Numerical Linear Algebra
Module 5: Eigenvalues

Dr. Zahra Lakdawala

December 2, 2020

Outline

Eigenvalue problems

Eigenvalue and Eigenvectors

- Eigenvalue problem of $m \times m$ matrix $\mathbf{A}$ is

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

  with eigenvalues λ and eigenvectors $\mathbf{x}$ (nonzero)
- The set of all the eigenvalues of $\mathbf{A}$ is the spectrum of $\mathbf{A}$
- Eigenvalue are generally used where a matrix is to be compounded iteratively
- Eigenvalues are useful for algorithmic and physical reasons
  - Algorithmically, eigenvalue analysis can reduce a coupled system to a collection of scalar problems
  - Physically, eigenvalue analysis can be used to study resonance of musical instruments and stability of physical systems

Eigenvalue Decomposition

- Eigenvalue decomposition of $\mathbf{A}$ is

$$\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1} \quad \text{or} \quad \mathbf{AX} = \mathbf{X}\Lambda$$

  with eigenvectors $\mathbf{x}$ i as columns of $\mathbf{X}$ and eigenvalues λ_i along diagonal of Λ. Alternatively,

$$\mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{x}_i$$

- Eigenvalue decomposition is change of basis to "eigenvector coordinates"

$$\mathbf{A}\mathbf{x} = \mathbf{b} \rightarrow (\mathbf{X}\mathbf{b}^{-1}) = \Lambda(\mathbf{X}^{-1}\mathbf{x})$$

- Note that eigenvalue decomposition may not exist
- Question: How does eigenvalue decomposition differ from SVD?

Geometric Multiplicity

- Eigenvectors corresponding to a single eigenvalue λ form an eigenspace $E_\lambda \subseteq \mathbb{C}^{m\times m}$
- Eigenspace is invariant in that $\mathbf{A}E_\lambda \subseteq E_\lambda$
- Dimension of E_λ is the maximum number of linearly independent eigenvectors that can be found
- Geometric multiplicity of λ is dimension of E_λ , i.e., $\dim(\text{null}(\mathbf{A}-\lambda\mathbf{I}))$

Algebraic Multiplicity

- The characteristic polynomial of A is degree m polynomial

$$p_{\mathbf{A}}(z) = det(z\mathbf{I} - \mathbf{A}) = (z - \lambda_1)(z - \lambda_2)\cdots(z - \lambda_m)$$

  which is monic in that coefficient of z^m is 1

- λ is eigenvalue of $\mathbf{A}$ iff $p_{\mathbf{A}}(\lambda) = 0$
  - If λ is eigenvalue, then by definition, $\lambda\mathbf{x} - \mathbf{A}\mathbf{x} = (\lambda\mathbf{I} - \mathbf{A})\mathbf{x} = 0$, so $(\lambda\mathbf{I} - \mathbf{A})$ is singular and its determinant is 0
- Algebraic multiplicity of λ is its multiplicity as a root of $p_{\mathbf{A}}$
- Any matrix $\mathbf{A}$ has m eigenvalues, counted with algebraic multiplicity
- Question: What are the eigenvalues of a triangular matrix?
- Question: How are geometric multiplicity and algebraic multiplicity related?

Similarity Transformations

- The map $\mathbf{A} \rightarrow \mathbf{Y}^{-1}\mathbf{AY}$ is a similarity transformation of $\mathbf{A}$ for any nonsingular $\mathbf{Y} \in \mathbb{C}^{m \times m}$
- $\mathbf{A}$ and $\mathbf{B}$ are similar if there is similarity transformation $\mathbf{B} = \mathbf{Y}^{-1}\mathbf{AY}$

Theorem

If $\mathbf{Y}$ is nonsingular, then $\mathbf{A}$ and $\mathbf{Y}^{-1}\mathbf{AY}$ have the same characteristic polynomials, eigenvalues, and algebraic and geometric multiplicities.

1. For characteristic polynomial:

$$det(z\mathbf{I} - \mathbf{Y}^{-1}\mathbf{AY}) = det(\mathbf{Y}^{-1}(z\mathbf{I} - \mathbf{A})\mathbf{Y}) = det(z\mathbf{I} - \mathbf{A})$$

   so algebraic multiplicities remain the same

2. If $\mathbf{x} \in \mathbf{E}_\lambda$ for $\mathbf{A}$, then $\mathbf{Y}^{-1}\mathbf{x}$ is in eigenspace of $\mathbf{Y}^{-1}\mathbf{AY}$ corresponding to λ, and vice versa, so geometric multiplicities remain the same

Algebraic Multiplicity $\geq$ Geometric Multiplicity

- Let n be be geometric multiplicity of λ for $\mathbf{A}$. Let $\hat{\mathbf{V}} \in \mathbb{C}^{m \times n}$ constitute of orthonormal basis of the $\mathbf{E}_\lambda$
- Extend $\hat{\mathbf{V}}$ to unitary $\mathbf{V} = [\hat{\mathbf{V}}, \tilde{\mathbf{V}}] \in \mathbb{C}^{m \times m}$ and form

$$\mathbf{B} = \mathbf{V}^* \mathbf{A} \mathbf{V} = \begin{bmatrix} \hat{\mathbf{V}}^* \mathbf{A} \hat{\mathbf{V}} & \hat{\mathbf{V}}^* \mathbf{A} \tilde{\mathbf{V}} \\ \tilde{\mathbf{V}}^* \mathbf{A} \hat{\mathbf{V}} & \tilde{\mathbf{V}}^* \mathbf{A} \tilde{\mathbf{V}} \end{bmatrix} = \begin{bmatrix} \lambda \mathbf{I} & \mathbf{C} \\ 0 & \mathbf{D} \end{bmatrix}$$

- $\det(z\mathbf{I} - \mathbf{B}) = \det(z\mathbf{I} - \lambda \mathbf{I}\ ) \det(z\mathbf{I} - \lambda \mathbf{D}) = (z - \lambda)^n \det(z\mathbf{I} - \lambda \mathbf{D})$, so the algebraic multiplicity of λ as an eigenvalue of $\mathbf{B}$ is $\geq n$
- $\mathbf{A}$ and $\mathbf{B}$ are similar, so the algebraic multiplicity of λ as an eigenvalue of $\mathbf{A}$ is at least $\geq n$
- Examples:

$$\mathbf{A} = \begin{bmatrix} 2 & & \\ & 2 & \\ & & 2 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 2 & 1 & \\ & 2 & 1 \\ & & 2 \end{bmatrix}$$

  Their characteristic polynomial is $(z-2)^3$, so algebraic multiplicity of $\lambda = 2$ is 3. But geometric multiplicity of $\mathbf{A}$ is 3 and that of $\mathbf{B}$ is 1.

Defective and Diagonalizable Matrices

- An eigenvalue of a matrix is defective if its algebraic multiplicity $>$ its geometric multiplicity
- A matrix is defective if it has a defective eigenvalue. Otherwise, it is called nondefective.

Theorem

An $m \times m$ matrix $\mathbf{A}$ is nondefective iff it has an eigenvalue decomposition $\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1}$.

Defective and Diagonalizable Matrices

- An eigenvalue of a matrix is defective if its algebraic multiplicity $>$ its geometric multiplicity
- A matrix is defective if it has a defective eigenvalue. Otherwise, it is called nondefective.

Theorem

An $m \times m$ matrix $\mathbf{A}$ is nondefective iff it has an eigenvalue decomposition $\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1}$.

- ($\Leftarrow$) Λ is nondefective, and $\mathbf{A}$ is similar to Λ, so $\mathbf{A}$ is nondefective.
- ($\Rightarrow$) A nondefective matrix has m linearly independent eigenvectors. Take them as columns of $\mathbf{X}$ to obtain $\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1}$.
- Nondefective matrices are therefore also said to be diagonalizable.

Determinant and Trace

- Determinant of $\mathbf{A}$ is $\det(\mathbf{A}) = \Pi_{j=1}^{m} \lambda_j$, because

$$det(\mathbf{A}) = (-1)^m det(-\mathbf{A}) = (-1)^m p_{\mathbf{A}}(0) = \Pi_{j=1}^{m} \lambda_j$$

- Trace of $\mathbf{A}$ is $\text{tr}(\mathbf{A}) = \sum_{j=1}^{m} \lambda_j$, since

$$p_{\mathbf{A}}(z) = det(z\mathbf{I} - \mathbf{A}) = z^m - \sum_{j=1}^{m} a_{jj} z^{m-1} + O(z^{m-2})$$

$$p_{\mathbf{A}}(z) = \Pi_{j=1}^{m}(z - \lambda_j) = z^m - \sum_{j=1}^{m} \lambda_j z^{m-1} + O(z^{m-2})$$

- Question: Are these results valid for defective or nondefective matrices?

Unitary Diagonalization

- A matrix $\mathbf{A}$ is unitarily diagonalizable if $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^*$ for a unitary matrix $\mathbf{Q}$
- A hermitian matrix is unitarily diagonalizable, with real eigenvalues
- A matrix $\mathbf{A}$ is normal if $\mathbf{A}^*\mathbf{A} = \mathbf{A}\mathbf{A}^*$
  - Examples of normal matrices include hermitian matrices, skew hermitian matrices
  - hermitian $\Leftrightarrow$ matrix is normal and all eigenvalues are real
  - skew hermitian $\Leftrightarrow$ matrix is normal and all eigenvalues are imaginary
  - If $\mathbf{A}$ is both triangular and normal, then $\mathbf{A}$ is diagonal
- Unitarily diagonalizable $\Leftrightarrow$ normal
  - "$\Rightarrow$" is easy. Prove "$\Leftarrow$" by induction using Schur factorization next

Schur Factorization

- Schur factorization is $\mathbf{A} = \mathbf{QTQ}^*$, where $\mathbf{Q}$ is unitary and $\mathbf{T}$ is upper triangular

Theorem

Every square matrix $\mathbf{A}$ has a Schur factorization.

Proof by induction on dimension of $\mathbf{A}$. Case $m = 1$ is trivial. For $m \geq 2$, let $\mathbf{x}$ be any unit eigenvector of $\mathbf{A}$, with corresponding eigenvalue λ. Let $\mathbf{U}$ be unitary matrix with $\mathbf{x}$ as first column. Then

$$\mathbf{U}^*\mathbf{AU} = \begin{bmatrix} \lambda & \mathbf{w}^* \\ 0 & \mathbf{C} \end{bmatrix}$$

By induction hypothesis, there is a Schur factorization $\tilde{\mathbf{T}} = \mathbf{V}^*\mathbf{CV}$. Let

$$\mathbf{Q} = \mathbf{U}\begin{bmatrix} 1 & 0 \\ 0 & \mathbf{V} \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} \lambda & \mathbf{w}^*\mathbf{V} \\ 0 & \tilde{\mathbf{T}} \end{bmatrix}$$

and then $\mathbf{A} = \mathbf{QTQ}^*$

Eigenvalue Revealing Factorization

- Eigenvalue-revealing factorization of square matrix $\mathbf{A}$
  - Diagonalization $\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1}$ (nondefective $\mathbf{A}$)
  - Unitary Diagonalization $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^*$ (normal $\mathbf{A}$)
  - Unitary triangularization (Schur factorization) $\mathbf{A} = \mathbf{Q}\mathbf{T}\mathbf{Q}^*$ (any $\mathbf{A}$)
  - Jordan normal form $\mathbf{A} = \mathbf{X}\mathbf{J}\mathbf{X}$, where $\mathbf{J}$ block diagonal with

$$\mathbf{J}_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix}$$

- In general, Schur factorization is used, because
  - Unitary matrices are involved, so algorithm tends to be more stable
  - If $\mathbf{A}$ is normal, then Schur form is diagonal

Eigenvalue algorithms

Eigenvalue Revealing Factorization

- Eigenvalue-revealing factorization of square matrix $\mathbf{A}$
  - Diagonalization $\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1}$ (nondefective $\mathbf{A}$)
  - Unitary Diagonalization $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^*$ (normal $\mathbf{A}$)
  - Unitary triangularization (Schur factorization) $\mathbf{A} = \mathbf{Q}\mathbf{T}\mathbf{Q}^*$ (any $\mathbf{A}$)
  - Jordan normal form $\mathbf{A} = \mathbf{X}\mathbf{J}\mathbf{X}$, where $\mathbf{J}$ block diagonal with

$$\mathbf{J}_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix}$$

- In general, Schur factorization is used, because
  - Unitary matrices are involved, so algorithm tends to be more stable
  - If $\mathbf{A}$ is normal, then Schur form is diagonal

"Obvious" Algorithms

- Most obvious method is to find roots of characteristic polynomial $p_{\mathbf{A}}(\lambda)$, but it is very ill-conditioned
- Another idea is power iteration, using fact that

$$\frac{\mathbf{x}}{\|\mathbf{x}\|}, \frac{\mathbf{Ax}}{\|\mathbf{Ax}\|}, \frac{\mathbf{A}^2\mathbf{x}}{\|\mathbf{A}^2\mathbf{x}\|}, \frac{\mathbf{A}^3\mathbf{x}}{\|\mathbf{A}^3\mathbf{x}\|}, \ldots$$

converge to an eigenvector corresponding to the largest eigenvalue of $\mathbf{A}$ in absolute value, but it may converge very slowly

"Obvious" Algorithms

- Most obvious method is to find roots of characteristic polynomial $p_{\mathbf{A}}(\lambda)$, but it is very ill-conditioned
- Another idea is power iteration, using fact that

$$\frac{\mathbf{x}}{\|\mathbf{x}\|}, \frac{\mathbf{A}\mathbf{x}}{\|\mathbf{A}\mathbf{x}\|}, \frac{\mathbf{A}^2\mathbf{x}}{\|\mathbf{A}^2\mathbf{x}\|}, \frac{\mathbf{A}^3\mathbf{x}}{\|\mathbf{A}^3\mathbf{x}\|}, \ldots$$

  converge to an eigenvector corresponding to the largest eigenvalue of $\mathbf{A}$ in absolute value, but it may converge very slowly
- Instead, compute a eigenvalue-revealing factorization, such as Schur factorization

$$\mathbf{A} = \mathbf{Q}\mathbf{T}\mathbf{Q}^*$$

  by introducing zeros, using algorithms similar to QR factorization

A Fundamental Difficulty

- However, eigenvalue-revealing factorization cannot be done in finite number of steps:

  Any eigenvalue solver must be iterative

- To see this, consider a general polynomial of degree m

$$p(z) = z^m + a_{m-1}z^{m-1} + \cdots + a_1 z + a_0$$

  There is no closed-form expression for the roots of p: (Abel, 1842) In general, the roots of polynomial equations higher than fourth degree cannot be written in terms of a finite number of operations

A Fundamental Difficulty Cont'd

- However, the roots of $p_{\mathbf{A}}$ are the eigenvalues of the companion matrix

$$\mathbf{A} = \begin{bmatrix} 0 & & & & -a_0 \\ 1 & 0 & & & -a_1 \\ & 1 & \ddots & & \vdots \\ & & \ddots & 0 & -a_{m-2} \\ & & & 1 & -a_{m-1} \end{bmatrix}$$

- Therefore, in general, we cannot find the eigenvalues of a matrix in a finite number of steps
- In practice, however, there are algorithms that converge to desired precision in a few iterations

Schur Factorization and Diagonalization

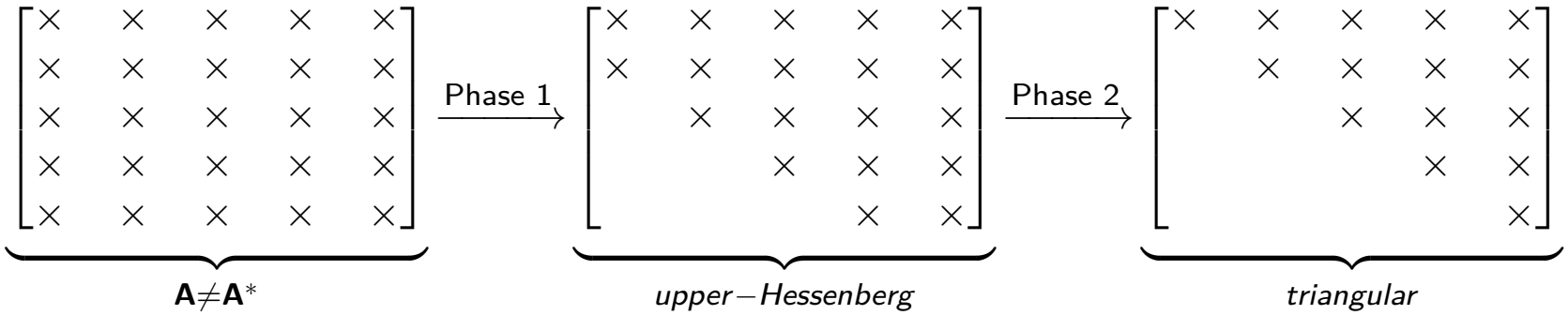
- Most eigenvalue algorithms compute Schur factorization $\mathbf{A} = \mathbf{QTQ}^*$ by transforming $\mathbf{A}$ with similarity transformations

$$\underbrace{\mathbf{Q}_j^* \cdots \mathbf{Q}_2^* \mathbf{Q}_1^*}_{\mathbf{Q}^*} \mathbf{A} \underbrace{\mathbf{Q}_1 \mathbf{Q}_2 \cdots \mathbf{Q}_j}_{\mathbf{Q}},$$

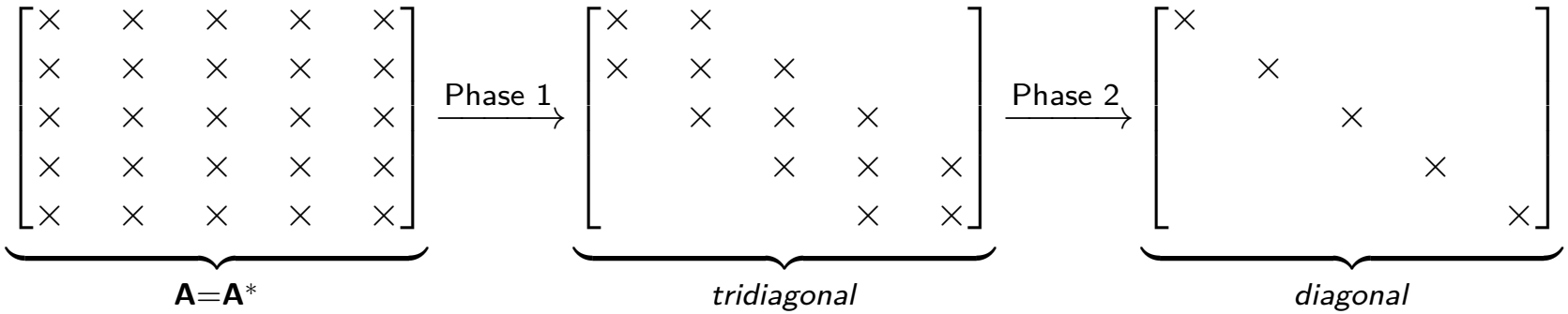
  where $\mathbf{Q}_i$ are unitary matrices, which converge to $\mathbf{T}$ as $j \to \infty$
- Note: Real matrices might need complex Schur forms and eigenvalues
- Question: For hermitian $\mathbf{A}$, what matrix will the sequence converge to?

Two Phases of Eigenvalue Computations

- General **A**: First convert to upper-Hessenberg form, then to upper triangular



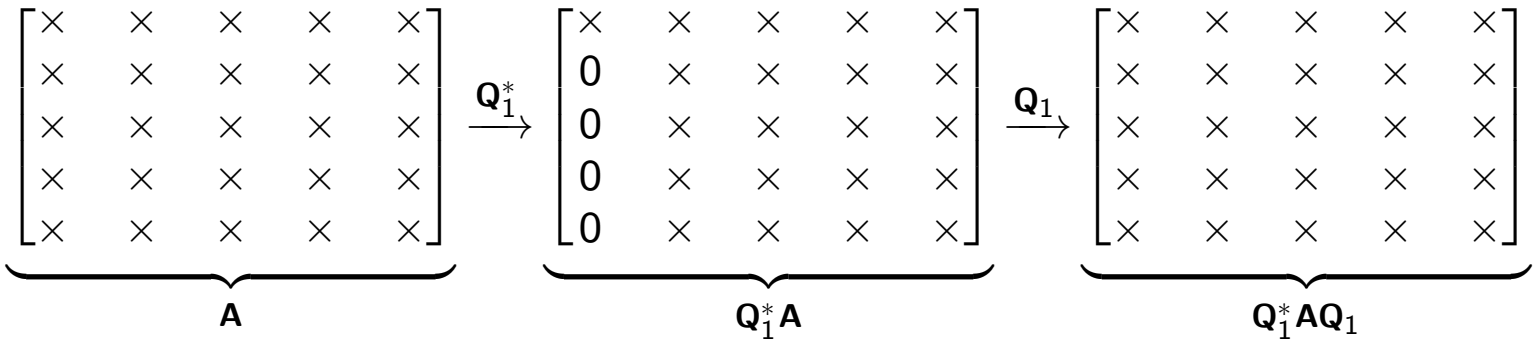
- Hermitian **A**: First convert to tridiagonal form, then to diagonal



- In general, phase 1 is direct and requires $O(m^3)$ flops, and phase 2 is iterative and requires $O(m)$ iterations, and $O(m^3)$ flops for non-Hermitian matrices and $O(m^2)$ flops for Hermitian matrices

Introducing Zeros by Similarity Transformations

- First attempt: Compute Schur factorization $\mathbf{A} = \mathbf{QTQ}^*$ by applying Householder reflectors from both left and right



- Unfortunately, the right multiplication destroys the zeros introduced by $\mathbf{Q}_1^*$ (has the effect of replacing each column by linear combination of all columns)
- This would not work because of Abel's theorem - no finite process can reveal the eigenvalues of $\mathbf{A}$
- However, the subdiagonal entries typically decrease in magnitude (even if it doesn't make it zero)

Hessenberg Form

The Hessenberg Form

- Second attempt: try to compute upper Hessenberg matrix $\mathbf{H}$ similar to $\mathbf{A}$:

$$\underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}}_{\mathbf{A}} \xrightarrow{\mathbf{Q}_1^*} \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}}_{\mathbf{Q}_1^*\mathbf{A}} \xrightarrow{\mathbf{Q}_1} \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}}_{\mathbf{Q}_1^*\mathbf{A}\mathbf{Q}_1}$$

- The zeros introduced by $\mathbf{Q}_1^*\mathbf{A}$ were not destroyed this time!
- Continue with remaining columns would result in Hessenberg form:

$$\xrightarrow{\mathbf{Q}_2^*} \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & 0 & \times & \times & \times \\ & 0 & \times & \times & \times \end{bmatrix}}_{\mathbf{Q}_2^*\mathbf{Q}_1^*\mathbf{A}\mathbf{Q}_1} \xrightarrow{\mathbf{Q}_2} \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{bmatrix}}_{\mathbf{Q}_2^*\mathbf{Q}_1^*\mathbf{A}\mathbf{Q}_1\mathbf{Q}_2}$$

The Hessenberg Form

- After $m-2$ steps, we obtain the Hessenberg form:

$$\underbrace{\mathbf{Q}_{m-2}^{*}\cdots\mathbf{Q}_{2}^{*}\mathbf{Q}_{1}^{*}}_{\mathbf{Q}^{*}}\mathbf{A}\underbrace{\mathbf{Q}_{1}\mathbf{Q}_{2}\cdots\mathbf{Q}_{m-2}}_{\mathbf{Q}}=\mathbf{H}=\begin{bmatrix}\times & \times & \times & \times & \times\\ \times & \times & \times & \times & \times\\ & \times & \times & \times & \times\\ & & \times & \times & \times\\ & & & \times & \times\end{bmatrix}$$

- For hermitian matrix $\mathbf{A}$, $\mathbf{H}$ is hermitian and hence is tridiagonal

Householder Reduction to Hessenberg

Algorithm: Householder Reduction to Hessenberg Form

for $k = 1$ to $m-2$

$\mathbf{x} = \mathbf{A}_{k+1:m,k}$

$\mathbf{v}_k = sign(\mathbf{x}_1)\|x\|_2 \mathbf{e}_1 + \mathbf{x}$

$\mathbf{v}_k = \mathbf{v}_k / \|\mathbf{v}_k\|_2$

$\mathbf{A}_{k+1:m,k:m} = \mathbf{A}_{k+1:m,k:m} - 2\mathbf{v}_k(\mathbf{v}_k^* \mathbf{A}_{k+1:m,k:m})$

$\mathbf{A}_{1:m,k+1:m} = \mathbf{A}_{1:m,k+1:m} - 2(\mathbf{A}_{1:m,k+1:m}\mathbf{v}_k)\mathbf{v}_k^*$

- Compare it to QR Factorization with Household Reflectors (Algorithm 10.1)
- Note: $\mathbf{Q}$ is never formed explicitly (as in Algorithm 10.1)
- Operation count

$$\sim \sum_{k=1}^{m-2} 4(m-k)^2 + 4m(m-k) \sim \frac{4m^3}{3} + 4m^3 - \frac{4m^3}{2} = \frac{10m^3}{3}$$

Reduction to Tridiagonal Form

- If **A** is hermitian, then

$$\underbrace{\mathbf{Q}_{m-2}^{*}\cdots\mathbf{Q}_{2}^{*}\mathbf{Q}_{1}^{*}}_{\mathbf{Q}^{*}}\mathbf{A}\underbrace{\mathbf{Q}_{1}\mathbf{Q}_{2}\cdots\mathbf{Q}_{m-2}}_{\mathbf{Q}}=\mathbf{H}=\begin{bmatrix}\times & \times & & & \\ \times & \times & \times & & \\ & \ddots & \ddots & \ddots & \\ & & \times & \times & \times \\ & & & \times & \times\end{bmatrix}$$

- For Hermitian A, operation count would be same as Householder QR: $\frac{4m^3}{3}$
  - First, taking advantage of sparsity, cost of applying right reflectors is also $4(m-k)^2$ instead of $4m(m-k)$, so cost is

$$\sim\sum_{k=1}^{m-2}8(m-k)^2\sim\frac{8m^3}{3}$$

  - Second, taking advantage of symmetry, cost is reduced by 50% to $\frac{4m^3}{3}$

Stability of Hessenberg Reduction

Theorem

Householder reduction to Hessenberg form is backward stable, in that

$$\tilde{\mathbf{Q}}\tilde{\mathbf{H}}\tilde{\mathbf{Q}}^* = \mathbf{A} + \delta\mathbf{A}, \qquad \frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|} = O(\varepsilon_{machine})$$

for some $\delta\mathbf{A} \in \mathbb{C}^{m\times m}$

Note: Similar to Householder QR, $\tilde{\mathbf{Q}}$ is exactly unitary based on some reflection vectors $\tilde{\mathbf{v}}_k$

Rayleigh Quotient, Inverse Iteration

Solving Eigenvalue Problems

- All eigenvalue solvers must be iterative
- Iterative algorithms have multiple facets:
  1. Basic idea behind the algorithms
  2. Convergence and techniques to speed-up convergence
  3. Efficiency of implementation
  4. Termination criteria
- We will focus on first two aspects

Simplification: Real Symmetric Matrices

- We will consider eigenvalue problems for real symmetric matrices, i.e. $\mathbf{A} = \mathbf{A}^{\mathbf{T}} \in \mathbb{R}^{m \times m}$, and $\mathbf{Ax} = \lambda \mathbf{x}$ for $\mathbf{x} \in \mathbb{R}^m$
- $\mathbf{A}$ has real eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_m$ and orthonormal eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \cdots \mathbf{q}_m$, where $\|\mathbf{q}_j\| = 1$
- Eigenvalues are often also ordered in a particular way (e.g., ordered from large to small in magnitude)
- In addition, we focus on symmetric tridiagonal form
  - Why? Because phase 1 of two-phase algorithm reduces matrix into tridiagonal form

Rayleigh Quotient

- The Rayleigh quotient of $\mathbf{x} \in \mathbb{R}^m$ is the scalar

$$r(\mathbf{x}) = \frac{\mathbf{x}^T \mathbf{A}\mathbf{x}}{\mathbf{x}^T \mathbf{x}}$$

- For an eigenvector $\mathbf{x}$, its Rayleigh quotient is $r(\mathbf{x}) = \mathbf{x}^T \lambda \mathbf{x} / \mathbf{x}^T \mathbf{x} = \lambda$, the corresponding eigenvalue of $\mathbf{x}$
- For general $\mathbf{x}$, $r(\mathbf{x}) = \alpha$ that minimizes $\|\mathbf{A}\mathbf{x} - \alpha \mathbf{x}\|_2$.
- $\mathbf{x}$ is eigenvector of $\mathbf{A} \leftrightarrow \nabla r(\mathbf{x}) = \frac{2}{\mathbf{x}^T \mathbf{x}}(\mathbf{A}\mathbf{x} - r(\mathbf{x})\mathbf{x}) = 0$ with $x \neq 0$
- $r(\mathbf{x})$ is smooth and $\nabla r(\mathbf{q}_j) = 0$ for any j, and therefore is quadratically accurate:

$$r(\mathbf{x}) - r(\mathbf{q}_J) = O(\|x - q_J\|^2) \text{ as } x \to \mathbf{q}_J \text{ for some } J$$

Power Iteration

- Simple power iteration for largest eigenvalue

Algorithm: Power Iteration

$v^{(0)} =$ some unit-length vector
for $k = 1, 2, \cdots$
$\mathbf{w} = \mathbf{A}\mathbf{v}^{(k-1)}$
$\mathbf{v}^{(k)} = \mathbf{w} / \|\mathbf{w}\|$
$\lambda^{(k)} = r(\mathbf{v}^{(k)}) = (\mathbf{v}^{(k)})^T \mathbf{A}\mathbf{v}^{(k)}$

- Termination condition is omitted for simplicity

Convergence of Power Iteration

- Expand initial $v^{(0)}$ in orthonormal eigenvectors $\mathbf{q}_i$, and apply $\mathbf{A}^k$:

$$\begin{aligned} v^{(0)} &= a_1\mathbf{q}_1 + a_2\mathbf{q}_2 + \cdots + a_m\mathbf{q}_m \\ v^{(k)} &= c_k\mathbf{A}^k\mathbf{v}^{(0)} \\ &= c_k(a_1\lambda_1^k\mathbf{q}_1 + a_2\lambda_2^k\mathbf{q}_2 + \cdots + a_m\lambda_m^k q_m) \\ &= c_k\lambda_1^k(a_1\mathbf{q}_1 + a_2(\lambda_2/\lambda_1)^k\mathbf{q}_2 + \cdots + a_m(\lambda_m/\lambda_1)^k q_m) \end{aligned}$$

Theorem

If $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_m| \geq 0$ and $\mathbf{q}_1^T\mathbf{v}^{(0)} \neq 0$, this gives

$$\left\|\mathbf{v}^{(k)} - (\pm\mathbf{q}_1)\right\| = O(|\lambda_2/\lambda_1|^k), \quad |\lambda^{(k)} - \lambda_1| = O(|\lambda_2/\lambda_1|^{2k})$$

as $k \to \infty$ where $\pm$ sign is chosen to be sign of $\mathbf{q}^T\mathbf{v}^{(k)}$

- It finds the largest eigenvalue (unless eigenvector is orthogonal to $\mathbf{v}^{(0)}$)
- Error reduces by only a constant factor ($\approx |\lambda_2/\lambda_1|$) each step, and very slowly especially when $|\lambda_2| \approx |\lambda_1|$

Limitations of the Power Iteration

- It can only find the eigenvector corresponding to the largest eigenvalue
- Convergence is linear, reducing the error only by a constant $\approx |\frac{\lambda_2}{\lambda_1}|$ at each iteration
- Quality of this algorithm depends on having a largest eigenvalue that is significantly larger than others
- Limited use, however powerful concept!!!

Inverse Iteration

- Apply power iteration on $(\mathbf{A}-\mu\mathbf{I})^{-1}$, with eigenvalues $\{(\lambda_j-\mu)-1\}$
- If $\mu \approx \lambda_J$ for some J, then $(\lambda_J-\mu)-1$ may be far larger than $(\lambda_j-\mu)^{-1}, j \neq J$, so power iteration may converge rapidly

Algorithm: Inverse Iteration

$v^{(0)}$ = some unit-length vector
for $k = 1, 2, \cdots$
  Solve $(\mathbf{A}-\mu\mathbf{I})\mathbf{w} = \mathbf{v}^{(k-1)}$ for $\mathbf{w}$
  $\mathbf{v}^{(k)} = \mathbf{w}/\|\mathbf{w}\|$
  $\lambda^{(k)} = r(\mathbf{v}^{(k)}) = (\mathbf{v}^{(k)})^T\mathbf{A}\mathbf{v}^{(k)}$

Convergence of Inverse Iteration

- Linear convergence (similar to the power iteration)
- Unlike power iteration, we can chose the eigenvector that will be found by suppling an estimate of μ of the corresponding eigenvalue.
- We can control the rate of linear convergence (chosing μ to be closer to the eigenvalue of **A**

Theorem

Suppose λ_J is the closest eigenvalue to μ and λ_K is the second closest, such that $|\mu - \lambda_J| < |\mu - \lambda_K| \leq |\mu - \lambda_j|$ for each $j \neq J$. The iterates converges to eigenvector $\mathbf{q}_J$ with

$$\left\|\mathbf{v}^{(k)} - (\pm \mathbf{q}_J)\right\| = O(|\frac{\mu - \lambda_J}{\mu - \lambda_K}|^k), |\lambda^{(k)} - \lambda_J| = O(|\frac{\mu - \lambda_J}{\mu - \lambda_K}|^{2k})$$

as $k \to \infty$.

- Standard method for determining eigenvector given eigenvalue (minus the Rayleigh quotient)

Rayleigh Quotient Iteration

- Parameter μ is constant in inverse iteration, but convergence is better for μ close to the eigenvalue
- Improvement: At each iteration, set μ to last computed Rayleigh quotient

Algorithm: Rayleigh Quotient Iteration

$\mathbf{v}^{(0)} =$ some unit-length vector
$\lambda^{(0)} = r(\mathbf{v}^{(0)}) = (\mathbf{v}^{(0)})^T \mathbf{A}\mathbf{v}^{(0)}$
for $k = 1, 2, \cdots$
  Solve $(\mathbf{A} - \lambda^{(k-1)}\mathbf{I})\mathbf{w} = \mathbf{v}^{(k-1)}$ for $\mathbf{w}$
  $\mathbf{v}^{(k)} = \mathbf{w}/\|w\|$
  $\lambda^{(k)} = r(\mathbf{v}^{(k)}) = (\mathbf{v}^{(k)})^T \mathbf{A}\mathbf{v}^{(k)}$

- Cost per iteration is linear for tridiagonal matrix

Convergence of Rayleigh Quotient Iteration

- Spectacular: Cubic convergence in Rayleigh quotient iteration

Theorem

Rayleigh Quotient Iteration converges to sn eigenvalue/eigenvector pair for all except a set of zero starting vectors. When it converges, the convergence is ultimately cubic in the sense that if λ_J is an eigenvalue of $\mathbf{A}$ and $\mathbf{v}^{(0)}$ is close to $\mathbf{q}_J$, then

$$\left\| v^{(k+1)} - (\pm \mathbf{q}_J) \right\| = O(\left\| \mathbf{v}^{(k)} - (\pm \mathbf{q}_J) \right\|^3)$$

and

$$|\lambda^{(k+1)} - \lambda_J| = O(|\lambda^{(k)} - \lambda_J|^3)$$

as $k \to \infty$.

- In other words, each iteration triples number of digits of accuracy
- Rayleigh quotient is great in finding largest (or smallest) eigenvalue and its corresponding eigenvector. What if we want to find all eigenvalues?

Operation Counts

In Rayleigh quotient iteration:

- if $\mathbf{A} \in \mathbb{R}^{m \times m}$ is full matrix, then solving $(\mathbf{A} - \mu \mathbf{I})\mathbf{w} = \mathbf{v}^{(k-1)}$ may take $O(m^3)$ flops per step
- if $\mathbf{A} \in \mathbb{R}^{m \times m}$ is upper Hessenberg, then each step takes $O(m^2)$ flops
- if $\mathbf{A} \in \mathbb{R}^{m \times m}$ is tridiagonal, then each step takes $O(m)$ flops

QR

QR Algorithm

- Most basic version of QR algorithm is remarkably simple:

> Algorithm: "Pure" QR Algorithm
>
> $\mathbf{A}^{(0)} = \mathbf{A}$
> for $k = 1, 2, \cdots$
> $\mathbf{Q}^{(k)}\mathbf{R}^{(k)} = \mathbf{A}^{(k-1)}$
> $\mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)}$

- With some suitable assumptions, $\mathbf{A}^{(k)}$ converge to Schur form of $\mathbf{A}$ (diagonal if $\mathbf{A}$ is symmetric)
- Similarity transformation of $\mathbf{A}$:

$$\mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)} = (\mathbf{Q}^{(k)})^T\mathbf{A}^{(k-1)}\mathbf{Q}^{(k)}$$

- But why does it work?

Unnormalized Simultaneous Iteration

- To understand QR algorithm, first consider simple algorithm
- Simultaneous iteration is power iteration applied to several vectors
- Start with linearly independent $\mathbf{v}_1^{(0)}, \cdots, \mathbf{v}_n^{(0)}$
- We know from power iteration that $\mathbf{A}^k\mathbf{v}_1$ converge to $\mathbf{q}_1$
- With some assumptions, the space $\langle \mathbf{A}^k\mathbf{v}_1^{(0)}, \cdots, \mathbf{A}^k\mathbf{v}_n^{(0)} \rangle$ should converge to $\langle \mathbf{q}_1, \cdots, \mathbf{q}_n \rangle$
- Notation: Define initial matrix $\mathbf{V}^{(0)}$ and matrix $\mathbf{V}^{(k)}$ at step k:

$$\mathbf{V}^{(0)} = [\mathbf{v}_1^{(0)} | \cdots | \mathbf{v}_n^{(0)}], \quad \mathbf{V}^{(k)} = \mathbf{A}^k\mathbf{V}^{(0)} = [\mathbf{v}_1^{(k)} | \cdots \mathbf{v}_n^{(k)}]$$

Unnormalized Simultaneous Iteration

- Define orthogonal basis for column space of $\mathbf{V}^{(k)}$ by reduced QR factorization $\hat{\mathbf{Q}}^{(k)}\hat{\mathbf{R}}^{(k)} = \mathbf{V}^{(k)}$
- We assume that
  1. leading $n+1$ eigenvalues are distinct, and
  2. all leading principal submatrices of $\hat{\mathbf{Q}}^T\mathbf{V}^{(0)}$ are nonsingular where $\hat{\mathbf{Q}} = [\mathbf{q}_1|\cdots|\mathbf{q}_n]$
- We then have columns of $\hat{\mathbf{Q}}^{(k)}$ converge to eigenvectors of $\mathbf{A}$:

$$\left\|\mathbf{q}_j^{(k)} - (\pm\mathbf{q}_j)\right\| = O(C^k),$$

  where $c = max_{1\leq k\leq n}|\lambda_{k+1}|/|\lambda_k|$
- Proof idea: Show that subspace of any leading j columns of $\mathbf{V}^{(k)} = \mathbf{A}^k\mathbf{V}^{(0)}$ converges to subspace of first j eigenvectors of $\mathbf{A}$, so does the subspace of any leading j columns of $\hat{\mathbf{Q}}^{(k)}$.

The Idea

- We know that other eigenvectors are orthogonal to the dominant one
- so we can use the power method, and force that the second vector is orthogonal to the first one
- this way we guarantee that they will converge to two different eigenvectors
- we can do this for many vectors, not just two
- this is called "Simaltaneous Iteration"

Simultaneous/Orthogonal Iteration

- Matrices $\mathbf{V}^{(k)} = \mathbf{A}^k \mathbf{V}^{(0)}$ are highly ill-conditioned
- Orthonormalize at each step rather than at the end

Algorithm: Simultaneous Iteration

Pick $\hat{\mathbf{Q}}^{(0)} \in \mathbb{R}^{m \times n}$
for $k = 1, 2, \cdots$
$\quad \mathbf{Z}^{(k)} = \mathbf{A}\hat{\mathbf{Q}}^{(k-1)}$;
$\quad \hat{\mathbf{Q}}^{(k)}\hat{\mathbf{R}}^{(k)} = \mathbf{Z}^{(k)}$

- Column spaces of $\hat{\mathbf{Q}}^{(k)}$ and $\mathbf{Z}^{(k)}$ are both equal to column space of $\mathbf{A}^k\hat{\mathbf{Q}}^{(0)}$, therefore same convergence as before

Simultaneous Iteration $\Leftrightarrow$ QR Algorithm

Simultaneous Iteration

Pick $\hat{\mathbf{Q}}^{(0)} \in \mathbb{R}^{m \times n}$

for $k = 1, 2, \cdots$

$\mathbf{Z} = \mathbf{A}\hat{\mathbf{Q}}^{(k-1)}$

$\hat{\mathbf{Q}}^{(k)}\hat{\mathbf{R}}^{(k)} = \mathbf{Z}$

"Pure" QR Algorithm

$\mathbf{A}^{(0)} = \mathbf{A}$

for $k = 1, 2, \cdots$

$\mathbf{Q}^{(k)}\mathbf{R}^{(k)} = \mathbf{A}^{(k-1)}$

$\mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)}$

- QR algorithm is equivalent to simultaneous iteration with $\hat{\mathbf{Q}}^{(0)} = \mathbf{I}$
- Since the matrices are now square, get rid of the hats. Replace $\hat{\mathbf{R}}^{(k)}$ by $\mathbf{R}^{(k)}$ and $\hat{\mathbf{Q}}^{(k)}$ by $\underline{\mathbf{Q}}^{(k)}$ (underline to differentiate between Simultaneous and QR algorithm).

Simultaneous Iteration $\Leftrightarrow$ QR Algorithm

- Further, we introduce a new statement $\mathbf{A}^{(k)} = (\underline{\mathbf{Q}}^{(k)})^T \mathbf{A} \underline{\mathbf{Q}}^{(k)}$ in simultaneous iteration

Simultaneous Iteration

Pick $\hat{\mathbf{Q}}^{(0)} \in \mathbb{R}^{m \times n}$

for $k = 1, 2, \cdots$

$\mathbf{Z} = \mathbf{A}\underline{\mathbf{Q}}^{(k-1)}$

$\mathbf{Z} = \underline{\mathbf{Q}}^{(k)}\mathbf{R}^{(k)}$

$\mathbf{A}^{(k)} = (\underline{\mathbf{Q}}^{(k)})^T \mathbf{A} \underline{\mathbf{Q}}^{(k)}$

"Pure" QR Algorithm

$\mathbf{A}^{(0)} = \mathbf{A}$

for $k = 1, 2, \cdots$

$\mathbf{A}^{(k-1)} = \mathbf{Q}^{(k)}\mathbf{R}^{(k)}$

$\mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)}$

$\underline{\mathbf{Q}}^{(k)} = \mathbf{Q}^{(1)}\mathbf{Q}^{(2)} \ldots \mathbf{Q}^{(k)}$

- Let's look at the sequence of $\mathbf{R}^{(k)}$: $\mathbf{R}^k = (\mathbf{Q}^{(k)})^T \mathbf{Z}^{(k)} = (\mathbf{Q}^{(k)})^T \mathbf{A}\mathbf{Q}^{(k-1)}$
- if $\mathbf{Q}_k$ converges to some $\mathbf{Q}$ then $\mathbf{Q}^T\mathbf{A}\mathbf{Q} = \mathbf{R}$ is upper triangular
- This is a Schur Decomposition of $\mathbf{A}$
- Thus, the eigenvalues of $\mathbf{A}$ are located on the main diagonal of $\mathbf{R}$
- And the columns of $\mathbf{Q}$ are the eigenvectors

Simultaneous Iteration $\Leftrightarrow$ QR Algorithm

- $\underline{\mathbf{Q}}^{(k)} = \mathbf{Q}^{(1)}\mathbf{Q}^{(2)}\dots\mathbf{Q}^{(k)}$. Let $\underline{\mathbf{R}}^{(k)} = \mathbf{R}^{(k)}\mathbf{R}^{(k-1)}\dots\mathbf{R}^{(1)}$

Theorem

Both schemes generate QR factorization of k-th power of A:

$$\mathbf{A}^k = \underline{\mathbf{Q}}^{(k)}\underline{\mathbf{R}}^{(k)}$$

and projection

$$\mathbf{A}^{(k)} = (\underline{\mathbf{Q}}^{(k)})^T\mathbf{A}\underline{\mathbf{Q}}^{(k)}$$

Proof by induction. For $k = 0$ it is trivial for both algorithms.
For $k \geq 1$ with simultaneous iteration, $A^{(k)}$ is given by definition, and

$$\mathbf{A}^k = \underline{\mathbf{Q}}^{(k-1)}\underline{\mathbf{R}}^{(k-1)} = \underline{\mathbf{Q}}^{(k)}\mathbf{R}^{(k)}\underline{\mathbf{R}}^{(k-1)} = \underline{\mathbf{Q}}^{(k)}\underline{\mathbf{R}}^{(k)}$$

For $k \geq 1$ with QR algorithm,

$$\mathbf{A}^k = \underline{\mathbf{Q}}^{(k-1)}\underline{\mathbf{R}}^{(k-1)} = \underline{\mathbf{Q}}^{(k-1)}\mathbf{A}^{(k-1)}\underline{\mathbf{R}}^{(k-1)} = \underline{\mathbf{Q}}^{(k)}\underline{\mathbf{R}}^{(k)}$$

and

$$\mathbf{A}^{(k)} = (\mathbf{Q}^{(k)})^T\mathbf{A}^{(k-1)}\mathbf{Q}^{(k)} = (\underline{\mathbf{Q}}^{(k)})^T\mathbf{A}\underline{\mathbf{Q}}^{(k)}$$

Convergence of the unshifted QR Algorithm

- Since $\underline{\mathbf{Q}}^{(k)} = \hat{\mathbf{Q}}^{(k)}$ in simultaneous iteration, column vectors of $\underline{\mathbf{Q}}^{(k)}$ converge linearly to eigenvectors if $\mathbf{A}$ has distinct eigenvalues
- $\mathbf{A}^{(k)} = (\underline{\mathbf{Q}}^{(k)})^T \mathbf{A} \underline{\mathbf{Q}}^{(k)}$, diagonal entries of $\mathbf{A}^{(k)}$ are Rayleigh quotients of column vectors of $\underline{\mathbf{Q}}^{(k)}$, so they converge linearly to eigenvalues of $\mathbf{A}$
- Off-diagonal entries of $\mathbf{A}^{(k)}$ converge to zeros, as they are generalized Rayleigh quotients involving approximations of distinct eigenvectors
- Overall, $\mathbf{A} = \underline{\mathbf{Q}}^{(k)} \mathbf{A}^{(k)} (\underline{\mathbf{Q}}^{(k)})^T$. For a symmetric matrix, it converges to eigenvalue decomposition of $\mathbf{A}$