}q$ are linearly independent and $q_1, \ldots, q_m$ are generated by the Arnoldi process. Then

1. *$\text{span}\{q_1, \ldots, q_k\} = \mathcal{K}_k(A, q)$ for $k = 1, \ldots, m$.*
2. *For $k = 1, \ldots, m-1$, $h_{k+1,k} > 0$.*
3. *$h_{m+1,m} = 0$ iff $q, Aq, \ldots, A^m q$ are linearly dependent which holds in turn iff the Krylov subspace $\mathcal{K}_m(A, q)$ is invariant under A.*

Matrix representation I

Arnoldi process:

$$Aq_k = \sum_{j=1}^{k+1} q_j h_{jk}$$

$$\Rightarrow AQ_m = Q_{m+1} H_{m+1,m}$$

where

$$Q_m = [q_1, \ldots, q_m] \in \mathbb{C}^{n,m} \quad \text{and}$$

$$H_{m+1,m} = \begin{bmatrix} h_{11} & h_{22} & \cdots & h_{1,m-1} & h_{1m} \\ h_{21} & h_{22} & \cdots & h_{2,m-1} & h_{2m} \\ 0 & h_{32} & \cdots & h_{3,m-1} & h_{3m} \\ \vdots & & \ddots & & \vdots \\ 0 & & & h_{m,m-1} & h_{mm} \\ 0 & 0 & & 0 & h_{m+1,m} \end{bmatrix} \in \mathbb{C}^{m+1,m}$$

Matrix representation II

Arnoldi in matrix form

The Arnoldi process can be written in the form

$$AQ_m = Q_m H_m + q_{m+1} h_{m+1,m} e_m^T \tag{1}$$

where H_m is square upper Hessenberg.

Proposition

Suppose $q_1, \ldots q_{m+1}$ are orthonormal vectors

$$Q_m = [q_1, \ldots, q_m]$$

and H_m is an upper Hessenberg matrix with $h_{j+1,j} > 0$ for $j = 1, \ldots, m$. If they satisfy (1) they must be vectors produced by the Arnoldi process.

Matrix representation II

Arnoldi in matrix form

The Arnoldi process can be written in the form

$$AQ_m = Q_m H_m + q_{m+1} h_{m+1,m} e_m^T \tag{1}$$

where H_m is square upper Hessenberg.

Proposition

Suppose $q_1, \ldots q_{m+1}$ are orthonormal vectors

$$Q_m = [q_1, \ldots, q_m]$$

and H_m is an upper Hessenberg matrix with $h_{j+1,j} > 0$ for $j = 1, \ldots, m$. If they satisfy (1) they must be vectors produced by the Arnoldi process.

How can Arnoldi deliver eigenvalues?

Theorem

If $h_{m+1,m} = 0$ then $q, Aq, \ldots, A^m q$ are linearly dependent, i.e. span$\{Q_m\}$ is invariant under A and

$$AQ_m = Q_m H_m$$

and the eigenvalues of H_m are the eigenvalues of A.

Remarks

1. for n steps we get $q_1, \ldots, q_n$ is an orthonormal basis of $\mathbb{C}^n$ and $Q_n^{-1} A Q_n = H_n$ is a unitary similarity transform
2. want to stop after m steps $m << n$
3. $h_{m+1,m} = 0$ only if we are lucky

How can Arnoldi deliver eigenvalues?

Theorem

If $h_{m+1,m} = 0$ then $q, Aq, \dots, A^m q$ are linearly dependent, i.e. $\text{span}\{Q_m\}$ is invariant under A and

$$AQ_m = Q_m H_m$$

and the eigenvalues of H_m are the eigenvalues of A.

Remarks

1. for n steps we get $q_1, \dots, q_n$ is an orthonormal basis of $\mathbb{C}^n$ and $Q_n^{-1} A Q_n = H_n$ is a unitary similarity transform
2. want to stop after m steps $m << n$
3. $h_{m+1,m} = 0$ only if we are lucky

How can Arnoldi deliver eigenvalues?

Theorem

If $h_{m+1,m} = 0$ then $q, Aq, \ldots, A^m q$ are linearly dependent, i.e. $\text{span}\{Q_m\}$ is invariant under A and

$$AQ_m = Q_m H_m$$

and the eigenvalues of H_m are the eigenvalues of A.

Remarks

1. for n steps we get $q_1, \ldots, q_n$ is an orthonormal basis of $\mathbb{C}^n$ and $Q_n^{-1} A Q_n = H_n$ is a unitary similarity transform
2. want to stop after m steps $m << n$
3. $h_{m+1,m} = 0$ only if we are lucky

How can Arnoldi deliver eigenvalues?

Theorem

If $h_{m+1,m} = 0$ then $q, Aq, \ldots, A^m q$ are linearly dependent, i.e. span$\{Q_m\}$ is invariant under A and

$$AQ_m = Q_m H_m$$

and the eigenvalues of H_m are the eigenvalues of A.

Remarks

1. for n steps we get $q_1, \ldots, q_n$ is an orthonormal basis of $\mathbb{C}^n$ and $Q_n^{-1} A Q_n = H_n$ is a unitary similarity transform
2. want to stop after m steps $m << n$
3. $h_{m+1,m} = 0$ only if we are lucky

How can Arnoldi deliver eigenvalues?

Theorem

If $h_{m+1,m} = 0$ then $q, Aq, \dots, A^m q$ are linearly dependent, i.e. $\mathrm{span}\{Q_m\}$ is invariant under A and

$$AQ_m = Q_m H_m$$

and the eigenvalues of H_m are the eigenvalues of A.

Remarks

1. for n steps we get $q_1, \dots, q_n$ is an orthonormal basis of $\mathbb{C}^n$ and $Q_n^{-1} A Q_n = H_n$ is a unitary similarity transform
2. want to stop after m steps $m << n$
3. $h_{m+1,m} = 0$ only if we are lucky

Error estimates (Ritz values and Ritz vectors)

Theorem

Let Q_m, H_m and $h_{m+1,m}$ be generated by the Arnoldi process. Let μ be an eigenvalue of H_m with associated eigenvector x normalised so that $\|x\|_2=1$. Let $y = Q_m x \in \mathbb{C}^n$ (also with $\|y\|_2=1$). Then

$$\|Ay - \mu y\|_2 = |h_{m+1,m}||x_m|,$$

where x_m denotes the mth (and last) component of x.

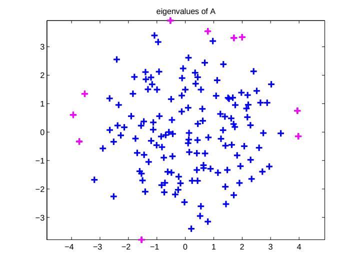
Definition

The vector y is called Ritz-vector of A associated with the subspace $\mathcal{K}_m(A, q)$ and the μ is called Ritz value.

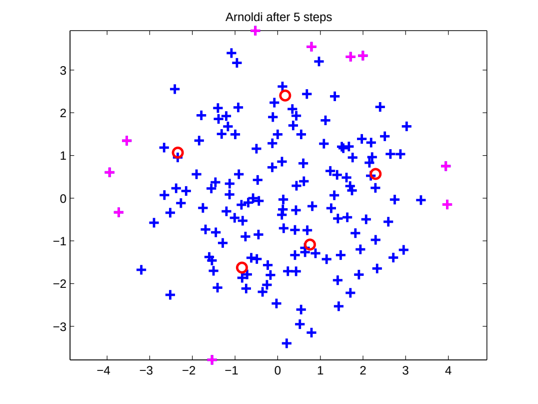
Example

random complex matrix of dimension $n = 144$ generated in MATLAB:

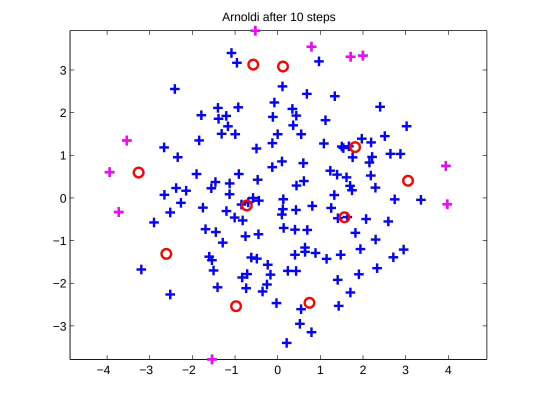
```
G=numgrid('N',14);B=delsq(G);A=sprandn(B)+i*sprandn(B)
```



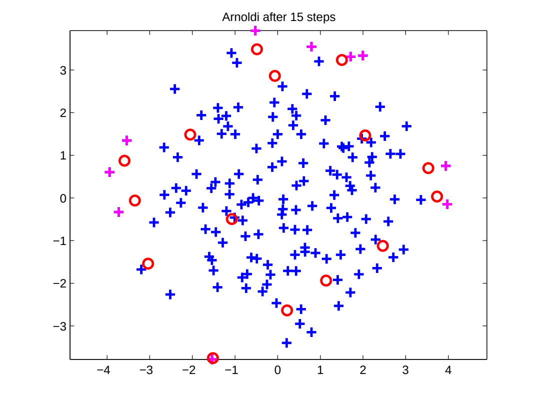
after 5 Arnoldi steps



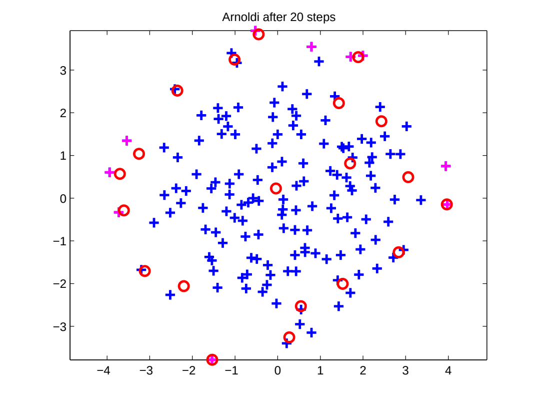
after 10 Arnoldi steps



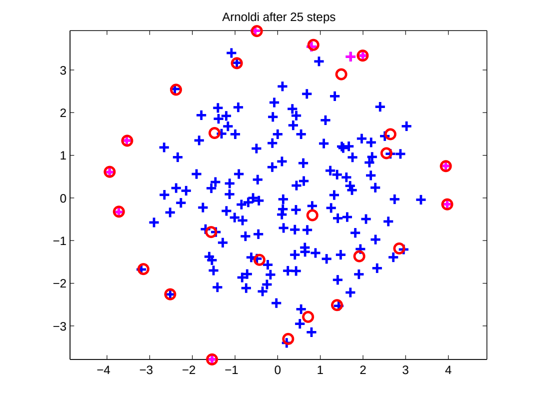
after 15 Arnoldi steps



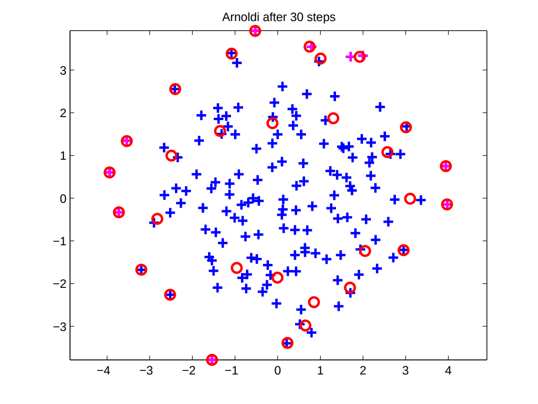
after 20 Arnoldi steps



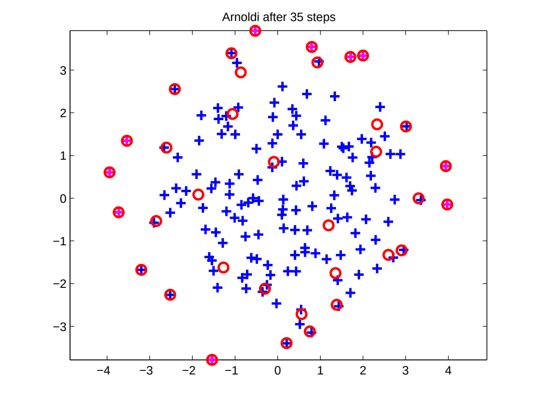
after 25 Arnoldi steps



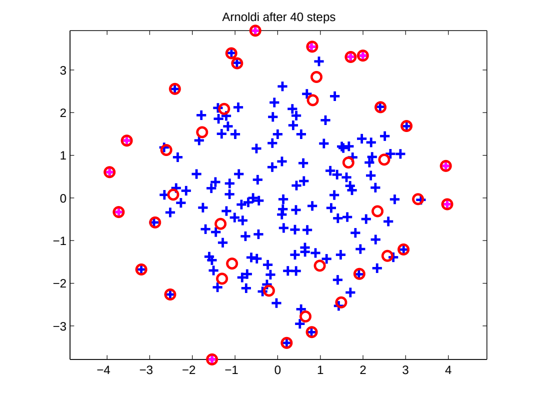
after 30 Arnoldi steps



after 35 Arnoldi steps



after 40 Arnoldi steps



Error estimates and true error

RESIDUAL $\lvert h_{41,40} \rvert \lvert x_{40} \rvert$	TRUE ERROR
$2.0 \cdot 10^{-6}$	$1.1 \cdot 10^{-7}$
$7.6 \cdot 10^{-4}$	$1.1 \cdot 10^{-4}$
$2.1 \cdot 10^{-4}$	$4.1 \cdot 10^{-5}$
$3.9 \cdot 10^{-4}$	$1.4 \cdot 10^{-5}$
$3.4 \cdot 10^{-4}$	$4.7 \cdot 10^{-5}$
$5.1 \cdot 10^{-3}$	$7.7 \cdot 10^{-4}$

Table: Best six Ritz pair residuals and corresponding Ritz value errors, $h_{41,40} \approx 2.8$

Why does Arnoldi approximate the outer eigenvalues?

- Let A have linearly independent eigenvectors $v_1, \ldots, v_n$ and associated eigenvalues $\lambda_1, \ldots, \lambda_n$
- starting vector $q = \sum_{i=1}^{n} c_i v_i$ and

$$\mathcal{K}_m(A, q) = \{p(A)q | q \in \mathcal{P}_{m-1}\}$$

$$\Rightarrow v = p(A)q = \sum_{i=1}^{n} c_i p(\lambda_i) v_i$$

- if $\exists p \in \mathcal{P}_{m-1}$ s.t. $p(\lambda_j)$ is much larger than $p(\lambda_i)$, $i \neq j$ then $\mathcal{K}_m(A, q)$ will contain a vector close to the eigenvector v_j

Why does Arnoldi approximate the outer eigenvalues?

- Let A have linearly independent eigenvectors $v_1, \ldots, v_n$ and associated eigenvalues $\lambda_1, \ldots, \lambda_n$
- starting vector $q = \sum_{i=1}^{n} c_i v_i$ and

$$\mathcal{K}_m(A, q) = \{p(A)q | q \in \mathcal{P}_{m-1}\}$$

$$\Rightarrow v = p(A)q = \sum_{i=1}^{n} c_i p(\lambda_i) v_i$$

- if $\exists p \in \mathcal{P}_{m-1}$ s.t. $p(\lambda_j)$ is much larger than $p(\lambda_i)$, $i \neq j$ then $\mathcal{K}_m(A, q)$ will contain a vector close to the eigenvector v_j

Why does Arnoldi approximate the outer eigenvalues?

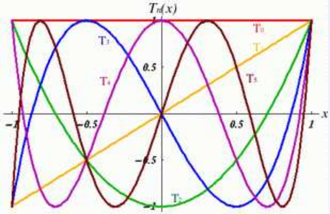
- Let A have linearly independent eigenvectors $v_1, \ldots, v_n$ and associated eigenvalues $\lambda_1, \ldots, \lambda_n$
- starting vector $q = \sum_{i=1}^{n} c_i v_i$ and

$$\mathcal{K}_m(A, q) = \{p(A)q | q \in \mathcal{P}_{m-1}\}$$

$$\Rightarrow v = p(A)q = \sum_{i=1}^{n} c_i p(\lambda_i) v_i$$

- if $\exists p \in \mathcal{P}_{m-1}$ s.t. $p(\lambda_j)$ is much larger than $p(\lambda_i)$, $i \neq j$ then $\mathcal{K}_m(A, q)$ will contain a vector close to the eigenvector v_j

Chebyshev polynomials



$$T_m(t) = \cos(m \cos^{-1}(t)), \quad |t| \leq 1$$

$$T_m(t) = \frac{1}{2}\left[(t + \sqrt{t^2 - 1})^m + (t + \sqrt{t^2 - 1})^{-m}\right] \geq \frac{1}{2} t^m, \quad |t| \geq 1$$

Convergence theory I

Let

$$A = A^T, \quad \lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n$$

Then

$$v = p_{m-1}(A)q = \sum_{i=1}^{n} c_i p_{m-1}(\lambda_i) v_i = c_1 p_{m-1}(\lambda_1) v_1 + \sum_{i=2}^{n} c_i p_{m-1}(\lambda_i) v_i$$

and

$$\|w_m - v_1\| \leq \sum_{i=2}^{n} \left\| \frac{c_i p_{m-1}(\lambda_i)}{c_1 p_{m-1}(\lambda_1)} \right\|.$$

Find

$$\min_{\substack{\tilde{p}_{m-1} \in \mathcal{P}_{m-1} \\ \tilde{p}(\lambda_1)=1}} \max_{\lambda_i \in [\lambda_n, \lambda_2]} \tilde{p}_{m-1}(\lambda), \quad \tilde{p}_{m-1}(\lambda) = \frac{p_{m-1}(\lambda)}{p_{m-1}(\lambda_1)}$$

Convergence theory II

Theorem (approximation theory)

The minimum

$$\min_{\substack{\tilde{p}_{m-1}\in\mathcal{P}_{m-1}\\ \tilde{p}(\lambda_1)=1}} \max_{\lambda_i\in[\lambda_n,\lambda_2]} \tilde{p}_{m-1}(\lambda), \quad \tilde{p}_{m-1}(\lambda) = \frac{p_{m-1}(\lambda)}{p_{m-1}(\lambda_1)}$$

is reached by the (scaled) Chebyshev polynomial

$$\hat{T}_{m-1} = \frac{T_{m-1}(1+2\frac{\lambda-\lambda_2}{\lambda_2-\lambda_n})}{T_{m-1}(1+2\frac{\lambda_1-\lambda_2}{\lambda_2-\lambda_n})}$$

$$\|w_m - v_1\| \le C\frac{T_{m-1}(1+2\frac{\lambda-\lambda_2}{\lambda_2-\lambda_n})}{T_{m-1}(1+2\frac{\lambda_1-\lambda_2}{\lambda_2-\lambda_n})} \le C\left(1+2\frac{\lambda_1-\lambda_2}{\lambda_2-\lambda_n}\right)^{-m}, m = 1, 2, \ldots$$

Shift-and-invert strategy

- can be used in conjunction with several iterative eigenvalue methods
- Power method, simultaneous iteration, Arnoldi and Lanzcos methods compute eigenvalues of A of largest modulus
- suppose we want eigenvalues near some target τ
- use new matrix $(A - \tau I)^{-1}$ that has the same eigenvectors and invariant subspaces as A, but different eigenvalues
- largest eigenvalue of $(A - \tau I)^{-1}$ corresponds to the eigenvalues of A that are closest to τ

Shift-and-invert strategy

- can be used in conjunction with several iterative eigenvalue methods
- Power method, simultaneous iteration, Arnoldi and Lanzcos methods compute eigenvalues of A of largest modulus
- suppose we want eigenvalues near some target τ
- use new matrix $(A-\tau I)^{-1}$ that has the same eigenvectors and invariant subspaces as A, but different eigenvalues
- largest eigenvalue of $(A-\tau I)^{-1}$ corresponds to the eigenvalues of A that are closest to τ

Shift-and-invert strategy

- can be used in conjunction with several iterative eigenvalue methods
- Power method, simultaneous iteration, Arnoldi and Lanzcos methods compute eigenvalues of A of largest modulus
- suppose we want eigenvalues near some target τ
- use new matrix $(A - \tau I)^{-1}$ that has the same eigenvectors and invariant subspaces as A, but different eigenvalues
- largest eigenvalue of $(A - \tau I)^{-1}$ corresponds to the eigenvalues of A that are closest to τ

The implicitely restarted Arnoldi process

- convergence depends very much on starting vector q
- find a better starting vector $\hat{q}$ (less storage)
- assume A is semisimple with linearly independent eigenvectors $v_1, \ldots, v_n$ and eigenvalues $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$ and $|\lambda_k| > |\lambda_{k+1}|$.
- Write $q = \sum_{i=1}^{n} c_i v_i$ and find $\hat{q}$, s.t. $\hat{q} = \sum_{i=1}^{n} \hat{c}_i v_i$ where $\hat{c}_1, \ldots, \hat{c}_k$ have been augmented and $\hat{c}_{k+1}, \ldots, \hat{c}_n$ have been diminished.
- Suppose $\hat{q} = p(A)q$ where p is some polynomial, then

$$\hat{q} = \sum_{i=1}^{n} c_i p(\lambda_i) v_i$$

and choose p s.t. $p(\lambda_1), \ldots, p(\lambda_k)$ are large in comparison with $p(\lambda_{k+1}), \ldots, p(\lambda_n)$

The implicitely restarted Arnoldi process

- convergence depends very much on starting vector q
- find a better starting vector $\hat{q}$ (less storage)
- assume A is semisimple with linearly independent eigenvectors $v_1, \ldots, v_n$ and eigenvalues $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$ and $|\lambda_k| > |\lambda_{k+1}|$.
- Write $q = \sum_{i=1}^{n} c_i v_i$ and find $\hat{q}$, s.t. $\hat{q} = \sum_{i=1}^{n} \hat{c}_i v_i$ where $\hat{c}_1, \ldots, \hat{c}_k$ have been augmented and $\hat{c}_{k+1}, \ldots, \hat{c}_n$ have been diminished.
- Suppose $\hat{q} = p(A)q$ where p is some polynomial, then

$$\hat{q} = \sum_{i=1}^{n} c_i p(\lambda_i) v_i$$

and choose p s.t. $p(\lambda_1), \ldots, p(\lambda_k)$ are large in comparison with $p(\lambda_{k+1}), \ldots, p(\lambda_n)$

The implicitely restarted Arnoldi process

- convergence depends very much on starting vector q
- find a better starting vector $\hat{q}$ (less storage)
- assume A is semisimple with linearly independent eigenvectors $v_1, \dots, v_n$ and eigenvalues $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$ and $|\lambda_k| > |\lambda_{k+1}|$.
- Write $q = \sum_{i=1}^{n} c_i v_i$ and find $\hat{q}$, s.t. $\hat{q} = \sum_{i=1}^{n} \hat{c}_i v_i$ where $\hat{c}_1, \dots, \hat{c}_k$ have been augmented and $\hat{c}_{k+1}, \dots, \hat{c}_n$ have been diminished.
- Suppose $\hat{q} = p(A)q$ where p is some polynomial, then

$$\hat{q} = \sum_{i=1}^{n} c_i p(\lambda_i) v_i$$

and choose p s.t. $p(\lambda_1), \dots, p(\lambda_k)$ are large in comparison with $p(\lambda_{k+1}), \dots, p(\lambda_n)$

Iteration of the IRA I

- make Arnoldi runs of length $m = k + j$ (to find k eigenvalues)

$$AQ_m = Q_m H_m + q_{m+1} h_{m+1,m} e_m^T$$

- find m eigenvalues $|\mu_1| \geq |\mu_2| \geq \ldots \geq |\mu_m|$ of H_m
- largest ones $|\mu_1| \geq |\mu_2| \geq \ldots \geq |\mu_k|$ are estimates for the largest eigenvalues of A, the smallest ones $|\mu_{k+1}| \geq |\ldots \geq |\mu_m|$ approximate the undesired part of the spectrum
- perform j iterations of the QR algorithm on H_m using j shifts $\nu_1, \ldots, \nu_j$ in the region of the spectrum we want to suppress (most popular choice is $\nu_1 = \mu_{k+1}, \ldots, \nu_j = \mu_m$)

Iteration of the IRA I

- make Arnoldi runs of length $m = k + j$ (to find k eigenvalues)

$$AQ_m = Q_m H_m + q_{m+1} h_{m+1,m} e_m^T$$

- find m eigenvalues $|\mu_1| \geq |\mu_2| \geq \ldots \geq |\mu_m|$ of H_m
- largest ones $|\mu_1| \geq |\mu_2| \geq \ldots \geq |\mu_k|$ are estimates for the largest eigenvalues of A, the smallest ones $|\mu_{k+1}| \geq |\ldots \geq |\mu_m|$ approximate the undesired part of the spectrum
- perform j iterations of the QR algorithm on H_m using j shifts $\nu_1, \ldots, \nu_j$ in the region of the spectrum we want to suppress (most popular choice is $\nu_1 = \mu_{k+1}, \ldots, \nu_j = \mu_m$)

Iteration of the IRA I

- make Arnoldi runs of length $m = k + j$ (to find k eigenvalues)

$$AQ_m = Q_m H_m + q_{m+1} h_{m+1,m} e_m^T$$

- find m eigenvalues $|\mu_1| \geq |\mu_2| \geq \ldots \geq |\mu_m|$ of H_m
- largest ones $|\mu_1| \geq |\mu_2| \geq \ldots \geq |\mu_k|$ are estimates for the largest eigenvalues of A, the smallest ones $|\mu_{k+1}| \geq |\ldots \geq |\mu_m|$ approximate the undesired part of the spectrum
- perform j iterations of the QR algorithm on H_m using j shifts $\nu_1, \ldots, \nu_j$ in the region of the spectrum we want to suppress (most popular choice is $\nu_1 = \mu_{k+1}, \ldots, \nu_j = \mu_m$)

Iteration of the IRA I

- make Arnoldi runs of length $m = k + j$ (to find k eigenvalues)

$$AQ_m = Q_m H_m + q_{m+1} h_{m+1,m} e_m^T$$

- find m eigenvalues $|\mu_1| \geq |\mu_2| \geq \ldots \geq |\mu_m|$ of H_m
- largest ones $|\mu_1| \geq |\mu_2| \geq \ldots \geq |\mu_k|$ are estimates for the largest eigenvalues of A, the smallest ones $|\mu_{k+1}| \geq | \ldots \geq |\mu_m|$ approximate the undesired part of the spectrum
- perform j iterations of the QR algorithm on H_m using j shifts $\nu_1, \ldots, \nu_j$ in the region of the spectrum we want to suppress (most popular choice is $\nu_1 = \mu_{k+1}, \ldots, \nu_j = \mu_m$)

Iteration of the IRA II

- QR algorithm with shifts is given by

$$p(H_m) = V_m R_m, \quad \text{where} \quad p(z) = (z - \nu_1) \cdots (z - \nu_j)$$

- combined effect is a similarity transform

$$\hat{H}_m = V_m^{-1} H_m V_m$$

- Let $\hat{Q}_m = Q_m V_m$ and $\hat{q}_1$ be the first column of $\hat{Q}_m$.
- Carry out another m Arnoldi steps with $\hat{q}_1$ as starting vector

Iteration of the IRA II

- QR algorithm with shifts is given by

$$p(H_m) = V_m R_m, \quad \text{where} \quad p(z) = (z - \nu_1) \cdots (z - \nu_j)$$

- combined effect is a similarity transform

$$\hat{H}_m = V_m^{-1} H_m V_m$$

- Let $\hat{Q}_m = Q_m V_m$ and $\hat{q}_1$ be the first column of $\hat{Q}_m$.
- Carry out another m Arnoldi steps with $\hat{q}_1$ as starting vector

Iteration of the IRA II

- QR algorithm with shifts is given by

$$p(H_m) = V_m R_m, \quad \text{where} \quad p(z) = (z-\nu_1)\cdots(z-\nu_j)$$

- combined effect is a similarity transform

$$\hat{H}_m = V_m^{-1} H_m V_m$$

- Let $\hat{Q}_m = Q_m V_m$ and $\hat{q}_1$ be the first column of $\hat{Q}_m$.
- Carry out another m Arnoldi steps with $\hat{q}_1$ as starting vector

Iteration of the IRA II

- QR algorithm with shifts is given by

$$p(H_m) = V_m R_m, \quad \text{where} \quad p(z) = (z - \nu_1) \cdots (z - \nu_j)$$

- combined effect is a similarity transform

$$\hat{H}_m = V_m^{-1} H_m V_m$$

- Let $\hat{Q}_m = Q_m V_m$ and $\hat{q}_1$ be the first column of $\hat{Q}_m$.
- Carry out another m Arnoldi steps with $\hat{q}_1$ as starting vector

Iteration of the IRA III

- don't need to start Arnoldi from scratch, since

$$A\hat{Q}_m = \hat{Q}_m \hat{H}_m + q_{m+1} h_{m+1,m} e_m^T V_m,$$

where $e_m^T V_m$ has exactly $m - j - 1$ leading zeros.

- drop last j entries to get

$$A\hat{Q}_k = \hat{Q}_k \hat{H}_k + \underbrace{(\hat{q}_{k+1} \hat{h}_{k+1,k} + q_{m+1} h_{m+1,m} V_{m,k})}_{\|\cdot\| = h_{k+1,k}^{\text{new}}} e_k^T,$$

- do another j steps of Arnoldi

Iteration of the IRA III

- don't need to start Arnoldi from scratch, since

$$A\hat{Q}_m = \hat{Q}_m \hat{H}_m + q_{m+1} h_{m+1,m} e_m^T V_m,$$

where $e_m^T V_m$ has exactly $m - j - 1$ leading zeros.

- drop last j entries to get

$$A\hat{Q}_k = \hat{Q}_k \hat{H}_k + \underbrace{(\hat{q}_{k+1} \hat{h}_{k+1,k} + q_{m+1} h_{m+1,m} V_{m,k})}_{\|\cdot\| = h_{k+1,k}^{\text{new}}} e_k^T,$$

- do another j steps of Arnoldi

Iteration of the IRA III

- don't need to start Arnoldi from scratch, since

$$A\hat{Q}_m = \hat{Q}_m \hat{H}_m + q_{m+1} h_{m+1,m} e_m^T V_m,$$

  where $e_m^T V_m$ has exactly $m - j - 1$ leading zeros.
- drop last j entries to get

$$A\hat{Q}_k = \hat{Q}_k \hat{H}_k + \underbrace{(\hat{q}_{k+1} \hat{h}_{k+1,k} + q_{m+1} h_{m+1,m} V_{m,k})}_{\|\cdot\| = h_{k+1,k}^{\text{new}}} e_k^T,$$

- do another j steps of Arnoldi

IRA Algorithm

```
take $k$ Arnoldi steps to produce $H_k$, $Q_k$
for $ii = 1$ to $imax$ do
    take another $j$ Arnoldi steps to produce $H_m$, $Q_m$, $m = k + j$
    find $j$ shifts $\nu_1, \ldots, \nu_j$
    take $j$ steps of the QR algorithm on $H_m$ with shifts $\nu_1, \ldots, \nu_j$
    to obtain the transformation matrix $V_m$
    $H_m \leftarrow V_m^{-1} H_m V_m$
    $Q_m = Q_m V_m$
    if $|h_{k+1,k}| \leq \text{tol}$ then
        span$\{q_1, \ldots, q_k\}$ is invariant under $A$
    end if
end for
```

Why IRA works

- want new starting vector $\hat{q} = p(A)q$, where p is choosen to suppress unwanted eigenvectors and enhance wanted ones
- Recall

$$AQ_m = Q_m H_m + q_{m+1} h_{m+1,m} e_m^T$$

and insert shift

$$(A - \nu_1 I)Q_m = Q_m(H_m - \nu_1 I) + E_1$$

where E_1 is zero except for the last column,

$$(A - \nu_2 I)(A - \nu_1 I)Q_m = Q_m(H_m - \nu_1 I)(H_m - \nu_1 I) + E_2$$

where E_2 is zero except for the last two columns.

Why IRA works

- want new starting vector $\hat{q} = p(A)q$, where p is choosen to suppress unwanted eigenvectors and enhance wanted ones
- Recall

$$AQ_m = Q_m H_m + q_{m+1} h_{m+1,m} e_m^T$$

and insert shift

$$(A - \nu_1 I)Q_m = Q_m(H_m - \nu_1 I) + E_1$$

where E_1 is zero except for the last column,

$$(A - \nu_2 I)(A - \nu_1 I)Q_m = Q_m(H_m - \nu_1 I)(H_m - \nu_1 I) + E_2$$

where E_2 is zero except for the last two columns.

Why IRA works continued

Theorem

Suppose $AQ_m = Q_m H_m + q_{m+1} h_{m+1,m} e_m^T$ *and let* p *be a polynomial of degree* $j \leq m$. *Then*

$$p(A)Q_m = Q_m p(H_m) + E_j,$$

where $E_j \in \mathcal{C}^{n,m}$ *is identically zero except in the last* j *columns.*

new starting vector

- $p(A)Q_m = Q_m V_m R_m + E_j$
- $p(A)Q_m e_1 = \hat{Q}_m R_m$
- $\hat{q}_1 = \frac{1}{r_{11}} p(A) q_1$

Why IRA works continued

Theorem

Suppose $AQ_m = Q_m H_m + q_{m+1} h_{m+1,m} e_m^T$ *and let* p *be a polynomial of degree* $j \leq m$. *Then*

$$p(A)Q_m = Q_m p(H_m) + E_j,$$

where $E_j \in \mathcal{C}^{n,m}$ *is identically zero except in the last* j *columns.*

new starting vector

- $p(A)Q_m = Q_m V_m R_m + E_j$
- $p(A)Q_m e_1 = \hat{Q}_m R_m$
- $\hat{q}_1 = \frac{1}{r_{11}} p(A) q_1$

Why IRA works continued

Theorem

Suppose $AQ_m = Q_m H_m + q_{m+1} h_{m+1,m} e_m^T$ *and let* p *be a polynomial of degree* $j \leq m$. *Then*

$$p(A)Q_m = Q_m p(H_m) + E_j,$$

where $E_j \in \mathcal{C}^{n,m}$ *is identically zero except in the last* j *columns.*

new starting vector

- $p(A)Q_m = Q_m V_m R_m + E_j$
- $p(A)Q_m e_1 = \hat{Q}_m R_m$
- $\hat{q}_1 = \frac{1}{r_{11}} p(A) q_1$

Why IRA works continued

Theorem

Suppose $AQ_m = Q_m H_m + q_{m+1} h_{m+1,m} e_m^T$ *and let* p *be a polynomial of degree* $j \leq m$. *Then*

$$p(A)Q_m = Q_m p(H_m) + E_j,$$

where $E_j \in \mathcal{C}^{n,m}$ *is identically zero except in the last* j *columns.*

new starting vector

- $p(A)Q_m = Q_m V_m R_m + E_j$
- $p(A)Q_m e_1 = \hat{Q}_m R_m$
- $\hat{q}_1 = \frac{1}{r_{11}} p(A) q_1$

Why IRA works continued

Theorem

Suppose $AQ_m = Q_m H_m + q_{m+1} h_{m+1,m} e_m^T$ *and let* p *be a polynomial of degree* $j \leq m$. *Then*

$$p(A)Q_m = Q_m p(H_m) + E_j,$$

where $E_j \in \mathcal{C}^{n,m}$ *is identically zero except in the last* j *columns.*

new starting vector

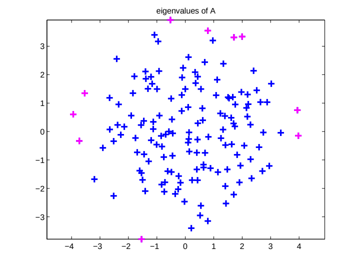
- $p(A)Q_m = Q_m V_m R_m + E_j$
- $p(A)Q_m e_1 = \hat{Q}_m R_m$
- $\hat{q}_1 = \frac{1}{r_{11}} p(A) q_1$

Tests

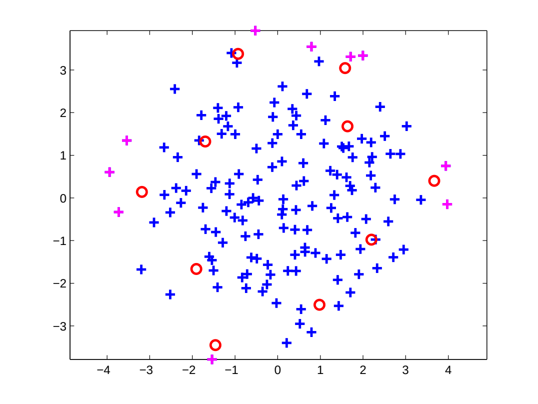
random complex matrix of dimension $n = 144$ generated in MATLAB:

`G=numgrid('N',14);B=delsq(G);A=sprandn(B)+i*sprandn(B)`,

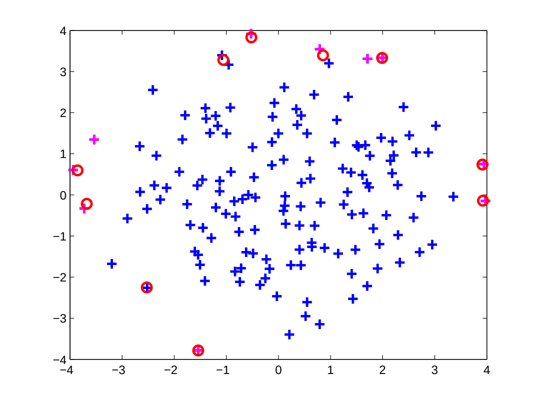
`tol = 0.1`



first 10 Arnoldi steps

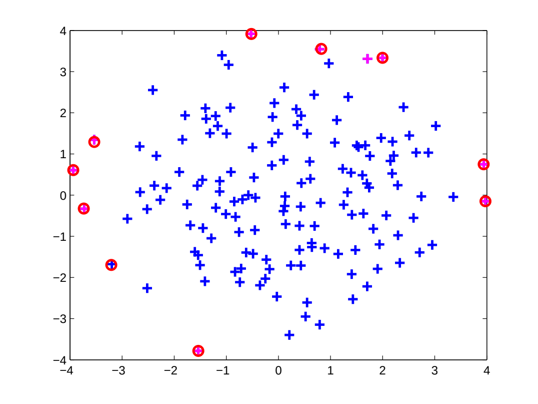


another 10 Arnoldi steps



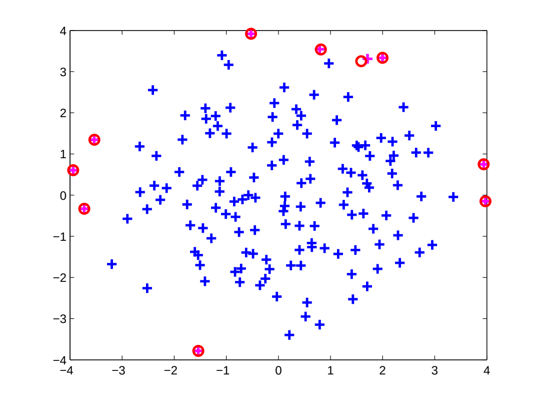
$$h_{11,10} = 1.56$$

first restart



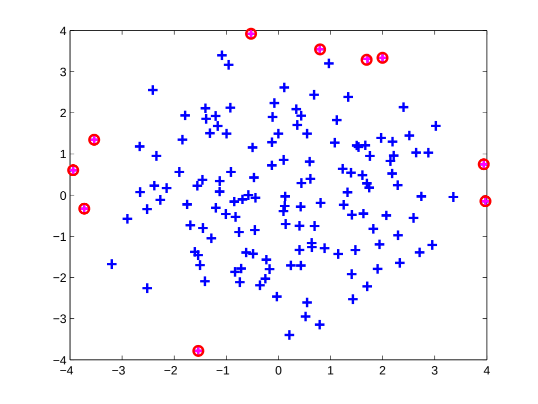
$$h_{11,10} = 0.62$$

second restart



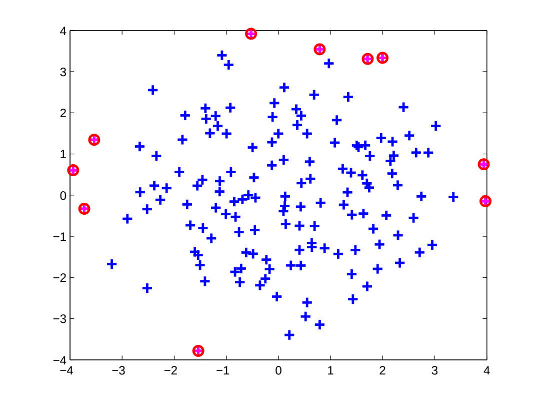
$$h_{11,10} = 0.81$$

third restart



$$h_{11,10} = 0.14$$

forth restart



$$h_{11,10} = 0.04 \leq \text{tol}$$

The Lanczos process

Arnoldi in matrix form

The Lanczos process can be written in the form

$$AQ_m = Q_m T_m + q_{m+1}\beta_m e_m^T$$

where T_m is square tridiagonal matrix:

$$T_m = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{m-1} \\ & & \beta_{m-1} & \alpha_m \end{bmatrix}$$

Thus we obtain only a three-term recurrence and therefore need to store only three vectors at each step.

The Lanczos algorithm

```
$q_1 = \frac{q}{\|q\|_2}$
for $k = 1$ to $m-1$ do
    $q_{k+1} \leftarrow Aq_k$
    $\alpha_k = q_k^* q_{k+1}$
    $q_{k+1} \leftarrow q_{k+1} - \alpha_k q_k$
    if $k > 1$ then
        $q_{k+1} \leftarrow q_{k+1} - q_{k-1}\beta_{k-1}$
    end if
    $\beta_k \leftarrow \|q_{k+1}\|_2$
    if $\beta_k = 0$ then
        span$\{q_1, \ldots, q_k\}$ is invariant under $A$
    end if
    $q_{k+1} = \frac{q_{k+1}}{\beta_k}$
end for
```

Reorthogonalisation

- store only three vectors q_{j-1}, q_j, q_{j+1} at each step
- roundoff errors will lead to

$$\tilde{q}_{k+1} = \frac{q_{k+1} + w_{k+1}}{\beta_k},$$

where $w_{k+1} \approx \mathbf{u}\|A\|_2$ is the roundoff error.

- check orthogonality

$$|\tilde{q}_{k+1}^T q_i| = \frac{q_{k+1}^T q_i + \mathbf{u}\|A\|_2}{|\beta_k|}$$

- orthogonality is lost, especially for converged eigenvectors

Reorthogonalisation

- store only three vectors q_{j-1}, q_j, q_{j+1} at each step
- roundoff errors will lead to

$$\tilde{q}_{k+1} = \frac{q_{k+1} + w_{k+1}}{\beta_k},$$

where $w_{k+1} \approx \mathbf{u}\|A\|_2$ is the roundoff error.

- check orthogonality

$$|\tilde{q}_{k+1}^T q_i| = \frac{q_{k+1}^T q_i + \mathbf{u}\|A\|_2}{|\beta_k|}$$

- orthogonality is lost, especially for converged eigenvectors

Reorthogonalisation

- store only three vectors q_{j-1}, q_j, q_{j+1} at each step
- roundoff errors will lead to

$$\tilde{q}_{k+1} = \frac{q_{k+1} + w_{k+1}}{\beta_k},$$

where $w_{k+1} \approx \mathbf{u}\|A\|_2$ is the roundoff error.

- check orthogonality

$$|\tilde{q}_{k+1}^T q_i| = \frac{q_{k+1}^T q_i + \mathbf{u}\|A\|_2}{|\beta_k|}$$

- orthogonality is lost, especially for converged eigenvectors

Reorthogonalisation

- store only three vectors q_{j-1}, q_j, q_{j+1} at each step
- roundoff errors will lead to

$$\tilde{q}_{k+1} = \frac{q_{k+1} + w_{k+1}}{\beta_k},$$

where $w_{k+1} \approx \mathbf{u}\|A\|_2$ is the roundoff error.

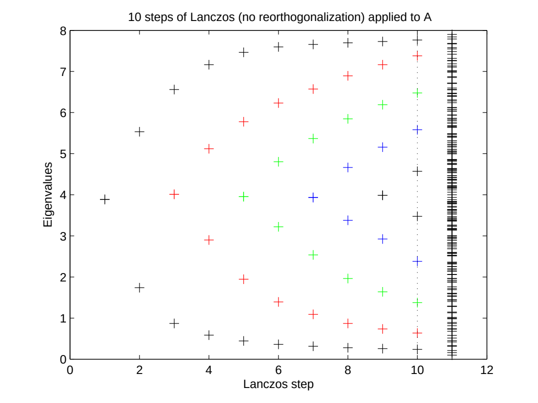
- check orthogonality

$$|\tilde{q}_{k+1}^T q_i| = \frac{q_{k+1}^T q_i + \mathbf{u}\|A\|_2}{|\beta_k|}$$

- orthogonality is lost, especially for converged eigenvectors

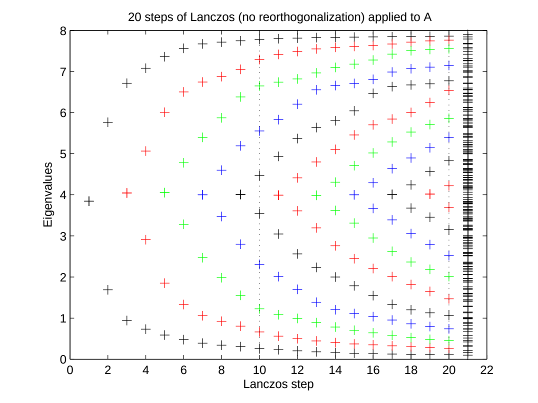
Examples without reorthogonalisation

first 10 Lanczos steps



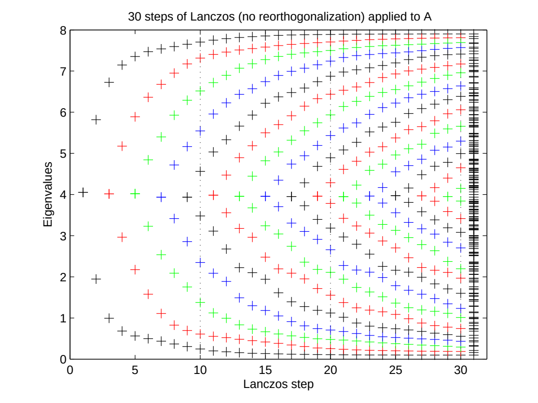
Examples without reorthogonalisation

first 20 Lanczos steps



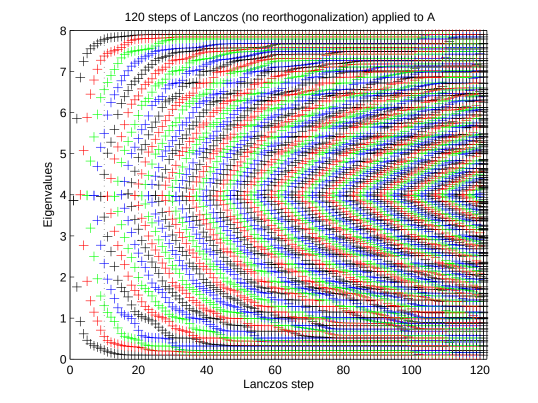
Examples without reorthogonalisation

first 30 Lanczos steps



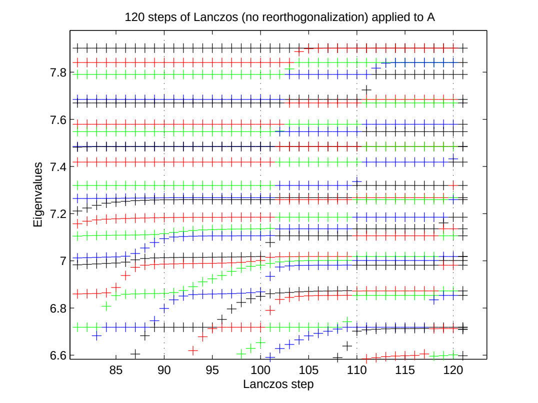
Examples without reorthogonalisation

after 120 Lanczos steps



Examples without reorthogonalisation

after 120 Lanczos steps



Examples without reorthogonalisation

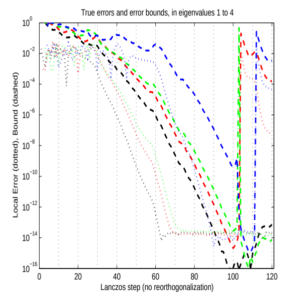


Figure: Error and error bounds

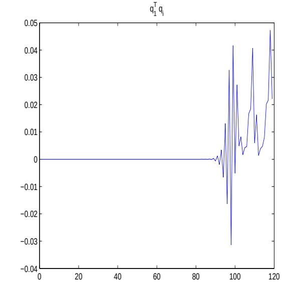
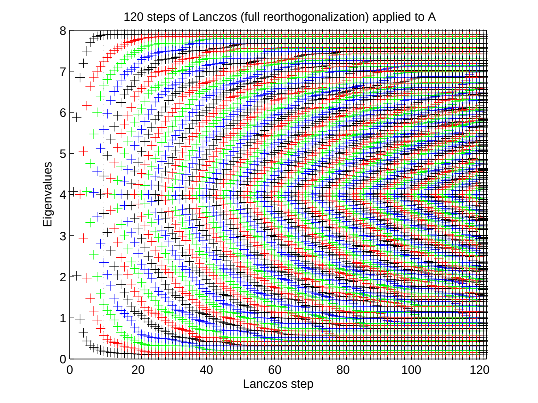


Figure: Loss of orthogonality

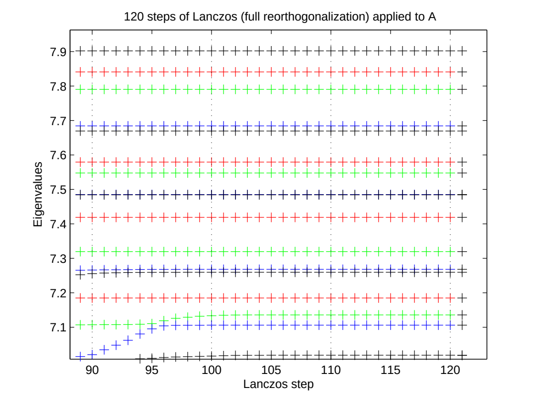
Examples with reorthogonalisation

after 120 Lanczos steps



Examples with reorthogonalisation

after 120 Lanczos steps



Examples with reorthogonalisation

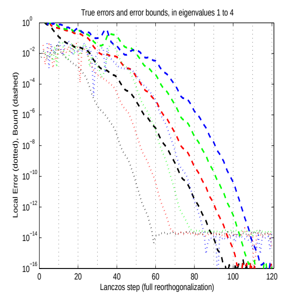


Figure: Error and error bounds

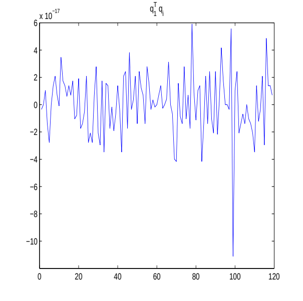


Figure: No loss of orthogonality

- W. E. Arnoldi, *The Principle of minimized iteration in the solution of the matrix eigenvalue problem*, Quarterly of Applied Mathematics, 9 (1951), pp. 17–29.
- J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- G. Golub and C. V. Loan, *Matrix Computations*, John Hopkins University Press, Baltimore, 3nd ed., 1996.
- C. Lanczos, *An iterative method for the solution of the eigenvalue problem of linear differential and integral operators*, Journal of Research of the National Bureau of Standards, 45 (1950), pp. 255–282.
- Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Halsted Press, New York, 1992.

- D. C. Sorensen, *Implicitly restarted Arnoldi/Lanczos methods for large scale eigenvalue calculations*, 1995.
  in Parallel Numerical Algorithms: Proceedings of an ICASE/LaRC Workshop, May 23-25, 1994, Hampton, VA, D. E. Keyes, A. Sameh, and V. Venkatakrishnan, eds., Kluwer.

- N. L. Trefethen and D. I. Bau, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

- D. S. Watkins, *Fundamentals of Matrix Computations*, Pure and Applied Mathematics. Wiley-Interscience [John Wiley & Sons], New York, 2nd ed., 2002.