Gaussian Elimination:

I) LU Factorization: (no pivots): $A = LU$ by triangular-triangularization

① Algorithm: $U = A \quad L = I$

for $k = 1$ to $m-1$
  for $j = k+1$ to m
    $l_{jk} = u_{jk} / u_{kk}$
    $u_{j,k:m} = u_{j,k:m} - l_{jk} u_{k,k:m}$

Work: $\Theta(\frac{2}{3} m^3)$

② Solution: $Ax = b$, $LUx = b$ → ① Solve $Ly = b$ for y; ② Solve $Ux = y$ for x

* This is not backward stable *

II) Pivoting: the diagonal entry is called the Pivot

① Complete Pivoting: find largest entry left in matrix and pivot around that → move colums, rows, etc → huge cost!

② Partial Pivoting: only pivot around the rows; ie: $L_{m-1} P_{m-1} \cdots L_2 P_2 L_1 P_1 A = U$

③ $PA = LU$ ① Permute row of A by P
② Apply GE w/o pivoting to PA

④ Algorithm: $U = A,\ L = I,\ P = I$

for $k = 1$ to $m-1$
  Select $i \geq k$ to max $|u_{ik}|$
  $u_{k,k:m} \leftrightarrow u_{i,k:m}$ (interchange 2 rows)
  $l_{k,1:k-1} \leftrightarrow l_{i,1:k-1}$
  $P_{k,:} \leftrightarrow P_{i,:}$
  for $j = k+1$ to m
    $l_{jk} = u_{jk} / u_{kk}$
    $u_{j,k:m} = u_{j,k:m} - l_{jk} u_{k,k:m}$

Work: $\sim \Theta(\frac{2}{3} m^3)$

III) Stability of G.E.: * not stable in general... BUT FAST

IV) Cholesky Factorization: * For Hermitian matrices * → Every Hermitian matrix has unique Cholesky factorization

① Basically use a "symmetric version of LU" to factor into Cholesky

② Idea:
$$A = \begin{bmatrix} a_{11} & w^* \\ w & K \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ w/\alpha & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & K - ww^*/a_{11} \end{bmatrix} \begin{bmatrix} \alpha & w^*/\alpha \\ 0 & I \end{bmatrix} \quad \text{where } \alpha = \sqrt{a_{11}}$$
K ↑ Hermitian
$$= R_1^* A_1 R_1$$

③ Work: $\sim \Theta(\frac{1}{3} m^3)$ * Stable and fast *

④ Algorithm: $Ax = b$
$R^* R x = b$
① Solve $R^* y = b$ for y
② Solve $Rx = y$ for x
$\Big\} \sim \Theta(m^2)$

Stability and Facts

I) Norms: $\|A\|_1 =$ max column sum $\qquad \|A\|_2 = \sqrt{\text{largest eigenvalue of } A^*A} =$ largest singular value

$\|A\|_\infty =$ max row sum

II) Stability:

1. Backward Stability: "A backward stable algorithm gives exactly the right answer to nearly the right question"
2. Condition Number: $K = \|A\| \|A^{-1}\|$; if $\|\cdot\|_2$ then $K = \frac{\sigma_1}{\sigma_m}$
3. Backward Stable Algorithms:
   1. QR for $Ax=b$ using Householder Reflectors
   2. Back substitution
   3. Least Squares w/
      a) Householder for QR
      b) QR w/ Gram-Schmidt if Q^*b is formed implicitly
      c) SVD w/ full rank

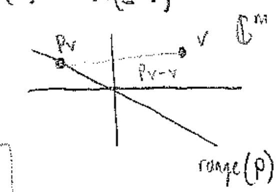
QR Factorization and Least Squares

I) Projectors: $P \in \mathbb{C}^{m\times m}$, then if $P^2 = P$, P is a projector. $\longrightarrow$ if $v \in \text{range}(P)$, then if $v = Px \Rightarrow Pv = P^2x = Px = v$

① $I-P$ is complimentary projector of $P \rightarrow I-P$ projects onto null(P) : range$(I-P)$ = null(P) and range(P) = null$(I-P)$

② A projector P is orthogonal iff $P = P^*$

③ if range$(P) = S_1$ and range$(I-P) = S_2$, then $S_1 + S_2 = \mathbb{C}^m$



II) QR Factorization: $A = QR$; Q: orthogonal matrix, R: upper Δ ; $A = [a_1\ a_2\ a_3 \cdots] = [q_1\ q_2 \cdots q_n]\begin{bmatrix} r_{11} & r_{12} & \cdots \\ & r_{22} & \cdots \\ 0 & & \ddots \end{bmatrix}$

① Gram-Schmidt: ① Classical (unstable) Algorithm : for $j=1$ to n
↑ due to rounding errors

$v_j = a_j$
for $i=1$ to $j-1$
$r_{ij} = q_i^* a_j$
$v_j = v_j - r_{ij}q_i$
$r_{jj} = \|v_j\|$
$q_j = v_j / r_{jj}$

* Every $A \in \mathbb{C}^{m\times n}$ $(m \geq n)$ has a full QR factorization and hence reduced *

* Each $A \in \mathbb{C}^{m\times n}$ $(m \geq n)$ of full rank has unique reduced QR factorization w/ $r_{jj} > 0$ *

② Solution of $Ax = b$ via QR : $Ax = b$; let $A = QR$

$Q^*A = Q^*QR = R$

$\Rightarrow Q^*Ax = Q^*b$

$Rx = Q^*b$

III) Modified Gram-Schmidt: "triangular orthogonalization" $\rightarrow$ making columns of a matrix orthonormal

① Algorithm: for $i=1$ to n
$v_i = a_i$
for $i=1$ to n
$r_{ii} = \|v_i\|$
$q_i = v_i / r_{ii}$
for $j = i+1$ to n
$r_{ij} = q_i^* v_j$
$v_j = v_j - r_{ij}q_i$

$\sim O(2mn^2)$ operations

$\sim \underbrace{AR_1R_2R_3 \cdots R_n}_{\hat{R}^{-1}} = \hat{Q}$

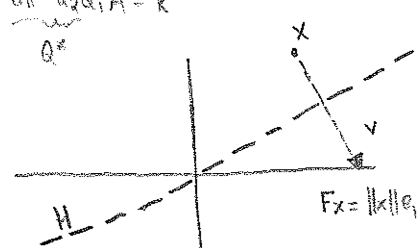
* • instead of computing a single orthogonal projector of rank $m-(j-1)$, MGS uses a sequence of $j-1$ projections of rank $m-1$. *

IV) Householder Triangularization: "orthogonal triangularization" $\rightarrow$ apply sequence of unitary matrix operations : $\underbrace{Q_n \cdots Q_2 Q_1}_{Q^*} A = R$

① Process:

$$\underset{A}{\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix}} \xrightarrow{Q_1} \underset{Q_1A}{\begin{bmatrix} \tilde{x} & \tilde{x} & \tilde{x} & \tilde{x} \\ 0 & \tilde{x} & \tilde{x} & \tilde{x} \\ 0 & \tilde{x} & \tilde{x} & \tilde{x} \\ 0 & \tilde{x} & \tilde{x} & \tilde{x} \end{bmatrix}} \xrightarrow{Q_2} \underset{Q_2Q_1A}{\begin{bmatrix} \tilde{x} & \tilde{x} & \tilde{x} & \tilde{x} \\ 0 & \tilde{x} & \tilde{x} & \tilde{x} \\ 0 & 0 & \tilde{x} & \tilde{x} \\ 0 & 0 & \tilde{x} & \tilde{x} \end{bmatrix}}$$

Q_1A (nonzero elements change) Q_2Q_1A (non-zero elements change)



② Householder Reflector: let $Q_k = \begin{bmatrix} I & 0 \\ 0 & F \end{bmatrix}$; I is $(k-1)\times(k-1)$ identity and $F = I - 2\dfrac{vv^*}{v^*v}$

• F reflects the space $\mathbb{C}^{m-k+1}$ across H orthogonal to $v = \|x\|e_1 - x$

③ Work: for $A \in \mathbb{C}^{m\times n}$: $\sim O\left(2mn^2 - \frac{2}{3}n^3\right)$

V) **Least-Squares:** for overdetermined systems → more rows than columns in $Ax=b$ ($A \in \mathbb{C}^{m\times n}$ $(m>n)$) → minimize 2-Norm of residual

① Orthogonal Projection / Normal Eqs: • find closest Ax in range(A) so $\|b-Ax\|_2$ is minimized

• $r = b - Ax$ must be orthogonal to range(A) ⇝ geometrically occurs provided $Ax = Pb$, where P is orthogonal projector that maps $\mathbb{C}^m$ onto range(A)

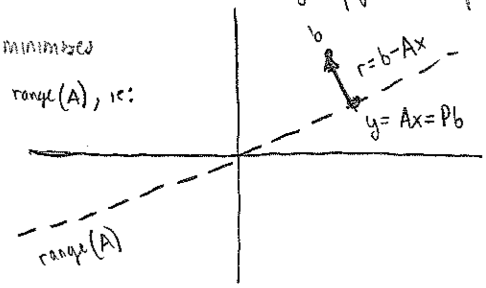
② Theorem: Let $A \in \mathbb{C}^{m\times n}$ $(m \geq n)$ and $b \in \mathbb{C}^m$. The $x \in \mathbb{C}^m$ that minimizes $\|r\| = \|b-Ax\|_2$ is such that if and only if $r \perp$ range(A), ie:

$$A^* r = 0$$

$$\longrightarrow \quad A^*A x = A^* b$$

$$\longrightarrow \quad x = (A^*A)^{-1} A^* b$$

← Pseudo-inverse



③ Solving w/ Cholesky: if A full rank → $A^*A = R^*R$, R: upper-Δ hence eqs become: $A^*Ax = A^*b$

$R^*Rx = A^*b$

① Algorithm: ① form A^*A and A^*b

② Compute Cholesky factorization

③ Solve Lower Δ sys: $Rw = A^*b$ for w

④ Solve Upper Δ sys: $R^*x = w$ for x

work: $\sim \theta(mn^2 + \frac{n^3}{3})$

* reduces to 2 Δ sys of eqs *

④ Solving w/ QR: using reduced $\hat{Q}\hat{R}$, $y = Pb = \hat{Q}\hat{Q}^*b$ since $y \in$ range(A), sys. $Ax = y$ has exact solution

$$Ax = y$$
$$\hat{Q}\hat{R}x = \hat{Q}\hat{Q}^*b$$
$$\hat{R}x = \hat{Q}^*b$$

work: $\sim \theta(2mn^2 - \frac{2}{3}n^3)$

* reduces to Δ-sys of eqs *

⑤ Solving w/ SVD: $A = \hat{U}\hat{\Sigma}V^*$, let $P = \hat{U}\hat{U}^*$

$$y = Pb = \hat{U}\hat{U}^*b$$

work: $\sim \theta(2mn^2 + 11n^3)$

* reduces to diagonal sys of eqs *

$$\longrightarrow \quad Ax = y$$
$$\hat{U}\hat{\Sigma}V^*x = \hat{U}\hat{U}^*b$$
$$\hat{\Sigma}V^*x = \hat{U}^*b$$

Algorithm: ① Compute reduced SVD: $A = \hat{U}\hat{\Sigma}V^*$

② Compute $\hat{U}^*b$

③ Solve diagonal sys: $\hat{\Sigma}w = \hat{U}^*b$ for w

④ Let $x = Vw$

⑥ Comparison:

① Speed → Cholesky

② Stability → QR (if A is not rank deficient)

③ Rank-Deficient → SVD

Eigenvalues

I) Eigenvalue Problem:

① Geometric Multiplicity: # of linearly independent eigenvectors for the same λ. This is also the dimension of the null-space of $A - \lambda I$

② Theorem: λ is eigenvalue $\iff \exists$ non-zero vector such that $\lambda x - Ax = 0 \iff \lambda I - A$ is non-singular $\iff |\lambda I - A| = 0$

③ Algebraic Multiplicity: multiplicity of the root, λ, for the characteristic polynomial

④ Similarity Transformation: If $X \in \mathbb{C}^{m \times m}$ is non-singular, the map $A \to X^{-1}AX$ is similarity transform

• Theorem: If X is non-singular, then A and $X^{-1}AX$ have the same eigenvalues, charateristic poly, geo. and alg. multiplicity

⑤ $\det(A) = \prod_{j=1}^{m} \lambda_j$ and $\operatorname{tr}(A) = \sum_{j=1}^{m} \lambda_j$ Note: algebraic multiplicity $\geq$ geo multiplicity

⑥ Decompositions:

① Diagonalizable: $A = X \Lambda X^{-1}$ ← A is non-defective (alg mult = geometric multiplicity)

② Unitary Diagonalizable: $A = Q \Lambda Q^*$ ← this is both an Eigenvalue decomp and SVD

Note: A matrix is unitary diagonalizable iff it is normal, ie: $A^*A = AA^*$

③ Schur Factorization: $A = QTQ^*$; $T \to$ upper triangular $\iff$ eigenvalues are along main diagonal ; * EVERY MATRIX *

II) Overview: An eigenvalue solver must be iterative

① most eigenvalue solvers generally compute a Schur Factorization

② Two Phases:

① Phase 1: $A \to$ Hessenberg Form (or tridiagonal if $A = A^*$) total work $\sim \Theta(m^3)$

② Phase 2: Hessenberg (tridiagonal) → Upper triangular (diagonal if $A = A^*$) total work $\sim \Theta(m^2)$ ... but could run ∞

III) Reduction to Hessenberg:

• Select a householder reflector Q_1^* that leaves first row unchanged and multiply it on right and left of A:

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

then just keep apply $Q_k^* \cdots Q_1^* A Q_1 Q_2 \cdots Q_k$

• operation count: $\sim \frac{10}{3} m^3$

• Note if A is Hermitian $\sim \frac{4}{3} m^3$

• This process is backward stable

IV) Rayleigh Quotient, Inverse Iteration: Real, symmetric matrices $\{\lambda_j\}$ - eigenvalues, $\{q_j\}$ eigenvectors (orthonormal)

① Rayleigh Quotient: $r(x) = \dfrac{x^T A x}{x^T x}$; if x is an eigenvector then $r(x) = \lambda$; Note: • $\nabla r(x) = \dfrac{2}{x^T x}\left(Ax - r(x)x\right) = 0$ if x is an eigenvector

* Rayleigh Quotient is quadratically accurate estimate of eigenvalues * • Eigenvectors are stationary pts of $r(x)$

② Power Iteration: $v^{(0)} =$ some v w/ $\|v\| = 1$

For $k = 1, 2, \ldots$

$w = Av^{(k-1)}$

$v^{(k)} = w / \|w\|$

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

• Finds eigenvector associated w/ largest eigenvalue

• If two eigenvalues are close in magnitude → convergence is slow

• Quadratic convergence

③ Inverse Iteration: $v^{(0)} =$ some v w/ $\|v\| = 1$

For $k = 1, 2, \ldots$

costs $\Theta(m^2)$ if done once fully → Solve $(A - \mu I) w = v^{(k-1)}$

linear convergence → $v^{(k)} = w / \|w\|$

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

quadratic convergence

• μ is eigenvalue estimate

• Note if we use $\mu = \lambda^{(k)}$ for each step becomes Rayleigh Quotient Iteration

• Cubic convergence → * this costs $\Theta(m^3)$ at each step *

V) QR Algorithm w/ Shifts: stable procedure for computing QR factorizations of $A, A^2, A^3, \ldots$ $\longrightarrow$ gives cubic convergence

* NOTE: STILL ASSUMING REAL | SYMMETRIC MATRICES *

① **Algorithm:** $(Q^{(0)})^T A^{(0)} Q^{(0)} = A$; $A^{(0)}$ is tridiagonalization of A

for $K = 1, 2, 3, \ldots$

pick $\mu^{(k)}$: $\mu^{(k)} = A_{mm}^{(k-1)}$ ← value

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

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

$\Bigg\}$ Pure QR Algorithm

② **Simultaneous Iteration:** **Algorithm:** Pick $\hat{Q}^{(0)} \in \mathbb{R}^{m \times m}$ (orthonormal column → usually pick I)

for $K = 1, 2, \ldots$

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

$\hat{Q}^{(k)} \hat{R}^{(k)} = Z$ reduced QR of Z * column space of $\hat{Q}^{(k)}$ and $Z^{(k)}$ are equal and equal to $A^k \hat{Q}^{(0)}$

③ Simultaneous Iteration $\Longleftrightarrow$ QR algorithm applied to full set of $n = m$ vectors (Identity)

④ **Theorem:** Simultaneous Iteration: $\underline{Q}^{(0)} = I$

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

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

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

QR: $A^{(0)} = A$

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

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

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

both: $\underline{R}^{(k)} = R^{(k)} R^{(k-1)} \cdots R^{(1)}$

* These two processes generate identical sequences of matrices $\underline{R}^{(k)}$, $\underline{Q}^{(k)}$, and $A^{(k)}$, namely those defined by the QR factorization of the K^{th} power of A,

$A^k = \underline{Q}^{(k)} \underline{R}^{(k)}$ $\longleftrightarrow$ constructs orthonormal bases for powers of A

together w/ the projection,

$A^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}$ ⟵ Why it finds eigenvalues → diagonal elements of $A^{(k)}$ are Rayleigh Quotients corresponding to columns of $\underline{Q}^{(k)}$

* Converges linearly *

VI) QR w/ Shifts:

Krylov Subspaces

I) Overview:

1) **Krylov Subspace:** Set of vectors $b, Ab, A^2b, \ldots,$

2) **Table:**

	$Ax=b$	$Ax=\lambda x$
$A=A^*$	Conjugate Gradients	Lanczos
$A \neq A^*$	GMRES BICG	Arnoldi

* Conjugate Gradients needs A to be symmetric positive definite as well

3) **Idea:** Projection of A into Krylov subspace to reduce to a sequence of matrix problems of dimensions $n=1,2,\ldots$

Note: • If A is hermitian $\longrightarrow$ reduced matrices are tridiagonal

• If A is not $\longrightarrow$ Hessenberg Form

• These methods typically reduce order of the problem, i.e. $\mathcal{O}(n^3) \to \mathcal{O}(n^2)$

II) Arnoldi Iteration:

Gram-Schmidt style iteration for transforming a matrix to Hessenberg form for finding eigenvalues/eigenvectors

Recall: Two steps to find eigenvalues iteratively: ① Reduce to Hessenberg (or tridiagonal)
② Take that matrix and reduce to upper triangular (or diagonal) } Schur Form

① Computing a Hessenberg reduction: $A = QHQ^*$ $\Leftarrow$ Householder reflectors / Arnoldi Iteration

② Orthogonal Similarity: $A=QHQ^*$ or $AQ=QH$; $A \in \mathbb{C}^{m\times m}$

Let $Q_n =$ matrix of first n-columns of Q ; $Q \in \mathbb{C}^{m\times n}$; $\tilde{H}_n = \begin{bmatrix} h_{11} & \cdots & \cdots & h_{1,n} \\ h_{21} & h_{22} & \cdots & \vdots \\ & 0 & \ddots & \\ & & & h_{n+1,n} \end{bmatrix}$

$$\Rightarrow \quad \underbrace{AQ_n}_{m\times n} = \underbrace{Q_{n+1}\tilde{H}_n}_{m\times n}$$

Note: $Aq_n = h_{1n}q_1 + h_{2n}q_2 + \ldots + h_{nn}q_n + h_{n+1,n}q_{n+1}$ $\leftarrow$ q_{n+1} depends on itself and previous Krylov vecs

③ **Algorithm:**

$b =$ anything, $q_1 = b/\|b\|$
for $n=1,2,\ldots$
  $v = Aq_n$
  for $j=1,2,\ldots n$
    $h_{jn} = q_j^* v$
    $v = v - h_{jn}q_j$ } Gram-Schmidt
  $h_{n+1,n} = \|v\|$
  $q_{n+1} = v/h_{n+1,n}$

clearly $q_1,\ldots,q_n$ form basis of $\mathcal{K}_n = \langle b, Ab, \ldots, A^{n-1}b\rangle$
$= \langle q_1, q_2, \ldots, q_n\rangle \subseteq \mathbb{C}^m$

* ④ In a nutshell, the Arnoldi process is the systematic construction of orthonormal bases for successive Krylov subspaces *

or

A computation of projections onto successive Krylov subspaces

⑤ Ritz Values: Eigenvalues of H_n are called Arnoldi eigenvalue estimates and some are good approximations of the eigenvalues for A

⑥ **Theorem**: The matrices Q_n generated by Arnoldi are ① reduced QR factors of the Krylov matrix, $K_n = Q_n R_n$,

② the Hessenberg matrices H_n are projections, $H_n = Q_n^* A Q_n$

③ Successive iterates via, $AQ_n = Q_{n+1}\tilde{H}_n$

III) How Arnoldi Locates E-vals:

Note: Ritz values (e-vals of H_n) of converge rapidly are usually the extreme eigenvalues of A (e-vals near edge of spectrum of A)

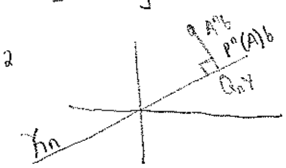
① Arnoldi and Polynomial Approximation: $x \in K_n \Rightarrow x = \sum_k c_k A^k b = q(A)b$ ← some poly. times b

- Arnoldi/Lanczos Approx Problem: Find p^n such that $\|p^n(A)b\|_2$ is a minimum ; $p^n \in P^n = \{$monic polys of deg $n\}$ (coeff 1 on x^n)

- Theorem: If K_n has full rank during Arnoldi iteration, $\|p^n(A)b\|_2 = \text{minimum}$ has unique solution → the characteristic polynomial of H_n

- Least Squares: If $p \in P^n$ $\quad p(A)b = A^n b - Q_n y$ for some $y \in \mathbb{C}^n$ and $Q_n = [q_1 \cdots q_n]$ (first n-columns of Q)

  → Find the point in K_n closest to $A^n b$, ie: $\min\|A^n b - Q_n y\|_2$



② How to find e-vals:

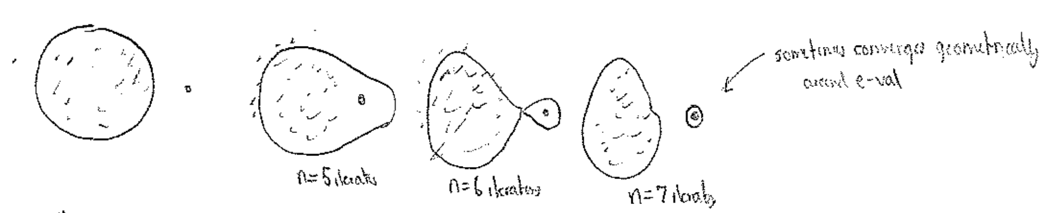
- Extreme Case: A is diagonalizable and has $n \ll m$ distinct eigenvalues, hence minimal degree n poly, then after n steps of Arnoldi all e-vals should be found exactly by getting characteristic poly of H_n from $p^n(A)$.

- Practical Case: We do same thing but get Ritz values and e-vals are approximate

③ Arnoldi Lemniscates (a curve $z \in \mathbb{C}$, $|p(z)| = C$) ⟶ Arnoldi: $C = \frac{\|p^n(A)b\|}{\|b\|}$

- as n increases (more iterations), lemniscates surround themselves around extreme e-vals then shrink rapidly around them

ie:



IV) GMRES: "generalized minimal residuals" ⟶ solving $Ax = b$

Note: exact solution $x_* = A^{-1}b$ **Idea**: At step n, approx x_* by $x_n \in K_n$ that minimizes norm of $r_n = b - Ax_n$ ← *Least Squares*

① Least-Squares: • $K_n = m \times n$ Krylov matrix ⟶ $AK_n = \begin{bmatrix} | & & & | \\ Ab, & A^2b, & \ldots, & A^n b \\ | & & & | \end{bmatrix}$; column space is AK_n

- wish to solve $\|AK_n c - b\|_2 = \min$ ⟶ ① could use QR for AK_n ② Once find c, $x_n = K_n c$ } QR is unstable unfortunately for this

- Use Arnoldi!
  a) use Arnoldi to construct sequence of Krylov Q_n's ; recall $q_1, q_2, \ldots, q_n$ span K_n
  b) Hence the least squares we solve is: $\|AQ_n y - b\| = \min$ then $x_n = Q_n y$
  c) Doing substitution, $AQ_n = Q_{n+1}\tilde{H}_n$ $\quad \|Q_{n+1}\tilde{H}_n y - b\| = \min$
  $\|\tilde{H}_n y - Q_{n+1}^* b\| = \min$; $Q_{n+1}^* b = \|b\| \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

⟹ **Final Least Squares**: $\|\tilde{H}_n y - \|b\| e_1\|_2 = \min$

② Mechanics of GMRES: ① Algorithm: $q_1 = b/\|b\|$

for $n=1,2,3,\dots$

[step n of Arnoldi]

Find y to minimize $\|\tilde{H}_n y - \|b\| e_1\|_2$

$x_n = Q_n y$

• each step minimizes the residual over all vectors $x_n \in \mathcal{K}_n$

• can solve least-squares step via QR factorization in $\theta(n^2)$

③ GMRES and Poly Approximation: Consider $P_n = \{\text{polys of deg} \le n, \text{ with } p(0)=1\}$

• $x_n = (A^0 + c_1 A + c_2 A^2 + \dots + c_{n-1}A^{n-1}) b = q_n(A) b$; note: the coeffs c_k are from $\vec{c}$ ← solution to unstable method of least-squares

• now residual, $r_n = b - Ax_n = [I - Aq_n(A)]b = p_n(A)b$

• Hence GMRES process chooses coeffcts of p_n to minimize the norm of the residual.

④ Convergence of GMRES: ① GMRES converges monotonically $\|r_{n+1}\| \le \|r_n\|$ (Note: $P_n \subseteq P_{n+1}$ but $P^n \not\subseteq P^{n+1}$)

② After m steps (recall $A \in \mathbb{C}^{m\times m}$), the process must converge (in absence of rounding error): $\|r_m\| = 0$

③ $\|r_n\| = \|p_n(A)b\| \le \|p_n(A)\|\,\|b\| \Rightarrow \dfrac{\|r_n\|}{\|b\|} \le \inf_{p_n \in P_n} \|p_n(A)\|$

V) Lanczos Iteration: Basically Arnoldi but for Hermitian matrices

① Simplifications: ① H_n is real and symmetric $\Rightarrow$ it is tridiagonal

② Ritz values are also real

③ Much cheaper than Arnoldi ⟶ instead of $(n+1)$ recurrence we have 3-term

② Iteration: Define $H_n = T_n = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \beta_2 & \\ & \beta_2 & \ddots & \ddots \\ & & \ddots & \beta_{n-1}\ \alpha_n \end{bmatrix}$

Algorithm: $\beta_0 = 0$, $q_0 = 0$, b = arbitrary, $q_1 = b/\|b\|$

for $n=1,2,\dots$

$v = Aq_n$

$\alpha_n = q_n^T v$

$v = v - \beta_{n-1} q_{n-1} - \alpha_n q_n$;

$\beta_n = \|v\|$

$q_{n+1} = v/\beta_n$

Note: $K_n = Q_n R_n$, $T_n = Q_n^* A Q_n \Rightarrow AQ_n = Q_{n+1}\tilde{T}_n$

$Aq_n = \beta_{n-1} q_{n-1} + \alpha_n q_n + \beta_{n+1} q_{n+1}$

VI) Lanczos and Gauss Quadrature:

① Orthogonal Polys on $[-1,1]$: $\beta_0 = 0$, $q_0(x) = 0$, $q_1(x) = 1/\sqrt{2}$

for $n=1,2,\dots$

$v(x) = xq_n$

$\alpha_n = \langle q_n, v\rangle$

$v = v - \beta_{n-1} q_{n-1}(x) - \alpha_n q_n(x)$

$\beta_n = \|v\|$

$q_{n+1} = v(x)/\beta_n$

* Produces Legendre Polys *

* Note: α_n's $= 0$, $\beta_n = \frac{1}{2}\left(1 - \frac{1}{4n^2}\right)^{-1/2}$

② Jacobi Matrices: $K_n = \begin{bmatrix} 1 & x & x^2 & \cdots & x^{n-1} \end{bmatrix}$ $Q_n = \begin{bmatrix} q_1(x) & q_2(x) & \cdots & q_n(x) \end{bmatrix}$

Jacobi Matrix ↙ $T_{ij} = \langle q_i, xq_j \rangle$

③ Characteristic Poly: For special choice of b and A $p(A)b = p(x)/\sqrt{2}$ hence we wish to find $\|p^n(x)\|$ = minimum ← coeff of 1 on highest degree

Theorem: Let $\{q_n(x)\}$ be sequence of orthogonal polys generated by iteration, $\{T_n\}$ is sequence of tridiag Jacobi matrices, p^n is characteristic poly of T_n. Then for $n=1,2,\dots$ $p^n(x) = C_n q_{n+1}(x)$, $C_n \in \mathbb{R}$, the zeros of $q_{n+1}(x)$ are eigenvalues of T_n ⟶ are distinct and lie in $(-1,1)$

④ Quadrature Formulas: Zeros of Legendre Polys are nodes of Gauss-Legendre quadrature formulas ; $I_n(f) = \sum_{j=1}^{n} \omega_j f(x_j) \approx \int_{-1}^{1} f(x)\,dx$

1) **Gauss-Quad Theorem:** $\{x_j\}$ set of special n distinct pts in $[-1,1]$, then $\exists$ unique choice of $\{\omega_j\}$ such that quadrature is exact for any poly of degree $\leq 2n-1$

* No quad formula can do better than $2n-1$, Newton Cotes are $n-1$ *

2) **Gauss-Quad Thm 2:** Let T_n be $n\times n$ Jacobi matrix from iteration, Let $T_n = VDV^T$ be orthogonal diagonalization, then nodes and weights of Gauss-Quad are given by:

$$x_j = \lambda_j$$

$$\omega_j = 2(v_j)_1^2 \quad \text{"2 times square of } 1^{st} \text{ component of } v_j\text{"} \quad \forall\ j=1,2,\dots,n$$

VII) Conjugate Gradients: Solve $Ax=b$ for symmetric positive definite systems

① Minimizing A-Norm: • $x^TAx > 0 \quad \forall$ nonzero $x \in \mathbb{R}^m$; define $\|x\|_A = \sqrt{x^TAx}$

• In a nutshell, conjugate gradient system of recursive formulas that generate unique sequence of $x_n \in K_n$ such that $\|e_n\|_A$ is minimized at each n.

② Algorithm: $x_0 = 0$, $r_0 = b$, $p_0 = r_0$

for $n = 1,2,\dots$

$\alpha_n = \dfrac{r_{n-1}^T r_{n-1}}{p_{n-1}^T A p_{n-1}}$; step length

$x_{n+1} = x_n + \alpha_n p_{n-1}$; approx solution

$r_n = r_{n-1} - \alpha_n A p_{n-1}$; residual

$\beta_n = \dfrac{r_n^T r_n}{r_{n-1}^T r_{n-1}^T}$; improvement this step

$p_n = r_n + \beta_n p_{n-1}$; search direction

Theorem: ① Identities of Subspaces:

$$X_n = \langle x_1, x_2, \dots, x_n\rangle = \langle p_0, p_1, \dots, p_{n-1}\rangle$$
$$= \langle r_0, r_1, \dots, r_{n-1}\rangle = \langle b, Ab, A^2b, \dots A^{n-1}b\rangle$$

② Residual are orthogonal: $r_n^T r_j = 0 \quad j<n$

③ Search direction "A-conjugate" $p_n^T A p_j = 0 \quad j<n$

③ Optimality of CG:

① Theorem: Apply CG to SPD $Ax=b$, if iteration has not converged then x_n is unique pt in K_n that minimizes $\|e_n\|_A$. The convergence is monotonic, $\|e_n\|_A \leq \|e_{n-1}\|_A$ and $e_n = 0$ for some $n < m$

Proof: Consider arbitrary pt. $x = x_n - \Delta x \in K_n$ then $e = x_* - x = x_* + \Delta x$

$$\|e\|_A^2 = (e_n+\Delta x)^T A (e_n + \Delta x) = e_n^T A e_n + (\Delta x)^T A (\Delta x) + \underbrace{2e_n^T A \Delta x}_{=0 \text{ by CG}}$$

$$= e_n^T A e_n + (\Delta x)^T A (\Delta x)$$

hence b/c A is positive def, $(\Delta x)^T A (\Delta x) > 0$ so error is minimized iff $\Delta x = 0$

② Optimization Algorithm: given A and b. and $x \in \mathbb{R}^m$, $\phi(x) = \frac{1}{2}x^TAx - x^Tb$

$$\|e_n\|_A^2 = e_n^T A e_n = (x_*-x_n)^T A (x_* - x_n)$$
$$= x_n^T A x_n - 2x_n^T A x_* + x_*^T A x_* \quad ; \quad Ax_* = b$$
$$= x_n^T A x_n - 2x_n^T b + x_*^T b$$
$$= 2\phi(x) + \text{constant}$$

Hence conjugate gradient can be seen as minimizing $\phi(x)$, $x \in \mathbb{R}^m$, where it achieves its minimum uniquely at x_*

④ Rate of Convergence:

① If A has only n-distinct e-vals, then CG converges in at most n steps

② If $\|A\|_2 = K$, then $\dfrac{\|e_n\|_A}{\|e_0\|_A} \leq 2\left(\dfrac{\sqrt{K}-1}{\sqrt{K}+1}\right)^n$

Lanczos Iteration:

- symmetric matrices to reduce to tridiagonal for eigenvalue solvers

(I) $AQ_n = Q_{n+1}T$; $T = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \beta_2 & \\ & \beta_2 & \ddots & \ddots \\ & & \ddots & \ddots \end{bmatrix}$

$$A\begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & \cdots & q_{n+1} \end{bmatrix} \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \beta_2 & \\ & \beta_2 & \ddots & \ddots \\ & & \ddots & \ddots \end{bmatrix}$$

$\underline{n=1}$: $Aq_1 = \alpha_1 q_1 + \beta_1 q_2$

$n=2$: $Aq_2 = \beta_1 q_1 + \alpha_2 q_2 + \beta_2 q_3$

$\vdots$

n: $Aq_n = \beta_{n-1} q_{n-1} + \alpha_n q_n + \beta_{n+1} q_{n+1}$

(II) Algorithm:

$\beta_0 = 0$, $\alpha_0 = 0$, $q_1 = b/\|b\|$

for $n = 1 \ldots$

$v = Aq_n$

$\alpha_n = q_n^T v$

$v = v - \alpha_n q_n - \beta_{n-1} q_{n-1}$

$\beta_n = \|v\|$

$q_{n+1} = v/\beta_n$

Arnoldi Iteration

- non-symmetric matrices

(I) $AQ_n = Q_{n+1}\tilde{H}_n$

$$A\begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix} = \begin{bmatrix} q_1 & \cdots & q_{n+1} \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ \vdots & & & \\ h_{n1} & \cdots & \cdots & h_{nn} \end{bmatrix}$$

$\underline{n=1}$: $Aq_1 = h_{11} q_1 + h_{12} q_2$

$\underline{n=2}$: $Aq_2 = h_{21} q_1 + h_{22} q_2 + h_{23} q_3$

$\vdots$

$\underline{n}$: $Aq_n = h_{n1} q_1 + h_{n2} q_2 + \ldots + h_{nn} q_n + h_{n,n+1} q_{n+1}$

(II) Algorithm: $q_1 = b/\|b\|$

for $n = 1, \ldots$

$v = Aq_n$

for $j = 1, \ldots n$

$h_{jn} = q_j^* v$

$v = v - h_{jn} q_j$

end

$h_{n,n+1} = \|v\|$

$q_{n+1} = v/h_{n,n+1}$

QR Factorization of Krylov:

- vectors q_j are orthonormal basis of Krylov subspaces : $\mathcal{K}_n = \langle b, Ab, A^2b, \ldots A^{n-1}b \rangle = \langle q_1, \ldots, q_n \rangle$

$$K_n = \begin{bmatrix} b & Ab & A^2b & \ldots \end{bmatrix} = Q_n R_n$$

$\longrightarrow$ projection of A onto the space gives. $H_n = Q_n A Q_n'$ where eigenvalues of H_n are good approximate ... eigenvalues

GMRES: solving $Ax=b$

- step n, approximate x_n by $\in K_n$ that minimizes the norm of the residual $r_n = b - Ax_n$

(I) let $c \in \mathbb{C}^m$ st $AK_n = \begin{bmatrix} Ab & A^2b & \cdots & A^n b \end{bmatrix}$ ← column space is $A\mathcal{K}_n$

$\therefore$ $\min \| AK_n c - b \|$ ← unstable

instead use Arnoldi to find Q_n ...

let $x_n = Q_n y$

$$\rightarrow \min \| AQ_n y - b \| \quad ; \text{ recall } AQ_n = Q_{n+1} \tilde{H}_n$$

$$\min \| Q_{n+1} \tilde{H}_n y - b \|$$

$$\min \| \tilde{H}_n y - Q_{n+1}^* b \| \quad ; \quad Q_{n+1} b = \|b\| \hat{e}_1$$

$$\min \| \tilde{H}_n y - \|b\| \hat{e}_1 \|$$

<u>Note:</u> $\| \tilde{H}_n y - \|b\| \hat{e}_1 \|_2 = \sqrt{(H_n y - \|b\| e_1)^T (H_n y - \|b\| e_1)} = f = \sqrt{y^T H_n^2 y - 2\|b\| y^T \tilde{H}_n^T \hat{e}_1 + \|b\|^2}$

$$\nabla f = \frac{1}{2} \frac{2 H_n^2 y - 2\|b\| H_n^T \hat{e}_1}{\sqrt{}} = 0$$

$$\therefore \boxed{H_n^2 y = 2\|b\| H_n^T \hat{e}_1}$$