Eigenvalue Problems

The Eigenvalue Decomposition

- Eigenvalue problem for $m \times m$ matrix A:

$$Ax = \lambda x$$

  with *eigenvalues* λ and *eigenvectors* x (nonzero)

- *Eigenvalue decomposition* of A:

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

  with eigenvectors as columns of X and eigenvalues on diagonal of Λ

- In "eigenvector coordinates", A is diagonal:

$$Ax = b \;\; \longrightarrow \;\; (X^{-1}b) = \Lambda(X^{-1}x)$$

Multiplicity

- The eigenvectors corresponding to a single eigenvalue λ (plus the zero vector) form an *eigenspace*
- Dimension of E_λ = $\dim(\text{null}(A - \lambda I))$ = *geometric multiplicity* of λ
- The *characteristic polynomial* of A is

$$p_A(z) = \det(zI - A) = (z - \lambda_1)(z - \lambda_2)\cdots(z - \lambda_m)$$

- λ is eigenvalue of $A \iff p_A(\lambda) = 0$
  - Since if λ is eigenvalue, $\lambda x - Ax = 0$. Then $\lambda I - A$ is singular, so $\det(\lambda I - A) = 0$
- Multiplicity of a root λ to p_A = *algebraic multiplicity* of λ
- Any matrix A has m eigenvalues, counted with algebraic multiplicity

Similarity Transformations

- The map $A \mapsto X^{-1}AX$ is a *similarity transformation* of A
- A and B are *similar* if there is a similarity transformation $B = X^{-1}AX$
- A and $X^{-1}AX$ have the same characteristic polynomials, eigenvalues, and multiplicities:
  - The characteristic polynomials are the same:

$$\begin{aligned} p_{X^{-1}AX}(z) &= \det(zI - X^{-1}AX) = \det(X^{-1}(zI - A)X) \\ &= \det(X^{-1})\det(zI - A)\det(X) = \det(zI - A) = p_A(z) \end{aligned}$$

  - Therefore, the algebraic multiplicities are the same
  - If E_λ is eigenspace for A, then $X^{-1}E_\lambda$ is eigenspace for $X^{-1}AX$, so geometric multiplicities are the same

Algebraic Multiplicity $\geq$ Geometric Multiplicity

- Let n first columns of $\hat{V}$ be orthonormal basis of the eigenspace for λ
- Extend $\hat{V}$ to square unitary V, and form

$$B = V^*AV = \begin{bmatrix} \lambda I & C \\ 0 & D \end{bmatrix}$$

- Since

$$\det(zI - B) = \det(zI - \lambda I)\det(zI - D) = (z - \lambda)^n \det(zI - D)$$

the algebraic multiplicity of λ (as eigenvalue of B) is $\geq n$

- A and B are similar; so the same is true for λ of A

Defective and Diagonalizable Matrices

- If the algebraic multiplicity for an eigenvalue $>$ its geometric multiplicity, it is a *defective eigenvalue*
- If a matrix has any defective eigenvalues, it is a *defective matrix*
- A *nondefective* or *diagonalizable* matrix has equal algebraic and geometric multiplicities for all eigenvalues
- The matrix A is nondefective $\iff A = X\Lambda X^{-1}$
  - ($\Longleftarrow$) If $A = X\Lambda X^{-1}$, A is similar to Λ and has the same eigenvalues and multiplicities. But Λ is diagonal and thus nondefective.
  - ($\Longrightarrow$) Nondefective A has m linearly independent eigenvectors. Take these as the columns of X, then $A = X\Lambda X^{-1}$.

Determinant and Trace

- The *trace* of A is $\operatorname{tr}(A) = \sum_{j=1}^{m} a_{jj}$
- The determinant and the trace are given by the eigenvalues:

$$\det(A) = \prod_{j=1}^{m} \lambda_j, \qquad \operatorname{tr}(A) = \sum_{j=1}^{m} \lambda_j$$

since $\det(A) = (-1)^m \det(-A) = (-1)^m p_A(0) = \prod_{j=1}^{m} \lambda_j$ and

$$p_A(z) = \det(zI - A) = z^m - \sum_{j=1}^{m} a_{jj} z^{m-1} + \cdots$$

$$p_A(z) = (z - \lambda_1) \cdots (z - \lambda_m) = z^m - \sum_{j=1}^{m} \lambda_j z^{m-1} + \cdots$$

Unitary Diagonalization and Schur Factorization

- A matrix A is *unitary diagonalizable* if, for a unitary matrix Q, $A = Q\Lambda Q^*$
- A hermitian matrix is unitarily diagonalizable, with real eigenvalues (because of the Schur factorization, see below)
- A is unitarily diagonalizable $\Longleftrightarrow$ A is normal ($A^*A = AA^*$)
- Every square matrix A has a Schur factorization $A = QTQ^*$ with unitary Q and upper-triangular T
- Summary, Eigenvalue-Revealing Factorizations
  - Diagonalization $A = X\Lambda X^{-1}$ (nondefective A)
  - Unitary diagonalization $A = Q\Lambda Q^*$ (normal A)
  - Unitary triangularization (Schur factorization) $A = QTQ^*$ (any A)

Eigenvalue Algorithms

- The most obvious method is ill-conditioned: Find roots of $p_A(\lambda)$
- Instead, compute Schur factorization $A = QTQ^*$ by introducing zeros
- However, this can not be done in a 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

Eigenvalue Algorithms

- (continued) However, the roots of p are the eigenvalues of the *companion matrix*

$$A = \begin{bmatrix} 0 & & & & & -a_0 \\ 1 & 0 & & & & -a_1 \\ & 1 & 0 & & & -a_2 \\ & & 1 & \ddots & & \vdots \\ & & & \ddots & 0 & -a_{m-2} \\ & & & & 1 & -a_{m-1} \end{bmatrix}$$

- Therefore, in general we cannot find the eigenvalues of a matrix in a finite number of steps (even in exact arithmetic)

- In practice, algorithms available converge in just a few iterations

Schur Factorization and Diagonalization

- Compute Schur factorization $A = QTQ^*$ by transforming A with similarity transformations

$$\underbrace{Q_j^* \cdots Q_2^* Q_1^*}_{Q^*} A \underbrace{Q_1 Q_2 \cdots Q_j}_{Q}$$

  which converge to a T as $j \to \infty$

- Note: Real matrices might need complex Schur forms and eigenvalues (or a *real Schur factorization* with 2×2 blocks on diagonal)

- For hermitian A, the sequence converges to a diagonal matrix

Two Phases of Eigenvalues Computations

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

$$\underset{A \neq A^*}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\text{Phase 1}} \underset{H}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}} \xrightarrow{\text{Phase 2}} \underset{T}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}}$$

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

$$\underset{A \neq A^*}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\text{Phase 1}} \underset{T}{\begin{bmatrix} \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}} \xrightarrow{\text{Phase 2}} \underset{D}{\begin{bmatrix} \times & & & & \\ & \times & & & \\ & & \times & & \\ & & & \times & \\ & & & & \times \end{bmatrix}}$$

Hessenberg/Tridiagonal Reduction

Introducing Zeros by Similarity Transformations

- Try computing the Schur factorization $A = QTQ^*$ by applying Householder reflectors from left and right that introduce zeros:

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{Q_1^*} \underset{Q_1^*A}{\begin{bmatrix} \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}} \xrightarrow{Q_1} \underset{Q_1^*AQ_1}{\begin{bmatrix} \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}$$

- The right multiplication destroys the zeros previously introduced

- We already knew this would not work, because of Abel's theorem

- However, the subdiagonal entries typically decrease in magnitude

The Hessenberg Form

- Instead, try computing an upper Hessenberg matrix H similar to A:

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{Q_1^*} \underset{Q_1^* A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}} \xrightarrow{Q_1} \underset{Q_1^* A Q_1}{\begin{bmatrix} \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}$$

- This time the zeros we introduce are not destroyed

- Continue in a similar way with column 2:

$$\underset{Q_1^* A Q_1}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{Q_1^*} \underset{Q_2^* Q_1^* A Q_1}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}} \xrightarrow{Q_1} \underset{Q_2^* Q_1^* A Q_1 Q_2}{\begin{bmatrix} \times & \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \times & \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}$$

The Hessenberg Form

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

$$\underbrace{Q_{m-2}^* \cdots Q_2^* Q_1^*}_{Q^*} A \underbrace{Q_1 Q_2 \cdots Q_{m-2}}_{Q} = H = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}$$

- For hermitian A, zeros are also introduced above diagonals

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{Q_1^*} \underset{Q_1^* A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \end{bmatrix}} \xrightarrow{Q_1} \underset{Q_1^* A Q_1}{\begin{bmatrix} \times & \mathbf{\times} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \times & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \end{bmatrix}}$$

producing a tridiagonal matrix T after $m-2$ steps

Householder Reduction to Hessenberg

Algorithm: Householder Hessenberg

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

$$x = A_{k+1:m,k}$$

$$v_k = \text{sign}(x_1)\|x\|_2 e_1 + x$$

$$v_k = v_k / \|v_k\|_2$$

$$A_{k+1:m,k:m} = A_{k+1:m,k:m} - 2v_k(v_k^* A_{k+1:m,k:m})$$

$$A_{1:m,k+1:m} = A_{1:m,k+1:m} - 2(A_{1:m,k+1:m} v_k) v_k^*$$

- Operation count (*not* twice Householder QR):

$$\sum_{k=1}^{m} 4(m-k)^2 + 4m(m-k) = \underbrace{4m^3/3}_{QR} + 4m^3 - 4m^3/2 = 10m^3/3$$

- For hermitian A, operation count is twice QR divided by two $= 4m^3/3$

Power Iteration

Real Symmetric Matrices

- We will only consider eigenvalue problems for real symmetric matrices
- Then $A = A^T \in \mathbb{R}^{m \times m}$, $x \in \mathbb{R}^m$, $x^* = x^T$, and $\|x\| = \sqrt{x^T x}$
- A then also has

real eigenvalues: $\lambda_1, \ldots, \lambda_m$

orthonormal eigenvectors: $q_1, \ldots, q_m$

- Eigenvectors are normalized $\|q_j\| = 1$, and sometimes the eigenvalues are ordered in a particular way
- Initial reduction to tridiagonal form assumed
  - Brings cost for typical steps down from $O(m^3)$ to $O(m)$

Rayleigh Quotient

- The *Rayleigh quotient* of $x \in \mathbb{R}^m$:

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

- For an eigenvector x, the corresponding eigenvalue is $r(x) = \lambda$
- For general x, $r(x) = \alpha$ that minimizes $\|Ax - \alpha x\|_2$
- x eigenvector of $A \Longleftrightarrow \nabla r(x) = 0$ with $x \neq 0$
- $r(x)$ is smooth and $\nabla r(q_j) = 0$, therefore quadratically accurate:

$$r(x) - r(q_J) = O(\|x - q_J\|^2) \text{ as } x \to q_J$$

Power Iteration

- Simple power iteration for largest eigenvalue:

Algorithm: Power Iteration

$v^{(0)} =$ some vector with $\|v^{(0)}\| = 1$

for $k = 1, 2, \ldots$

$w = Av^{(k-1)}$	apply A
$v^{(k)} = w/\|w\|$	normalize
$\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$	Rayleigh quotient

- Termination conditions usually omitted

Convergence of Power Iteration

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

$$\begin{aligned} v^{(0)} &= a_1 q_1 + a_2 q_2 + \cdots + a_m q_m \\ v^{(k)} &= c_k A^k v^{(0)} \\ &= c_k (a_1 \lambda_1^k q_1 + a_2 \lambda_2^k q_2 + \cdots + a_m \lambda_m^k q_m) \\ &= c_k \lambda_1^k (a_1 q_1 + a_2 (\lambda_2/\lambda_1)^k q_2 + \cdots + a_m (\lambda_m/\lambda_1)^k q_m) \end{aligned}$$

- If $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_m| \geq 0$ and $q_1^T v^{(0)} \neq 0$, this gives:

$$\|v^{(k)} - (\pm q_1)\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right), \qquad |\lambda^{(k)} - \lambda_1| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)$$

- Finds the largest eigenvalue (unless eigenvector orthogonal to $v^{(0)}$)
- Linear convergence, factor $\approx \lambda_2/\lambda_1$ at each iteration

Inverse Iteration

- Apply power iteration on $(A - \mu I)^{-1}$, with eigenvalues $(\lambda_j - \mu)^{-1}$

Algorithm: Inverse Iteration

$v^{(0)} =$ some vector with $\|v^{(0)}\| = 1$

for $k = 1, 2, \ldots$

Solve $(A - \mu I)w = v^{(k-1)}$ for w — apply $(A - \mu I)^{-1}$

$v^{(k)} = w / \|w\|$ — normalize

$\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$ — Rayleigh quotient

- Converges to eigenvector q_J if the parameter μ is close to λ_J:

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

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

$v^{(0)} =$ some vector with $\|v^{(0)}\| = 1$

$\lambda^{(0)} = (v^{(0)})^T A v^{(0)}$ = corresponding Rayleigh quotient

for $k = 1, 2, \ldots$

Solve $(A - \lambda^{(k-1)} I) w = v^{(k-1)}$ for w	apply matrix
$v^{(k)} = w / \|w\|$	normalize
$\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$	Rayleigh quotient

Convergence of Rayleigh Quotient Iteration

- Cubic convergence in Rayleigh quotient iteration:

$$\|v^{(k+1)} - (\pm q_J)\| = O(\|v^{(k)} - (\pm q_J)\|^3)$$

  and

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

- Proof idea: If $v^{(k)}$ is close to an eigenvector, $\|v^{(k)} - q_J\| \le \epsilon$, then the accurate of the Rayleigh quotient estimate $\lambda^{(k)}$ is $|\lambda^{(k)} - \lambda_J| = O(\epsilon^2)$. One step of inverse iteration then gives

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

QR Algorithm

The QR Algorithm

- Remarkably simple algorithm: QR factorize and multiply in reverse order:

Algorithm: “Pure” QR Algorithm

$A^{(0)} = A$

for $k = 1, 2, \ldots$

$Q^{(k)} R^{(k)} = A^{(k-1)}$ QR factorization of $A^{(k-1)}$

$A^{(k)} = R^{(k)} Q^{(k)}$ Recombine factors in reverse order

- With some assumptions, $A^{(k)}$ converge to a Schur form for A (diagonal if A symmetric)
- Similarity transformations of A:

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

Unnormalized Simultaneous Iteration

- To understand the QR algorithm, first consider a simpler algorithm
- *Simultaneous Iteration* is power iteration applied to several vectors
- Start with linearly independent $v_1^{(0)}, \ldots, v_n^{(0)}$
- We know from power iteration that $A^k v_1^{(0)}$ converges to q_1
- With some assumptions, the space $\langle A^k v_1^{(0)}, \ldots, A^k v_n^{(0)} \rangle$ should converge to $q_1, \ldots, q_n$
- Notation: Define initial matrix $V^{(0)}$ and matrix $V^{(k)}$ at step k:

$$V^{(0)} = \left[\begin{array}{c|c|c} v_1^{(0)} & \cdots & v_n^{(0)} \end{array} \right], \quad V^{(k)} = A^k V^{(0)} = \left[\begin{array}{c|c|c} v_1^{(k)} & \cdots & v_n^{(k)} \end{array} \right]$$

Unnormalized Simultaneous Iteration

- Define well-behaved basis for column space of $V^{(k)}$ by $\hat{Q}^{(k)}\hat{R}^{(k)} = V^{(k)}$
- Make the assumptions:
  - The leading $n + 1$ eigenvalues are distinct
  - All principal leading principal submatrices of $\hat{Q}^T V^{(0)}$ are nonsingular, where columns of $\hat{Q}$ are $q_1, \ldots, q_n$

  We then have that the columns of $\hat{Q}^{(k)}$ converge to eigenvectors of A:

$$\|q_j^{(k)} - \pm q_j\| = O(C^k)$$

  where $C = \max_{1 \leq k \leq n} |\lambda_{k+1}|/|\lambda_k|$
- *Proof.* Textbook / Black board

Simultaneous Iteration

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

Algorithm: Simultaneous Iteration

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

for $k = 1, 2, \ldots$

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

$$\hat{Q}^{(k)}\hat{R}^{(k)} = Z$$ Reduced QR factorization of Z

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

Simultaneous Iteration $\Longleftrightarrow$ QR Algorithm

- The QR algorithm is equivalent to simultaneous iteration with $\hat{Q}^{(0)} = I$
- Notation: Replace $\hat{R}^{(k)}$ by $R^{(k)}$, and $\hat{Q}^{(k)}$ by $\underline{Q}^{(k)}$

Simultaneous Iteration:

$$\begin{aligned} \underline{Q}^{(0)} &= I \\ Z &= A\underline{Q}^{(k-1)} \\ Z &= \underline{Q}^{(k)} R^{(k)} \\ A^{(k)} &= (\underline{Q}^{(k)})^T A \underline{Q}^{(k)} \end{aligned}$$

Unshifted QR Algorithm:

$$\begin{aligned} A^{(0)} &= A \\ A^{(k-1)} &= Q^{(k)} R^{(k)} \\ A^{(k)} &= R^{(k)} Q^{(k)} \\ \underline{Q}^{(k)} &= Q^{(1)} Q^{(2)} \cdots Q^{(k)} \end{aligned}$$

- Also define $\underline{R}^{(k)} = R^{(k)} R^{(k-1)} \cdots R^{(1)}$
- Now show that the two processes generate same sequences of matrices

Simultaneous Iteration $\Longleftrightarrow$ QR Algorithm

- Both schemes generate the QR factorization $A^k = \underline{Q}^{(k)}\underline{R}^{(k)}$ and the projection $A^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}$

- *Proof.* $k = 0$ trivial for both algorithms.

  For $k \geq 1$ with simultaneous iteration, $A^{(k)}$ is given by definition, and

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

  For $k \geq 1$ with unshifted QR, we have

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

  and

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

Simultaneous *Inverse* Iteration $\Longleftrightarrow$ QR Algorithm

- Last lecture we showed that "pure" QR $\Longleftrightarrow$ simultaneous iteration applied to I, and the first column evolves as in power iteration

- But it is also equivalent to simultaneous *inverse* iteration applied to a "flipped" I, and the last column evolves as in inverse iteration

- To see this, recall that $A^k = \underline{Q}^{(k)}\underline{R}^{(k)}$ with

$$\underline{Q}^{(k)} = \prod_{j=1}^{k} Q^{(j)} = \left[\begin{array}{c|c|c|c} q_1^{(k)} & q_2^{(k)} & \cdots & q_m^{(k)} \end{array}\right]$$

- Invert and use that A^{-1} is symmetric:

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

Simultaneous *Inverse* Iteration $\Longleftrightarrow$ QR Algorithm

- Introduce the "flipping" permutation matrix

$$P = \begin{bmatrix} & & & 1 \\ & & 1 & \\ & \cdots & & \\ 1 & & & \end{bmatrix}$$

and rewrite that last expression as

$$A^{-k} P = [\underline{Q}^{(k)} P][P(\underline{R}^{(k)})^{-T} P]$$

- This is a QR factorization of $A^{-k}P$, and the algorithm is equivalent to simultaneous iteration on A^{-1}
- In particular, the last column of $\underline{Q}^{(k)}$ evolves as in inverse iteration

The Shifted QR Algorithm

- Since the QR algorithm behaves like inverse iteration, introduce shifts $\mu^{(k)}$ to accelerate the convergence:

$$A^{(k-1)} - \mu^{(k)} I = Q^{(k)} R^{(k)}$$
$$A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I$$

- We then get (same as before):

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

and (different from before):

$$(A - \mu^{(k)} I)(A - \mu^{(k-1)} I) \cdots (A - \mu^{(1)} I) = \underline{Q}^{(k)} \underline{R}^{(k)}$$

- Shifted simultaneous iteration – last column of $\underline{Q}^{(k)}$ converges quickly

Choosing $\mu^{(k)}$: The Rayleigh Quotient Shift

- Natural choice of $\mu^{(k)}$: Rayleigh quotient for last column of $\underline{Q}^{(k)}$

$$\mu^{(k)} = \frac{(q_m^{(k)})^T A q_m^{(k)}}{(q_m^{(k)})^T q_m^{(k)}} = (q_m^{(k)})^T A q_m^{(k)}$$

- Rayleigh quotient iteration, last column $q_m^{(k)}$ converges cubically
- Convenient fact: This Rayleigh quotient appears as m, m entry of $A^{(k)}$ since $A^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}$
- The *Rayleigh quotient shift* corresponds to setting $\mu^{(k)} = A_{mm}^{(k)}$

Choosing $\mu^{(k)}$: The Wilkinson Shift

- The QR algorithm with Rayleigh quotient shift might fail, e.g. with two symmetric eigenvalues
- Break symmetry by the *Wilkinson shift*

$$\mu = a_m - \text{sign}(\delta) b_{m-1}^2 \Big/ \left(|\delta| + \sqrt{\delta^2 + b_{m-1}^2} \right)$$

where $\delta = (a_{m-1} - a_m)/2$ and $B = \begin{bmatrix} a_{m-1} & b_{m-1} \\ b_{m-1} & a_m \end{bmatrix}$ is the lower-right submatrix of $A^{(k)}$

- Always convergence with this shift, in worst case quadratically

A Practical Shifted QR Algorithm

Algorithm: "Practical" QR Algorithm

$(Q^{(0)})^T A^{(0)} Q^{(0)} = A$ — $A^{(0)}$ is a tridiagonalization of A

for $k = 1, 2, \ldots$

Pick a shift $\mu^{(k)}$ — e.g., choose $\mu^{(k)} = A_{mm}^{(k-1)}$

$Q^{(k)} R^{(k)} = A^{(k-1)} - \mu^{(k)} I$ — QR factorization of $A^{(k-1)} - \mu^{(k)} I$

$A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I$ — Recombine factors in reverse order

If any off-diagonal element $A_{j,j+1}^{(k)}$ is sufficiently close to zero,

set $A_{j,j+1} = A_{j+1,j} = 0$ to obtain

$$\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} = A^{(k)}$$

and now apply the QR algorithm to A_1 and A_2

Stability and Accuracy

- The QR algorithm is backward stable:

$$\tilde{Q}\tilde{\Lambda}\tilde{Q}^T = A + \delta A, \qquad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{\text{machine}})$$

where $\tilde{\Lambda}$ is the computed Λ and $\tilde{Q}$ is an exactly orthogonal matrix

- The combination with Hessenberg reduction is also backward stable
- Can be shown (for normal matrices) that $|\tilde{\lambda}_j - \lambda_j| \leq \|\delta A\|_2$, which gives

$$\frac{|\tilde{\lambda}_j - \lambda_j|}{\|A\|} = O(\epsilon_{\text{machine}})$$

where $\tilde{\lambda}_j$ are the computed eigenvalues