CS450: Numerical Analysis

Howard Hao Yang

Assistant Professor, ZJU-UIUC Institute

17/11/2023

Eigenvalue Problems: Setup

- Given an $n \times n$ matrix $\boldsymbol{A}$
  - $\boldsymbol{x} \neq \boldsymbol{0}$ is called an eigenvector of if there exists a λ such that

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}$$

  - In that case, λ is called an eigenvalue
  - An eigenvector of a matrix determines a direction in which the matrix expands or shrinks any vector lying in that direction by a scalar multiple, given by the eigenvalue
  - For LU and QR, we can obtain exact answers (except rounding).
  - For eigenvalue problems: not possible—we must perform approximations.

Recap: Power Method

$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$

$err\_k \sim \left(\frac{|\lambda_2|}{|\lambda_1|}\right)^k$

- Algorithm details

Algorithm 4.1 Power Iteration

$\boldsymbol{x}_0 =$ arbitrary nonzero vector
for $k = 1, 2, \ldots$
  $\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1}$ $\sim$ $x_k = \frac{x_k}{\|x_k\|_2}$ { generate next vector }
end

- This algorithm finds the eigenvector corresponding to the maximum eigenvalue (a.k.a. the leading eigenvalue)
- More concretely, this algorithm converges to a multiple of $\boldsymbol{x}_1$

Recap: Rayleigh Quotient Iteration

- In summary, compute the shift σ using Rayleigh quotient as $\sigma_k = \frac{x_k^T A x_k}{x_k^T x_k}$
- Then, apply inverse iteration with that shift: $\boldsymbol{x_{k+1}} = (\boldsymbol{A} - \sigma_k \boldsymbol{I})^{-1} \boldsymbol{x_k}$
- More concretely

Algorithm 4.4 Rayleigh Quotient Iteration

$\boldsymbol{x}_0 =$ arbitrary nonzero vector
for $k = 1, 2, \ldots$
  $\sigma_k = \boldsymbol{x}_{k-1}^T \boldsymbol{A} \boldsymbol{x}_{k-1} / \boldsymbol{x}_{k-1}^T \boldsymbol{x}_{k-1}$ { compute shift }
  Solve $(\boldsymbol{A} - \sigma_k \boldsymbol{I})\, \boldsymbol{y}_k = \boldsymbol{x}_{k-1}$ for $\boldsymbol{y}_k$ { generate next vector }
  $\boldsymbol{x}_k = \boldsymbol{y}_k / \|\boldsymbol{y}_k\|_\infty$ { normalize }
end

Eigenvalue Decomposition

- If a $n \times n$ matrix $\boldsymbol{A}$ has distinct eigenvectors, it has an eigenvalue decomposition as

$$\boldsymbol{A} = \boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^{-1} \rightsquigarrow \lambda_1 x_1 x_1^T + \lambda_2 x_2 x_2^T + \cdots + \lambda_n x_n x_n^T$$

where $\boldsymbol{X}$ are the right eigenvectors, $\boldsymbol{X}^{-1}$ are left eigenvectors, and $\boldsymbol{D}$ are eigenvalues

$$\boldsymbol{A}\boldsymbol{X} = [\boldsymbol{A}\boldsymbol{x}_1, \ldots, \boldsymbol{A}\boldsymbol{x}_n] = \boldsymbol{X}\boldsymbol{D} = [d_{11}\boldsymbol{x}_1, \ldots, d_{nn}\boldsymbol{x}_n]$$

$A x_1 = \lambda_1 x_1, \quad A x_2 = \lambda_2 x_2 \quad \cdots \quad A x_n = \lambda_n x_n$

$$A[x_1, x_2] = [x_1, x_2]\begin{bmatrix}\lambda_1 & 0 \\ 0 & \lambda_2\end{bmatrix} \Rightarrow A[x_1, x_2, \cdots, x_n] = [x_1, x_2, \cdots, x_n]\begin{bmatrix}\lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \\ & & & \lambda_n\end{bmatrix}$$

$A \quad X \qquad\qquad X \qquad\qquad D$

Similar Matrices

- Definition (similar matrices): matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are said to be similar, if there exist a nonsingular matrix $\boldsymbol{S}$ such that $\boldsymbol{A} = \boldsymbol{SBS^{-1}}$
- Theorem: suppose matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are similar matrices with $\boldsymbol{A} = \boldsymbol{SBS^{-1}}$, and λ is an eigenvalue of $\boldsymbol{A}$ with associated eigenvector $\boldsymbol{x}$. Then, λ is also an eigenvalue of $\boldsymbol{B}$ with associated eigenvector $\boldsymbol{S^{-1}x}$
- --> Similar matrices have same eigenvalues
- Is the reverse true?

$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ vs $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

(λ, x)

$Ax = \lambda x$

$S^{-1}SBS^{-1}x = \lambda x$ ← S^{-1}

$\Rightarrow B(S^{-1}x) = \lambda(S^{-1}x)$

Similar Matrices (Cont'd)

- Are the following matrices similar to each other?

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

$B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} A \quad \Rightarrow \quad B \neq SAS^{-1}$

Similar Matrices (Cont'd)

$A = XDX^{-1}$

- Theorem: an $n \times n$ matrix $\boldsymbol{A}$ is similar to a diagonal matrix $\boldsymbol{D}$ if and only if $\boldsymbol{A}$ has n linearly independent eigenvectors
- Theorem (Schur): let $\boldsymbol{A}$ be an arbitrary $n \times n$ matrix. A nonsingular matrix $\boldsymbol{U}$ exists with the property that

$$\boldsymbol{T} = \boldsymbol{U}^{-1}\boldsymbol{A}\boldsymbol{U}$$

where $\boldsymbol{T}$ is an upper triangular matrix whose diagonal entries consist of the eigenvalues of $\boldsymbol{A}$

Symmetric Matrices

ZJU-UIUC INSTITUTE
Zhejiang University/University of Illinois at Urbana-Champaign Institute

- Theorem: an $n \times n$ matrix $\boldsymbol{A}$ is symmetric if and only if there exists a diagonal matrix $\boldsymbol{D}$ and an orthogonal matrix $\boldsymbol{Q}$ with $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^{\boldsymbol{T}}$
- Theorem: let $\boldsymbol{A}$ be a symmetric $n \times n$ matrix. There exists n eigenvectors of $\boldsymbol{A}$ that form an orthonormal set, and the eigenvalues of $\boldsymbol{A}$ are real.

Sensitivity

- Recall condition number
  - The condition number is a property of the problem, measuring its sensitivity to perturbations in input
  - How much a small change in the input leads to changes in the output
- Investigate the condition number/sensitivity of an eigenvalue problem
- Setting: Suppose $\boldsymbol{A}$ is nonsingular. Suppose that $\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X} = \boldsymbol{D}$. For a certain perturbation in the input, i.e., $\boldsymbol{A} \to \boldsymbol{A} + \boldsymbol{E}$, how much would it affect the eigenvalues $\boldsymbol{D}$?

Sensitivity (Cont'd)

- Suppose $\boldsymbol{A}$ is not defective. Suppose that $\boldsymbol{X^{-1}AX = D}$. For a certain perturbation in the input, i.e., $\boldsymbol{A \rightarrow A + E}$, how much would it affect the eigenvalues $\boldsymbol{D}$?
- Recall: in linear system or linear least square, the sensitivity depends on $\boldsymbol{A}$, would it be similar case here?
- Note that $\boldsymbol{X^{-1}(A + E)X = D + F}$
  - $\boldsymbol{A + E}$ and $\boldsymbol{D + F}$ have the same eigenvalues ?
  - $\boldsymbol{D + F}$ is not necessarily diagonal

$$X^{-1}(A+E)X = X^{-1}AE + X^{-1}EX$$
$$= D + X^{-1}EX$$
$$= F$$

Sensitivity (Cont'd)

- Suppose $\boldsymbol{v}$ is an eigen vector of the perturbed matrix $\boldsymbol{A}+\boldsymbol{E}$ with an eigenvalue μ, satisfying $(\boldsymbol{D}+\boldsymbol{F})\boldsymbol{v}=\mu\boldsymbol{v}$ it gives

$$\boldsymbol{F}\boldsymbol{v}=(\mu\boldsymbol{I}-\boldsymbol{D})\boldsymbol{v} \Rightarrow (\mu\boldsymbol{I}-\boldsymbol{D})^{-1}\boldsymbol{F}\boldsymbol{v}=\boldsymbol{v} \text{ (when is it invertible?)}$$

$$\Rightarrow \|\boldsymbol{v}\| \le \left\|(\mu\boldsymbol{I}-\boldsymbol{D})^{-1}\right\| \cdot \|\boldsymbol{F}\| \cdot \|\boldsymbol{v}\|$$

$$\Rightarrow \left\|(\mu\boldsymbol{I}-\boldsymbol{D})^{-1}\right\|^{-1} \le \|\boldsymbol{F}\|$$

- Let λ_k be the entry in $\boldsymbol{D}$ that is closest to μ, then

$$|\lambda_k-\mu| = \|(\mu\boldsymbol{I}-\boldsymbol{D})^{-1}\|^{-1} \le \|\boldsymbol{F}\| = \|\boldsymbol{X}^{-1}\boldsymbol{E}\boldsymbol{X}\| \le \|\boldsymbol{X}^{-1}\|\|\boldsymbol{E}\|\|\boldsymbol{X}\| = \kappa(\boldsymbol{X})\|\boldsymbol{E}\|$$

Sensitivity (Cont'd)

- Let λ_k be the entry in $\boldsymbol{D}$ that is closest to μ, then

$$|\lambda_k - \mu| \leq \kappa(\boldsymbol{X})\|\boldsymbol{E}\|$$

- The conditioning depends on the eigenvectors
  - The solutions are sensitive to input perturbations if the eigenvectors are nearly linearly dependent
  - If $\boldsymbol{X}$ is orthogonal (e.g., for symmetric matrix $\boldsymbol{A}$), then eigenvalues are always well-conditioned
  - This bound is in terms of all eigenvalues, so may overestimate for each individual eigenvalue

Rethinking Power Method

- All power iteration methods compute one eigenvalue at a time.
- What if I want multiple (or, all the) eigenvalues?
- One approach is via deflation (how to do it?), but it is not widely used in practice (why?)
- Can I make this process "faster" by obtaining several (or all of the) eigenvalues/eigenvectors simultaneously?

$A - \lambda_1 x_1 x_1^T - \lambda_2 x_2 x_2^T$

QR Method

Goal: get eigenvectors $x_1, x_2, \cdots, x_n$ and $\lambda_1, \lambda_2, \cdots, \lambda_n$ from A

- Given an $n \times n$ matrix $\boldsymbol{A}$, do the following
  - Set $\boldsymbol{X}_0 = \boldsymbol{A}$
  - For $k = 1, 2, \cdots$ perform QR factorization as follows

$$\boldsymbol{Q}_k \boldsymbol{R}_k = \boldsymbol{X}_{k-1}$$

  - Update $\boldsymbol{X}_k$ as

$$\boldsymbol{X}_k = \boldsymbol{R}_k \boldsymbol{Q}_k$$

  - Repeat until converge

Step-1: $Q_1 R_1 = X_0 = A$

$X_1 = R_1 Q_1$

Step2: $Q_2 R_2 = X_1$

$X_2 = R_2 Q_2$

⋮

Recall: QR Factorization

- For a rectangular matrix $\boldsymbol{A} \in R^{m \times n}$ with $m \geq n$. The QR factorization to this matrix can be written as

$$\boldsymbol{H}_n \cdots \boldsymbol{H}_1 \boldsymbol{A} = \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix}$$

- The product of successive Householder transformations $\boldsymbol{H}_n \cdots \boldsymbol{H}_1$ is itself an orthogonal matrix. Thus, if we take $\boldsymbol{Q}^T = \boldsymbol{H}_n \cdots \boldsymbol{H}_1$, or equivalently, $\boldsymbol{Q} = \boldsymbol{H}_1 \cdots \boldsymbol{H}_n$, then

$$\boldsymbol{A} = \boldsymbol{Q} \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix}$$

Illustrative Example

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

```
% Define your matrix
A = [2, -1, 0; -1, 2, -1; 0, -1, 2];

% Set convergence tolerance and maximum iterations
tolerance = 1e-8;
max_iterations = 1000;

% Initialize variables
n = size(A, 1);
Q = eye(n);
iterations = 0;

while iterations < max_iterations
    [Q, R] = qr(A);
    A = R * Q;
    iterations = iterations + 1;
    if norm(tril(A,-1), 'fro') < tolerance
        break;
    end

    fprintf('Intermediate evaluation of eigenvalues:\n')
    disp(diag(A));

    fprintf('\nIntermediate evaluation of eigenvectors:\n');
    disp(Q);

end
```

① ②

```
Intermediate evaluation of eigenvalues:
    2.8000
    2.3429
    0.8571


Intermediate evaluation of eigenvectors:
   -0.8944   -0.3586    0.2673
    0.4472   -0.7171    0.5345
         0    0.5976    0.8018

Intermediate evaluation of eigenvalues:
    3.1429
    2.2484
    0.6087


Intermediate evaluation of eigenvectors:
   -0.9661   -0.2467   -0.0761
    0.2582   -0.9231   -0.2849
         0   -0.2949    0.9555
```

③

```
Intermediate evaluation of eigenvalues:
    3.3084
    2.1039
    0.5876


Intermediate evaluation of eigenvectors:
   -0.9845   -0.1745    0.0155
    0.1752   -0.9807    0.0871
         0    0.0884    0.9961

Intermediate evaluation of eigenvalues:
    3.3761
    2.0380
    0.5859


Intermediate evaluation of eigenvectors:
   -0.9937   -0.1118   -0.0028
    0.1118   -0.9934   -0.0253
         0   -0.0255    0.9997
```

```
Intermediate evaluation of eigenvalues:
    3.4096
    2.0046
    0.5858


Intermediate evaluation of eigenvectors:
   -0.9992   -0.0401   -0.0001
    0.0401   -0.9992   -0.0022
         0   -0.0022    1.0000

Intermediate evaluation of eigenvalues:
    3.4126
    2.0016
    0.5858


Intermediate evaluation of eigenvectors:
   -0.9997   -0.0236    0.0000
    0.0236   -0.9997    0.0006
         0    0.0006    1.0000

Intermediate evaluation of eigenvalues:
    3.4137
    2.0005
    0.5858


Intermediate evaluation of eigenvectors:
   -0.9999   -0.0139   -0.0000
    0.0139   -0.9999   -0.0002
         0   -0.0002    1.0000

Intermediate evaluation of eigenvalues:
    3.4140
    2.0002
    0.5858


Intermediate evaluation of eigenvectors:
   -1.0000   -0.0081    0.0000
    0.0081   -1.0000    0.0001
         0    0.0001    1.0000
```

QR Method (Cont'd)

- Given an $n \times n$ matrix $\boldsymbol{A}$, do the following
  - Set $\boldsymbol{X}_0 = \boldsymbol{A}$
  - For $k = 1, 2, \cdots$ perform QR factorization as follows

$$\boldsymbol{Q}_k \boldsymbol{R}_k = \boldsymbol{X}_{k-1}$$

  - Update $\boldsymbol{X}_k$ as

$$\boldsymbol{X}_k = \boldsymbol{R}_k \boldsymbol{Q}_k = \boldsymbol{Q}_k^T \boldsymbol{Q}_k \boldsymbol{R}_k \boldsymbol{Q}_k = \boldsymbol{Q}_k^T (\boldsymbol{Q}_k \boldsymbol{R}_k) \boldsymbol{Q}_k = \boldsymbol{Q}_k^T \boldsymbol{X}_{k-1} \boldsymbol{Q}_k$$

$$= \boldsymbol{Q}_k^T \boldsymbol{Q}_{k-1}^T \boldsymbol{X}_{k-2} \boldsymbol{Q}_{k-1} \boldsymbol{Q}_k = \cdots = \boldsymbol{Q}_k^T \boldsymbol{Q}_{k-1}^T \cdots \boldsymbol{Q}_1^T \boldsymbol{A} \boldsymbol{Q}_1 \cdots \boldsymbol{Q}_{k-1} \boldsymbol{Q}_k$$

similar

  - Repeat until converge
- What is going on?
- What is the relationship between $\boldsymbol{X}_k$ and $\boldsymbol{X}_{k-1}$?

QR Method (Cont'd)

- What is the relationship between $\boldsymbol{X}_k$ and $\boldsymbol{X}_{k-1}$?

$$\boldsymbol{X}_k = \boldsymbol{R}_k\boldsymbol{Q}_k = \boldsymbol{Q}_k^T\boldsymbol{Q}_k\boldsymbol{R}_k\boldsymbol{Q}_k = \boldsymbol{Q}_k^T\boldsymbol{X}_{k-1}\boldsymbol{Q}_k$$

- $\boldsymbol{X}_k$ and $\boldsymbol{X}_{k-1}$ are similar to each other

- What is the relationship between $\boldsymbol{X}_k$ and $\boldsymbol{A}$?

$$\boldsymbol{X}_k = \boldsymbol{Q}_k^T\boldsymbol{X}_{k-1}\boldsymbol{Q}_k = \boldsymbol{Q}_k^T\boldsymbol{Q}_{k-1}^T\boldsymbol{X}_{k-2}\boldsymbol{Q}_{k-1}\boldsymbol{Q}_k = \boldsymbol{Q}_k^T\boldsymbol{Q}_{k-1}^T \dots \boldsymbol{Q}_1^T\boldsymbol{A}\boldsymbol{Q}_1 \dots \boldsymbol{Q}_{k-1}\boldsymbol{Q}_k$$

- $\boldsymbol{X}_k$ and $\boldsymbol{A}$ are similar to each other ($\boldsymbol{X}_k = \tilde{\boldsymbol{Q}}_k^T\boldsymbol{A}\tilde{\boldsymbol{Q}}_k$, where $\tilde{\boldsymbol{Q}}_k = \boldsymbol{Q}_1 \dots \boldsymbol{Q}_{k-1}\boldsymbol{Q}_k$)

QR Method (Cont'd)

- Theorem: Suppose the eigenvalues of matrix $\boldsymbol{A}$ satisfies $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0$. Let $\boldsymbol{Y}$ be a matrix such that the i-th row $\boldsymbol{y}_i^T$ of $\boldsymbol{Y}$ satisfies $\boldsymbol{y}_i^T \boldsymbol{A} = \lambda_i \boldsymbol{y}_i^T$. If $\boldsymbol{Y}$ has an LU factorization, then the entries under the diagonal of the matrix $\boldsymbol{X}_k = [x_{ij}^{(k)}]$ produced by the QR method tend to zero as $k \to \infty$. At the same time,

$$x_{ii}^{(k)} \to \lambda_i$$

for $i = 1, 2, \ldots, n$.

QR Iteration: Incorporating a Shift

- How can we accelerate the convergence of QR iteration by using shifts?
- Recall: any variant of power iteration has the convergence rate depending on the ratio of magnitudes of successive eigenvalues
- Similar to Rayleigh quotient iteration, introduce a shift operation to QR

Algorithm 4.8 QR Iteration with Shifts

$\boldsymbol{A}_0 = \boldsymbol{A}$
for $k = 1, 2, \ldots$
  Choose shift σ_k
  Compute QR factorization { normalize }
    $\boldsymbol{Q}_k \boldsymbol{R}_k = \boldsymbol{A}_{k-1} - \sigma_k \boldsymbol{I}$
  $\boldsymbol{A}_k = \boldsymbol{R}_k \boldsymbol{Q}_k + \sigma_k \boldsymbol{I}$ { generate next matrix }
end

QR Iteration: Incorporating a Shift (Cont'd)

Algorithm 4.8 QR Iteration with Shifts

$\boldsymbol{A}_0 = \boldsymbol{A}$
for $k = 1, 2, \ldots$
  Choose shift σ_k
  Compute QR factorization { normalize }
    $\boldsymbol{Q}_k \boldsymbol{R}_k = \boldsymbol{A}_{k-1} - \sigma_k \boldsymbol{I}$
  $\boldsymbol{A}_k = \boldsymbol{R}_k \boldsymbol{Q}_k + \sigma_k \boldsymbol{I}$ { generate next matrix }
end

- Why it works? – notice that $\boldsymbol{A}_k \sim \boldsymbol{A}_{k-1}$
  - Calculate $\boldsymbol{R}_k \boldsymbol{Q}_k = \boldsymbol{Q}_k^T (\boldsymbol{A}_{k-1} - \sigma_k \boldsymbol{I}) \boldsymbol{Q}_k$
  - Compute $\boldsymbol{A}_k = \boldsymbol{R}_k \boldsymbol{Q}_k + \sigma_k \boldsymbol{I} = \boldsymbol{Q}_k^T \boldsymbol{A}_{k-1} \boldsymbol{Q}_k - \boldsymbol{Q}_k^T \sigma_k \boldsymbol{I} \boldsymbol{Q}_k + \sigma_k \boldsymbol{I} = \boldsymbol{Q}_k^T \boldsymbol{A}_{k-1} \boldsymbol{Q}_k$
- Such a shifting operation does not change anything

QR Iteration: Incorporating a Shift (Cont'd)

Algorithm 4.8 QR Iteration with Shifts

$\boldsymbol{A}_0 = \boldsymbol{A}$
for $k = 1, 2, \ldots$
  Choose shift σ_k
  Compute QR factorization { normalize }
    $\boldsymbol{Q}_k \boldsymbol{R}_k = \boldsymbol{A}_{k-1} - \sigma_k \boldsymbol{I}$
  $\boldsymbol{A}_k = \boldsymbol{R}_k \boldsymbol{Q}_k + \sigma_k \boldsymbol{I}$ { generate next matrix }
end

- Why it works?
- QR iteration on $\boldsymbol{A}$ is also performing inverse iteration on $\boldsymbol{A}_k$ at each step
- Therefore, $\boldsymbol{A}_k - \sigma_k \boldsymbol{I}$ can accelerate using similar argument before
- Which σ_k to choose? The left most.

Example (2)

- Apply QR iteration to the following matrix

$$\boldsymbol{A} = \begin{bmatrix} 2.9766 & 0.3945 & 0.4198 & 1.1159 \\ 0.3945 & 2.7328 & -0.3097 & 0.1129 \\ 0.4198 & -0.3097 & 2.5675 & 0.6079 \\ 1.1159 & 0.1129 & 0.6079 & 1.7231 \end{bmatrix}$$

- This matrix has eigenvalues $\lambda_1 = 4, \lambda_2 = 3, \lambda_3 = 2, \lambda_4 = 1$, use $\sigma_1 = 1.7231$ as the shift for the first iteration

$$\boldsymbol{A}_1 = \begin{bmatrix} 3.8816 & -0.0178 & 0.2355 & 0.5065 \\ -0.0178 & 2.9528 & -0.2134 & -0.1602 \\ 0.2355 & -0.2134 & 2.0404 & -0.0951 \\ 0.5065 & -0.1602 & -0.0951 & 1.1253 \end{bmatrix}$$

Example (2)

- Next shift factor is $\sigma_2 = 1.1253$, giving the following

$$\boldsymbol{A}_2 = \begin{bmatrix} 3.9946 & -0.0606 & 0.0499 & 0.0233 \\ -0.0606 & 2.9964 & -0.0882 & -0.0103 \\ 0.0499 & -0.0882 & 2.0081 & -0.0252 \\ 0.0233 & -0.0103 & -0.0252 & 1.0009 \end{bmatrix}$$

- Finally, use $\sigma_3 = 1.0009$ and we have

$$\boldsymbol{A}_3 = \begin{bmatrix} 3.9980 & -0.0426 & 0.0165 & 0.0000 \\ -0.0426 & 3.0000 & -0.0433 & 0.0000 \\ 0.0165 & -0.0433 & 2.0020 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

Learning Objectives

- Variants of power methods
- Simultaneous iterations
- QR iteration
- QR iteration with a shift