Real Asymmetric Matrix Eigenvalue Analysis

Heewook Lee

Computational Mechanics Laboratory
Department of Mechanical Engineering and Applied Mechanics
University of Michigan
Ann Arbor, MI

Contents of Presentation

- **Numerical Algorithms**
  - **QR**
  - **QZ**

- **Examples**

Eigenvalue problem

$$\mathbf{Ax} = \lambda\mathbf{Bx}$$

$$\mathbf{Cx} = \lambda\mathbf{x}, \quad \mathbf{C} = \mathbf{B}^{-1}\mathbf{A}$$

- **C => Hessenberg form**

  O(n^3) => O(n^2)
- **QR algorithm**

Hessenberg Matrix

- **Elementary similarity transformation**

- **Orthogonal transformation**

Hessenberg Matrix

- **Maximum**

  ***i*+1 = *m* ?**

$$\mathbf{C} = \begin{bmatrix} 2 & -3 & 1 & -2 \\ 1 & 1 & 2 & 3 \\ 4 & 3 & 2 & -1 \\ 1 & -3 & 1 & 4 \end{bmatrix}$$

m

i

- **row & column interchange**

$$\mathbf{C} = \begin{bmatrix} 2 & 1 & -3 & -2 \\ 4 & 2 & 1 & -1 \\ 1 & 2 & 3 & 3 \\ 1 & 1 & -3 & 4 \end{bmatrix}$$

Hessenberg Matrix

$$y = \frac{a_{31}}{\max(c_{i1})}, \quad i = 2,\ldots,4$$

$$a_{3j} = a_{3j} - y \times a_{2j},$$

$$a_{j2} = a_{j2} + y \times a_{j3}, \quad j = 1,\ldots,4$$

$$\mathbf{C} = \begin{bmatrix} 2 & 0.25 & -3 & -2 \\ 4 & 2.25 & 1 & -1 \\ 0 & 2.1875 & 2.75 & -2.75 \\ 1 & 0.25 & -3 & 4 \end{bmatrix}$$

$$y = \frac{a_{41}}{\max(c_{i1})}, \quad i = 2,\ldots,4$$

$$a_{4j} = a_{4j} - y \times a_{2j},$$

$$a_{j2} = a_{j2} + y \times a_{j4}, \quad j = 1,\ldots,4$$

$$\mathbf{C} = \begin{bmatrix} 2 & -0.25 & -3 & -2 \\ 4 & 2 & 1 & -1 \\ 0 & 1.5 & 2.75 & -2.75 \\ 0 & 0.9375 & -3.25 & 5 \end{bmatrix}$$

Hessenberg Matrix

- **Maximum**

  ***i*+1 = *m* ?**

$$\mathbf{C}=\begin{bmatrix}2 & -0.25 & -3 & -2\\ 4 & 2 & 1 & -1\\ 0 & 1.5 & 2.75 & -2.75\\ 0 & 0.9375 & -3.25 & 5\end{bmatrix}$$

$$y=\frac{a_{42}}{\max(c_{i2})},\quad i=3,4$$

$$\mathbf{C}=\begin{bmatrix}2 & -0.25 & -4.25 & -2\\ 4 & 2 & 0.375 & -1\\ 0 & 1.5 & 1.03125 & -2.75\\ 0 & 0 & -0.76953 & 6.71875\end{bmatrix}$$

$$a_{4j}=a_{4j}-y\times a_{3j},\quad j=2,\dots,4$$

$$a_{j3}=a_{j3}+y\times a_{j4},\quad j=1,\dots,4$$

QR algorithm

- **Single shift QR**
- **Complex version of two step QR**

$$\begin{bmatrix} \lambda_1 & & & X \\ & \lambda_2 & & \\ & & \lambda_3 & \\ 0 & & & \lambda_4 \end{bmatrix}$$

- **Double shift QR**

$$\begin{bmatrix} \lambda_1 & & X \\ & \boxed{\begin{matrix} \lambda_2 & \\ X & \lambda_3 \end{matrix}} & \\ 0 & & \lambda_4 \end{bmatrix}$$

Single shift QR algorithm

$$\mathbf{C}_{k+1} = \mathbf{Q}_k{}^H \mathbf{C}_k \mathbf{Q}_k$$

$\mathbf{Q}_k$ is unitary matrix

$$\mathbf{Q}_k{}^H \mathbf{C}_k = \mathbf{R}_k$$

$\mathbf{R}_k$ is upper triangular matrix

$$\mathbf{Q}_k{}^H (\mathbf{C}_k - \zeta_k \mathbf{I}) = \mathbf{R}_k$$

Complex version of two step QR

$$\mathbf{C}_{k+1} = \mathbf{Q}_k{}^H \mathbf{C}_k \mathbf{Q}_k$$

$$\mathbf{Q}_k{}^H (\mathbf{C}_k - \zeta_k \mathbf{I}) = \mathbf{R}_k$$

$$\zeta_{k+1} = \overline{\zeta}_k$$

$$\mathbf{C}_{k+2} = \mathbf{Q}_{k+1}{}^H \mathbf{C}_{k+1} \mathbf{Q}_{k+1}$$

$$\mathbf{Q}_{k+1}{}^H (\mathbf{C}_{k+1} - \zeta_{k+1} \mathbf{I}) = \mathbf{R}_{k+1}$$

Double shift QR algorithm

$$\mathbf{C}_{k+2} = \mathbf{Q}_{k+1}^T \mathbf{Q}_k^T \mathbf{C}_k \mathbf{Q}_k \mathbf{Q}_{k+1}$$

$$\mathbf{Q}_{k+1}^T \mathbf{Q}_k^T (\mathbf{C}_k - \zeta_k \mathbf{I})(\mathbf{C}_k - \zeta_{k+1} \mathbf{I}) = \mathbf{R}_k \mathbf{R}_{k+1}$$

$$\mathbf{Q}_{k+1} \mathbf{Q}_k = \mathbf{Q},\ \ \mathbf{R}_k \mathbf{R}_{k+1} = \mathbf{R},\ \ (\mathbf{C}_k - \zeta_k \mathbf{I})(\mathbf{C}_k - \zeta_{k+1} \mathbf{I}) = \Gamma$$

$$\mathbf{C}_k \mathbf{Q} = \mathbf{Q} \mathbf{C}_{k+2}, \quad \mathbf{Q}^T \Gamma = \mathbf{R}$$

Double shift QR algorithm

If a matrix $\mathbf{H}$ is derived from $\mathbf{C}_k$ such that

$$\mathbf{C}_k\hat{\mathbf{Q}} = \hat{\mathbf{Q}}\mathbf{H} \quad \text{or} \quad \hat{\mathbf{Q}}^T\mathbf{C}_k\hat{\mathbf{Q}} = \mathbf{H}$$

where $\hat{\mathbf{Q}}$ is orthogonal

$\mathbf{H}$ is upper Hessenberg.

If $\hat{\mathbf{Q}}$ has the same first column as $\mathbf{Q}$, then

$$\hat{\mathbf{Q}} = \mathbf{Q} \quad \text{and} \quad \mathbf{C}_{k+2} = \mathbf{H}$$

Double shift QR algorithm

$$\mathbf{P}_r = \mathbf{I} - 2\mathbf{w}_r\mathbf{w}_r^T$$

$$\mathbf{P}_{n-1}^T \cdots \mathbf{P}_3^T \mathbf{P}_2^T \mathbf{P}_1^T \mathbf{C}_k \mathbf{P}_1 \mathbf{P}_2 \mathbf{P}_3 \cdots \mathbf{P}_{n-1} = \mathbf{H}$$

$$(\gamma_{11}, \gamma_{21}, \gamma_{31}, 0, \cdots, 0)^T$$

$$\gamma_{11} = c_{11}^2 - c_{11}(\zeta_1 + \zeta_2) + \zeta_1\zeta_2 + c_{12}c_{21}$$

$$\gamma_{21} = c_{21}(c_{11} + c_{22} - \zeta_1 - \zeta_2)$$

$$\gamma_{31} = c_{32}c_{21}$$

Double shift QR

$$\mathbf{C}=\begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & . & . & c_{1n} \\ c_{21} & c_{22} & c_{23} & c_{24} & . & . & c_{2n} \\ & c_{32} & c_{33} & c_{34} & . & . & c_{3n} \\ & & c_{43} & c_{44} & . & . & c_{4n} \\ & & & c_{54} & . & . & . \\ & & & & . & . & . \\ & & & & & . & c_{nn} \end{bmatrix}$$

$$\longrightarrow \qquad \mathbf{P}_1^T\mathbf{C}_k\mathbf{P}_1=\begin{bmatrix} \underline{c'_{11}} & \underline{c'_{12}} & \underline{c'_{13}} & \underline{c_{14}} & . & . & c_{1n} \\ \underline{c'_{21}} & \underline{c'_{22}} & \underline{c'_{23}} & \underline{c_{24}} & . & . & c_{2n} \\ \underline{c'_{31}} & \underline{c'_{32}} & \underline{c'_{33}} & \underline{c_{34}} & . & . & c_{3n} \\ c'_{41} & c'_{42} & c'_{43} & c_{44} & . & . & c_{4n} \\ & & & c_{54} & . & . & . \\ & & & & . & . & . \\ & & & & & . & c_{nn} \end{bmatrix}$$

Double shift QR

$$\begin{bmatrix} h & h & h & z & z & z & z & z & z \\ h & h & h & z & z & z & z & z & z \\ & h & h & z & z & z & z & z & z \\ & & d & d & d & z & z & z & z \\ & & d & d & d & z & z & z & z \\ & & d & d & d & a & a & a & a \\ & & & & & a & a & a & a \\ & & & & & & a & a & a \\ & & & & & & & a & a \end{bmatrix} \longrightarrow \begin{bmatrix} h & h & h & h' & z' & z' & z & z & z \\ h & h & h & h' & z' & z' & z & z & z \\ & h & h & h' & z' & z' & z & z & z \\ & & \underline{h} & \underline{h'} & \underline{z'} & \underline{z'} & \underline{z} & \underline{z} & \underline{z} \\ & & \underline{0} & \underline{d'} & \underline{d'} & \underline{d'} & \underline{z} & \underline{z} & \underline{z} \\ & & \underline{0} & \underline{d'} & \underline{d'} & \underline{d'} & \underline{z} & \underline{z} & \underline{z} \\ & & & d' & d' & d' & a & a & a \\ & & & & & & a & a & a \\ & & & & & & & a & a \end{bmatrix}$$

$$(0, 0, \ldots, 0, \alpha_r, \beta_r, \chi_r, 0, \cdots, 0)^T$$

Shifting Rule

Before each iteration, calculate the roots of the last principal 2×2 submatrix.

$\longrightarrow$ η_k and η_{k+1}

The choice of the origin shifts ζ_k and ζ_{k+1} for the iteration depends on;

$$\rho_1 = \left| \frac{\eta_k - \eta_{k-2}}{\eta_k} \right|, \qquad \rho_2 = \left| \frac{\eta_{k+1} - \eta_{k-1}}{\eta_{k+1}} \right|.$$

If they are both greater than 0.5, $\zeta_k = \zeta_{k+1} = 0$

if they are both less than 0.5, $\zeta_k = \eta_k$ and $\zeta_{k+1} = \eta_{k+1}$

otherwise we set both ζ_k and ζ_{k+1} to be the real part of either η_k or η_{k+1},

whichever corresponds to the quantity less than 0.5

Double shift QR

$$\mathbf{C}=\begin{bmatrix} 2 & -0.25 & -4.25 & -2 \\ 4 & 2 & 0.375 & -1 \\ 0 & 1.5 & 1.03125 & -2.75 \\ 0 & 0 & -0.76953 & 6.71875 \end{bmatrix}$$

$$\mathbf{C}=\begin{bmatrix} -1.659 & 3.425 & 7.386 & -4.548 \\ -1.391 & 3.195 & -.871 & -3.081 \\ -.888 & 3.401 & 3.919 & 1.955 \\ -.281 & -.200 & -.283 & 4.209 \end{bmatrix}$$

Double shift QR

$$\mathbf{C}=\begin{bmatrix} 1.809 & 3.425 & .5607 & 6.136 \\ .4191 & -4.338 & -2.886 & .5140 \\ .0719 & 4.510 & 1.925 & .7322 \\ 1E8 & -2E-9 & \underline{0} & \underline{4.179} \end{bmatrix}$$

$$\mathbf{C}=\begin{bmatrix} -1.354 & -6.977 & -.2425 & 6.136 \\ 0 & 4.111 & -2.773 & .5140 \\ 0 & 4.590 & 2.063 & .7322 \\ 0 & 0 & 0 & 4.179 \end{bmatrix}$$

Double shift QR algorithm

$$\lambda = -1.3537, \qquad 3.08773 \pm 3.4176, \qquad 4.1791$$

$$\mathbf{x} = \begin{Bmatrix} .3703 \\ .6166 \\ -.3944 \\ -.2809 \end{Bmatrix} \quad \begin{Bmatrix} .0520 \pm .6335i \\ -.1717 \mp .7388i \\ -.2325 \pm .3059i \\ -1.0068 \pm .3820i \end{Bmatrix} \quad \begin{Bmatrix} .6869 \\ .5560 \\ .4434 \\ -.8327 \end{Bmatrix}$$

MATLAB

Eig(A,B) where A is the same as C and B is identity matrix

```
Eigenvectors =

   -0.20097194777325 - 0.50105007827491i        0.54730339411280 - 0.09937518469598i
   -0.08340643758769 - 0.20794345949761i       -0.26613462932493 + 0.48878652750776i
    0.29513087471146 + 0.73580093895674i       -0.07049854771963 + 0.36026022392661i
   -0.06432521788507 - 0.16037141408775i        0.23622760645172 + 0.43621405391860i

    0.10493664007399 - 0.10402029787529i        0.12611544780603 + 0.54176685609719i
    0.25531452586401 - 0.25308503315463i        0.34292600567206 - 0.43833966855946i
    0.62818884756511 - 0.62270329028621i        0.30264859155449 - 0.20775298843654i
   -0.18321243924275 + 0.18161256631669i        0.49415131026397 + 0.04359662200692i

Eigenvalues =
    -1.35370498575813 + 10E-17i                 3.08728955138727 - 3.41762519028197i
     4.17912588298358 - 10E-15i                 3.08728955138727 + 3.41762519028197i
```

Nastran upper Hessenberg method

$$\lambda = -1.3537 + 10^{-9} i, \qquad 3.08773 \pm 3.4176, \qquad 4.1791 + 2 \times 10^{-9} i$$

$$\mathbf{x} = \begin{Bmatrix} -.695 - 4.48 \times 10^{-17} i \\ -.229 - 5.43 \times 10^{-17} i \\ 1 \\ -.268 + 6.29 \times 10^{-18} i \end{Bmatrix} \begin{Bmatrix} 1 \\ -.621 \pm .779 i \\ -.223 \pm .653 i \\ .227 \pm .822 i \end{Bmatrix} \begin{Bmatrix} .045 - 3.67 \times 10^{-18} i \\ .467 - 2.37 \times 10^{-17} i \\ 1 \\ -.242 + 6.92 \times 10^{-18} i \end{Bmatrix}$$

Complex Eigenvalue Solver

	MSC/NASTRAN		QZ algorithm	
	Real	**Imaginary**	**Real**	**Imaginary**
1	-1.31×10^{-2}	1.67×10^{3}	0	1.67×10^{3}
2	2.79×10^{-2}	2.38×10^{3}	0	2.38×10^{3}
3	-1.77×10^{-2}	2.66×10^{3}	0	2.66×10^{3}
29	-4.46×10^{1}	1.53×10^{4}	4.46×10^{1}	1.53×10^{4}
30	4.46×10^{1}	1.53×10^{4}	-4.46×10^{1}	1.53×10^{4}
31	-1.07×10^{-3}	1.71×10^{4}	0	1.71×10^{4}
32	-1.95×10^{-3}	1.76×10^{4}	0	1.76×10^{4}
73	2.06×10^{-3}	3.88×10^{4}	0	3.88×10^{4}
74	-2.42×10^{1}	3.90×10^{4}	2.42×10^{1}	3.90×10^{4}
75	2.42×10^{1}	3.90×10^{4}	-2.42×10^{1}	3.90×10^{4}
76	1.44×10^{-3}	4.04×10^{4}	0	4.04×10^{4}

QZ algorithm

$$\mathbf{Ax} = \lambda \mathbf{Bx}$$

$$\mathbf{QAZy} = \lambda \mathbf{QBZy}, \qquad \mathbf{x} = \mathbf{Zy}$$

- **A is reduced to upper Hessenberg**
  B is reduced to upper triangular
- **A is reduced to quasi-triangular**
  (generalization of QR)

Conclusions

- **For a real asymmetric eigenvalue problem, double shift QR or QZ algorithm should be used**

- **Other methods produce fictitious complex eigenvalues**