VII. Matrix Eigenvalue Problems

2008. 10

담당교수**:** 주 한 규
joohan@snu.ac.kr, x9241, Rm 32-205
원자핵공학과

VII. Matrix Eigenvalue Problems

1. Power Method

7.1.1 Basic Formulation

7.1.2 Inverse Power Method

7.1.3 Method of Deflation and Decontamination

2. Acceleration Methods

7.2.1 Chebyshev Acceleration Method

7.2.2 Wielandt Shift Method

3. QR Method

7.3.1 QR Factorization

7.3.2 Householder Transformation

7.3.3 QR Method

1.1 Basic Formulation (1/4)

□ Basics of Matrix Eigenvalue Problems

- **Form:** $Ax = \lambda x$ or $Mx = \lambda Fx \longrightarrow F^{-1}Mx = \lambda x$ or $\frac{1}{\lambda}x = M^{-1}Fx$
- **Eigenvalue**

$$(A - \lambda I)x = 0\,;\ \text{Non trivial solution} \rightarrow Det(A - \lambda I) = 0$$

  - Characteristic polynomial (n-th order) having n roots (can be repeated)
- **Eigenvector**

$$\text{Solve } (A - \lambda_i I)x = 0 \text{ for nontririvial solution} \qquad x = (\chi_1, \chi_2, \cdots \chi_n)^T$$

  - Gauss elimination would lead the last to be a null equation $0 \cdot \chi_n = 0$
  - The last entry is free to choose → magnitude of eigenvector not unique
  - Eigenvectors are linearly independent each other

□ Power Method

- **In most cases, finding only the largest eigenvalue and corresponding eigenvector is sufficient. E.g. Spectral radius**
- **Find those by repeating only matrix vector multiplication**

$$x^{(k)} = Ax^{(k-1)} \text{ and } \qquad \lambda^{(k)} = \frac{< x^{(k)}, x^{(k)} >}{< x^{(k)}, x^{(k-1)} >}$$

1.1 Basic Formulation (2/4)

□ Convergence

$$x^{(k)} = Ax^{(k-1)} \rightarrow \quad x^{(k)} = AAx^{(k-2)} = A^k x^{(0)}$$

- **Expansion of the initial (guess) vector in terms of eigenvectors which are mutually linearly independent**

$$x^{(0)} = c_1 u_1 + c_2 u_2 + \cdots + c_n u_n \quad ; \text{ eigenvectors numbered such that } |\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$$

$$x^{(k)} = A^k (c_1 u_1 + c_2 u_2 + \cdots + c_n u_n) = c_1 A^k u_1 + c_2 A^k u_2 + \cdots + c_n A^k u_n$$

$$= c_1 \lambda_1^k u_1 + c_2 \lambda_2^k u_2 + \cdots + c_n \lambda_n^k u_n = \lambda_1^k \left(c_1 u_1 + c_2 \left[\frac{\lambda_2}{\lambda_1} \right]^k u_2 + \cdots + c_n \left[\frac{\lambda_n}{\lambda_1} \right]^k u_n \right)$$

$$\approx c_1 \lambda_1^k u_1 \text{ as } k \rightarrow \infty \quad \because \quad \left| \frac{\lambda_i}{\lambda_1} \right|^k < 1 \quad \forall i$$

- **Convergence Rate**

$$\text{determined by dominance ratio } \sigma = \left| \frac{\lambda_2}{\lambda_1} \right|$$

How much the fundamental (largest) eigenvalue dominates over other eigenvalues

1.1 Basic Formulation (3/4)

□Determination of Eigenvalue

For a large k, $x^{(k)} = Ax^{(k-1)} \cong \lambda_1 x^{(k-1)} \quad \because x^{(k)} \to u_1$

Inner Product with $x^{(k)} \to < x^{(k)}, x^{(k)} > \cong \lambda_1 < x^{(k)}, x^{(k-1)} >$

$$\lambda_1 \cong \frac{< x^{(k)}, x^{(k)} >}{< x^{(k)}, x^{(k-1)} >} = \lambda^{(k)}$$

□Eigenvector Scaling

- **The iteration scheme above will result in continuously increasing (eig. val. > 1.0) infinitely or decreasing (eig. val <1.0)**
  - Truncation error would ruin the iteration scheme
- **Scale the eigenvector at each step by dividing the current estimate of the eigenvalue**

$$\hat{x}^{(k-1)} = \frac{x^{(k-1)}}{\lambda^{(k-1)}} \rightarrow x^{(k)} = A\hat{x}^{(k-1)} = \frac{1}{\lambda^{(k-1)}} Ax^{(k-1)}; \ \lambda^{(k)} = \frac{< x^{(k)}, x^{(k)} >}{< x^{(k)}, \hat{x}^{(k-1)} >} = \lambda^{(k-1)} \frac{< x^{(k)}, x^{(k)} >}{< x^{(k)}, x^{(k-1)} >}$$

$$\cong \lambda_1 \hat{x}^{(k-1)}$$

1.1 Basic Formulation (4/4)

Convergence After Scaling

$$x^{(k)} = \frac{1}{\lambda^{(k-1)}} A x^{(k-1)} = \frac{1}{\lambda^{(k-1)}\lambda^{(k-2)}} A^2 x^{(k-2)} = \frac{1}{\prod_{i=0}^{k-1} \lambda^{(i)}} A^k x^{(0)} \cong \frac{\lambda_1^k}{\prod_{i=0}^{k-1} \lambda^{(i)}} c_1 u_1$$

Scaling by Normalization

- **Divide the eigenvector by the maximum entry (*l*$_{\circledcirc}$ *norm*)**
- **The resulting eigenvector will always have 1.0 as the max. val**
- **The max. entry after the matrix-vector multiplication is the new eigenvalue** $\because \left(x^{(k)} = A[\cdots 1 \cdots]^T = \lambda [\cdots 1 \cdots]^T = [\cdots \lambda \cdots]^T \right)$

Power Iteration Sequence

0) **Make an initial guess of eigenvector, e.g. all entries of 1.0, and eigenvalue, e.g. 1.0.**
1) **Scale the eigenvector with the eigenvalue**
2) **Perform matrix-vector multiplication to determine new vector**
3) **Obtain the inner products and take the ratio as eigenvalue**
4) **Repeat Steps 1-3 until the change in eigenvalue is less than ♏**

1.2 Inverse Power Method

- **Objective: Find the minimum eigenvalue**
- **Method: Apply the power method to the inverse**
  - **Eigenvalue of the Inverse**

$$\det(A^{-1}-\tilde{\lambda}I)=0 \rightarrow \det\left(A^{-1}(I-\tilde{\lambda}A)\right)=0 \rightarrow \det(A^{-1})\det\left((I-\tilde{\lambda}A)\right)=0 \rightarrow \det\left(\tilde{\lambda}(\frac{I}{\tilde{\lambda}}-A)\right)=0$$

$$\rightarrow \tilde{\lambda}^n \det\left((\frac{1}{\tilde{\lambda}}I-A)\right)=0 \rightarrow \det(A-\lambda I)=0 \quad \leftarrow \frac{1}{\tilde{\lambda}}=\lambda;\text{ inverse of original eigevalue}$$

    - Minimum eigenvalue of original matrix = maximum eigenvalue of inverse
  - **Application**

$$x^{(k)}=A^{-1}\hat{x}^{(k-1)};\ \lambda^{(k)}=\frac{<x^{(k)},x^{(k)}>}{<x^{(k)},\hat{x}^{(k-1)}>}$$

    - Since finding the inverse is difficult, solve the following instead

$$Ax^{(k)}=\hat{x}^{(k-1)}$$

    - If A is LU factored, the repeated solution of the linear system with changed RHS will be easy.

1.3 Method of Deflation and Decontamination

- **Objective: Find the second eigenvalue**
- **Method: Apply the power method to a deflated matrix**
  - **Deflation : Remove the components contributing to the first eigenvector**
  - **After finding the first eigenvector,**

$$\text{Let } \hat{u}_1 = \frac{u_1}{\|u_1\|} \text{ be normalized eigenvector} \longrightarrow \;<\hat{u}_1, \hat{u}_1> = \hat{u}_1^T \hat{u}_1 = 1$$

  - **Construct a new matrix**

$$B = A - \lambda_1 \hat{u}_1 \hat{u}_1^T \longrightarrow B\hat{u}_1 = A\hat{u}_1 - \lambda_1 \hat{u}_1 \hat{u}_1^T \hat{u}_1 = \lambda_1 \hat{u}_1 - \lambda_1 \hat{u}_1 \left(\hat{u}_1^T \hat{u}_1\right) = o$$

  - **Power method applied to B**

$$x^{(k)} = B^k x^{(0)} = \sum_{i=1}^{n} c_i B^k u_i = o + \sum_{i=2}^{n} c_i \lambda_i^k u_i = \lambda_2^k \sum_{i=2}^{n} c_i \left(\frac{\lambda_i}{\lambda_2}\right)^k u_i \cong c_2 \lambda_2^k u_2$$

- **Successive Deflation→ Lower eigenvalues**
- **Drawback: B becomes full → Flops increases for Bx even though A is sparse**

1.3 Method of Deflation and Decontamination

- **Decontamination: Remove the first eigenvector component from the k-th vector**
- **For Symmetric matrices**
  - **Eigenvectors are orthogonal**

$$x^{(k-1)} = c_1^{(k-1)}u_1 + c_2^{(k-1)}u_2 + \cdots c_n^{(k-1)}u_n \longrightarrow c_1^{(k-1)} = \frac{< x^{(k-1)}, u_1 >}{< u_1, u_1 >}$$

  - **Decontaminated vector**

$$\tilde{x}^{(k-1)} = c_2^{(k-1)}u_2 + \cdots c_n^{(k-1)}u_n = x^{(k-1)} - c_1^{(k-1)}u_1 = x^{(k-1)} - \frac{u_1^T x^{(k-1)}}{u_1^T u_1}u_1$$

  - **Power method with decontaminated vector**

scaling

$$\hat{x}^{(k-1)} = \frac{1}{\lambda_2^{(k-1)}}\tilde{x}^{(k-1)}; \longrightarrow x^{(k)} = A\hat{x}^{(k-1)} \cong c_2\lambda_2^k u_2$$

- **Successive Decontamination→ Lower eigenvalues**
- **Can work for non-symmetric matrices**

2.1 Chebyshev Acceleration Method (1/2)

☐ Example of Large Dominance Ratio Cases (Causing Slow Convergence of Power Method)

- **Consider an eigenvalue problem in one-d particle diffusion**

$$-D\frac{d^2\phi}{dx^2}+\sigma_A\phi=\lambda\sigma_S\phi\,,\quad x\in[0,a],\quad \phi(0)=0,\quad \phi(a)=0$$

$$A\phi^{(n)}=\lambda^{(n-1)}S\phi^{(n-1)}$$

- Discretization would lead to $A\phi=\lambda S\phi$ $\quad min\ Eig.\frac{1}{\lambda}\phi=A^{-1}S\phi \qquad \frac{1}{\lambda}\phi^{(n)}=A^{-1}S\phi^{(n-1)}$
- Rearrange after dividing by D, $\frac{d^2\phi}{dx^2}+\frac{\lambda\sigma_S-\sigma}{D}\phi=0$
- Let $B^2=\frac{\lambda\sigma_S-\sigma_A}{D}\rightarrow\frac{d^2\phi}{dx^2}+B^2\phi=0$

0 Flux Boundary Condition $\rightarrow\ B_n=\frac{n\pi}{a}$

- Eigenvalue $\lambda_n=\frac{\sigma_A+DB_n^2}{\sigma_S}=\frac{\sigma_A+\frac{n^2\pi^2}{a^2}}{\sigma_S}$
- Dominance Ratio $\sigma=\frac{\frac{1}{\lambda_2}}{\frac{1}{\lambda_1}}=\frac{\sigma_A+D\frac{\pi^2}{a^2}}{\sigma_A+D\frac{4\pi^2}{a^2}}$

Need minimum Eigenvalue for least adjusment

- $\sigma_A\rightarrow 0,\ \sigma\rightarrow 0.25.$

Conversely, as $a\rightarrow\infty$ or $D\downarrow$ or $\sigma_A\uparrow,\ \sigma\rightarrow 1.0$

- **Diffusion problems for large domain or weak diffusivity have large dominance ratio → Slow convergence of power method**

Chebyshev Acceleration Method

- Single Parameter Method
  - Extrapolation of eigenvector using the current estimate by power method and the previous iterate

$$x^{(k)} = \omega^{(k)} x_{Pow}^{(k)} + \left(1-\omega^{(k)}\right) x^{(k-1)} = \omega^{(k)} \frac{1}{\lambda^{(k-1)}} A x^{(k-1)} + \left(1-\omega^{(k)}\right) x^{(k-1)}$$

  - The extrapolation parameter is the single parameter and it is iteration dependent

- Two Parameter Method
  - Extrapolation using two previous iterates

$$x^{(k)} = \alpha^{(k)} x_{Pow}^{(k)} + \left(1-\alpha^{(k)} + \beta^{(k)}\right) x^{(k-1)} - \beta^{(k)} x^{(k-2)}$$

Single Parameter Chebyshev Acceleration (1/6)

- Eigenvector Extrapolation

$$x^{(k)} = \omega^{(k)} x_{Pow}^{(k)} + (1-\omega^{(k)}) x^{(k-1)} = \omega^{(k)} \frac{1}{\lambda^{(k-1)}} A x^{(k-1)} + (1-\omega^{(k)}) x^{(k-1)}$$

$$= \left(\omega^{(k)} \frac{1}{\lambda^{(k-1)}} A + (1-\omega^{(k)}) I \right) x^{(k-1)} \quad \rightarrow x^{(k)} = \prod_{p=1}^{k} \left(\omega^{(p)} \frac{1}{\lambda^{(p-1)}} A + (1-\omega^{(p)}) I \right) x^{(0)}$$

– For $x^{(0)} = c_1 u_1 + c_2 u_2 + \cdots + c_n u_n$

$$x^{(k)} = \sum_{i=1}^{n} \left[c_i \left\{ \prod_{p=1}^{k} \left(\omega^{(p)} \frac{1}{\lambda^{(p-1)}} A + \left(1-\omega^{(p)}\right) I \right) \right\} u_i \right] = \sum_{i=1}^{n} \left[c_i \left\{ \prod_{p=1}^{k} \left(\omega^{(p)} \frac{\lambda_i}{\lambda^{(p-1)}} + \left(1-\omega^{(p)}\right) \right) \right\} u_i \right]$$

– Note that $\prod_{p=1}^{k} \left(\omega^{(p)} \frac{\lambda_i}{\lambda^{(p-1)}} + \left(1-\omega^{(p)}\right) \right)$ is a k-th order polynomial of λ_i

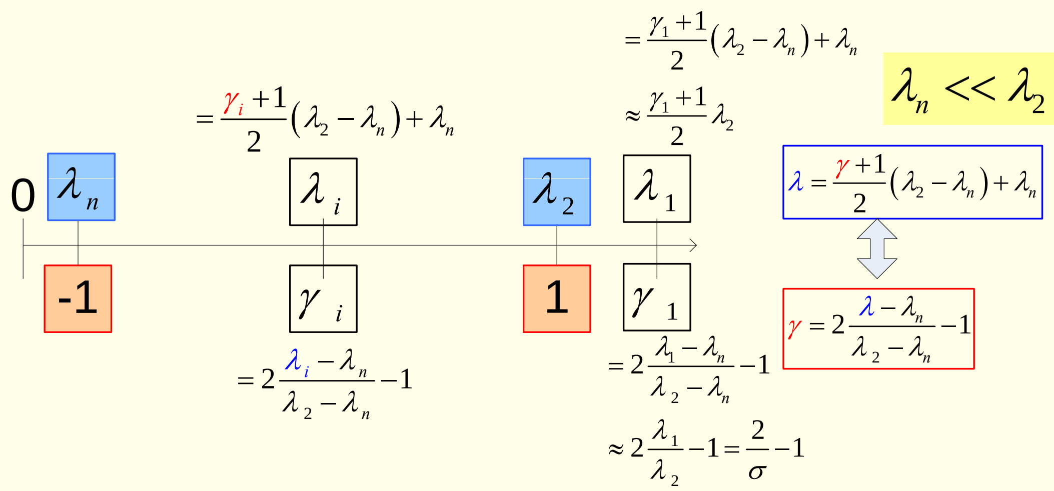
and $\lambda^{(p-1)} \cong \lambda^*$, the largest eigenvalue we seek to find so that $\frac{\lambda_i}{\lambda^{(p-1)}} \leq 1.0$

Single Parameter Chebyshev Acceleration (2/6)

- Change of Variable

$$\gamma = 2\frac{\lambda-\lambda_n}{\lambda_2-\lambda_n}-1 \;\leftarrow\; \begin{pmatrix} \lambda_i : \lambda_n \to \lambda_2 \\ \gamma_i : -1 \to 1 \end{pmatrix}$$

$$\lambda_1 > \lambda_2 \to \gamma_1 = 2\frac{\lambda_1}{\lambda_2} - 1 = \frac{2}{\sigma} - 1 > 1$$



- Passage to Chebyshev Polynomial

Assume the minimum eigenvalue $\lambda_n << 1$, $\gamma_i \cong 2\frac{\lambda_i}{\lambda_2} - 1 \;\to\; \lambda_i = \frac{\gamma_i + 1}{2}\lambda_2$

Then, $\prod_{p=1}^{k}\left(\omega^{(p)}\frac{\lambda_i}{\lambda^{(p-1)}} + \left(1-\omega^{(p)}\right)\right) = \prod_{p=1}^{k}\left(\omega^{(p)}\frac{\lambda_2}{\lambda^{(p-1)}}\frac{\gamma_i+1}{2} + \left(1-\omega^{(p)}\right)\right) \equiv \eta(\gamma_i)$

$\to$ a k-th order polynomial of $\gamma_i, -1 < \gamma_i \le 1, \;\; \forall i > 1$

Since $\lambda^{(p-1)} \cong \lambda_1$, $\eta(\gamma_i) \cong \prod_{p=1}^{k}\left(\omega^{(p)}\frac{\lambda_2}{\lambda_1}\frac{\gamma_i+1}{2} + \left(1-\omega^{(p)}\right)\right) = \prod_{p=1}^{k}\left(\omega^{(p)}\sigma\frac{\gamma_i+1}{2} + \left(1-\omega^{(p)}\right)\right)$

Single Parameter Chebyshev Acceleration (3/6)

- New Form Extrapolation Expression

Error Term

$$x^{(k)} = \sum_{i=1}^{n}\left[c_i\left\{\prod_{p=1}^{k}\left(\omega^{(p)}\frac{1}{\lambda^{(p-1)}}A + \left(1-\omega^{(p)}\right)I\right)\right\}u_i\right] = \sum_{i=1}^{n}[c_i\eta(\gamma_i)u_i] = c_1\eta(\gamma_1)u_1 + \sum_{i=2}^{n}c_i\eta(\gamma_i)u_i$$

$$= \eta(\gamma_1)\left(c_1u_1 + \sum_{i=2}^{n}c_i\frac{\eta(\gamma_i)}{\eta(\gamma_1)}u_i\right) \qquad \eta(\gamma_i) = \prod_{p=1}^{k}\left(\omega^{(p)}\sigma\frac{\gamma_i+1}{2} + \left(1-\omega^{(p)}\right)\right)$$

- Minimax Problem
  - For the maximum convergence in K iterations, try to minimize the error

    $\rightarrow$ Minimize the maximum value of $\frac{\eta(\gamma_i)}{\eta(\gamma_1)}$ for all possible values of $|\gamma_i| \le 1$

  - Make $\eta(\gamma)$ be the Chebyshev polynomial of Order K by properly choosing $\omega^{(p)}$

- How to Choose the extrapolation parameter?

At each step p, make $\omega^{(p)}\sigma\frac{\xi_p+1}{2} + (1-\omega^{(p)}) = 0$ with ξ_p be the $p-$th root of the Chebyshev Polynomial!

Single Parameter Chebyshev Acceleration (4/6)

- Roots of Chebyshev Polynomial of Order K

$$T_K(x) = \cos(K\cos^{-1}x) = \cos(K\theta); x = \cos\theta$$

$$K\theta = p\pi - \frac{1}{2}\pi \rightarrow \xi_p = \cos\theta = \cos^{-1}\frac{2p-1}{2K}\pi$$

- Optimum Extrapolation Parameter

$$\omega^{(p)}\sigma\frac{\xi_p+1}{2} + (1-\omega^{(p)}) = 0$$

$$\omega^{(p)} = \frac{1}{1-\sigma\frac{\xi_p+1}{2}} = \frac{1}{1-\frac{\sigma}{2}\left(\cos(\frac{2p-1}{2K}\pi)\right)}$$

– By choosing the extrapolation parameter this way,
  the K-th order chebyshev polynomial will be obtained after the K-th iteration

– But this requires the dominance ratio to be known.

– Dominance ratio can be estimated during the iteration by using the pseudo error vectors

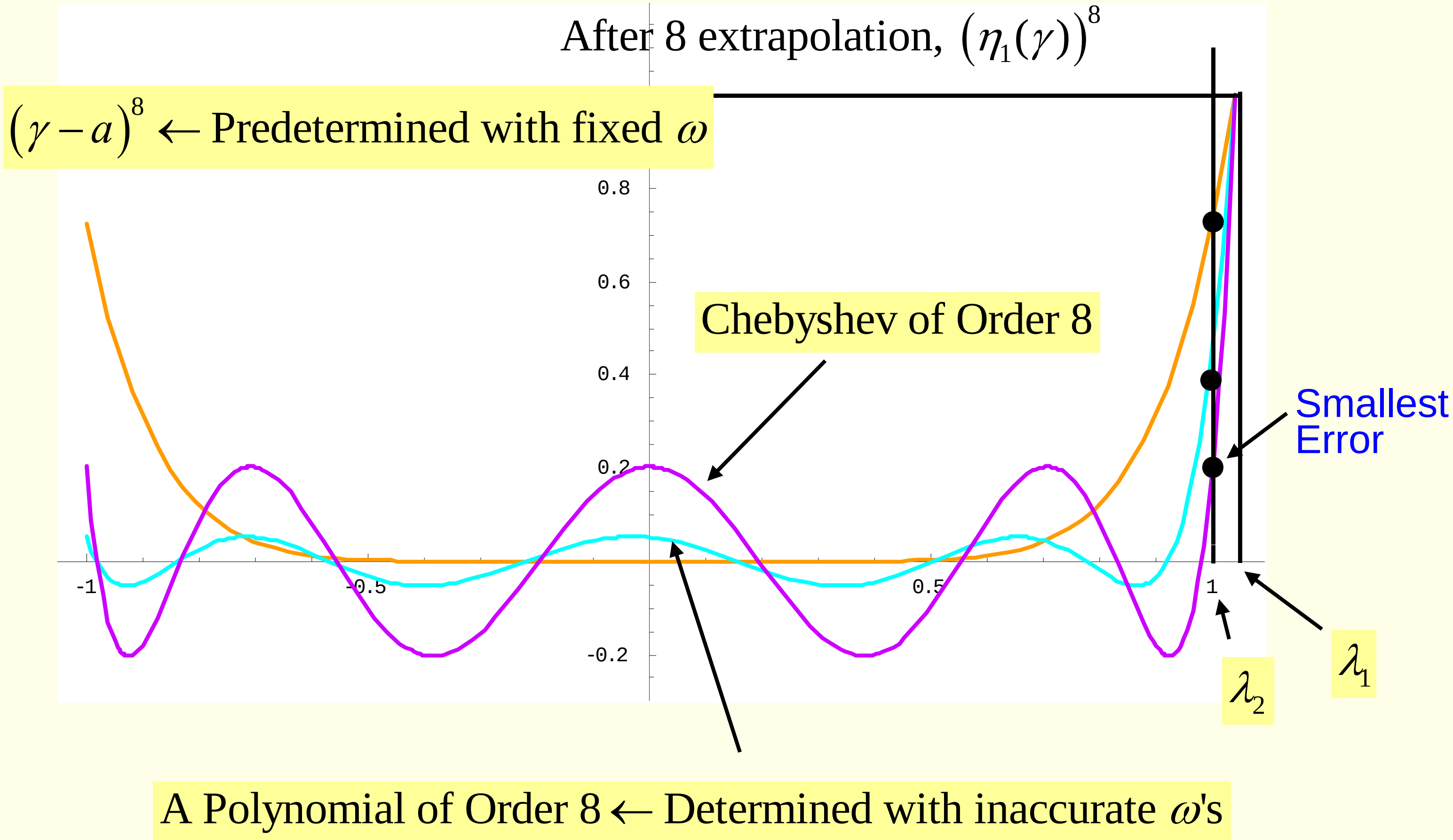
$$\tilde{e}^{(p)} = x^{(p)} - x^{(p-1)}$$

Single Parameter Chebyshev Acceleration (5/6)

- Examples for a system $\quad \sigma = 0.98 \rightarrow \gamma_1 = \frac{2}{\sigma} - 1 = 1.0408$
  - $K = 1 \rightarrow \xi_1 = 0,\ \theta_1 = \frac{1}{1 - \frac{\sigma}{2}} = 1.9608;$ Predetermined Fixed Parameter Extrapolation
  - $K = 8 \rightarrow \theta^{(p)} = 1.009, \cdots, 33.996;$ Extrapolation with Cheby
  - $K = 8,$ but with $\sigma = 0.95 \rightarrow \theta^{(p)} = 1.009, \cdots, 16.912;$ Extrapolation with Not Quite Cheby

Single Parameter Chebyshev Acceleration (6/6)

- Examples for a system $\eta_1(\gamma) = \omega^{(1)}\sigma\frac{\gamma+1}{2} + \left(1-\omega^{(1)}\right) = c(\gamma - a)$



2.2 Wielandt Shift Method

□Eigenvalue Shift

- **Consider an Eigenvalue Problem with Shifted Matrix**

$$A' = A - \alpha I$$

- If x is an eigenvector of A, it is also an eigenvector of A'.

$$A'x = (A - \alpha I)x = Ax - \alpha x = (\lambda - \alpha)x$$

$$A'x = \lambda' x$$

- Eigenvaue is shifted by α.

□Convergence of Power Method with Shifted Matrix

$$\sigma' = \frac{\lambda_2 - \alpha}{\lambda_1 - \alpha} < \frac{\lambda_2}{\lambda_1} = \sigma'$$

$$\text{Ex)}\ \sigma = \frac{\lambda_2}{\lambda_1} = \frac{0.99}{1.00} \rightarrow \sigma' = \frac{0.99 - 0.9}{1.00 - 0.9} = \frac{0.09}{0.10} = 0.9 < 0.99$$

- **Dominance ratio can be made significantly smaller if ♋ is chosen to be close to ●$_1$**

2.2 Wielandt Shift Method

☐ Inverse Power Method with Eigenvalue Shift

$$(A-\alpha I)x^{(k)} = x^{(k-1)} \Leftrightarrow x^{(k)} = (A-\alpha I)^{-1}x^{(k-1)} \quad \rightarrow x^{(k)} = (A-\alpha I)^{-k}x^{(0)}$$

$x^{(0)} = c_1u_1 + c_2u_2 + \cdots + c_nu_n$ with eigenvectors numbered such that $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$

$$(A-\alpha I)^{-1}u_i = \frac{u_i}{\lambda_1 - \alpha} \quad \rightarrow x^{(k)} = \frac{c_1u_1}{(\lambda_1-\alpha)^k} + \cdots + \frac{c_{n-1}u_{n-1}}{(\lambda_{n-1}-\alpha)^k} + \frac{c_nu_n}{(\lambda_n-\alpha)^k}$$

- If α is chosen close to λ_n, the last term dominates largely.

☐ Application Sequence

- Solve $(A-\alpha I)x^{(k)} = \hat{x}^{(k-1)}$
- Obtain $\lambda'^{(k)} = \dfrac{< x^{(k)}, \hat{x}^{(k-1)} >}{< x^{(k)}, x^{(k)} >} \rightarrow \lambda^{(k)} = \lambda'^{(k)} + \alpha$
- Scale $\hat{x}^{(k)} = \lambda'^{(k)}x^{(k)}$, then repeat iteration until convergence.
- Then repeat iteration until convergence ($|\Delta\lambda| < \varepsilon$)

3. Q-R Method

QR Factor of a Matrix

$$A = QR = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ q_{n1} & \cdots & & q_{nn} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & & \\ & & \ddots & \\ & & & r_{nn} \end{bmatrix}$$

- Q is orthogonal (unitary):

$$Q = [q_1, q_2, \cdots, q_n]$$

$$Q^T Q = I \rightarrow Q^T = Q^{-1}$$

$$\because < q_i, q_j >= \delta_{ij}$$

Similarity Transform

$A' = S^{-1}AS$ which is *similar* to A in that the eigenvalues are unchanged

- For two eigenvalue problems, $Ax = \lambda x$ and $A'x' = \lambda' x'$

$$Det(A' - \lambda' I) = Det(S^{-1}AS - \lambda' I) = Det\left(S^{-1}(AS - \lambda' S)\right) = Det\left(S^{-1}(A - \lambda' I)S\right)$$

$$= Det(S^{-1})Det(A - \lambda' I)Det(S) = Det(A - \lambda' I) = 0 \rightarrow \text{Same Chracteristic Eqn.} \rightarrow \lambda' = \lambda$$

$$A'x' = \lambda' x' \rightarrow S^{-1}ASx' = \lambda' x' \rightarrow ASx' = \lambda' Sx' \rightarrow Ax = \lambda x \rightarrow x = Sx' \text{ or } x' = S^{-1}x$$

- Eigenvalue unchanged, but eigvector changes to $x' = S^{-1}x$
- Diagonalization: $S = [u_1 \cdots u_n] \rightarrow S^{-1}AS = D$

3. Q-R Method

- **Similarity Transform Using Q (Orthogonal Transform)**

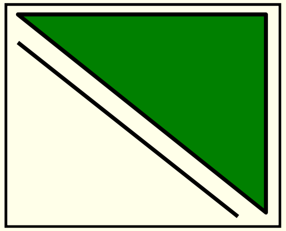
  $A' = Q^{-1}AQ = Q^{-1}QRQ = RQ$ $\leftarrow QR$ factors reversed!
  with smaller entries in lower diagonal

- **Q-R Algorithm with Repeated Orthogonal Transform**

  0. Let $A_1 = A$
  1. For A_k, determine Q_k and R_k such that $A_k = Q_k R_k$
  2. Set $A_{k+1} = R_k Q_k$ (Orthogonal transform)
  3. Repeat Steps 1 and 2 until lower-diagonal entries of A_k vanish

  $\rightarrow$ Eigenvalues are diagonal entries of A_k

Hessenberg Matrix



- **Can be applicable to any matrix to find all the eigenvalues and eigenvectors**
- **For efficient application, need to transform the matrix first into Hessenberg form (Upper triangular+one band below the diagonal) by Householder Transform**

3.1 QR Factorization

□ Objective and Results

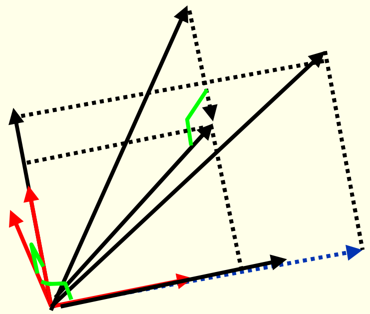
For $A = (v_1, v_2, \cdots, v_n)$,

Generate a set of orthonormal vectors $\{q_1, q_2, \cdots, q_n\}$ from $\{v_1, v_2, \cdots, v_n\}$

Let r_i be a vector consisting of the components of v_i with repect to $q_1, \cdots, q_n$
and $R = (r_1, \cdots, r_n)$, $Q = (q_1, \cdots, q_n)$

Then $A = QR$ and R is an upper triangular matrix.

□ Gram-Schmidt Orthogonalization Process



- Suppose that $q_1, \cdots, q_{k-1}$ are known at the k-th step.
- Determine the components of v_k w.r.t $q_1, \cdots, q_{k-1}$ by

$$r_{ik} = q_i^T v_k = < q_i, v_k >, \quad i = 1, \cdots, k-1$$

- Determine the new orthonormal vector q_k by

$$\tilde{q}_k = v_k - \sum_{i=1}^{k-1} r_{ik} q_i \qquad \rightarrow q_k = \frac{\tilde{q}_k}{\|\tilde{q}_k\|} \qquad \rightarrow r_{kk} = q_k^T v_k \quad \rightarrow r_k \text{ has only k elements}$$

3.1 QR Factorization

- After all orthogonalization

$$v_j = r_{1j}q_1 + r_{2j}q_2 + \cdots + r_{jj}q_j$$

$$\rightarrow v_j = [q_1, q_2, \cdots, q_n] \begin{bmatrix} r_{1j} \\ \vdots \\ r_{jj} \\ 0 \\ 0 \end{bmatrix}$$

$$Note : Bc = [u_1, u_2, \cdots, u_n] \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \leftarrow \text{Matrix - vector Product}$$

$\rightarrow$ Linear combination of column vector u_j's

with coefficients c_j

$$A = [v_1, v_2, \cdots, v_n] = [q_1, q_2, \cdots, q_n] \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots & \cdots & r_{11} \\ 0 & r_{22} & r_{23} & & & \\ 0 & 0 & r_{33} & & & \\ \vdots & \vdots & 0 & \ddots & & \\ 0 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & r_{nn} \end{bmatrix} = QR$$

3.2 Householder Transformation

□ Objective: Transform a Matrix into Hessenberg Form

- **Make the lower diagonal entries zero except the one right below the diagonal entry**

□ Householder Matrix

- For a normalize vector, $\hat{v} = \dfrac{v}{\|v\|}$ and $\hat{v}^T\hat{v} = 1$, define $H = I - 2\hat{v}\hat{v}^T$

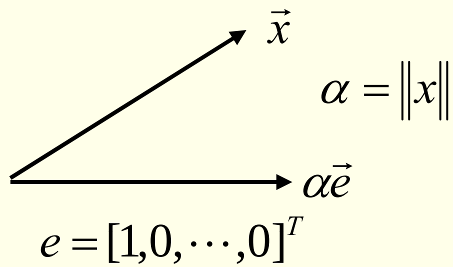
- **Properties**

$$H^T = \left(I - 2\hat{v}\hat{v}^T\right)^T = I - 2\hat{v}\hat{v}^T = H \rightarrow \text{Symmetric}$$

$$H^T H = \left(I - 2\hat{v}\hat{v}^T\right)\left(I - 2\hat{v}\hat{v}^T\right) = I - 4\hat{v}\hat{v}^T + 4\hat{v}\hat{v}^T\hat{v}\hat{v}^T = I - 4\hat{v}\hat{v}^T + 4\hat{v}\hat{v}^T = I$$
$$\rightarrow H^{-1} = H \rightarrow \text{Orthonormal}$$

□ Rotation of Vector

- **Which matrix rotates a vector x onto a vector on x axis with the same length?**

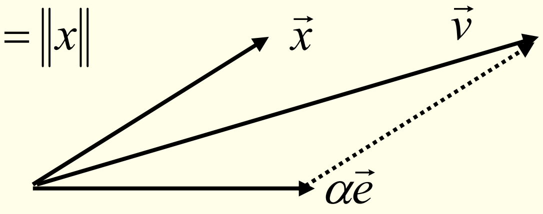


3.2 Householder Transformation

□Rotation of Vector

- Let $v = x + \alpha e$
- Then $H = I - 2\hat{v}\hat{v}^T$ rotates x to $-\alpha e$.
- Proof

$\alpha = \|x\|$



$$\alpha e = [\alpha, 0, \cdots, 0]^T$$

$$\alpha^2 = x^T x \qquad e^T x = x^T e, \; e^T e = 1$$

$$\|v\|^2 = (x+\alpha e)^T (x+\alpha e) = x^T x + \alpha e^T x + \alpha e x^T + \alpha^2 e^T e = 2\alpha^2 + 2\alpha e^T x = 2\alpha(\alpha + e^T x)$$

$$Hx = \left(I - 2\hat{v}\hat{v}^T\right) x = x - 2\frac{vv^T}{\|v\|^2} x = x - \frac{1}{\alpha(\alpha + e^T x)} vv^T x$$

$$vv^T x = (x+\alpha e)(x+\alpha e)^T x = (xx^T + \alpha e x^T + \alpha x e^T + \alpha^2 e e^T) x$$

$$= xx^T x + \alpha e x^T x + \alpha x e^T x + \alpha^2 e e^T x = \alpha^2 x + \alpha^3 e + \alpha(e^T x) x + \alpha^2 (e^T x) e$$

$$= \alpha(\alpha + e^T x) x + \alpha^2 (\alpha + e^T x) e = \alpha(\alpha + e^T x)(x + \alpha e)$$

$$\therefore Hx = x - \frac{1}{\alpha(\alpha + e^T x)} \alpha(\alpha + e^T x)(x + \alpha e) = -\alpha e \longrightarrow x = \begin{bmatrix} \chi_1 \\ \chi_2 \\ \vdots \\ \chi_n \end{bmatrix} \rightarrow Hx \rightarrow \begin{bmatrix} -\alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

3.2 Householder Transformation

□ Similarity Transform with Householder Matrix

- For Householder Matrix of Order n -1
  - Construct H out of the first column n-1 entries below diagonal of $A_1 = A$

$$U_1 = \begin{bmatrix} 1 & \begin{bmatrix} & o^T & \end{bmatrix} \\ \begin{bmatrix} \\ o \\ \\ \end{bmatrix} & \begin{bmatrix} & & \\ & H & \\ & & \end{bmatrix} \end{bmatrix} \quad A_1 = \begin{bmatrix} a_{11} & \begin{bmatrix} & y^T & \end{bmatrix} \\ \begin{bmatrix} \\ x \\ \\ \end{bmatrix} & \begin{bmatrix} & & \\ & B & \\ & & \end{bmatrix} \end{bmatrix} \qquad U_1 = U_1^{-1}$$

$$U_1^{-1}A = \begin{bmatrix} a_{11} & * & * & & * \\ -\alpha & * & & & * \\ 0 & * & & & \\ \vdots & \vdots & & & \\ 0 & * & * & & * \end{bmatrix} \qquad U_1^{-1}A = \begin{bmatrix} a_{11} + o^T x & y^T + o^T B \\ a_{11} o + Hx & oy^T + HB \end{bmatrix} = \begin{bmatrix} a_{11} & y^T \\ Hx & HB \end{bmatrix}$$

$$A_2 = U_1^{-1} A U_1 = \begin{bmatrix} a_{11} & * & * & & * \\ -\alpha & * & & & * \\ 0 & * & & & \\ \vdots & \vdots & & & \\ 0 & * & * & & * \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & H & \\ 0 & & & \end{bmatrix} = \begin{bmatrix} a_{11} & * & * & & * \\ -\alpha & * & & & * \\ 0 & * & & & \\ \vdots & \vdots & & & \\ 0 & * & * & & * \end{bmatrix}$$

U_1 multiplied additionally for similarity transform

3.2 Householder Transformation

□ Subsequent Householder Transformation with Reduced Order

– Construct H out of the second column n-2 entries below diagonal of A_2

$$U_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & & & \\ 0 & 0 & & H & \\ 0 & 0 & & & \end{bmatrix} \qquad U_2^{-1}\left(U_1^{-1} A U_1\right)U_2 = \begin{bmatrix} a_{11} & * & * & & * \\ * & * & & & * \\ 0 & * & * & & \\ \vdots & 0 & * & * & \\ 0 & 0 & * & * & * \end{bmatrix}$$

– Continue n-2 steps of Householder Transformation

$$U^{-1} = U_{n-2}^{-1} U_{n-3}^{-1} \cdots U_1^{-1} \rightarrow U = U_1 U_2 \cdots U_{n-2}$$

$A' = U^{-1} A U$ is a similarity transform of A which appears as Hessenberg form.

3.2 Householder Transformation

□Householder Transformation for Symmetric Matrix

$$U_1 = \begin{bmatrix} 1 & \begin{bmatrix} o^T \end{bmatrix} \\ \begin{bmatrix} o \end{bmatrix} & \begin{bmatrix} H \end{bmatrix} \end{bmatrix} \qquad A_1 = \begin{bmatrix} a_{11} & \begin{bmatrix} x^T \end{bmatrix} \\ \begin{bmatrix} x \end{bmatrix} & \begin{bmatrix} B \end{bmatrix} \end{bmatrix}$$

$$U_1 A_1 = \begin{bmatrix} a_{11} + o^T x & \begin{bmatrix} x^T + o^T B \end{bmatrix} \\ \begin{bmatrix} a_{11} o + Hx \end{bmatrix} & \begin{bmatrix} o x^T + HB \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_{11} & \begin{bmatrix} x^T \end{bmatrix} \\ \begin{bmatrix} Hx \end{bmatrix} & \begin{bmatrix} HB \end{bmatrix} \end{bmatrix}$$

$$U_1 A_1 U_1 = \begin{bmatrix} a_{11} & \begin{bmatrix} x^T \end{bmatrix} \\ \begin{bmatrix} Hx \end{bmatrix} & \begin{bmatrix} HB \end{bmatrix} \end{bmatrix} \begin{bmatrix} 1 & \begin{bmatrix} o^T \end{bmatrix} \\ \begin{bmatrix} o \end{bmatrix} & \begin{bmatrix} H \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_{11} & \begin{bmatrix} x^T H \end{bmatrix} \\ \begin{bmatrix} Hx \end{bmatrix} & \begin{bmatrix} HBH \end{bmatrix} \end{bmatrix}$$

3.2 Householder Transformation

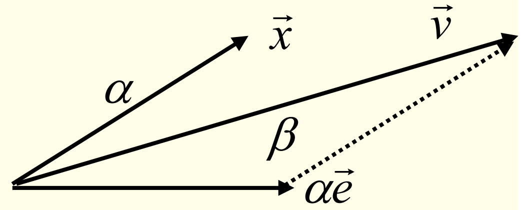
□Householder Transformation for Symmetric Matrix

$$Hx = \begin{bmatrix} 1 \\ -\alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} \longrightarrow \begin{bmatrix} a_{11} & \begin{bmatrix} -\alpha & 0 & \cdots & 0 \end{bmatrix} \\ \begin{bmatrix} -\alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} & \begin{bmatrix} \\ HBH \\ \\ \end{bmatrix} \end{bmatrix}$$

- **Elements on the right of the diagonal were also eliminated except the one right next to the diagonal.**
- **Continued Householder transformation with lower dimensions yields tridiagonal matrix**

Practical Householder Algorithm

$$x^{(k)} = \begin{bmatrix} a_{k+1,k} \\ a_{k+2,k} \\ \vdots \\ a_{n,k} \end{bmatrix},\ e^{(k)} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \alpha = \left\| x^{(k)} \right\| = \sqrt{\sum_{i=k+1}^{n} a_{ik}^2},\ v = x^{(k)} + \alpha e^{(k)}$$



$$\beta = \|v\| = \sqrt{(a_{k+1,k} + \alpha)^2 + \sum_{i=k+2}^{n} a_{ik}^2} = \sqrt{a_{k+1,k}^2 + 2\alpha a_{k+1,k} + \alpha^2 + \sum_{i=k+2}^{n} a_{ik}^2} \quad = \sqrt{2(\alpha a_{k+1,k} + \alpha^2)}$$

$$\hat{v} = \frac{1}{\beta} \begin{bmatrix} a_{k+1,k} + \alpha \\ a_{k+2,k} \\ \vdots \\ a_{n,k} \end{bmatrix} \qquad H_k = I - 2\hat{v}\hat{v}^T \qquad U_k = \begin{bmatrix} I_k & O_k^T \\ O_k & H_k \end{bmatrix} \longrightarrow A_{k+1} = U_k A_k U_k$$

$$\dim(I_k) = k$$

$$\dim(H_k) = n - k$$

- What if $\beta = 0$? $\leftarrow \alpha = -a_{k+1,k} \rightarrow a_{k+1,k} < 0$ and $a_{i,k} = 0 \quad \forall i \geq k+2$

Algorithm breaks down. $\longrightarrow$ Remedy: Choose $\alpha = -\left\| x^{(k)} \right\|$ if $a_{k+1,k} < 0$.

3.2 Householder Transform

MATLAB Script

```
%function H=householder(A)
%
% dimension of input matrix
n=length(A);
%
% matrix to contain householder transform matrix at each step
U=zeros(n);
H=A;
In=eye(n); %identity matrix of dimension n
for k=1:n-2
   nmk=n-k;
   kp1=k+1;
   x=H(k+1:n,k);
   alpha=norm(x);
   if(x(1)<0) alpha=-alpha; end %make sure the same direction
   v=x;
   v(1)=v(1)+alpha;
   beta=norm(v);
   vhat=v/beta;
   Hk=eye(nmk)-2*vhat*vhat'; %householder matrix of order n-k
%    U=In;
%    U(kp1:n,kp1:n)=Hk;
%    H2=U*H*U;
   H(kp1,k)=-alpha;
   H(kp1+1:n,k)=0; %zero out the lower diagonal entries
   H(kp1:n,kp1:n)=Hk*H(kp1:n,kp1:n)*Hk; %update the remainder
   H(k,kp1:n)=H(k,kp1:n)*Hk; %update the row vector
%    H2-H
   pause
end
```

Q-R Factorization of a Hessenberg Matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \ddots & & \vdots \\ 0 & \ddots & \ddots & \vdots \\ 0 & 0 & a_{nn-1} & a_{nn} \end{bmatrix} = QR = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ q_{n1} & \cdots & & q_{nn} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & & \\ & & \ddots & \\ & & & r_{nn} \end{bmatrix}$$

$$\longrightarrow Q^{-1}A = R$$

$$\begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \\ 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} a'_{12} \\ 0 \\ a'_{32} \\ 0 \\ 0 \end{bmatrix}$$

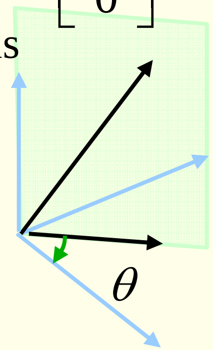
- Which operation on A can make A to upper triagonal matrix?

  $\rightarrow$ Eliminate the subdiagonal entry one-by-one.

- Plane Rotation

  - Rotate the plane normal to the xy-plane by θ so that it is aligned with the x-axis to remove $y-$component in the rotated vector

$$\text{First Rotation Matrix, } P_1 = \begin{bmatrix} \cos\theta & -\sin\theta & & \\ \sin\theta & \cos\theta & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$



3.3 Q-R Method

Determination of Rotation Angle

$$\begin{bmatrix} \cos\theta & -\sin\theta & & \\ \sin\theta & \cos\theta & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{21} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} a'_{11} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$a_{11}\sin\theta + a_{21}\cos\theta = 0$$

$$\sin\theta = -\frac{a_{12}}{\sqrt{a_{11}^2 + a_{21}^2}}$$

$$a_{11}^2(1-\mu^2) = a_{21}^2\mu^2 \rightarrow \mu = \frac{a_{11}}{\sqrt{a_{11}^2 + a_{21}^2}} = \cos\theta$$

$$a'_{11} = a_{11}\frac{a_{11}}{\sqrt{a_{11}^2 + a_{21}^2}} + a_2\frac{a_{21}}{\sqrt{a_{11}^2 + a_{21}^2}}$$

Repeated Plane Rotation

$$P_k = \begin{bmatrix} I_k & & & \\ & \cos\theta & -\sin\theta & \\ & \sin\theta & \cos\theta & \\ & & & I_{n-k-2} \end{bmatrix}$$

← k - th row

Properties of Rotation Matrix P_k

- P is an orthonormal matrix $\rightarrow P^{-1} = P^T$

Final Form

$$P_{n-1}P_{n-2}\cdots P_1 A = R$$

$$A = P_1^{-1}P_2^{-1}\cdots P_{n-1}^{-1}R = P_1^T P_2^T \cdots P_{n-1}^T R$$

$$Q = P_1^T P_2^T \cdots P_{n-1}^T$$

3.3 Q-R Method

□ Eigenvalue Shift

- Consider QR factorization of $H_k - \lambda^{(k)} I = Q_k R_k$
- Similarity Transform : $Q_k^{-1}\left(H_k - \lambda^{(k)} I\right) Q_k = R_k Q_k$

  $Q_k^{-1} H_k Q_k = R_k Q_k + \lambda^{(k)} I \equiv H_{k+1} \rightarrow$ Similar to H_k
- Shifted Factorization with the last diagonal entry of H as a guess of eigenvalue can accelerate convergence

□ Summary of QR Factorization

1. Obtain a Hessenberg from by a Series of Householder Transformation : $H_1 = U^{-1} A U$
2. Shift Eigenvalue : $\tilde{H}_k = H_k - \lambda^{(k)} I; \lambda^{(k)} = H_k(n,n)$
3. Perform Q - R Factorization by Successvie Plane Rotation : $\tilde{H}_k = Q_k R_k$
4. Determine New Matrix by $H_{k+1} = R_k Q_k + \lambda^{(k)} I$
5. Repeat Steps 2 - 4 until subdiagonal entries are small enough.

3.3 Q-R Method

□Determination of Eigenvector after Getting Eigenvalues

- Inverse Power Method for $A-\tilde{\lambda}_k I$ with $\tilde{\lambda}_k \approx \lambda_k$

$$\left(A-\tilde{\lambda}_k I\right)x^{(1)} = \hat{x}^{(0)} \rightarrow x^{(1)} = \left(A-\tilde{\lambda}_k I\right)^{-1}\hat{x}^{(0)} = \sum_{i=1}^{n}\frac{c_i u_i}{\lambda_i - \tilde{\lambda}_k} \cong \frac{c_i u_i}{\lambda_k - \tilde{\lambda}_k}$$

$\rightarrow$ Single step of inverse power method is sufficient to obtain eigenvector!

3.3 Q-R Method

MATLAB Script

```
%function lam=qralgo(H);
% assume hessenberg form for H
n=length(H);
% identity matrix of dimension n
In=eye(n);
while (1>0)
    %factorize by plane rotation
    lam=H(n,n); %guess of eigenvector
    Hk=H-lam*In; %shift eigenvalue
    Q=In; %reset Q to identity matrix before factorization
    for i=1:n-1
      ip1=i+1;
      d=Hk(i,i); %diagonnal entry
      s=Hk(i+1,i); %subdiagonal entry to eliminate
      afac=1/sqrt(d*d+s*s);
      cosv=d*afac; %cosine value
      sinv=-s/d*cosv; %sine value, this way ensures s*sinv+d*cosv=0
      Hk(i,i)=d*cosv-s*sinv; %new diagonal value after rotation
      Hk(i+1,i)=0; %subdiagonal entry now being 0
      P=In; %rotation matrix initially identity
      P(i,i)=cosv;P(i,ip1)=-sinv; %fill rotation matrix
      P(ip1,i)=sinv;P(ip1,i+1)=cosv;
      Hk(i:ip1,i+1:n)=P(i:ip1,i:ip1)*Hk(i:ip1,i+1:n); %update entries on
        plane
      Pt=P'; %transpose of rotation matrix
      Q=Q*Pt; %Q matrix being constructed stepwise
    end
    R=Hk; %upper trianglular matrix
    H=R*Q+lam*In; %new matrix
    diag(H) %current guess of eigenvalues
    pause
end
```