Iterative Techniques For Solving Eigenvalue Problems

P. H. M. W. in 't panhuis

CASA
Center for Analysis, Scientific Computing and Applications
Department of Mathematics and Computer Science

9-November-2005

Outline

Introduction

Eigenvalue Problem

- For a given matrix $A \in \mathbb{C}^{n\times n}$ find a non-zero vector $x \in \mathbb{C}^n$ and a scalar $\lambda \in \mathbb{C}$ such that $A\boldsymbol{x} = \lambda\boldsymbol{x}$.
- The vector $\boldsymbol{x}$ is the (right) eigenvector of A associated with the eigenvalue λ of A.

Introduction

Eigenvalue Problem

- For a given matrix $A \in \mathbb{C}^{n \times n}$ find a non-zero vector $x \in \mathbb{C}^{n}$ and a scalar $\lambda \in \mathbb{C}$ such that $A\boldsymbol{x} = \lambda \boldsymbol{x}$.
- The vector $\boldsymbol{x}$ is the (right) eigenvector of A associated with the eigenvalue λ of A.

Approximation of Eigenvalues

- There are two classes of numerical methods:
- Partial methods: computation of extremal eigenvalues.
  $\Rightarrow$ The power method.
- Global methods: approximation of whole spectrum.
  $\Rightarrow$ The QR method

Schur Decomposition

Schur Decomposition

- Given $A \in \mathbb{C}^{n\times n}$, there exists U unitary such that

$$U^{-1}AU = U^{H}AU = \begin{bmatrix} \lambda_1 & b_{12} & \cdots & b_{1n} \\ 0 & \lambda_2 & \cdots & b_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} =: T.$$

  where λ_i are the eigenvalues of A.
- Unfortunately U cannot be determined in a direct way for $n \geq 5$ (Abel's theorem).
- Therefore we have to resort to iterative techniques.

The QR Iteration

The QR Iteration (for real matrices)

- Let $A \in \mathbb{R}^{n \times n}$, $Q^{(0)} \in \mathbb{R}^{n \times n}$ orthogonal and $T^{(0)} := {Q^{(0)}}^T A Q^{(0)}$, then for $k = 1, 2, \ldots$:

  determine $Q^{(k)}$, $R^{(k)}$, such that
  $Q^{(k)} R^{(k)} = T^{(k-1)}$ (QR factorization);
  then, let
  $T^{(k)} = R^{(k)} Q^{(k)}$.

- $Q^{(k)}$ is orthogonal and $R^{(k)}$ is upper triangular.

The QR Iteration

The QR Iteration (for real matrices)

- Let $A \in \mathbb{R}^{n\times n}$, $Q^{(0)} \in \mathbb{R}^{n\times n}$ orthogonal and $T^{(0)} := {Q^{(0)}}^T A Q^{(0)}$, then for $k = 1, 2, \ldots$:

  determine $Q^{(k)}$, $R^{(k)}$, such that
  $Q^{(k)} R^{(k)} = T^{(k-1)}$ (QR factorization);
  then, let
  $T^{(k)} = R^{(k)} Q^{(k)}$.
- $Q^{(k)}$ is orthogonal and $R^{(k)}$ is upper triangular.
- Every matrix $T^{(k)}$ is orthogonally similar to A as $T^{(k)} = (Q^{(0)} Q^{(1)} \ldots Q^{(k)})^T A (Q^{(0)} Q^{(1)} \ldots Q^{(k)})$

Convergence

- If A has real eigenvalues, distinct in modulo, then $T^{(k)}$ will converge to the Schur form

  $\Rightarrow$ Shift techniques can be used in remaining cases

Convergence

- If A has real eigenvalues, distinct in modulo, then $T^{(k)}$ will converge to the Schur form
  - $\Rightarrow$ Shift techniques can be used in remaining cases
- If A has complex eigenvalues

$$T^{(k)} \rightarrow \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ 0 & R_{22} & \cdots & R_{2m} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & R_{mm} \end{bmatrix},$$

  where each R_{ii} is either a real number or a matrix of order 2 having complex conjugate eigenvalues.
- This is known as the real Schur decomposition

The Hessenberg-QR iteration

- Naive implementation $\rightarrow$ Start with $Q^{(0)} = I$ and $T^{(0)} = A$ and perform the QR-factorizations using the modified Gram-Schmidt procedure.
  - $\Rightarrow$ $2n^3$ flops per iteration

The Hessenberg-QR iteration

- Naive implementation → Start with $Q^{(0)} = I$ and $T^{(0)} = A$ and perform the QR-factorizations using the modified Gram-Schmidt procedure.
  $\Rightarrow$ $2n^3$ flops per iteration
- Computational costs can be reduced if in stead we start iterating from a matrix $T^{(0)}$ in upper Hessenberg form:

$$t_{ij}^{(0)} = 0, \text{ for } i > j + 1.$$

- Reduction into upper Hessenberg form is realized with Householder matrices (n^3 flops).

The Hessenberg-QR iteration

- Naive implementation → Start with $Q^{(0)} = I$ and $T^{(0)} = A$ and perform the QR-factorizations using the modified Gram-Schmidt procedure.

  ⇒ $2n^3$ flops per iteration
- Computational costs can be reduced if in stead we start iterating from a matrix $T^{(0)}$ in upper Hessenberg form:

$$t_{ij}^{(0)} = 0, \text{ for } i > j + 1.$$

- Reduction into upper Hessenberg form is realized with Householder matrices (n^3 flops).
- QR factorization is carried out using Givens matrices.

  ⇒ n^2 flops per iteration

Householder Reflection Matrix

- For any vector $\boldsymbol{v} \in \mathbb{R}^n$ the orthogonal and symmetric matrix

$$P = I - 2\boldsymbol{v}\boldsymbol{v}^T / \|\boldsymbol{v}\|_2^2$$

is called the Householder reflection matrix.

Householder Reflection Matrix

- For any vector $\boldsymbol{v} \in \mathbb{R}^n$ the orthogonal and symmetric matrix

$$P = I - 2\boldsymbol{v}\boldsymbol{v}^T / \|\boldsymbol{v}\|_2^2$$

is called the Householder reflection matrix.

- The vector $\boldsymbol{y} = P\boldsymbol{x}$ is the reflection of $\boldsymbol{x}$ with respect to the hyperplane spanned by the vectors orthogonal to $\boldsymbol{v}$.
- If $\boldsymbol{v} = \boldsymbol{x} \pm \|\boldsymbol{x}\|_2 \boldsymbol{e}_m$, then $P = P(\boldsymbol{x})$ and

$$P\boldsymbol{x} = [0, \ldots, 0, \underbrace{\pm\|\boldsymbol{x}\|_2}_{m}, 0, \ldots, 0]^T.$$

Reducing a Matrix in Hessenberg Form

- To obtain the upper Hessenberg form we use the Householder matrices $P_{(k)}$, for a given $\boldsymbol{x}$ defined by

$$P_{(k)}(\boldsymbol{x}) = \begin{bmatrix} I_k & 0 \\ 0 & R_{n-k} \end{bmatrix}, \quad R_{n-k} = I_{n-k} - 2\frac{\boldsymbol{v}_{(k)}\boldsymbol{v}_{(k)}{}^T}{\|\boldsymbol{v}_{(k)}\|_2^2},$$

Reducing a Matrix in Hessenberg Form

- To obtain the upper Hessenberg form we use the Householder matrices $P_{(k)}$, for a given $\boldsymbol{x}$ defined by

$$P_{(k)}(\boldsymbol{x}) = \begin{bmatrix} I_k & 0 \\ 0 & R_{n-k} \end{bmatrix}, \quad R_{n-k} = I_{n-k} - 2\frac{\boldsymbol{v}_{(k)}\boldsymbol{v}_{(k)^T}}{\|\boldsymbol{v}_{(k)}\|_2^2},$$

where

$$\boldsymbol{v}_{(k)} = \boldsymbol{x}_{(n-k)} \pm \|\boldsymbol{x}_{(n-k)}\|_2 \boldsymbol{e}_1^{(n-k)}, \qquad \boldsymbol{x} = [\boldsymbol{x}_{(k)}, \boldsymbol{x}_{(n-k)}].$$

Reducing a Matrix in Hessenberg Form

- To obtain the upper Hessenberg form we use the Householder matrices $P_{(k)}$, for a given $\boldsymbol{x}$ defined by

$$P_{(k)}(\boldsymbol{x}) = \begin{bmatrix} I_k & 0 \\ 0 & R_{n-k} \end{bmatrix}, \quad R_{n-k} = I_{n-k} - 2\frac{\boldsymbol{v}_{(k)}\boldsymbol{v}_{(k)}^T}{\|\boldsymbol{v}_{(k)}\|_2^2},$$

where

$$\boldsymbol{v}_{(k)} = \boldsymbol{x}_{(n-k)} \pm \|\boldsymbol{x}_{(n-k)}\|_2 \boldsymbol{e}_1^{(n-k)}, \qquad \boldsymbol{x} = [\boldsymbol{x}_{(k)}, \boldsymbol{x}_{(n-k)}].$$

Then

$$P_{(k)}\boldsymbol{x} = \begin{bmatrix} x_1, \ldots, x_k, \pm\|\boldsymbol{x}_{(n-k)}\|_2, 0, \ldots, 0 \end{bmatrix}^T.$$

Reducing a Matrix in Hessenberg Form (continued)

- Choose $P_{(k)}$ to set to zero the elements $k+2,\dots,n$ in the k-th column of A and define

$$Q_{(k)} = P_{(1)}P_{(2)}\cdots P_{(k)}.$$

Then

$$A^{(n-2)} = Q_{(n-2)}^T A Q_{(n-2)}, \qquad (n \geq 3).$$

is in upper Hessenberg form

Reducing a Matrix in Hessenberg Form (continued)

- Choose $P_{(k)}$ to set to zero the elements $k+2, \ldots, n$ in the k-th column of A and define

$$Q_{(k)} = P_{(1)} P_{(2)} \cdots P_{(k)}.$$

Then

$$A^{(n-2)} = Q^T_{(n-2)} A Q_{(n-2)}, \qquad (n \geq 3).$$

is in upper Hessenberg form

- $$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}$$

Reducing a Matrix in Hessenberg Form (continued)

- Choose $P_{(k)}$ to set to zero the elements $k+2, \ldots, n$ in the k-th column of A and define

$$Q_{(k)} = P_{(1)} P_{(2)} \cdots P_{(k)}.$$

Then

$$A^{(n-2)} = Q^T_{(n-2)} A Q_{(n-2)}, \qquad (n \geq 3).$$

is in upper Hessenberg form

- $$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}$$

Reducing a Matrix in Hessenberg Form (continued)

- Choose $P_{(k)}$ to set to zero the elements $k+2, \ldots, n$ in the k-th column of A and define

$$Q_{(k)} = P_{(1)} P_{(2)} \cdots P_{(k)}.$$

Then

$$A^{(n-2)} = Q_{(n-2)}^T A Q_{(n-2)}, \qquad (n \geq 3).$$

is in upper Hessenberg form

- $$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \xrightarrow[P_{(1)}]{} \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{bmatrix}$$

Reducing a Matrix in Hessenberg Form (continued)

- Choose $P_{(k)}$ to set to zero the elements $k+2,\dots,n$ in the k-th column of A and define

$$Q_{(k)} = P_{(1)}P_{(2)}\cdots P_{(k)}.$$

Then

$$A^{(n-2)} = Q_{(n-2)}^T A Q_{(n-2)}, \qquad (n \geq 3).$$

is in upper Hessenberg form

- $$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \xrightarrow[P_{(2)},\,P_{(1)}]{} \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \end{bmatrix}$$

Reducing a Matrix in Hessenberg Form (continued)

- Choose $P_{(k)}$ to set to zero the elements $k+2,\dots,n$ in the k-th column of A and define

$$Q_{(k)} = P_{(1)}P_{(2)}\cdots P_{(k)}.$$

Then

$$A^{(n-2)} = Q_{(n-2)}^T A Q_{(n-2)}, \qquad (n \geq 3).$$

is in upper Hessenberg form

- $$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \xrightarrow{\overline{P_{(3)}, P_{(2)}, P_{(1)}}} \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$$

Reducing a Matrix in Hessenberg Form (continued)

- Choose $P_{(k)}$ to set to zero the elements $k+2, \ldots, n$ in the k-th column of A and define

$$Q_{(k)} = P_{(1)} P_{(2)} \cdots P_{(k)}.$$

Then

$$A^{(n-2)} = Q_{(n-2)}^T A Q_{(n-2)}, \qquad (n \geq 3).$$

is in upper Hessenberg form

- $$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \xrightarrow{\hspace{2cm}} \overline{P_{(3)}, P_{(2)}, P_{(1)}} \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$$
- If A is symmetric, then $A^{(n-2)}$ is even tridiagonal.

Givens Elementary Matrices

- The Givens matrices are orthogonal rotation matrices defined as

$$G(i,k,\theta) = \begin{bmatrix} I_{i-1} & & & & \\ & \cos(\theta) & & \sin(\theta) & \\ & & I_{k-i-1} & & \\ & -\sin(\theta) & & \cos(\theta) & \\ & & & & I_{n-k} \end{bmatrix}$$

- Given $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y} = (G(i,k,\theta))^T \boldsymbol{x}$ is equivalent to rotating $\boldsymbol{x}$ counterclockwise by an angle θ in the coordinate plane (x_i, x_k) and

$$y_j = \begin{cases} x_j & j \neq i,k, \\ cx_i - sx_k & j = i, \\ sx_i + cx_k & j = k, \end{cases} \qquad \text{where} \quad \begin{array}{l} c = \cos(\theta), \\ s = \sin(\theta). \end{array}$$

QR-iteration using Givens rotations

- Choosing $\theta = \arctan(-x_k/x_i)$ yields

$$\boldsymbol{y} = \left[\; x_1, \ldots, \underbrace{\sqrt{x_i^2 + x_k^2}}_{i}, x_{i+1}, \ldots, \underbrace{0}_{k}, \ldots, x_n \; \right]^T.$$

- Start with $T^{(0)}$ in upper Hessenberg form and for each $k \geq 1$ compute $R^{(k)}$ as

$$R^{(k)} = Q^{(k)^T} T^{(k-1)} = G_{n-1}^{(k)^T} \cdots G_1^{(k)^T} T^{(k-1)},$$

where $G_j^{(k)} = G(j, j+1, \theta_j)^{(k)}$ and θ_j is chosen such that $R_{j+1,j}^{(k)} = 0$.

QR-iteration using Givens rotations (continued)

- Next complete the orthogonal similarity transformation by

$$T^{(k)} = R^{(k)} Q^{(k)}.$$

- $R^{(k)}$ is upper triangular and $Q^{(k)}$ is orthogonal and upper Hessenberg .
- The iterations do not always converge to the real Schur decomposition
  - Resort to shift techniques.

Stability

- The algorithm is well-conditioned with respect to rounding errors.
  - $\Rightarrow$ Reduction into Hessenberg form:

$$\hat{H} = Q^T(A+E)Q, \qquad \|E\|_F \leq cn^2\mathrm{u}\|A\|_F.$$

  - $\Rightarrow$ QR iteration into real Schur form:

$$\hat{T} = Q^T(A+E)Q, \qquad \|E\|_2 \sim \mathrm{u}\|A\|_2.$$

- u is the machine roundoff unit.

The QR Iteration with Shifting Techniques

Shifting Techniques

- The QR iteration does not always converge to the real Schur form.
- The single shift technique accelerates the convergence, when there are eigenvalues with moduli close to each other.
- The double shift technique guarantees convergence to the (approximate) Schur form, even for complex eigenvalues.

The QR-Method with Single Shift

- Given $\mu \in \mathbb{R}$, $T^{(0)} = Q^{(0)^T} A Q^{(0)}$ in upper Hessenberg form,

  determine $Q^{(k)}$, $R^{(k)}$, such that
  $Q^{(k)} R^{(k)} = T^{(k-1)} - \mu I$ (QR factorization);
  then, let
  $T^{(k)} = R^{(k)} Q^{(k)} + \mu I$.

The QR-Method with Single Shift

- Given $\mu \in \mathbb{R}$, $T^{(0)} = Q^{(0)^T} A Q^{(0)}$ in upper Hessenberg form,
  determine $Q^{(k)}$, $R^{(k)}$, such that
  $Q^{(k)} R^{(k)} = T^{(k-1)} - \mu I$ (QR factorization);
  then, let
  $T^{(k)} = R^{(k)} Q^{(k)} + \mu I$.
- Assume μ is fixed and the eigenvalues are order by

$$|\lambda_1 - \mu| \geq |\lambda_2 - \mu| \geq \cdots \geq |\lambda_n - \mu|,$$

then for $k \to \infty$

$$t_{i,i-1}^{(k)} = \mathcal{O}\left(\frac{|\lambda_j - \mu|}{|\lambda_{j-1} - \mu|}\right), \qquad i = 2, 3 \ldots .$$

The QR-Method with Single Shift(continued)

- If μ is chosen such such that

$$|\lambda_n - \mu| < |\lambda_i - \mu|, \qquad i = 1, \ldots, n-1,$$

then $t^{(k)}_{n,n-1}$ rapidly tends to zero as $k \to \infty$.

The QR-Method with Single Shift(continued)

- If μ is chosen such such that

$$|\lambda_n - \mu| < |\lambda_i - \mu|, \qquad i = 1, \ldots, n-1,$$

  then $t_{n,n-1}^{(k)}$ rapidly tends to zero as $k \to \infty$.
- The QR method with single shift takes $\mu = t_{nn}^{(k)}$.
- The sequence $\{t_{n,n-1}^{(k)}\}$ converges quadratic

$$\frac{t_{n,n-1}^{(k)}}{\| T^{(0)} \|_2} = \eta_k < 1 \quad \Rightarrow \quad \frac{t_{n,n-1}^{(k+1)}}{\| T^{(0)} \|_2} = \mathcal{O}\left(\eta_k^2\right).$$

The QR-Method with Double Shift

- In the case of complex eigenvalues a complex shift is needed.
- The double shift technique detects complex eigenvalues and removes the 2×2 diagonal blocks of the real Schur form.
- The convergence is again quadratic.

The QR-Method with Double Shift

- In the case of complex eigenvalues a complex shift is needed.
- The double shift technique detects complex eigenvalues and removes the 2×2 diagonal blocks of the real Schur form.
- The convergence is again quadratic.
- Procedure:
  - $\Rightarrow$ Perform single-shift QR iterations until a 2×2 diagonal block $R_{kk}^{(k)}$ is detected, with complex conjugate eigenvalues $\lambda^{(k)}$ and $\bar{\lambda}^{(k)}$.
  - $\Rightarrow$ Perform double shift strategy.
  - $\Rightarrow$ Continue single-shift QR iterations until the next diagonal block is detected.

The QR-Method with Double Shift (continued)

- Double shift strategy:
  - determine $Q^{(k)}$, $R^{(k)}$, such that
    $Q^{(k)}R^{(k)} = T^{(k-1)} - \lambda^{(k)} I$ (first QR factorization);
    then, let
    $T^{(k)} = R^{(k)}Q^{(k)} + \lambda^{(k)} I$.

The QR-Method with Double Shift (continued)

- Double shift strategy:
  - determine $Q^{(k)}$, $R^{(k)}$, such that
    $Q^{(k)}R^{(k)} = T^{(k-1)} - \lambda^{(k)} I$ (first QR factorization);
    then, let
    $T^{(k)} = R^{(k)}Q^{(k)} + \lambda^{(k)} I$.
  - determine $Q^{(k+1)}$, $R^{(k+1)}$, such that
    $Q^{(k+1)}R^{(k+1)} = T^{(k)} - \bar{\lambda}^{(k)} I$ (second QR factorization);
    then, let
    $T^{(k+1)} = R^{(k+1)}Q^{(k+1)} + \bar{\lambda}^{(k)} I$.
- There exist special matrices for which even this method fails to converge.

Methods for Eigenvalues of Symmetric Matrices

The Jacobi method

- The Jacobi method generates a sequence of matrices converging to the diagonal Schur form of A.

Methods for Eigenvalues of Symmetric Matrices

The Jacobi method

- The Jacobi method generates a sequence of matrices converging to the diagonal Schur form of A.
- Given $A^{(0)} = A$, for any k, a pair of indices p, q is fixed and G_{pq} is the Givens matrix $G(p, q, \theta)$. θ is chosen such, that the matrix

$$A^{(k)} = G_{pq}^T A^{(k-1)} G_{pq}$$

  has the property $a_{pq}^{(k)} = 0$.
- If $a_{pq}^{(k-1)} = 0$, the choice $\theta = 0$ satisfies. Otherwise a complicated algebraic expression has to be solved.

The Row-Cyclic Jacobi method

- The optimal choice for p and q seems to be

$$|a_{pq}^{(k-1)}| = \max_{i \neq j} |a_{ij}^{(k-1)}|.$$

$\Rightarrow$ n^2 flops for choosing p, q and n flops for updating $A^{(k)}$.

The Row-Cyclic Jacobi method

- The optimal choice for p and q seems to be

$$|a_{pq}^{(k-1)}| = \max_{i \neq j} |a_{ij}^{(k-1)}|.$$

$\Rightarrow$ n^2 flops for choosing p, q and n flops for updating $A^{(k)}$.

- In the row cyclic Jacobi method p and q are chosen by a row-sweeping of $A^{(k)}$:
  - $\Rightarrow$ For any i-th row, $p := i$ and $q := i+1, \ldots n$.
  - $\Rightarrow$ Each complete sweep requires $N := n(n-1)/2$ Jacobi transformations.
- If $|\lambda_i - \lambda_j| > \delta$, for $i \neq j$, then the convergence is quadratic:

The Method of Sturm Sequences

- Finds the eigenvalues of $T = \begin{bmatrix} d_1 & b_1 & & & \\ b_1 & d_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-2} & d_{n-1} & b_{n-1} \\ & & & b_{n-1} & d_n \end{bmatrix}$.

The Method of Sturm Sequences

- Finds the eigenvalues of $T = \begin{bmatrix} d_1 & b_1 & & & \\ b_1 & d_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-2} & d_{n-1} & b_{n-1} \\ & & & b_{n-1} & d_n \end{bmatrix}$.
- Let T_i be the principal minor of order i and define $p_0(x) = 1$ and $p_i(x) = \det(T_i - xI)$.
- The Sturm sequence is given by

$$p_1(x) = d_1 - x$$
$$p_i(x) = (d_i - x)p_{i-1}(x) - b_{i-1}^2 p_{i-2}(x), \quad i = 2, \ldots, n.$$

- p_n is the characteristic polynomial of T.

Property of Sturm Sequences

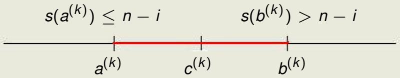
- For $\mu \in \mathbb{R}$ define $S_\mu = \{p_0(\mu), p_1(\mu), \ldots, p_n(\mu)\}$. Then the number of sign changes $s(\mu)$ in S_μ equals the number of eigenvalues less than μ.

Property of Sturm Sequences

- For $\mu \in \mathbb{R}$ define $S_\mu = \{p_0(\mu), p_1(\mu), \ldots, p_n(\mu)\}$. Then the number of sign changes $s(\mu)$ in S_μ equals the number of eigenvalues less than μ.

Givens method for the calculation of eigenvalues

- The eigenvalue λ_i can be found using a bisection method:

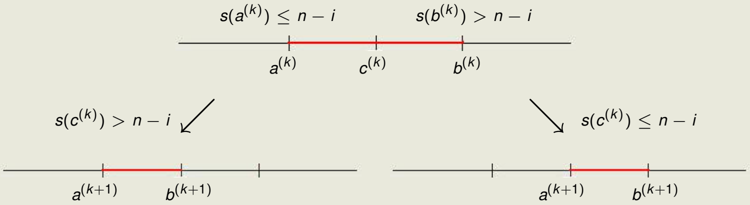


Property of Sturm Sequences

- For $\mu \in \mathbb{R}$ define $S_\mu = \{p_0(\mu), p_1(\mu), \ldots, p_n(\mu)\}$. Then the number of sign changes $s(\mu)$ in S_μ equals the number of eigenvalues less than μ.

Givens method for the calculation of eigenvalues

- The eigenvalue λ_i can be found using a bisection method:

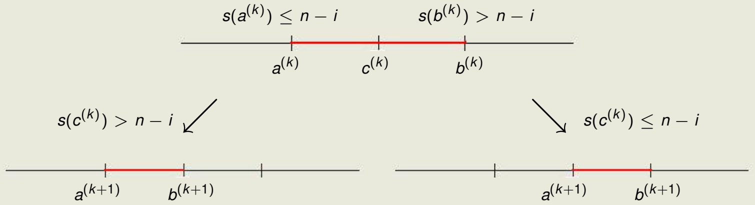


Property of Sturm Sequences

- For $\mu \in \mathbb{R}$ define $S_\mu = \{p_0(\mu), p_1(\mu), \ldots, p_n(\mu)\}$. Then the number of sign changes $s(\mu)$ in S_μ equals the number of eigenvalues less than μ.

Givens method for the calculation of eigenvalues

- The eigenvalue λ_i can be found using a bisection method:



- $c^{(k)}$ approximates λ_i within $(|a^{(0)}| + |b^{(0)}|) \cdot 2^{-(k+1)}$.

Summarizing

- The QR-method for matrices in upper Hessenberg form.
  - Converges to real Schur form.
- The QR-method with single shift.
  - Useful if there are eigenvalues with moduli close to each other.
- The QR-method with double shift.
  - Useful if there are complex eigenvalues.
  - Converges to approximate Schur form.
- The Jacobi method
  - Calculates diagonal Schur form of a symmetric matrix
- The method of Sturm sequences.
  - Calculates eigenvalues of a real, tridiagonal and symmetric matrix.