CS 450: Numerical Anlaysis[1]

Eigenvalue Problems

University of Illinois at Urbana-Champaign

[1] *These slides have been drafted by Edgar Solomonik as lecture templates and supplementary material for the book "Scientific Computing: An Introductory Survey" by Michael T. Heath (slides).*

Eigenvalues and Eigenvectors

- A matrix $\boldsymbol{A}$ has eigenvector-eigenvalue pair (eigenpair) $(\lambda, \boldsymbol{x})$ if

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

  - *For any scalar α, $\alpha\boldsymbol{x}$ is also an eigenvector of $\boldsymbol{A}$ with eigenvalue λ*
  - *Generally, an eigenvalue λ is associated with an eigenspace $\mathcal{X} \subseteq \mathbb{C}^n$ such that each $\boldsymbol{x} \in \mathcal{X}$ is an eigenvector of $\boldsymbol{A}$ with eigenvalue λ.*
  - *The dimensionality of an eigenspace is at most the multiplicity of an eigenvalue (when less, matrix is defective, otherwise matrix is diagonalizable).*
- Each $n \times n$ matrix has up to n eigenvalues, which are either real or complex
  - *The conjugate of any complex eigenvalue of a real matrix is also an eigenvalue.*
  - *The dimensionalities of all the eigenspaces (multiplicity associated with each eigenvalue) sum up to n for a diagonalizable matrix.*
  - *If the matrix is real, real eigenvalues are associated with real eigenvectors, but complex eigenvalues may not be.*

Eigenvalue Decomposition

- If a matrix $\boldsymbol{A}$ is diagonalizable, it has an *eigenvalue decomposition*

$$\boldsymbol{A} = \boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^{-1}$$

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

$$\boldsymbol{A}\boldsymbol{X} = \begin{bmatrix}\boldsymbol{A}\boldsymbol{x}_1 & \cdots \boldsymbol{A}\boldsymbol{x}_n\end{bmatrix} = \boldsymbol{X}\boldsymbol{D} = \begin{bmatrix}d_{11}\boldsymbol{x}_1 & \cdots & d_{nn}\boldsymbol{x}_n\end{bmatrix}.$$

  - *If $\boldsymbol{A}$ is symmetric, its right and left singular vectors are the same, and consequently are its eigenvectors.*
  - *More generally, any normal matrix, $\boldsymbol{A}^H\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^H$, has unitary eigenvectors.*
- $\boldsymbol{A}$ and $\boldsymbol{B}$ are *similar*, if there exist $\boldsymbol{Z}$ such that $\boldsymbol{A} = \boldsymbol{Z}\boldsymbol{B}\boldsymbol{Z}^{-1}$
  - *Normal matrices are unitarily similar ($\boldsymbol{Z}^{-1} = \boldsymbol{Z}^H$) to diagonal matrices*
  - *Symmetric real matrices are orthogonally similar ($\boldsymbol{Z}^{-1} = \boldsymbol{Z}^T$) to real diagonal matrices*
  - *Hermitian matrices are unitarily similar to real diagonal matrices*

Similarity of Matrices

matrix	*similarity*	*reduced form*
SPD	orthogonal	real positive diagonal
real symmetric	orthogonal	real tridiagonal real diagonal
Hermitian	unitary	real diagonal
normal	unitary	diagonal
real	orthogonal	real Hessenberg
diagonalizable	invertible	diagonal
arbitrary	unitary invertible	triangular bidiagonal

Canonical Forms

- Any matrix is *similar* to a bidiagonal matrix, giving its *Jordan form*:

$$\boldsymbol{A} = \boldsymbol{X} \begin{bmatrix} \boldsymbol{J}_1 & & \\ & \ddots & \\ & & \boldsymbol{J}_k \end{bmatrix} \boldsymbol{X}^{-1}, \quad \forall i, \quad \boldsymbol{J}_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix}$$

  the Jordan form is unique modulo ordering of the diagonal Jordan blocks.

- Any diagonalizable matrix is *unitarily similar* to a triangular matrix, giving its *Schur form*:

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{T}\boldsymbol{Q}^H$$

  where $\boldsymbol{T}$ is upper-triangular, so the eigenvalues of $\boldsymbol{A}$ is the diagonal of $\boldsymbol{T}$. Columns of $\boldsymbol{Q}$ are the Schur vectors.

Eigenvectors from Schur Form

- Given the eigenvectors of one matrix, we seek those of a similar matrix:
  Suppose that $\boldsymbol{A} = \boldsymbol{S}\boldsymbol{B}\boldsymbol{S}^{-1}$ *and* $\boldsymbol{B} = \boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^{-1}$ *where* $\boldsymbol{D}$ *is diagonal,*
  - *The eigenvalues of* $\boldsymbol{A}$ *are* $\{d_{11}, \ldots, d_{nn}\}$
  - $\boldsymbol{A} = \boldsymbol{S}\boldsymbol{B}\boldsymbol{S}^{-1} = \boldsymbol{S}\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^{-1}\boldsymbol{S}^{-1}$ *so* $\boldsymbol{S}\boldsymbol{X}$ *are the eigenvectors of* $\boldsymbol{A}$
- Its easy to obtain eigenvectors of triangular matrix $\boldsymbol{T}$:
  - *One eigenvector is simply the first elementary vector.*
  - *The eigenvector associated with any diagonal entry (eigenvalue* λ*) may be obtaining by observing that*

$$\boldsymbol{0} = (\boldsymbol{T} - \lambda\boldsymbol{I})\boldsymbol{x} = \begin{bmatrix} \boldsymbol{U}_{11} & \boldsymbol{u} & \boldsymbol{T}_{13} \\ & 0 & \boldsymbol{v}^T \\ & & \boldsymbol{U}_{33} \end{bmatrix} \begin{bmatrix} -\boldsymbol{U}_{11}^{-1}\boldsymbol{u} \\ 1 \\ \boldsymbol{0} \end{bmatrix},$$

so it suffices to solve $\boldsymbol{U}_{11}\boldsymbol{y} = -\boldsymbol{u}$ *to obtain eigenvector* $\boldsymbol{x}$*.*

Rayleigh Quotient

- For any vector $\boldsymbol{x}$, the *Rayleigh quotient* provides an estimate for some eigenvalue of $\boldsymbol{A}$:

$$\rho_{\boldsymbol{A}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^H \boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^H \boldsymbol{x}}.$$

  - *If $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{A}$, then $\rho_{\boldsymbol{A}}(\boldsymbol{x})$ is the associated eigenvalue.*
  - *Moreover, for $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, the Rayleigh quotient is the best possible eigenvalue estimate given $\boldsymbol{x}$ and $\boldsymbol{y}$, as it is the solution to $\boldsymbol{x}\alpha \cong \boldsymbol{y}$.*
  - *The normal equations for this scalar-output least squares problem are*

$$\boldsymbol{x}^T \boldsymbol{x}\alpha = \boldsymbol{x}^T \boldsymbol{y} \quad \Rightarrow \quad \alpha = \frac{\boldsymbol{x}^T \boldsymbol{y}}{\boldsymbol{x}^T \boldsymbol{x}} = \frac{\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}.$$

Perturbation Analysis of Eigenvalue Problems

- Suppose we seek eigenvalues $\boldsymbol{D} = \boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}$, but find those of a slightly perturbed matrix $\boldsymbol{D} + \boldsymbol{\delta D} = \hat{\boldsymbol{X}}^{-1}(\boldsymbol{A} + \boldsymbol{\delta A})\hat{\boldsymbol{X}}$:

  Note that the eigenvalues of $\boldsymbol{X}^{-1}(\boldsymbol{A} + \boldsymbol{\delta A})\boldsymbol{X} = \boldsymbol{D} + \boldsymbol{X}^{-1}\boldsymbol{\delta A}\boldsymbol{X}$ are also $\boldsymbol{D} + \boldsymbol{\delta D}$. So if we have perturbation to the matrix $||\boldsymbol{\delta A}||_F$, its effect on the eigenvalues corresponds to a (non-diagonal/arbitrary) perturbation $\boldsymbol{\delta \hat{A}} = \boldsymbol{X}^{-1}\boldsymbol{\delta A}\boldsymbol{X}$ of a diagonal matrix of eigenvalues $\boldsymbol{D}$, with norm

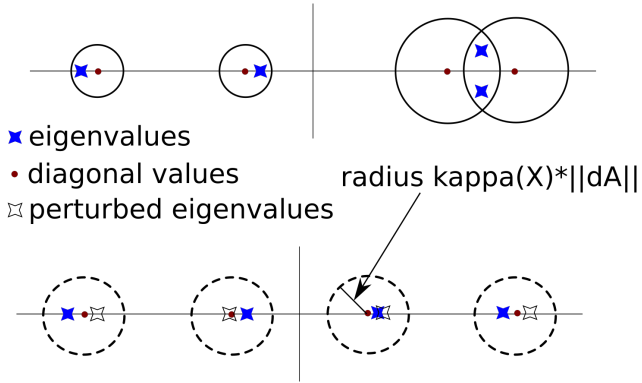
$$||\boldsymbol{\delta \hat{A}}||_F \leq ||\boldsymbol{X}^{-1}||_2||\boldsymbol{\delta A}||_F||\boldsymbol{X}||_2 = \kappa(\boldsymbol{X})||\boldsymbol{\delta A}||_F.$$

- Gershgorin's theorem allows us to bound the effect of the perturbation on the eigenvalues of a (diagonal) matrix:
  Given a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, let $r_i = \sum_{j\neq i} |a_{ij}|$, define the Gershgorin disks as

$$D_i = \{z \in \mathbb{C} : |z - a_{ii}| \leq r_i\}.$$

  The eigenvalues $\lambda_1, \ldots, \lambda_n$ of any matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ are contained in the union of the Gershgorin disks, $\forall i \in \{1, \ldots, n\}, \lambda_i \in \bigcup_{j=1}^{n} D_j$.

Gershgorin Theorem Perturbation Visualization



- Top corresponds to Gershgorin disks on complex plane of 4-by-4 real matrix.
- Bottom part corresponds to bounds on Gershgorin disks of $\boldsymbol{X}^{-1}(\boldsymbol{A}+\boldsymbol{\delta A})\boldsymbol{X}$, which contain the eigenvalues $\boldsymbol{D}$ of $\boldsymbol{A}$ and the perturbed eigenvalues $\boldsymbol{D}+\boldsymbol{\delta D}$ of $\boldsymbol{A}+\boldsymbol{\delta A}$ provided that $||\boldsymbol{\delta A}||$ is sufficiently small.

Conditioning of Particular Eigenpairs

- Consider the effect of a matrix perturbation on an eigenvalue λ associated with a right eigenvector $\boldsymbol{x}$ and a left eigenvector $\boldsymbol{y}$, $\lambda = \boldsymbol{y}^H \boldsymbol{A}\boldsymbol{x}/\boldsymbol{y}^H\boldsymbol{x}$

  For a sufficiently small perturbation $\boldsymbol{\delta A}$, the eigenvalue λ is perturbed to an eigenvalue $\hat{\lambda}$ of $\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{\delta A}$. The eigenvalue perturbation,

$$|\hat{\lambda} - \lambda| = |\boldsymbol{y}^H \boldsymbol{\delta A}\boldsymbol{x}/\boldsymbol{y}^H\boldsymbol{x}| \leq \frac{||\boldsymbol{\delta A}||}{|\boldsymbol{y}^H\boldsymbol{x}|},$$

  is small if $\boldsymbol{x}$ is near-parallel to $\boldsymbol{y}$ and large if they are near-perpendicular.

- A more accurate eigenvalue approximation than Rayleigh quotient for a normalized perturbed eigenvector (e.g. iterative guess) $\hat{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\delta x}$, can be obtain with an estimate of both eigenvectors (also $\hat{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{\delta y}$),

$$|\hat{\lambda}_{xAx} - \lambda| \approx |\boldsymbol{\delta x}^H \boldsymbol{A}\boldsymbol{x} + \boldsymbol{x}^H \boldsymbol{A}\boldsymbol{\delta x}| \leq |\lambda|||\boldsymbol{\delta x}|| + \Big(|\lambda||\boldsymbol{y}^H\boldsymbol{x}| + |1 - \boldsymbol{y}^H\boldsymbol{x}| \cdot ||\boldsymbol{A}||\Big)||\boldsymbol{\delta x}||$$

$$|\hat{\lambda}_{yAx} - \lambda| \approx \left|\frac{\boldsymbol{\delta y}^H \boldsymbol{A}\boldsymbol{x} + \boldsymbol{y}^H \boldsymbol{A}\boldsymbol{\delta x}}{\boldsymbol{y}^H\boldsymbol{x}}\right| \leq |\lambda|\frac{||\boldsymbol{\delta x}|| + ||\boldsymbol{\delta y}||}{|\boldsymbol{y}^H\boldsymbol{x}|}$$

Power Iteration

- *Power iteration* can be used to compute the largest eigenvalue of a real symmetric matrix $\boldsymbol{A}$:

  $$\boldsymbol{x}^{(i)} = \boldsymbol{A}\boldsymbol{x}^{(i-1)} \quad \textit{(typically with normalization of } \boldsymbol{x}^{(i)} \textit{ at each step)}.$$

  For a random $\boldsymbol{x}^{(0)}$, power iteration converges eigenvalue of $\boldsymbol{A}$ with largest modulus, $\lim_{i\to\infty} \rho_{\boldsymbol{A}}(\boldsymbol{x}^{(i)}) = \lambda_{max}(\boldsymbol{A})$. If this eigenvalue has multiplicity one, power iteration converges to dominant eigenvector.

- The error of power iteration decreases at each step by the ratio of the largest eigenvalues:

  Assuming $\boldsymbol{A}$ is diagonalizable with eigenvectors $\boldsymbol{U}$ and $\boldsymbol{V}^H = \boldsymbol{U}^{-1}$,

  $$\boldsymbol{x}^{(k)} = \boldsymbol{A}^k\boldsymbol{x}^{(0)} = (\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^H)^k\boldsymbol{x}^{(0)} = \boldsymbol{U}\boldsymbol{D}^k\boldsymbol{V}^H\boldsymbol{x}^{(0)} = \sum_{i=1}^{n} \boldsymbol{u}_i \underbrace{\lambda_i^k \boldsymbol{v}_i^H \boldsymbol{x}^{(0)}}_{\alpha^{(i,k)}}.$$

  The coefficient $\alpha^{(i,k)}$ associated with the maximum λ_i and dominant eigenvector $\boldsymbol{u}_i$ grows relatively, since $|\alpha^{(i,k)}/\alpha^{(j,k)}| = (|\lambda_i|/|\lambda_j|)^k \underbrace{|\alpha^{(i,0)}/\alpha^{(j,0)}|}_{\textit{constant}}$.

Inverse and Rayleigh Quotient Iteration

Activity: *Inverse Iteration with a Shift*
Activity: *Rayleigh Quotient Iteration*

- *Inverse iteration* uses LU/QR/SVD of $\boldsymbol{A}$ to run power iteration on $\boldsymbol{A}^{-1}$
  - *For a randomly chosen* $\boldsymbol{x}^{(0)}$*, solving*

$$\boldsymbol{A}\boldsymbol{x}^{(i)} = \boldsymbol{x}^{(i-1)} \quad \textit{(typically with normalization of } \boldsymbol{x}^{(i)} \textit{ at each step).}$$

    converges to $\lim_{i\to\infty} \rho_{\boldsymbol{A}}(\boldsymbol{x}^{(i)}) = \lambda_{min}(\boldsymbol{A})$ *provided there is a unique eigenvalue with minimum magnitude.*
  - *Inverse iteration on* $\boldsymbol{A} - \sigma\boldsymbol{I}$ *converges to the eigenvalue closes to* σ*, since all eigenvalues are shifted by* σ*.*

- *Rayleigh quotient iteration* provides rapid convergence to an eigenpair

$$(\boldsymbol{A} - \rho_{\boldsymbol{A}}(\boldsymbol{x}^{(i-1)})\boldsymbol{I})\boldsymbol{x}^{(i)} = \boldsymbol{x}^{(i-1)},$$

since at each step the relative magnitude largest eigenvalue of $(\boldsymbol{A} - \rho_{\boldsymbol{A}}(\boldsymbol{x}^{(i-1)})\boldsymbol{I})^{-1}$ *grows. Formally, it achieves cubic convergence, but requires matrix refactorization at each step.*

Deflation

- Power, inverse, and Rayleigh-quotient iteration compute a single eigenpair, to obtain further eigenpairs, can perform *deflation*
  - *Given eigenvalue λ_1 and right eigenvector $\boldsymbol{x}_1$, seek $\boldsymbol{v}$ so that $\boldsymbol{B} = \boldsymbol{A} - \lambda_1 \boldsymbol{u}\boldsymbol{v}^T$ has eigenvalues $\lambda_2, \ldots, \lambda_n$, where*

$$\boldsymbol{A} = \boldsymbol{X}\boldsymbol{D}\underbrace{\boldsymbol{Y}^T}_{\boldsymbol{X}^{-1}} = \sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i \boldsymbol{y}_i^T.$$

  - *Ideal choice would be $\boldsymbol{v} = \boldsymbol{y}_1^T$, i.e. the left eigenvector associated with λ_1, as then the $n-1$ other eigenvectors of $\boldsymbol{B}$ would be the same as those of $\boldsymbol{A}$.*
  - *For symmetric matrices $\boldsymbol{y}_1 = \boldsymbol{x}_1$, but for nonsymmetric, obtaining $\boldsymbol{y}_1$ may require more work.*
  - *Good alternative choice for nonsymmetric is to select $\boldsymbol{v} = \boldsymbol{x}_1$, as then the Schur vectors are unmodified, since for $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{T}\boldsymbol{Q}^T$, with $t_{11} = \lambda_1$, $\boldsymbol{q}_1 = \boldsymbol{x}_1$, we get*

$$\boldsymbol{B} = \boldsymbol{Q}\boldsymbol{T}\boldsymbol{Q}^T - \lambda_1 \boldsymbol{q}_1 \boldsymbol{q}_1^T = \boldsymbol{Q}(\boldsymbol{T} - \lambda_1 \boldsymbol{Q}^T \boldsymbol{q}_1 \boldsymbol{q}_1^T \boldsymbol{Q})\boldsymbol{Q}^T = \boldsymbol{Q}(\boldsymbol{T} - \lambda_1 \boldsymbol{e}_1 \boldsymbol{e}_1^T)\boldsymbol{Q}^T.$$

Direct Matrix Reductions

Demo: *Householder Similarity Transforms*

- We can always compute an orthogonal similarity transformation to reduce a general matrix to *upper-Hessenberg* (upper-triangular plus the first subdiagonal) matrix $\boldsymbol{H}$, i.e. $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{H}\boldsymbol{Q}^T$:

  We can perform successive two-sided application of Householder reflectors

$$\boldsymbol{A} = \begin{bmatrix} h_{11} & a_{12} & \cdots \\ a_{21} & a_{22} & \\ \vdots & & \ddots \end{bmatrix} = \boldsymbol{Q}_1 \begin{bmatrix} h_{11} & a_{12} & \cdots \\ h_{21} & t_{22} & \cdots \\ \mathbf{0} & \vdots & \ddots \end{bmatrix} = \boldsymbol{Q}_1 \begin{bmatrix} h_{11} & h_{12} & \cdots \\ h_{21} & h_{22} & \cdots \\ \mathbf{0} & \vdots & \ddots \end{bmatrix} \boldsymbol{Q}_1^T = \cdots$$

  subsequent columns can be reduced by induction, so we can always stably reduce to upper-Hessenberg with roughly the same cost as QR.

- In the symmetric case, Hessenberg form implies tridiagonal:

  If $\boldsymbol{A} = \boldsymbol{A}^T$ then $\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{A}\boldsymbol{Q}^T = \boldsymbol{H}^T$, and a symmetric upper-Hessenberg matrix must be tridiagonal.

Simultaneous and Orthogonal Iteration

Demo: *Orthogonal Iteration*
Activity: *Orthogonal Iteration*

- *Simultaneous iteration* provides the main idea for computing many eigenvectors at once:
  - *Initialize* $\boldsymbol{X}_0 \in \mathbb{R}^{n \times k}$ *to be random and perform*

$$\boldsymbol{X}_{i+1} = \boldsymbol{A}\boldsymbol{X}_i.$$

  - *Observe that* $\lim_{i \to \infty} \text{span}(\boldsymbol{X}_i) = \mathbb{S}$ *where* $\mathbb{S}$ *is the subspace spanned by the* k *eigenvectors of* $\boldsymbol{A}$ *with the largest eigenvalues in magnitude.*
  - *Can use this to compute the right singular vectors of matrix* $\boldsymbol{M}$ *by using* $\boldsymbol{A} = \boldsymbol{M}^T\boldsymbol{M}$ *(no need to form* $\boldsymbol{A}$*, just multiply* $\boldsymbol{X}_i$ *by* $\boldsymbol{M}^T$ *then* $\boldsymbol{M}$*).*
  - *Small number of iterations suffice to obtain reasonable low-rank approximation of* $\boldsymbol{M}$*, and ultimately* $\boldsymbol{X}$ *converge to singular vectors in truncated SVD.*
- Orthogonal iteration performs QR at each step to ensure stability

$$\boldsymbol{Q}_{i+1}\boldsymbol{R}_{i+1} = \boldsymbol{A}\boldsymbol{Q}_i$$

  - $\boldsymbol{Q}_i$ *has the same span as* $\boldsymbol{X}_i$ *in orthogonal iteration.*
  - *QR has cost* $O(nk^2)$ *while product has cost* $O(n^2k)$ *per iteration.*

QR Iteration

- QR iteration reformulates orthogonal iteration for $n = k$ to reduce cost/step,
  - *Orthogonal iteration computes* $\hat{\boldsymbol{Q}}_{i+1}\hat{\boldsymbol{R}}_{i+1} = \boldsymbol{A}\hat{\boldsymbol{Q}}_i$
  - *QR iteration computes* $\boldsymbol{A}_{i+1} = \boldsymbol{R}_i\boldsymbol{Q}_i$ *at iteration* i
- Using induction, we assume $\boldsymbol{A}_i = \hat{\boldsymbol{Q}}_i^T\boldsymbol{A}\hat{\boldsymbol{Q}}_i$ and show that QR iteration obtains $\boldsymbol{A}_{i+1} = \hat{\boldsymbol{Q}}_{i+1}^T\boldsymbol{A}\hat{\boldsymbol{Q}}_{i+1}$
  - *QR iteration performs QR to obtain* $\boldsymbol{Q}_i\boldsymbol{R}_i = \boldsymbol{A}_i$
  - *Orthogonal iteration performs QR*

$$\hat{\boldsymbol{Q}}_{i+1}\hat{\boldsymbol{R}}_{i+1} = \boldsymbol{A}\hat{\boldsymbol{Q}}_i \underbrace{=}_{\text{inductive assumption}} \hat{\boldsymbol{Q}}_i\boldsymbol{A}_i = \underbrace{\hat{\boldsymbol{Q}}_i\boldsymbol{Q}_i}_{\hat{\boldsymbol{Q}}_{i+1}}\underbrace{\boldsymbol{R}_i}_{\hat{\boldsymbol{R}}_{i+1}}$$

consequently, we can observe that $\boldsymbol{R}_i = \underbrace{\boldsymbol{Q}_i^T\hat{\boldsymbol{Q}}_i^T}_{\hat{\boldsymbol{Q}}_{i+1}^T}\boldsymbol{A}\hat{\boldsymbol{Q}}_i$

  - *QR iteration performs product* $\boldsymbol{A}_{i+1} = \boldsymbol{R}_i\boldsymbol{Q}_i = \hat{\boldsymbol{Q}}_{i+1}^T\boldsymbol{A}\hat{\boldsymbol{Q}}_{i+1}$

QR Iteration with Shift

- QR iteration can be accelerated using shifting:

$$\boldsymbol{Q}_i\boldsymbol{R}_i = \boldsymbol{A}_i - \sigma_i\boldsymbol{I}$$
$$\boldsymbol{A}_{i+1} = \boldsymbol{R}_i\boldsymbol{Q}_i + \sigma_i\boldsymbol{I}$$

note that $\boldsymbol{A}_{i+1}$ is similar to $\boldsymbol{A}_i$, since we can reorganize the above as

$$\boldsymbol{R}_i\boldsymbol{Q}_i = \boldsymbol{Q}_i^T(\boldsymbol{A}_i - \sigma_i\boldsymbol{I})\boldsymbol{Q}_i,$$
$$\boldsymbol{Q}_i(\boldsymbol{A}_{i+1} - \sigma_i\boldsymbol{I})\boldsymbol{Q}_i^T = \boldsymbol{Q}_i\boldsymbol{R}_i,$$

and observe that $\boldsymbol{R}_i\boldsymbol{Q}_i$ is similar to $\boldsymbol{Q}_i\boldsymbol{R}_i$.

- The shift is typically selected to accelerate convergence with respect to a particular eigenvalue:

  We can select the shift as the bottom right element of $\boldsymbol{A}_i$, which would be the smallest eigenvalue if $\boldsymbol{A}_i$ is triangular (we have converged). Such shifting should accelerate convergence of the last column of $\boldsymbol{A}_i$, once finished we should operate only on the first $n-1$ columns, and so on.

QR Iteration Complexity

- QR iteration is accelerated by first reducing to upper-Hessenberg or tridiagonal form:

  Reduction to upper-Hessenberg or tridiagonal in the symmetric case, costs $O(n^3)$ operations and can be done in a similar style to Householder QR.

  Given an upper-Hessenberg matrix, $\boldsymbol{H}_i = \boldsymbol{A}_i$

  - *reduction to upper-triangular requires $n-1$ Givens rotations, if $\boldsymbol{G}_i$ rotates the $(i+1)$th row into the ith to eliminate the ith element on the first subdiagonal, $\boldsymbol{R}_i = \boldsymbol{G}_1^T \cdots \boldsymbol{G}_{n-1}^T \boldsymbol{H}_i$*
  - *computation of $\boldsymbol{H}_{i+1} = \boldsymbol{R}\boldsymbol{Q}$ can be done by application of the $n-1$ Givens rotations to $\boldsymbol{R}$ from the right $\boldsymbol{H}_{i+1} = \boldsymbol{R}_i \boldsymbol{G}_{n-1} \cdots \boldsymbol{G}_1$.*

  Both cost $O(n^2)$, for $O(n^3)$ overall if QR iteration converges in $O(n)$ steps.

  Given a tridiagonal matrix, the same two general steps are required, but now each step costs $O(n)$, so overall the eigenvalues and eigenvectors of a tridiagonal matrix can be computed with $O(n^2)$ work.

Solving Tridiagonal Symmetric Eigenproblems

A variety of methods exists for the tridiagonal eigenproblem:

- QR iteration *requires $O(1)$ QR factorizations per eigenvalue, $O(n^2)$ cost to get eigenvalues, $O(n^3)$ for eigenvectors. The last cost is not optimal.*
- Divide and conquer *reduces tridiagonal $\boldsymbol{T}$ by a similarity transformation to a rank-1 perturbation of identity, then computes its eigenvalues using roots of secular equation*

$$
\begin{aligned}
\boldsymbol{T} &= \begin{bmatrix} \boldsymbol{T}_1 & t_{n/2+1,n/2}\boldsymbol{e}_{n/2}\boldsymbol{e}_1^T \\ t_{n/2+1,n/2}\boldsymbol{e}_1\boldsymbol{e}_{n/2}^T & \boldsymbol{T}_2 \end{bmatrix} \\
&= \begin{bmatrix} \hat{\boldsymbol{T}}_1 & \\ & \hat{\boldsymbol{T}}_2 \end{bmatrix} + t_{n/2+1,n/2} \begin{bmatrix} \boldsymbol{e}_{n/2} \\ \boldsymbol{e}_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_{n/2}^T & \boldsymbol{e}_1^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q}_1\boldsymbol{D}_1\boldsymbol{Q}_1^T & \\ & \boldsymbol{Q}_2\boldsymbol{D}_2\boldsymbol{Q}_2^T \end{bmatrix} + \cdots \\
&= \begin{bmatrix} \boldsymbol{Q}_1 & \\ & \boldsymbol{Q}_2 \end{bmatrix} \Bigg(\underbrace{\begin{bmatrix} \boldsymbol{D}_1 & \\ & \boldsymbol{D}_2 \end{bmatrix} + t_{n/2+1,n/2} \begin{bmatrix} \boldsymbol{Q}_1^T\boldsymbol{e}_{n/2} \\ \boldsymbol{Q}_2^T\boldsymbol{e}_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_{n/2}^T\boldsymbol{Q}_1 & \boldsymbol{e}_1^T\boldsymbol{Q}_2 \end{bmatrix}}_{\boldsymbol{D}+\alpha\boldsymbol{u}\boldsymbol{u}^T} \Bigg) \begin{bmatrix} \boldsymbol{Q}_1^T & \\ & \boldsymbol{Q}_2^T \end{bmatrix}
\end{aligned}
$$

Solving the Secular Equation for Divide and Conquer

To solve the eigenproblem at each step, the divide and conquer method needs to diagonalize a rank-1 perturbation of a diagonal matrix

$$\boldsymbol{A} = \boldsymbol{D} + \alpha \boldsymbol{u}\boldsymbol{u}^T.$$

- *The zeros of the characteristic polynomial define the eigenvalues,*

$$f(\lambda) = \det(\boldsymbol{D} + \alpha \boldsymbol{u}\boldsymbol{u}^T - \lambda \boldsymbol{I}) = 1 + \alpha \boldsymbol{u}^T (\boldsymbol{D} - \lambda \boldsymbol{I})^{-1} \boldsymbol{u} = 1 + \alpha \sum_{i=1}^{n} \frac{u_i^2}{d_{ii} - \lambda} = 0.$$

- *This nonlinear equation can be solved efficiently by a variant of Newton's method (covered in the next chapter) that uses hyperbolic rather than linear extrapolations at each step.*
- *Major alternatives to divide and conquer include spectral bisection and the MRRR algorithm.*

Introduction to Krylov Subspace Methods

- *Krylov subspace methods* work with information contained in the $n \times k$ matrix

$$\boldsymbol{K}_k = \begin{bmatrix} \boldsymbol{x_0} & \boldsymbol{A x_0} & \cdots & \boldsymbol{A}^{k-1}\boldsymbol{x_0} \end{bmatrix}$$

We seek to best use the information from the matrix vector product results (columns of $\boldsymbol{K}_k$) to solve eigenvalue problems.

- The matrix $\boldsymbol{K}_n^{-1}\boldsymbol{A}\boldsymbol{K}_n$ is a *companion matrix* $\boldsymbol{C}$:

Letting $\boldsymbol{k}_n^{(i)} = \boldsymbol{A}^{i-1}\boldsymbol{x}$, we observe that

$$\boldsymbol{A}\boldsymbol{K}_n = \begin{bmatrix} \boldsymbol{A}\boldsymbol{k}_n^{(1)} & \cdots & \boldsymbol{A}\boldsymbol{k}_n^{(n-1)} & \boldsymbol{A}\boldsymbol{k}_n^{(n)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{k}_n^{(2)} & \cdots & \boldsymbol{k}_n^{(n)} & \boldsymbol{A}\boldsymbol{k}_n^{(n)} \end{bmatrix},$$

therefore premultiplying by $\boldsymbol{K}_m^{-1}$ transforms the first $n-1$ columns of $\boldsymbol{A}\boldsymbol{K}_n$ into the last $n-1$ columns of $\boldsymbol{I}$,

$$\begin{aligned} \boldsymbol{K}_n^{-1}\boldsymbol{A}\boldsymbol{K}_n &= \begin{bmatrix} \boldsymbol{K}_n^{-1}\boldsymbol{k}_n^{(2)} & \cdots & \boldsymbol{K}_n^{-1}\boldsymbol{k}_n^{(n)} & \boldsymbol{K}_n^{-1}\boldsymbol{A}\boldsymbol{k}_n^{(n)} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_n & \boldsymbol{K}_n^{-1}\boldsymbol{A}\boldsymbol{k}_n^{(n)} \end{bmatrix} \end{aligned}$$

Krylov Subspaces

- Given $\boldsymbol{Q}_k\boldsymbol{R}_k = \boldsymbol{K}_k$, we obtain an orthonormal basis for the Krylov subspace,

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{x}_0) = span(\boldsymbol{Q}_k) = \{p(\boldsymbol{A})\boldsymbol{x}_0 : deg(p) < k\},$$

  where p is any polynomial of degree less than k.

- The Krylov subspace includes the $k-1$ approximate dominant eigenvectors generated by $k-1$ steps of power iteration:
  - *The approximation obtained from $k-1$ steps of power iteration starting from $\boldsymbol{x}_0$ is given by the Rayleigh-quotient of $\boldsymbol{y} = \boldsymbol{A}^k\boldsymbol{x}_0$.*
  - *This vector is within the Krylov subspace, $\boldsymbol{y} \in \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{x}_0)$.*
  - *Consequently, Krylov subspace methods will generally obtain strictly better approximations of the dominant eigenpair than power iteration.*

Krylov Subspace Methods

- The $k \times k$ matrix $\boldsymbol{H}_k = \boldsymbol{Q}_k^T \boldsymbol{A} \boldsymbol{Q}_k$ minimizes $||\boldsymbol{A}\boldsymbol{Q}_k - \boldsymbol{Q}_k \boldsymbol{H}_k||_2$:
  The minimizer $\boldsymbol{X}$ for the linear least squares problem $\boldsymbol{Q}_k \boldsymbol{X} \cong \boldsymbol{A}\boldsymbol{Q}_k$ is (via the normal equations) $\boldsymbol{X} = \boldsymbol{Q}_k^T \boldsymbol{A} \boldsymbol{Q}_k = \boldsymbol{H}_k$.
- $\boldsymbol{H}_k$ is Hessenberg, because the companion matrix $\boldsymbol{C}_k$ is Hessenberg:

$$\boldsymbol{H}_k = \boldsymbol{Q}_k^T \boldsymbol{A} \boldsymbol{Q}_k = \boldsymbol{R}_k \boldsymbol{K}_k^{-1} \boldsymbol{A} \boldsymbol{K}_k \boldsymbol{R}_k^{-1} = \boldsymbol{R}_k \boldsymbol{C}_k \boldsymbol{R}_k^{-1}$$

  is a product of three matrices: upper-triangular $\boldsymbol{R}_k$, upper-Hessenberg $\boldsymbol{C}_k$, and upper-triangular $\boldsymbol{R}_k^{-1}$, which results in upper-Hessenberg $\boldsymbol{H}_k$.

Rayleigh-Ritz Procedure

Demo: *Arnoldi vs Power Iteration*
Activity: *Computing the Maximum Ritz Value*

- The eigenvalues/eigenvectors of $\boldsymbol{H}_k$ are the *Ritz values/vectors*:

$$\boldsymbol{H}_k = \boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^{-1}$$

eigenvalue approximations based on Ritz vectors $\boldsymbol{X}$ are given by $\boldsymbol{Q}_k\boldsymbol{X}$.

- The Ritz vectors and values are the *ideal approximations* of the actual eigenvalues and eigenvectors based on only $\boldsymbol{H}_k$ and $\boldsymbol{Q}_k$:

Assuming $\boldsymbol{A}$ is a symmetric matrix with positive eigenvalues, the largest Ritz value $\lambda_{max}(\boldsymbol{H}_k)$ will be the maximum Rayleigh quotient of any vector in $\mathcal{K}_k = span(\boldsymbol{Q}_k)$,

$$\max_{\boldsymbol{x}\in span(\boldsymbol{Q}_k)} \frac{\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} = \max_{\boldsymbol{y}\neq 0} \frac{\boldsymbol{y}^T\boldsymbol{Q}_k^T\boldsymbol{A}\boldsymbol{Q}_k\boldsymbol{y}}{\boldsymbol{y}^T\boldsymbol{y}} = \max_{\boldsymbol{y}\neq 0} \frac{\boldsymbol{y}^T\boldsymbol{H}_k\boldsymbol{y}}{\boldsymbol{y}^T\boldsymbol{y}} = \lambda_{max}(\boldsymbol{H}_k),$$

which is the best approximation to $\lambda_{max}(\boldsymbol{A}) = \max_{\boldsymbol{x}\neq 0} \frac{\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}}$ available in $\mathcal{K}_k$. The quality of the approximation can also be shown to be optimal for other eigenvalues/eigenvectors.

Arnoldi Iteration

Demo: *Arnoldi Iteration*
Demo: *Arnoldi Iteration with Complex Eigenvalues*

- Arnoldi iteration computes $\boldsymbol{H} = \boldsymbol{H}_n$ directly using the recurrence $\boldsymbol{q}_i^T \boldsymbol{A} \boldsymbol{q}_j = h_{ij}$, where $\boldsymbol{q}_l$ is the lth column of $\boldsymbol{Q}_n$:

  We have that

$$\boldsymbol{q}_i^T \boldsymbol{A} \boldsymbol{q}_j = \boldsymbol{q}_i^T (\boldsymbol{Q}_n \boldsymbol{H}_n \boldsymbol{Q}_n^T) \boldsymbol{q}_j = \boldsymbol{e}_i^T \boldsymbol{H}_n \boldsymbol{e}_j = h_{ij}.$$

- After each matrix-vector product, orthogonalization is done with respect to each previous vector:

  Given $\boldsymbol{u}_j = \boldsymbol{A} \boldsymbol{q}_j$, compute $h_{ij} = \boldsymbol{q}_i^T \boldsymbol{u}_j$ for each $i \leq j$, forming a column of the $\boldsymbol{H}$ matrix at a time.

Lanczos Iteration

- Lanczos iteration provides a method to reduce a symmetric matrix to a tridiagonal matrix:

  Arnoldi iteration on a symmetric matrix will result in an upper-Hessenberg matrix $\boldsymbol{H}_n$ as before, except that it must also be symmetric, since

  $$\boldsymbol{H}_n^T = (\boldsymbol{Q}_n^T \boldsymbol{A} \boldsymbol{Q}_n)^T = \boldsymbol{Q}_n^T \boldsymbol{A}^T \boldsymbol{Q}_n = \boldsymbol{Q}_n^T \boldsymbol{A} \boldsymbol{Q}_n = \boldsymbol{H}_n,$$

  which implies that $\boldsymbol{H}_n$ must be tridiagonal.

- After each matrix-vector product, it suffices to orthogonalize with respect to two previous vectors:

  Since $h_{ij} = 0$ if $|i - j| > 1$, given $\boldsymbol{u}_j = \boldsymbol{A}\boldsymbol{q}_j$, it suffices to compute only $h_{jj} = \boldsymbol{q}_j^T \boldsymbol{u}_j$ and $h_{j-1,j} = h_{j,j-1} = \boldsymbol{q}_{j-1}^T \boldsymbol{u}_j$.

Cost Krylov Subspace Methods

- The cost of matrix-vector multiplication when the matrix has m nonzeros *is m multiplications and at most m additions, so roughly $2m$ in total.*
- The cost of orthogonalization at the kth iteration of a Krylov subspace method is
  - *$O(nk)$ for k inner products in Arnoldi,*
  - *$O(n)$ in Lanczos, since only 2 orthogonalizations needed.*
  - *For Arnoldi with k-dimensional subspace, orthogonalization costs $O(nk^2)$, matrix-vector products cost $O(mk)$, so generally desire $nk < m$.*

Restarting Krylov Subspace Methods

- In finite precision, Lanczos generally loses orthogonality, while orthogonalization in Arnoldi can become prohibitively expensive:
  - *Arnoldi cost of orthogonalization dominates if $k > m/n$.*
  - *In Lanczos, reorthogonalizing iterate to previous guesses can ensure orthogonality.*
  - *Selective orthogonalization strategies control when and with respect to what previous columns of $\boldsymbol{Q}$, each new iterate $\boldsymbol{u}_j = \boldsymbol{A}\boldsymbol{q}_j$ should be orthogonalized.*

- Consequently, in practice, low-dimensional Krylov subspace methods are constructed repeatedly using carefully selected new starting vectors:

  If we wish to find a particular eigenvector isolate some eigenspaces, restarting is beneficial
  - *can orthogonalize to previous eigenvector estimates to perform deflation,*
  - *can pick starting vector as Ritz vector estimate associated with desired eigenpair,*
  - *given new starting vector, can discard previous Krylov subspace, which helps make storing the needed parts of $\boldsymbol{Q}$ possible.*

Generalized Eigenvalue Problem

- A generalized eigenvalue problem has the form $\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{B}\boldsymbol{x}$,

$$\boldsymbol{A}\boldsymbol{X} = \boldsymbol{B}\boldsymbol{X}\boldsymbol{D}$$
$$\boldsymbol{B}^{-1}\boldsymbol{A} = \boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^{-1}$$

  Generalized eigenvalue problems arise frequently, especially in solving partial differential equations.

- When $\boldsymbol{A}$ and $\boldsymbol{B}$ are symmetric and $\boldsymbol{B}$ is SPD, we can perform Cholesky on $\boldsymbol{B}$, multiply $\boldsymbol{A}$ by the inverted factors, and diagonalize it:

$$\boldsymbol{A}\boldsymbol{X} = \boldsymbol{L}\boldsymbol{L}^T\boldsymbol{X}\boldsymbol{D}$$
$$\underbrace{\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-T}}_{\tilde{\boldsymbol{A}}}\underbrace{\boldsymbol{L}^T\boldsymbol{X}}_{\tilde{\boldsymbol{X}}} = \underbrace{\boldsymbol{L}^T\boldsymbol{X}}_{\tilde{\boldsymbol{X}}}\boldsymbol{D}$$

- Alternative canonical forms and methods exist that are specialized to the generalized eigenproblem.