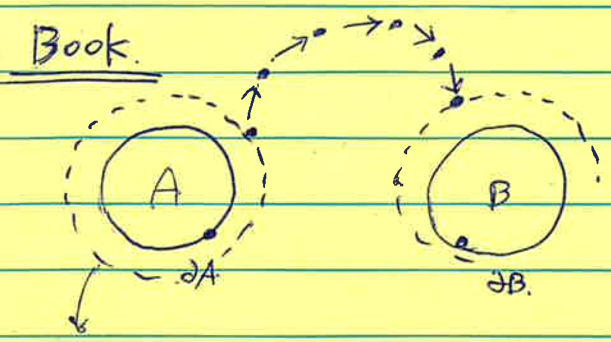
CME302 09/23/2024.

- **Book.**



"$\partial A + \partial A$".

Hill's relation

$$\text{rate} = \left[\int_{\partial A + \partial A} f(x)\,dx\right] \cdot j_{(\partial A + \partial A)}$$

?

- Solve linear systems $A\vec{x} = b$.
- Find approximate solutions. (e.g. linear regression).
- Evolution of systems. (e.g. $x_{n+1} = A x_n$).
- Matrices (e.g. dense (expensive), ~~cheap~~ sparse (cheap)).
- Solve PDE via discretization.

e.g.) $-y''(x_i) = \dfrac{-y_{i-1} + 2y_i - y_{i+1}}{h^2}$ (centered finite difference).

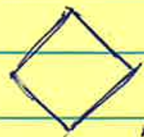
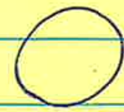
↳ Tri-diagonal matrix.

$Ay = f$
- Storage required on computer. (space)
- Efficient algorithms to solve (time)
- Accuracy

CME 302

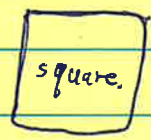
09/25/2024.

- Measure things. (norms)

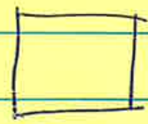




$\|x\|_1 = \sum_{i=1}^{n} |x_i| = 1$ $\qquad \|x\|_2 = 1$ $\qquad \|x\|_p = 1$ $(p > 2)$

Then,



$\equiv$



$\|x\|_\infty = 1$ $\qquad \max(x_i) = 1$

Note : Cauchy – Schwartz. Inequality.

$|x^T y| \le \|x\|_2 \|y\|_2$ at $\mathbb{R}^n \setminus \mathbb{C}^n$ $\rightarrow$ Expanded to Hölder inequality.

- Matrix norms.

There is an x such that. $\|Ax\|$ is maximum.

$\|A\| = \max_{\|x\|=1} \|Ax\|$ using induced norm or natural norm.

$\|A\|_1, \|A\|_2, \|A\|_3$.

Also, $\|A\|_F = \left(\sum_{i,j} |a_{ij}|^2 \right)^{1/2}$ (Frobenius norm).

Note : $\|A\| \|B\| \geq \|AB\|$ holds for sub-multiplicative norm. (SMN)

e.g.) $\max_{ij} |a_{ij}|$ is not SMN, however, $\sqrt{mn} \cdot \max_{ij} |a_{ij}|$ is SMN.

· **Orthogonal matrices.** (rotation, reflection, permutation). isometry !

— $\|x\|_2 = \|Qx\|_2$ and Q of size n, square

$$Q = \underbrace{H_1}_{\text{reflection}} \cdots H_K \quad (\text{always!}) \quad (1 \le k \le n)$$

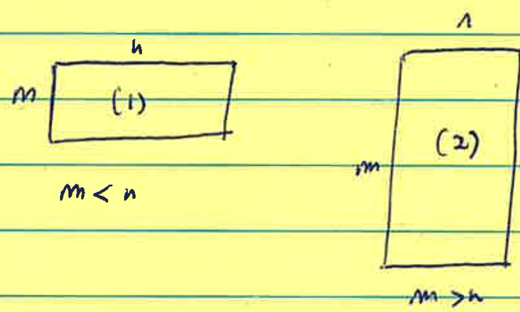
— Also $Q^T Q = I$. $\Leftrightarrow$ $q_i^T q_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$

($Q \in \mathbb{R}^{m \times n}$).

when Q is complex, $Q^H Q = I$ $\left(Q^H = (Q^T)^* \right) = Q^+$ in Q.M.

($Q \in \mathbb{C}^{m \times n}$)

① Note :



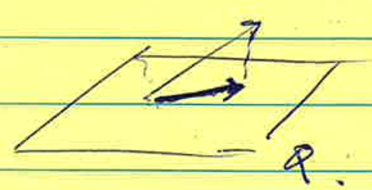
In (1), columns can't be independent each other $\rightarrow$ orthogonal X

$\Rightarrow$ $m \ge n$ to be orthogonal Q.

② Note: $Q^T Q = I$, $Q Q^T = I$

Always True | True if Q is square, False otherwise.

What is QQ^T if $Q =$  $\equiv$ projection to subspace Q



100000 $10^8 - 1500$.
$10^5 - 15s$.

- Decomposition. (eigenvalues, singular values, etc.)

$Ax = \lambda x \longrightarrow A^n x = \lambda^n x$ (λ greatest, $n \gg 1$). after decay...

$x_1 \sim x_n$: linearly independent eigenvectors. $X = [x_1, \cdots, x_n]$

$AX = X\Lambda$ where $\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix}$ $(Ax_i = \lambda_i x_i)$

X is invertible. $A = X \Lambda X^{-1}$ eigen decomp.

$A^k = X \Lambda^k X^{-1}$ where $\begin{pmatrix} \lambda_1^k & & 0 \\ & \ddots & \\ 0 & & \lambda_n^k \end{pmatrix}$.

$\max(\lambda_i) = 1$: oscillatory
$\max(\lambda_i) < 1$: decaying
$\max(\lambda_i) > 1$: amplifying.

A, $a_{ij} = a_{ji}$ (symmetric, hermitian)

$A^T = A$ $\quad A^\dagger = A$.

(Eigenvectors are orthogonal
Eigenvalues are real) $A = Q \Lambda Q^T$

For Symmetric matrix,
1. Apply sequence of reflections Q^T
2. Scale axis Λ
3. Apply reflections in reverse order Q

Note: symmetric positive definite : $\lambda_i > 0$.. $\rightarrow$ Apply conjugate gradient

< Schur decomposition > — easily get eig. vals.

$A = Q T Q^H$ (Q: unitary) $\quad T = \begin{bmatrix} \lambda_1 & & \text{////} \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix}$ upper triangular.

always exists #.

E.g.) $\begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix}$ There is no second eigenvector

+ Jordan decomposition

09/27/2024.

- $\boxed{\text{Block matrices.}}$

$$\begin{bmatrix} \text{---} \end{bmatrix}\begin{bmatrix} | \end{bmatrix} = \begin{bmatrix} \circ \end{bmatrix} \approx \left[\begin{array}{c|c} a_{11} & r_1^T \\ \hline c_1 & X_1 \end{array}\right]\left[\begin{array}{c|c} b_{11} & r_2^T \\ \hline c_2 & X_2 \end{array}\right] = ?$$

$$\Rightarrow \left[\begin{array}{c|c} a_{11} \cdot b_{11} + r_1^T c_2 & a_{11} r_2^T + r_1^T X_2 \\ \hline c_1 b_{11} + X c_2 & c_1 r_2^T + X_1 X_2 \end{array}\right]$$

$\rightsquigarrow$ Recursive operations.

Example. DIY

- $\boxed{\text{Schur decomposition}}$ (A: $n \times n$ matrix)

$A = QRQ^T$ (where Q: orthogonal & R: upper triangular)

$\rightarrow$ Prove that such decomposition always exists.

Induction: (i) $n=1$. A is scalar, $Q = [1]$, $R = A$. $\rightarrow$ True.

(ii) Assume $n \times n$ matrix A exists, show for size $n+1 \times n+1$.

We already know that.

$\exists \lambda, \vec{v}$ s.t. $A\vec{v} = \lambda \vec{v}$ when $\vec{v} \neq \vec{0}$, define $q = \vec{v}/\|\vec{v}\|$ ($\lambda \neq 0$).

Let $\{\vec{q}_2, \vec{q}_3, \cdots \vec{q}_{n+1}\}$ be orthonormal basis for the subspace $\perp$ to $\vec{v}$

Let $Q = [q \; q_2, \cdots, q_{n+1}]$ and $Q^TQ = I$ (trivial).

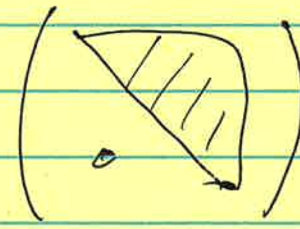
$$\text{Consider } Q^T A Q = \begin{pmatrix} \text{---}\, q^T \,\text{---} \\ \text{---}\, q_2^T \,\text{---} \\ \vdots \\ \text{---}\, q_{n+1}^T \,\text{---} \end{pmatrix} A \begin{pmatrix} | & | & & | \\ q & q_2 & \cdots & q_{n+1} \\ | & | & & | \end{pmatrix} = \begin{pmatrix} \text{---}\, q^T \,\text{---} \\ \vdots \\ \text{---}\, q_{n+1}^T \,\text{---} \end{pmatrix} (\underbrace{Aq}_{\lambda q} \cdots A q_{n+1})$$

$$= \begin{pmatrix} q^T \lambda q & (\quad) \\ q_2^T \lambda q & \\ \vdots & \\ q_{n+1}^T \lambda q & \end{pmatrix} = \left(\begin{array}{c|c} \lambda & w^T \\ \hline 0 & \\ \vdots & B \\ 0 & \end{array}\right)$$

We need to show $Q^T A Q = R$ is upper triangular.

Since from the induction, β shall have its own Schur Decomposition, so that

$$\left(\begin{array}{c|c} \lambda & w^T \\ \hline 0 & \\ \vdots & \beta_1 \\ 0 & \end{array}\right) \rightarrow \left(\begin{array}{cc|c} \lambda & & w_1^T \\ 0 & \lambda' & w_2^T \\ \hline \vdots & 0 & \\ \vdots & \vdots & \beta_2 \\ 0 & 0 & \end{array}\right) \rightarrow \cdots$$



Therefore, Schur decomposition always exists, for $n \times n$ matrix A.

To be specific, $\beta = \tilde{Q}_2^T R_2 \tilde{Q}_2$

$$Q_2 = \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & \tilde{Q}_2 \end{array}\right) \qquad Q_2^T \begin{bmatrix} \lambda & w^T \\ 0 & \beta \end{bmatrix} Q_2 = \begin{pmatrix} 1 & 0 \\ 0 & \tilde{Q}_2^T \end{pmatrix}\begin{pmatrix} \lambda & w^T \\ 0 & \beta \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & \tilde{Q}_2 \end{pmatrix}$$

$n+1 \times n+1$

$$= \begin{pmatrix} \lambda & \sim\sim \\ 0 & \tilde{Q}_2^T \beta \tilde{Q}_2 \end{pmatrix}$$

$\tilde{Q}_2^T \beta \tilde{Q}_2$ " R_2 (upper triangular).

Why? $Q_2^T Q_2 = I \rightarrow Q_2$ is orthogonal matrix.

$$Q_2^T Q_1^T A Q_1 Q_2 = \begin{pmatrix} \lambda & \text{---} \\ 0 & R_2 \end{pmatrix} \text{ which is upper-triangular.}$$

$Q_1 Q_2$ is orthogonal (Q_1, Q_2 are orthogonal). — Note.

$\because (Q_1 Q_2)^T (Q_1 Q_2) = Q_2^T Q_1^T Q_1 Q_2 = Q_2^T I \cdot Q_2 = Q_2^T Q_2 = I.$

- Define $\boxed{\text{Nilpotent matrix}}$

A is nilpotent if $\exists\, k>0$ s.t $A^k = 0$.

Problem 1

Suppose A is nilpotent, $(A \neq 0)$. (A is $n\times n$) show that A is not diagonalizable.

Pf) suppose A is diagonalizable, $A = S\Lambda S^{-1}$. Λ is diagonal matrix.

Since $A^k = 0, \Rightarrow (S\Lambda S^{-1})^k = 0 \Rightarrow (S\Lambda S^{-1})(S\Lambda S^{-1})\cdots(S\Lambda S^{-1})$

$= S\Lambda^k S^{-1} = 0$.

$\Rightarrow \Lambda^k = 0$, hence, A is non-diagonalizable. #

$(\because \lambda_1 = \lambda_2 = \cdots = \lambda_n) = 0 \rightarrow A = 0$ contradiction!

Problem 2

N is strictly upper-triangular. $N = \begin{pmatrix} 0 & & * \\ & \ddots & \\ 0 & & 0 \end{pmatrix}$ then N is Nilpotent

Pf1) Think of $N^2 = \begin{pmatrix} 0 & 0 & * \\ & \ddots & \\ 0 & & 0 \end{pmatrix} \cdots\cdots N^k = \begin{pmatrix} 0 & 0 & 0 & * \\ & \ddots & & 0 \\ 0 & & & 0 \end{pmatrix} \rightarrow 0$.

Trivial #

Pf2) $|N - \lambda I| = 0 = (\lambda)^n = 0 \rightarrow$ Cayley-Hamilton $\rightarrow N^n = 0$

Problem 3.

Show that any matrix A can be written as $A = D + N$ where $\begin{cases} D: \text{diagonalizable} \\ N: \text{nilpotents.} \end{cases}$

$$A = Q\begin{bmatrix} \ddots & * \\ 0 & \ddots \end{bmatrix}Q^T = Q\left(\begin{bmatrix} \ddots & 0 \\ 0 & \ddots \end{bmatrix} + \begin{bmatrix} 0 & * \\ 0 & 0 \end{bmatrix}\right)Q^T$$

$$= \underbrace{Q\begin{bmatrix} \ddots & 0 \\ 0 & \ddots \end{bmatrix}Q^T}_{D\ (\text{diagonalized!})} + \underbrace{Q\begin{bmatrix} 0 & * \\ 0 & 0 \end{bmatrix}Q^T}_{N} \qquad N^k = Q\underbrace{\begin{bmatrix} 0 & * \\ 0 & 0 \end{bmatrix}^k}_{=0}Q^T = 0$$

09/30/2024

- $Ax = \lambda x$ such that $X = [x_1, \cdots, x_n]$, then
  $AX = X\Lambda \Rightarrow \boxed{A = X \Lambda X^{-1}}$ (X singular, X^{-1} issue!)

- $A = QTQ^H$ where Q : unitary, $Q^H Q = I$, $T = \begin{bmatrix} \lambda_1, \dots, \lambda_n & /// \\ & \end{bmatrix}$

  Schur decomposition.

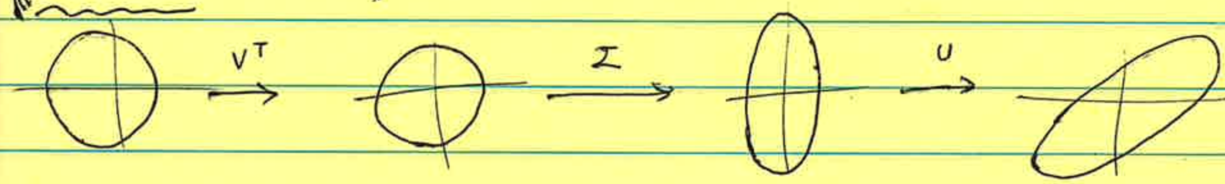
① A is symmetric, $A = Q\Lambda Q^T$ (Schur $\equiv$ eigen).

② Singular value decomposition. $A = U \Sigma V^T$ (U, V orthogonal)

$$\Sigma = \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \end{pmatrix}$$

A can be described in sequence : V^T: reflections
Σ : scaling < for any matrix A > !
U : reflections.

(U, V, are orthogonal, $\sigma_i > 0$)



For symm $A = Q\Lambda Q^T = U\Sigma V^T$ (not always).

only for positive-definite A ($\because$ $\sigma_i > 0$ $\lambda_i > 0$)

- Rank

① Rank 1 matrix : $A = u v^T$, Storage : $O(n)$ ($\because$ store u and v).
Cost : $O(n)$ ($\because$ $Ax = (uv^T)x = u\underbrace{(v^T x)}_{\text{dot product}}$)

$A = uv^T \rightarrow$ How to do SVD

$$A = \underbrace{\begin{bmatrix} \frac{u}{\|u\|} & \cdots \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \|u\|\|v\| & & 0 \\ & 0 & \\ 0 & & 0 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} \frac{v^T}{\|v\|} \\ \vdots \end{bmatrix}}_{V^T}$$

Storage ~~O(mn)~~ vs $O(n^2)$
$\sigma_1 = \|u\|\|v\|$ $O(rn)$
$\sigma_i = 0$ $(i \geq 2)$
Very rapid decay. after $i \geq r$

- $Ax = b$. — How to solve.

A square, non-singular, ① $Lx = b$. $\Rightarrow \sum_{j=1}^{i} l_{ij} x_j = b_i$

$l_{11} x_1 = b_1 \Rightarrow x_1 = b_1 / l_{11}$

$l_{21} x_1 + l_{22} x_2 = b_2$

$\Rightarrow x_i = \frac{1}{l_{ii}} \left(b_i - \sum_{j=1}^{i-1} l_{ij} x_j \right)$.

$x_1 \to x_2 \cdots \to x_n$

② $Ux = b \Rightarrow \Sigma$

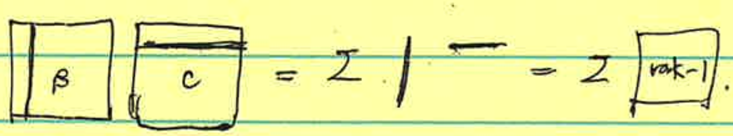
$u_{nn} x_n = b_n \to x_{n-1} \cdots \to x_1 \cdots$

same as L ①.

Therefore, L, U are same algorithm. Now, to solve $Ax = b$,

$A = L \cdot U \Rightarrow LUx = b \to$ Run ① and ② consecutively.

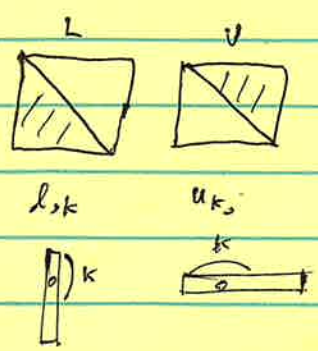
- $A = B \cdot C$ = sum of rank-1 matrices



$\to$ Apply this to LU factorization.

$b_{,k}$: column k of B. $\}$ $\Rightarrow A = \sum_{k=1}^{n} b_{,k} c_{k,}$

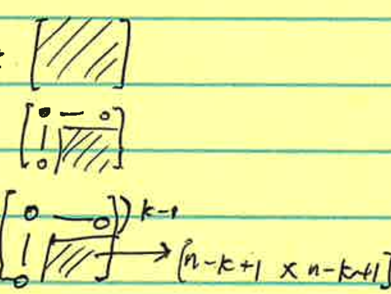
$c_{k,}$: ~~column~~ row k of c



$l_{,1} \cdot u_{1,} =$

$l_{,2} \; u_{2,} =$

$l_{,k} \; u_{k,} =$



$\therefore \; a_{,1} = l_{,1} u_{11}$

$a_{1,} = l_{11} u_{1,}$ $\to$ Continue...

< Algorithm for LU decomposition >

LU $\rightarrow$ $a_{11} = l_{11} u_{11}$: (convention : $l_{11} = 1$). $\Rightarrow$ ($u_{11} = a_{11}$)

Then, $u_{11} = a_{11}$ and $a_{\cdot 1} = l_{\cdot 1} u_{11} \Rightarrow l_{\cdot 1} = \frac{1}{u_{11}} \cdot a_{\cdot 1} = \frac{1}{a_{11}} \cdot a_{\cdot 1}$

Recall . $A = \sum_{k=1}^{n} l_{\cdot k} u_{k \cdot}$

Then, $A - l_{\cdot 1} u_{1 \cdot} = \sum_{k=2}^{n} l_{\cdot k} u_{k \cdot}$

Next, repeat for $k = 2, 3, \ldots$

Elasticity equation $\leftrightarrow$ SDE ?

$\cdot a_{11}$ pivot $\Rightarrow$ All pivots must be non-zero. $A = LU$ pivot at step k u_{kk}.

A non-singular, $A = LU \Rightarrow \det(A) = \det(L)\det(U)$

$$= \underbrace{l_{11} l_{22} \cdots l_{nn}}_{=1 \text{ (convention)}} \cdot u_{11} u_{22} \cdots u_{nn}.$$

$\Rightarrow u_{11} \cdots u_{nn} \neq 0$ ($\because A$ is non-singular).

If $a_{11} = 0$, LU does not exist. $\rightarrow$ what should we do? $\rightarrow$ Pivoting.

$A = \begin{pmatrix} 0 & 1 \\ 1 & \pi \end{pmatrix}$: non-singular, no LU

If we do $\begin{pmatrix} \varepsilon & 1 \\ 1 & \pi \end{pmatrix}$. unstable ... $\rightarrow$ Do row pivoting.

LU decomposition — examples

10/02/2024.

$$A = \begin{pmatrix} \varepsilon & 1 \\ 1 & \pi \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 \\ 1/\varepsilon & 1 \end{pmatrix} \qquad \text{Then, } LU = \begin{pmatrix} \varepsilon & 1 \\ \sim 1 & 1/\varepsilon + \pi - 1/\varepsilon \end{pmatrix}$$

$$U = \begin{pmatrix} \varepsilon & 1 \\ 0 & \pi - \frac{1}{\varepsilon} \end{pmatrix}$$

$1/\varepsilon \cdot \varepsilon$ → it's okay (numerically). division is okay.

→ This is <u>not okay</u> (numerically). <u>Subtractions</u> are <u>dangerous</u>

If $\varepsilon = 2^{-100}$, $\pi - 2^{100} = 0$ → Error!

Let $\tilde{A} = \begin{pmatrix} \varepsilon & 1 \\ 1 & 0 \end{pmatrix}$

Thus f.

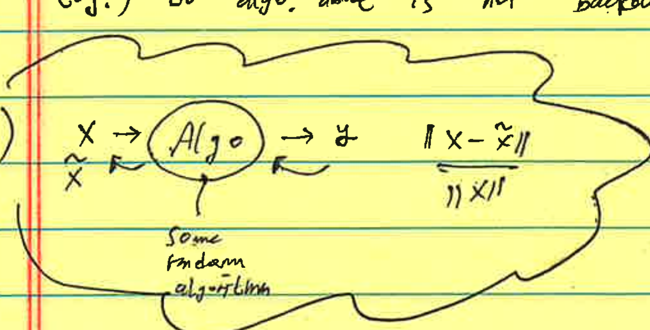
Define Backward error : Relative error $\frac{\|x - \tilde{x}\|}{\|x\|}$ when $f(\tilde{x}) = \tilde{f}(x)$.
$f: \mathbb{R} \to \mathbb{R}$.

We say $\tilde{f}$ is backward stable if backward error $= O(\mu)$.

E.g.) LU algo. above is not backward stable. $\tilde{A} \neq A$

Huge discrepancy.

Fig.)



Q) How to stabilize LU algo. above?

④

CME 302 — 10/07/2024.

- Floating point arithmetic.

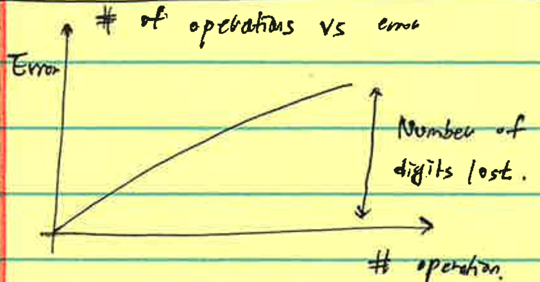
Problem : $(a+b)+c$ where $a=1$, $c=-1$

$$\begin{array}{l} 1.0 \text{ ——— } 0 \\ + \ 0. \text{ ——— } [\text{ lost }] \\ \hline -1.0 \text{ ——— } \\ \hline \end{array}$$

[lost] → This part wrong.

E.g.)		$(a+b)+c$
$a=1$	$b=10^{-14}$	$0.9992 \cdot 10^{-14}$
$c=-1$	$b=10^{-15}$	$1.11 \cdot 10^{-15}$
	$b=10^{-16}$	

# of operations vs error

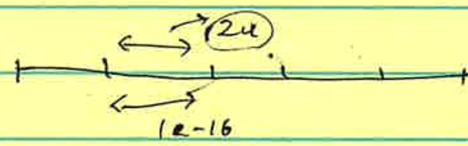


<How to estimate roundoff errors>

$x_1 + \cdots + x_n$

$S_i = \sum_{j=1}^{i} x_j$ $\qquad \dfrac{u \max|S_i|}{|S_n|}$

Error $\sim u \cdot \max|S_i|$



Example 1)

$(x-10)^6 = x^6 - 60x^5 \cdots\cdots$

Near $x=10$, $(x-10)^6$ power is fine.

In RHS, $-20000x^3$ & $150000x^2$ will give huge result $\sim 10^8$.

☆ $x = 10.01 \rightarrow (x-10)^6 = 10^{-12}$

$20000x^3$

$\boxed{\dfrac{u \max|S_i|}{|S_n|}} = \dfrac{10^{-16} \cdot 10^8}{10^{-12}} = 10^4$ your relative error.

$\qquad \| $

$\qquad 10^{-8}$ your real error

Note: $10^{-16} \cdot 10^8 = 10^{-8}$

- Stability of LU.

$Ax = b.$

$\|x - \tilde{x}\|_2$ forward error.

$(A+E)\tilde{x} = b$ —— (1)

Q: Is there an approx. problem that algorithm solves exactly (e.g. (1)).

A: Perturb $A \rightarrow A+E$ that $\tilde{x}$ has to be exactly the solution.

Backward error analysis.

→ Backward error bound $\|E\|_2 \le$ instead of $\|x-\tilde{x}\|_2 \le$

$Ax = b$, $A \to A+E$ (perturbation), $x \to \tilde{x}$

$$f: \begin{cases} A \to x \\ A+E \to \tilde{x} \end{cases}$$

→ For LU factor.,

$$|M| = \begin{pmatrix} |m_{11}| & |m_{12}| & \cdots \\ |m_{21}| & & \\ \vdots & & \end{pmatrix}$$

$$\left.\begin{array}{l} Ax = b,\ A = LU, \\ \uparrow \text{ LU + solve} \end{array}\right\} \Rightarrow |E| \le n \cdot u \left((2|A|) + 4 \cdot |\tilde{L}||\tilde{U}| \right) + O(u^2)$$

n: size of matrix / u: unit round off

$|\tilde{L}|, |\tilde{U}|$: numerical LU factors.

→ Forward error.

$$\frac{\|x-\tilde{x}\|_2}{\|x\|_2} \le \frac{k_2(A)}{1 - k_2(A)\,\|E\|/\|A\|} \cdot \frac{\|E\|}{\|A\|}$$

$\|E\|$ → perturbation

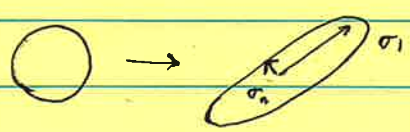
$\frac{\|x-\tilde{x}\|_2}{\|x\|_2}$ → Error in solution

$k_2(A) = \|A\|_2 \, \|A^{-1}\|_2$ → Condition number.

$Ax = b$

$$\left.\begin{array}{l} A: \to \|A\|_2 = \sigma_1 \quad (\text{from SVD}). \\ \quad \to \|A^{-1}\| = 1/\sigma_n \quad (\ \text{"}\ ) \end{array}\right\} \quad k_2(A) = \sigma_1/\sigma_n$$

Error $\sim u\,k(A) = u\,\sigma_1/\sigma_n$ → If you meet this, you are at your optimum.

A:  ill-conditioned = ellipsoid is very flat

$x = V\Sigma^{-1}U^T b \Rightarrow$ if $1/\sigma_n$ is very large → result in x changes drastically.

If $A = \begin{pmatrix} \varepsilon & 1 \\ 1 & \pi \end{pmatrix}$ $L = \begin{pmatrix} 1 & 0 \\ 1/\varepsilon & 1 \end{pmatrix}$ $U = \begin{pmatrix} \varepsilon & 1 \\ 0 & \pi - \varepsilon^{-1} \end{pmatrix}$ neglected $\because 1/\varepsilon$

Then $|\tilde{L}||\tilde{U}|$ dominates the error term in $|E|$
$\simeq \max|s_i|$

⇒ Pivoting (row) will fix.

CME302. 10/09/2024.

- **LU factorization.**

$l_{ik} = a_{ik}/a_{kk}$ ($l_{ik} >> 1$ when $a_{kk} << 1$)

$\tilde{L}$ $\tilde{U}$ ↳ causes round-off errors. when $l_{ik} >> 1$

Q) How to solve this errors?

A) Permute order of equations (row pivoting).

⇒ $PAx = Pb$. (P : permutation matrix)

⇒ Perform row pivoting using the largest entry in the column.

$\left\{ i = \underset{i \geq k}{\operatorname{argmax}} |a_{ik}| \right\}$

⇒ with pivoting, $|l_{ik}| = \frac{|a_{ik}|}{|a_{kk}|} \leq 1$ (small number) so that $l_{ij} \in O(1)$

$u_{ij} \in O(\|A\|_2)$.

Therefore, $\tilde{L} \cdot \tilde{U}$ ⇒ errors are small

Q) Is it true $PA = LU$ always exists?

A) If you encounter zero, it implies that whole column $i \geq k$ is zero.

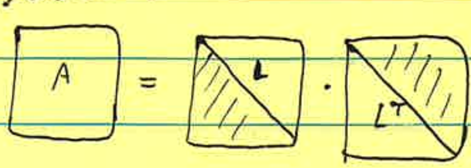
⇒ Proceed with the method.

- **Cholesky factorization.**

symm. pos. def. $A = Q \Lambda Q^T$ ($\lambda_i > 0$)

$a_{ij} = a_{ji}$ and $x^T A x > 0$ (for $x \neq 0$).

⇒ $A = LL^T$ where L is non-singular, LL^T is positive definite. (Can we?)



⇒ $a_{ij} = \sum_{j=1}^{i} (l_{ij})^2$

Since $|l_{ij}| < \sqrt{a_{ii}}$ ⇒ Pivoting is not required!

- How to do Cholesky factorization?

$$A = \sum_{k=1}^{n} l_{:,k}\, l_{:,k}^T$$

$k=1 \rightarrow l_{:1} \cdot l_{:1}^T$ contributes (only among n) to row 1, col 1 of matrix A.

$\Rightarrow a_{:1} = l_{:1}\, l_{11} = l_{11}\, l_{:1} \Rightarrow a_{11} = (l_{11})^2 \Rightarrow a_{11} > 0$, then $l_{11} = \sqrt{a_{11}}$

$\Rightarrow \underbrace{l_{i1} = a_{i1} / \sqrt{a_{11}}}$

$k>1 \rightarrow A - l_{:1}\, l_{:1}^T \rightarrow$ Repeat $l_{:2} \rightarrow A - l_{:1} l_{:1}^T - l_{:2} l_{:2}^T \rightarrow \cdots$

But we need assumption that all pivots must be > 0 $\Leftarrow$ sym. pos. def.

Pf) $A^T = A$, $x^T A x > 0$. $\Rightarrow$ all pivots > 0.

$\rightsquigarrow$ MATH ...

- {Storage of Cholesky. $\Rightarrow$ Half of LU}
- {Computational time of Cholesky $\rightarrow$ Half of LU $O(\frac{1}{2}n^3)$}
- {Advantage $\rightarrow$ No pivoting!}.

Pf) Induction. (n : size of matrix).

$A = LL^T$?

(i) $n=1 \rightarrow x^T A x > 0 \Rightarrow 1^T A\, 1 > 0 \Rightarrow [a_{11}] > 0$ $\therefore$ True. ($\because a_{11} = (l_{11})^2$)

(ii) Assume True for '$n-1$'

$$A = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right)$$

(Block matrices).

Corollary 1). if $x = \begin{pmatrix} y \\ 0 \end{pmatrix} \Rightarrow x^T A x = (y^T \;\; 0)\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}\begin{pmatrix} y \\ 0 \end{pmatrix} > 0$

$\Rightarrow y^T A_{11}\, y > 0$

Thus. A_{11} is sym. and pos. def $\Rightarrow$ A_{11} is SPD.

(for any block of A.

Collorary 2) $A = \underbrace{\begin{pmatrix} I & 0 \\ A_{21}A_{11}^{-1} & I \end{pmatrix}}^{①} \begin{pmatrix} A_{11} & 0 \\ 0 & S \end{pmatrix} \overbrace{\begin{pmatrix} I & A_{11}^{-1}A_{21}^T \\ 0 & I \end{pmatrix}}^{③}$

where $S = A_{22} - A_{21}A_{11}^{-1}A_{21}^T \Rightarrow$ Schur complement.

Note: $③ = ①^T$ ⟨ S exists ∵ A_{11} is non-singular. (SPD)⟩

Collory 3) S is SPD. where $S = A_{22} - A_{21}A_{11}^{-1}A_{21}^T$.

$\Rightarrow X^TAX = \begin{pmatrix} A_{11} & 0 \\ 0 & S \end{pmatrix}$ where $X = \begin{pmatrix} I & -A_{11}^{-1}A_{21}^T \\ 0 & I \end{pmatrix}$

and since $x = \begin{pmatrix} 0 \\ y \end{pmatrix}$, $x^T \begin{pmatrix} A_{11} & 0 \\ 0 & S \end{pmatrix} x = y^TSy > 0$.

$= x^TX^TAXx = (Xx)^TA(Xx) > 0$.

Conclusion: when A_{11} is scalar, we know from assumption that S is S.P.D and pivots > 0, and using $n=1$ logic for A_{11},

$A = A_{11}$

S

$A_{11} = L_{11}L_{11}^T$

$S = L_{22}L_{22}^T$ (∵ S is SPD)

$A = \begin{pmatrix} I & 0 \\ A_{21}A_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} L_{11}L_{11}^T & 0 \\ 0 & L_{22}L_{22}^T \end{pmatrix} \begin{pmatrix} I & A_{11}^{-1}A_{21}^T \\ 0 & I \end{pmatrix}$

L_1 L_2

Note: $A_{11} = L_{11}L_{11}^T$
$\downarrow$
$A_{11}^{-1}L_{11} = L_{11}^{-T}$

$= \begin{pmatrix} L_{11} & 0 \\ A_{21}A_{11}^{-1}L_{11} & L_{22} \end{pmatrix} \begin{pmatrix} L_{11}^T & L_{11}^TA_{11}^{-1}A_{21}^T \\ 0 & L_{22}^T \end{pmatrix} = \begin{pmatrix} L_{11} & 0 \\ A_{21}L_{11}^{-T} & L_{22} \end{pmatrix} \begin{pmatrix} L_{11}^T & L_{11}^TA_{21} \\ 0 & L_{22}^T \end{pmatrix}$

$\| \qquad\qquad \|$

$L \qquad\qquad L^T$ ∴ Exists for n

QR decomposition

10/14/2024.

- $A = Q \cdot R$ $\rightsquigarrow$ Related to, $\|Ax - b\|_2$: least squares.

Q: orthogonal, R: 

Ax: Linear model. b: observation.

This line "filters out" the noise!

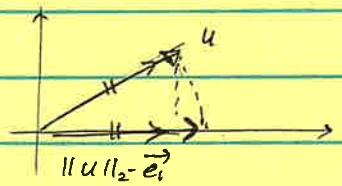
implies, $Q^T A = R$ (upper triangular) = "orthogonal triangulization"

Q: square matrix.

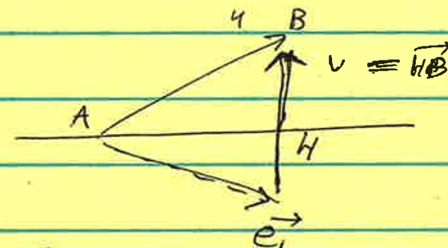
- Examples.

① $A = u =$ (m×1) $\Rightarrow Q^T A = R = \begin{pmatrix} x \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ when $x = \|u\|_2$. $\Rightarrow R = \|u\|_2 \cdot \vec{e_1}$.

②



$\rightarrow$ How to do? $\rightarrow$



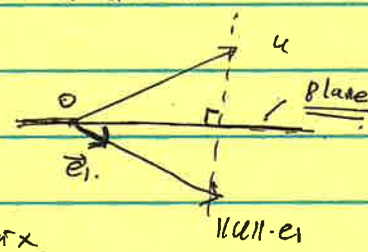
$\overrightarrow{HB} = \dfrac{u^T v}{\|v\|_2} v$ so that $\underline{\overrightarrow{AH} = \vec{u} - \dfrac{u^T v}{\|v\|} v}$.

$\Rightarrow$ ~~QOU~~ $Q^T u = u - 2 \cdot \dfrac{u^T v}{\|v\|} v = \|u\|_2 \vec{e_1}$ $\rightarrow$ "Reflected u"

where $v = u - \|u\|_2 e_1$

Basically, take $v = u - \|u\|_2 e_1$ and calculate,

$Q^T u = u - 2 \cdot \dfrac{u^T v}{\|v\|} \cdot v = \|u\|_2 e_1$ #

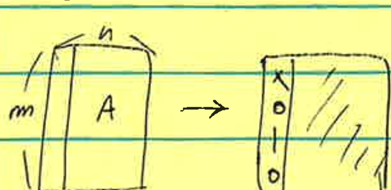


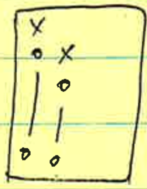
Hence, we found $R = \|u\|_2 \cdot e_1$ for given Q. matrix.

This is called, Householder reflection. (most comp. efficient way to transform $u \rightarrow \|u\|_2 e_1$)

$a_{,i} \rightarrow Q_1^T a_{,i}$ (cost : $m \times n$)

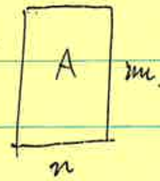
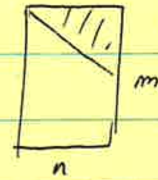
$= O(mn)$





$Q_2^T = \begin{pmatrix} 1 & 0 \text{ — } 0 \\ 0 & \\ | & \boxed{\times} \\ 0 & \end{pmatrix}$ → Householder Reflection.

(keep going) - - - - → $Q_i^T = \begin{pmatrix} I_{i-1} & 0 \text{ — } 0 \\ 0 & \\ | & \boxed{} \\ 0 & \end{pmatrix}$ → Householder R.

$\Rightarrow Q_n^T Q_{n-1}^T \cdots Q_1^T \cdot$  $=$  $= R$

sequence of operations

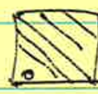
$\Rightarrow A = \underbrace{Q_1 Q_2 \cdots Q_n}_{=Q} \cdot R$ (just store $Q_1 \sim Q_n$ not Q!)

$\begin{cases} m > n : Q_n \\ m = n : Q_{n-1} \\ m < n : Q_{m-1} \to \text{underdetermined} \ldots \end{cases}$

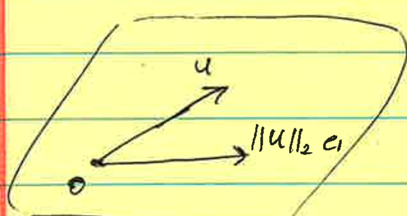
Cost : $O(nm) \times n \Rightarrow O(mn^2)$

If $m = n \to$ same as LU $(m > n)$

Advantage : No pivoting issue.

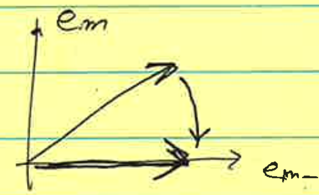
- Given's method $\leadsto$  $\approx$ upper Hessenberg.

– Given's rotations.



on span$\{e_{m-1}, e_m\}$

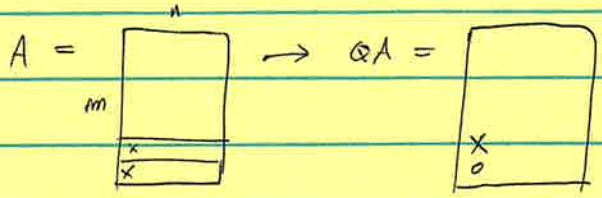
$\begin{array}{c} m-1 \\ m \end{array} \begin{bmatrix} \times \\ \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times \\ 0 \end{bmatrix}$ → zero out e_m components



we can define $\cos = c = \dfrac{x}{\|u\|_2}$ for $u = \begin{pmatrix} x \\ y \end{pmatrix}$

$\sin = s = \dfrac{y}{\|y\|_2}$

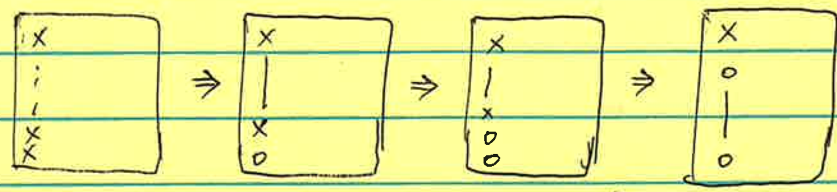
where $Q = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}$ so that, $Qu = \begin{pmatrix} \|u\|_2 \\ 0 \end{pmatrix}$.



where $Q = \begin{pmatrix} I & 0 \\ 0 & \begin{array}{cc} c & s \\ -s & c \end{array} \end{pmatrix}$

$\Rightarrow$ Cost $= O(n)$

keep iterating for $\underset{O(n)}{m-1}$, $\underset{O(n)}{m-2}$, $\cdots$ $\Rightarrow$

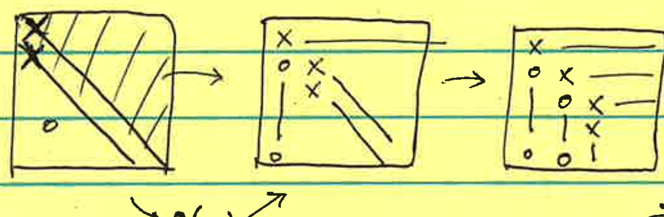


total : $m-1$ times.

$\Rightarrow$ Cost $= O(mn)$ but "ordering" of non-zero entries is a key "

$\rightarrow$ Total cost : $O(mn^2)$. ($\because$ for n columns)

- Upper Hessenberg.



$\cdots$ n times (all columns)

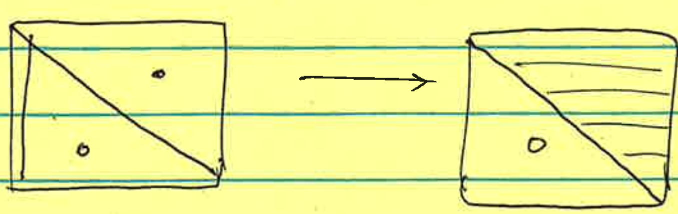
$\Rightarrow$ Total cost : $O(n^2)$

$\rightarrow$ Total cost drops from $O(mn^2)$ to $O(n^2)$ for given's for upper Hessenberg.

$\boxed{\text{Q) What is advantage of upper Hessenberg?}}$

- Think.

$A = D + u e_1^T$,



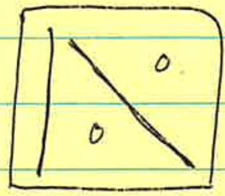
How many sequences?

Givons.

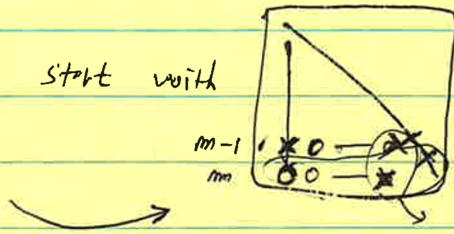
10/16/2024.

How to do Q.R. factorization for

$A = D + u e_1^T$



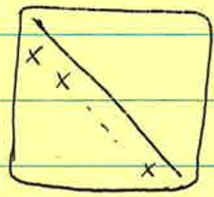
start with



apply Givon's to m-1th, mth

two new non-zero entries.

keep iterating for $\underline{(m, m-1)} \longrightarrow (m-1, m-2), \longrightarrow \cdots \longrightarrow \underline{(2, 1)}$

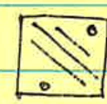


→ Final result (upper - Hessenburg).

Cost : general : $O(mn^2)$

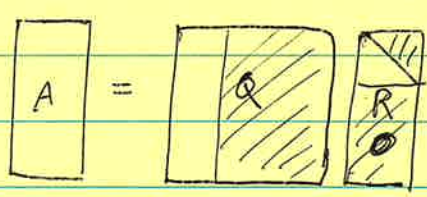
upper - Hess : $O(mn)$

Tri-diag : $O(m)$

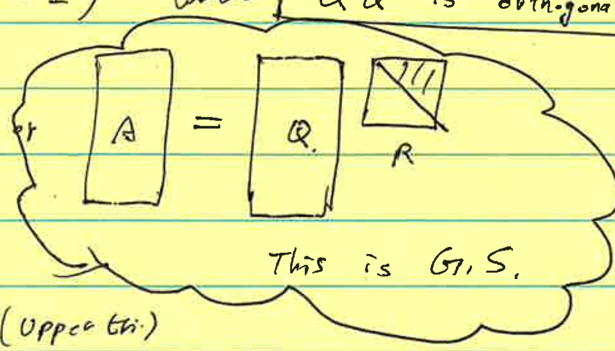


Gramm - Schmidt.

$A = Q R$ $\quad (Q^T Q = I)$ where $\boxed{Q Q^T \text{ is orthogonal projection}}$



or



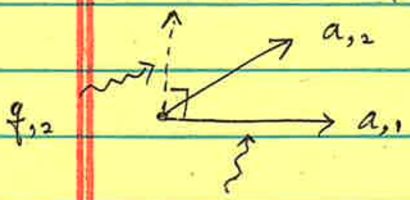
This is Householder & Givons

because they are transformations (upper tri)

span (A) = span (Q) → not good notation

span $\{ a_{,1}, a_{,2}, \cdots, a_{,n} \}$ = span $\{ q_{,1}, q_{,2}, \cdots, q_{,n} \}$.

- Procedure (Note : $r_{ii} > 0$ and A is full-rank)

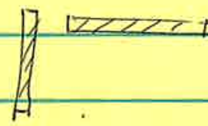


$q_{,1}$ (∵ R is upper tri.)
normalized

$$\boxed{q_{,2} = \frac{a_{,2} - P_{a_{,1}}(a_{,2})}{\| a_{,2} - P_{a_{,1}}(a_{,2}) \|_2}}$$

⇒ G.S. applies triangular trasformation to make coloumns of A orthogonal.

$$A = \sum_{k=1}^{n} q_{,k} \cdot r_{k,}$$



And note that $r_{k,}$ is $\overset{n}{\longleftrightarrow}$ $(\underbrace{0 \text{ ——— } 0}_{k-1} \; \times \text{ ——— } \times)$

① Therefore, only contribution to $a_{,1}$ is $q_{,1}$

⇒ $a_{,1} = r_{11} q_{,1}$ and since $(r_{11} > 0)$ and $\| q_{,1} \|_2 = 1$

$r_{11} = \| a_{,1} \|_2$ and $q_{,1} = a_{,1} / \| a_{,1} \|_2$. —— (1).

② Look at $a_{,j} = \sum_{k=1}^{j} q_{,k} r_{kj}$ (column j of A)

$$q_{,1}^T a_{,j} = \sum_{k=1}^{j} \underbrace{q_{,1}^T q_{,k}}_{=1 \text{ if } k=1} r_{kj} = r_{1j} = q_{,1}^T a_{,j}$$

Thus, if we have $q_{,1}^T$ ⇒ r_{1j} is obtained!

Then,

③ $A - q_{,1} r_{1,}$ → Repeat ①, ②.

$$a_{,j} - q_{,1} \underbrace{r_{1j}}_{= q_{,1}^T a_{,j}} \iff \text{makes } a_{,j} \perp \text{ to } a_{,1}$$

- Algorithm.

loop $k : 1 \rightarrow n$.

$r_{kk} = \|a_{,k}\|_2$

$q_{,k} = a_{,k} / r_{kk}$

loop $j : k+1 \rightarrow n$

$r_{kj} = q_{,k}^T a_{,j}$

$a_{,j} \leftarrow a_{,j} - r_{kj} q_{,k}$.

<Another method>

$A = Q \cdot R$.

A^TA is symmetric matrix.

If A is full rank, A^TA is positive definite,

$\Rightarrow A^TA = $ S.P.D.

Note that

$A^TA = R^TR$

$R^T \to L$, $R \to U$ ($\because$ definition of QR)

$\Rightarrow$ Cholesky factorization #

- Cholesky factorization is unique. $\Rightarrow$ R^T is a chol. factor.

So, ① Calculate A^TA $O(mn^2)$.

② $A^TA = R^TR$ $O(n^3)$ by running Cholesky

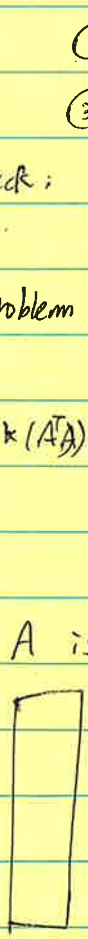
③ $Q = AR^{-1}$ $\rightarrow$ we can arrive at Q matrix.

check; $Q^TQ = R^{-T}(A^TA)R^{-1} = R^{-T}R^TRR^{-1} = I$ #

Problem with this method : Cholesky condition # bad $\rightarrow$ Not orthogonal Q

$$\kappa(A^TA) = \kappa(A)^2 = (\sigma_1/\sigma_n)^2 \quad \left(\because A = U\Sigma V^T \quad A^TA = V\Sigma^2V^T \text{ and } \kappa(A) = \sigma_1/\sigma_n\right).$$

- If A is thin $\rightarrow$ $\begin{cases} A^TA \text{ cheap.} \\ \text{Cholesky cheap.} \\ \kappa(A) \text{ not too large.} \end{cases}$ $\Rightarrow$ Algorithms will be accurate

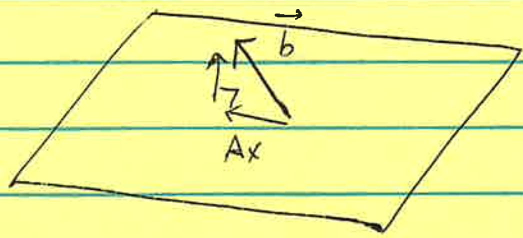


- Original G.S. $\Rightarrow$ Cost $=$ ~~OOA~~ $O(mn^2)$

Explicit R^{-1} for thin

AR^{-1} very fast ...

Projections — Midterm.



$\vec{b} - A\vec{x} \perp \text{range}(A)$

($\because$ l_2 norm minimization)

$\Rightarrow A^T(b - Ax) = 0$

$\Rightarrow \underline{A^TAx = A^Tb}$ //

Pf)1 $\text{range}(A) = Ay$

$\therefore (Ay)^T \cdot (b - Ax) = 0$ for $\forall y \Rightarrow y^T [\underbrace{A^T(b - Ax)}_{=0}] = 0$. #

Pf)2 $A^T(b - Ax) = 0$

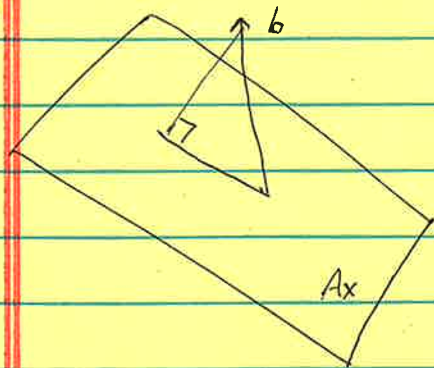
$a_{,j}^T (b - Ax) = 0$ (for $1 \le j \le n$) $\rightarrow$ trivial #

$\underbrace{A^TA}_{n\times n} x = A^Tb$

$n \times n \rightarrow$ Cholesky. $\longrightarrow$ Solve $\longrightarrow$ x : least square solution.

Least Squares.

10/21/2024



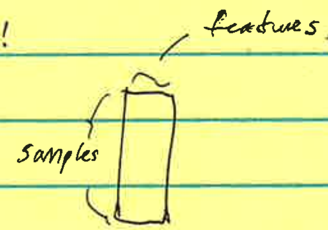
- $\|Ax - b\|_2$ minimized

  $\Leftrightarrow \quad b - Ax \perp \text{range}(A).$

  $\Rightarrow \quad A^T(b - Ax) = 0$

  $\Rightarrow \quad \underbrace{(A^TA)}_{n\times n,\ \text{SPD}} x = A^Tb$

  $n\times n$, SPD. $\rightarrow$ Use Cholesky!



- Condition # : $\underset{\substack{\| \\ V\Sigma^2V^T}}{\kappa(A^TA)} = \kappa(A)^2 \rightarrow$ quadratic increase!

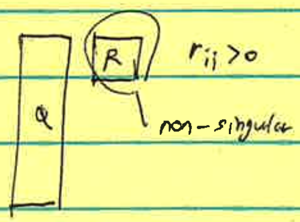
Thus, method of normal eq. works well on, $\begin{cases} A \text{ is well-conditioned.} \Rightarrow x = (A^TA)^{-1}A^Tb \\ A \text{ is very tall} \end{cases}$

※ A^TA : Dot product between colums of A

- QR fact.

$x = (A^TA)^{-1}A^Tb$, A is full-col-rank, $A = Q\cdot R$.

$\Rightarrow (A^TA)^{-1}A^T = (R^T \underbrace{Q^T \cdot Q}_{=I} \cdot R)^{-1} R^TQ^T = (R^TR)^{-1} \cdot R^TQ^T$



$r_{ii} > 0 \quad \Rightarrow \quad = R^{-1}R^{-T}R^TQ^T$

$\Rightarrow \quad \boxed{x = R^{-1}Q^Tb}$

Q) when is $(AB)^{-1} = B^{-1}A^{-1}$ true? $\langle A, B$ are square$\rangle$

$(C^TC)^{-1} = C^{-1}C^{-T}$ true? $\langle C$ is square$\rangle$

- $x = R^{-1} Q^T b$.

Q : principal directions of A "col span".

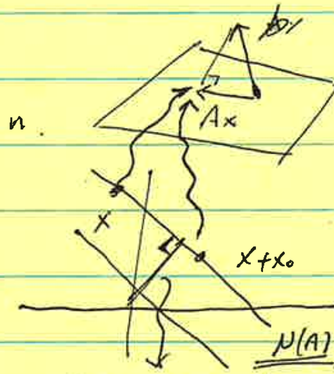
QR : R action of A in Q basis.

$Q^T b$: components of b in basis Q. (Q is orthog. $\Rightarrow$ no distortion).

$\Rightarrow$ $\underline{R^{-1} Q^T b}$: undoes action of A in this Q subspace.

$\Rightarrow$ R^{-1} : $\kappa(A) = \kappa(R)$. $\rightarrow$ Good!

- Rank deficient $N(A)$ non-trivial. $\text{rank}(A) = r < n$.



R singular, A^TA singular.

$\Rightarrow x_0 \in N(A)$, $A(x+x_0) = Ax + Ax_0 = Ax$

$\|Ax-b\|_2 + \lambda \|x\|_2 \rightarrow \|x\|_2$ minimized!

$\Rightarrow$ min. $\|Ax - b\|_2$

$x \perp N(A)$

solution is unique.

$$\Rightarrow \|Ax - b\|_2 = \|U\Sigma V^T x - b\|_2$$
$$= \|\Sigma V^T x - U^T b\|_2$$
$$\Rightarrow V^T x = \Sigma^{-1} U^T b.$$

Note : $A = U\Sigma V^T \Rightarrow \Sigma(V^T x) = U^T b$

$\Rightarrow \underset{\sim}{V^T x} = \Sigma^{-1} U^T b$.

$x : \mathbb{R}^n \rightarrow V^T x : \mathbb{R}^r$ $(r<n)$. so that $V^T x_0 = 0 \longrightarrow x_0 \in N(A)$

$\because V^T(x+x_0) = V^T(x)$.

$$\begin{array}{cc} & \begin{array}{cc} r & n-r \end{array} \\ n & \left[\begin{array}{c|c} V & V_0 \end{array}\right] \end{array}$$

$x = Vy + V_0 y_0$.

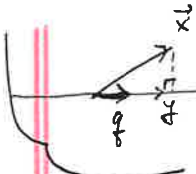
$V^T x = y \quad (\because V^T V_0 = 0)$.

orthogonal

$V^T x = y = \Sigma^{-1} U^T b$ (unique)

$\|x\|_2^2 = \|y\|^2 + \|y_0\|^2 \rightarrow y_0 = 0$. for min $\|x\|_2$.

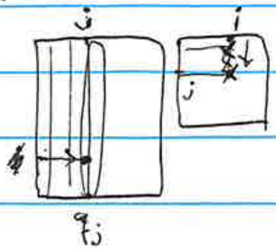
$\Rightarrow$ Solution is, $x = V\Sigma^{-1}U^T b$. (where $A = U\Sigma V^T$)



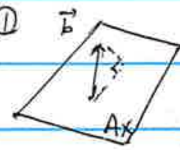
$\cos\theta = \frac{q^T x}{\|x\|}$

$\vec{y} = \underbrace{(q^T x)}_{\text{dot product.}} q$

- $A = QR \rightarrow a_i = \sum_{j=1}^{i} r_{ji} q_j$ normalized $\|q_j\|_2 = 1$

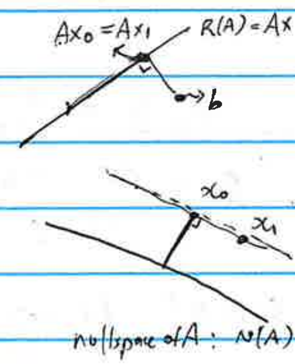


- ① $\|Ax - b\|_2$ minimize ② $\|x\|_2$

①  $(\vec{b} - Ax) \perp Ax$

$\Rightarrow x^T A^T (b - Ax) = 0$

$\Rightarrow A^T A x = A^T b$.

②  if $\operatorname{rank}(A) \neq m$. not full rank!

$x_0 \perp N(A)$

$x_1 = x_1 + v \quad v \in N(A)$

Also, $A = QR \rightarrow R^T \underbrace{Q^T Q}_{=I} R x = R^T Q^T b$

$\Rightarrow R x = Q^T b$.

Since $\operatorname{rank}(A) = \operatorname{rank}(R) \Rightarrow Q^T(Ax - b) = 0$

- power iteration.

$A = X \Lambda X^{-1} \Rightarrow A^k = X \Lambda^k X^{-1}$

$\Rightarrow A^k / \lambda_1^k = x_1 y_1^T$.

- Orthogonal iteration.

$Z = A Q_k$, $Q_{k+1} R_{k+1} = Z \rightsquigarrow A = Q T Q^H$

$\Rightarrow T_k = Q_k^H A Q_k \rightsquigarrow T_\infty = \begin{bmatrix} \lambda_1 & \\ 0 & \lambda_n \end{bmatrix}$

~~$Q_k Q_{k+1} R_{k+1} = A Q_{k+2}$~~ $A Q_k = Q_{k+1} R_{k+1}$

$A Q_{k+1} = Q_{k+2} R_{k+2} \rightarrow A Q = Q R \Rightarrow A = Q R Q^H \rightarrow$ Schur decomp. (T)

- QR iteration.

$T_k = U_{k+1} R_{k+1}$ $\quad U_k$ is orthogonal basis.

$T_{k+1} = R_{k+1} U_{k+1} \longrightarrow T_\infty = \begin{bmatrix} \lambda_1 & \\ 0 & \lambda_n \end{bmatrix}$

$Q^H \quad \simeq U_{k+1}^H T_k U_{k+1}$ $\quad Q$,

$T_\infty = U_\infty^H \cdots U_1^H A (U_1 \cdots U_\infty)$ $\quad O(n^4) \quad o(n^3)$ 3.

$\operatorname{span}(Q) = \operatorname{span}(A)$. $\quad Q^T(Ax - b) = 0$

$A_k = Q_k R_k$

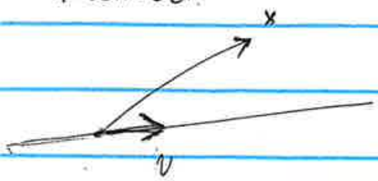
$\rightarrow A_{k+1} = R_k Q_k$

Note: $A_{k+1} = R_k Q_k = Q_k^T A_k Q_k$

$\rightarrow$ Stable (orthogonal). $\downarrow$ Similarity Transformation

Upper Hessen Accelerates

- Householder



$$P = I - \frac{vv^T}{v^T v}$$

$$Q_1^T = I - \frac{a_1 a_1^T}{a_1^T a_1} \qquad \text{project } a_1 \to e_1$$

- Gram-Schmidt.

for $k = 1:n$. ($A \in \mathbb{R}^{m\times n}$).

1) $r_{ik} = q_i^T a_k$
2) $r_{kk} = a_{kk} - \sum_{i<k} r_{ik} q_i$
3) Normalize q_i

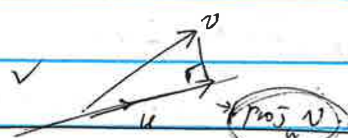
① $q_1 \to q_1/\|q_1\|$

② $a_2 - (q_1 \cdot a_2) q_1 = q_2 \to q_2/\|q_2\|$

③ $a_3 - (q_1 \cdot a_3) q_1 - (q_2 \cdot a_3) q_2 = q_3 \to q_3/\|q_3\|$

...

$$\text{proj}_u v = \frac{v^T u}{u^T u} \cdot u \quad ✓$$



- Givens. $G^T \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix}$, $G = \begin{pmatrix} c & -s \\ +s & c \end{pmatrix}$ $\quad c = a/r$, $s = -b/r$ $\quad \|r\| = \left\| \begin{pmatrix} a \\ b \end{pmatrix} \right\|$

Cholesky (SPD) $\quad A = Q \Lambda Q^T \quad (\lambda_i > 0)$

$a_{ij} = a_{ji}$ and $x^T A x > 0 \quad (x \neq 0)$

$A = LL^T$, L is non-singular

$$a_{ij} = \sum_{j=1}^{i} (l_{ij})^2$$

$$A = U\Sigma V^T \to A^T(Ax - b) = 0 \to V\Sigma^2 \underbrace{V^T x}_{y} = V\Sigma U^T b \to \Sigma y = U^T b.$$

$$\Rightarrow y = \Sigma^{-1} U^T b.$$

$$x = V\Sigma^{-1} U^T b$$

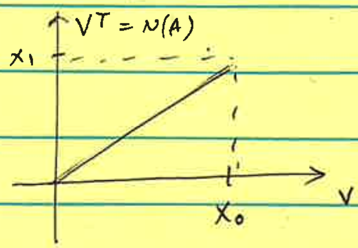
$A = U\Sigma V^T \to$ find $x \perp N(A)$. $\Rightarrow x = Vy$ (best x).

SVD and QR.

10/23/2024.

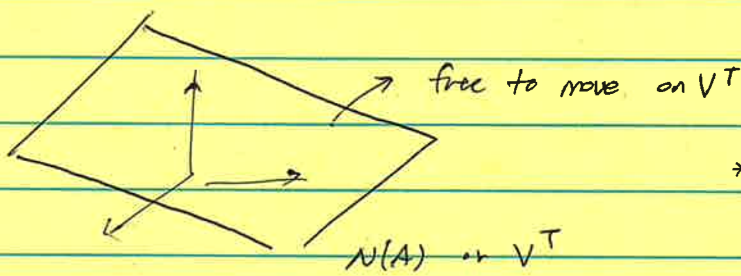
- $V^T x = \Sigma^{-1} U^T b$ ∵ $b - Ax \in N(A^T) = U^T$

$\Rightarrow \underset{\text{spans}}{\underbrace{U^T}} \underset{\text{error.}}{\underbrace{(b - Ax)}} = 0$



$x = \underset{V}{\underbrace{x_0}} + \underset{V^T}{\underbrace{x_1}} \Rightarrow V^T x = V^T(x_0 + x_1) = V^T(Vz + x_1) = z = y$

free move



* just minimize $\|x\|$.

- Eigen-Values.

$Ax = \lambda x$ ~ different from ~ $Ax = b$.

Applications : Companion matrix.

$$A = \begin{pmatrix} 0 & 1 & & & \\ 0 & 0 & 1 & & \\ \vdots & & \ddots & \ddots & \\ & & & 0 & 1 \\ a_0 & a_1 & \cdots & & a_{n-1} \end{pmatrix}$$ so that, $$u = \begin{pmatrix} 1 \\ z \\ \vdots \\ \vdots \\ z^{n-1} \end{pmatrix}$$ then,

$$Au = \begin{pmatrix} z \\ \vdots \\ z^{n-1} \\ a_0 \quad a_1 \cdot z \quad \cdots \quad a_{n-1} z^{n-1} \end{pmatrix} = Zu$$

$\Rightarrow -a_0 - a_1 z - \cdots a_{n-1} z^{n-1} = z\, z^{n-1}$

$\Rightarrow \underline{z^n + a_{n-1} z^{n-1} + \cdots + a_0 = 0}$ → Abel - Ruffini Theorem.

Finding Eigenvalue ≡ Finding root of polynomial.

10/30/2024

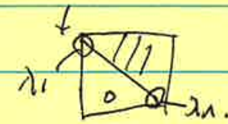
- Effect of power.

$A^k x_1 = \lambda_1^k x_1$.

Recall schur decomposition, $A = QTQ^H$

$Q_0 \to AQ_0 \to Q_1R_1 = AQ_0 \to \cdots \; Z = AQ_k.\; (Q_{k+1}R_{k+1} = Z)$

Then, $Q_k \rightsquigarrow$ converges to $\rightsquigarrow Q$ s.t. $A = QTQ^H$



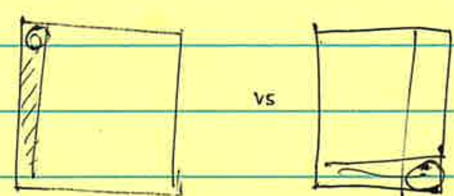
Summary : 1) Multiply by A.

2) QR factorization.

$\Rightarrow$ $T_k = Q_k^H A Q_k$ and $T = Q^H A Q$

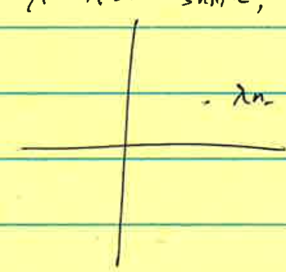
- Rate of convergence.

λ_i, λ_{i+1}.



$\left|\left(\frac{\lambda_2}{\lambda_1}\right)\right|$ $\quad$ $\left|\frac{\lambda_{n-1}}{\lambda_n}\right|$.

$A - \lambda I$ shift, $\lambda_1 - \lambda, \cdots, \lambda_n - \lambda$.



observe, $\left(\frac{|\lambda_n - \lambda|}{|\lambda_{n-1} - \lambda|}\right)^k$.

Algorithm : $\lambda = [T_k]_{nn}$ such that.

- QR iteration with shift.

$T_k = Q_k^H A Q_k$

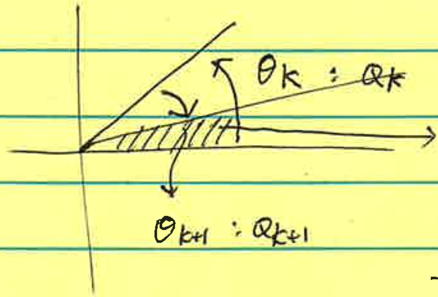
$T_{k+1} = Q_{k+1}^H A Q_{k+1}$

How to do without computing Q_k?

$T_k = Q_k^H A Q_k = \underbrace{Q_k^H Q_{k+1}} R_{k+1}$

$= \underbrace{U_{k+1}} \underbrace{R_{k+1}}_{\text{Upper tri.}}$ } This is literally QR fact. ($\because$ uniqueness of QR).

Therefore, $T_k = Q_k^H T Q_k = Q_k^H Q_{k+1} R_{k+1} = U_{k+1} \cdot R_{k+1}$ —— ①



$\theta_k - \theta_{k+1} : U_{k+1} = Q_k^H \underbrace{Q_{k+1}}_{\text{correction.}}$

→ It will be <u>identity matrix</u>. I.

(making T_k more upper-triangular).

Now., $T_{k+1} = Q_{k+1}^H A Q_{k+1} = Q_{k+1}^H Q_k T_k \cdot Q_k^H Q_{k+1}$

$\Rightarrow T_{k+1} = U_{k+1}^H T_k U_{k+1}$

→ This implies : U_k gets closer to identity matrix.

T_k gets closer to upper-triangular.

$= R_{k+1} U_{k+1}$ ($\because$ ①) ($\because$ $T_k = U_{k+1} R_{k+1}$)

$\therefore$ QR iteration.

① QR fact. : $T_k = U_{k+1} R_{k+1}$

② Update T_{k+1} : $T_{k+1} = R_{k+1} U_{k+1}$.

Repeat! (using magic U_k).

where $U_k = Q_{k-1}^H Q_k \Leftrightarrow Q_k = U_1 \cdots U_k$.

$$\Rightarrow T_\infty = \underbrace{U_\infty^H \cdots U_1^H}_{= Q^H} A \underbrace{U_1 \cdots U_\infty}_{= Q}$$

which agrees with $A = Q T Q^H$ (schur decomposition).

Comp. cost. $\left\{ \begin{array}{l} QR : O(n^3). \\ RQ \quad O(n^3) \end{array} \right\} O(n^3) \times$ <u>iterations</u>.

Do this for $\lambda_1 \sim \lambda_n \Rightarrow \boxed{O(n^4) \text{ complexity}}$

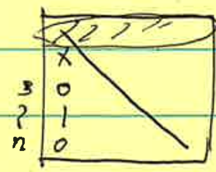
Now using upper-Hessenberg QR $O(n^3) \to O(n^2)$

ex) $A = QTQ^H$

QR

$A = Q \underbrace{H}_{\text{upper Hessenberg.}} Q^H$

$H = Q^H A Q$.



row 1 is untouched.

$Q_1^H A$: Householder trans. when ~~$Q_1^H A$~~ $Q_1^H A = \begin{pmatrix} x & \\ x & \\ 0 & \\ \vdots & \\ 0 & \end{pmatrix}$

$$Q_1^H = \begin{pmatrix} 1 & 0 \; \text{—} \; 0 \\ 0 & \\ \vdots & \boxed{H.T.} \\ 0 & \end{pmatrix} \Rightarrow Q_1^H A Q_1 = \begin{pmatrix} x & \\ x & \\ 0 & \\ \vdots & \\ 0 & \end{pmatrix}$$

Now, $Q^H A Q$ is upper-Hessenberg.

$\downarrow$

QR iter (matrix stays upper-Hessenberg).

QR factorization.

Factorization

$A = QR$, $Q^TQ = I \Rightarrow R = Q^TA$.

① Householder : project $a_1 \to e_1$

$$P = I - \frac{vv^T}{v^Tv} \quad \rightsquigarrow \quad Q_1^T = I - \frac{a_1a_1^T}{a_1^Ta_1}$$

② Givens.

$$G^T\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix} \quad \text{where} \quad G = \begin{pmatrix} c & -s \\ +s & c \end{pmatrix} \quad \begin{matrix} c = a/r \\ s = -b/r \end{matrix} \quad r = \left\|\begin{pmatrix} a \\ b \end{pmatrix}\right\|.$$

③ Gram - Schmidt.

for $k = 1 \sim n$

1) $r_{ik} = q_i^T a_k$

2) $r_{kk} = a_{kk} - \sum_{i<k} r_{ik} q_i$

3) Normalize q_i.

16.

Iterations.

① Power

$$\underbrace{A = X\Lambda X^{-1}}_{\text{eigen - decomposition.}} \to A^k = X\Lambda^k X^{-1} \to \frac{A^k}{\lambda^k} = x_1y_1^T + \cancel{\theta}.$$

② orthogonal.

$A = Q_1R_1$.

$Q_{k+1}R_{k+1} = AQ_k$

$Q_{k+2}R_{k+2} = AQ_{k+1}$

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

$$QR = AQ \to \underbrace{A = Q^TRQ}_{= \text{schur decomposition.}}$$

③ QR iteration.

In ②. $T_k = \underbrace{Q_k^TA}_{}Q_k \to T_{k+1} = \underbrace{Q_{k+1}^TA}_{}Q_{k+1}$

$= Q_{k+1}R_{k+1}$ $\quad = R_{k+1}Q_k^T \cdot Q_{k+1} = R_{k+1}U_{k+1}$

$= \underbrace{Q_kQ_{k+1}}_{U_{k+1}}R_{k+1}$

Example). $A \in \mathbb{R}^{n\times n}$ $a_n \in \text{span}\{a_1 \sim a_{n-1}\}$

Show that last row of $A^{(1)}$ is zero; $A^{(1)} = R_i Q_i$

Since $\text{rank}(A) = n-1$, $\rightarrow R_{nn} = 0 \rightarrow$ Bot. row of $A = 0$. #

- Least Squares. (LS)

$Ax = b$ where $b \notin \text{col}(A)$, find $\min \|Ax - b\|$

① Normal equations

$$A^T(Ax-b) = 0 \Rightarrow \underbrace{A^TA}_{\text{SPD} \leftarrow \text{cholesky.}}x = A^Tb. \Rightarrow x = (A^TA)^{-1} \cdot A^Tb.$$

② QR. (A should be full column rank)

Note $\text{span}(A) = \text{span}(Q)$.

$$Q^T(Ax-b) = 0 \Rightarrow \underbrace{Q^TQ}Rx = Q^Tb$$

$$\Rightarrow Rx = Q^Tb.$$

③ SVD

$$A = U\Sigma V^T \rightarrow A^T(Ax-b) = 0 \Rightarrow V\Sigma \cancel{U^T} \Sigma y = V\Sigma U^Tb.$$

$$\Rightarrow V\Sigma^2 y = V\Sigma U^Tb$$

$A = U\Sigma V^T$

$$\Rightarrow \Sigma y = U^Tb$$

find $x \perp N(A)$

$$\Rightarrow \underbrace{y = \Sigma^{-1}U^Tb}$$

best x

$x = Vy$

$$\Rightarrow x = V\Sigma^{-1}U^Tb. \text{ best } x$$

$N(A)$: $Az = 0 \equiv U\Sigma V^Tz = 0$

$z \perp x \Leftrightarrow z^Tx = 0$ or $x^Tz = 0$ $\rightarrow x \in \{V\Sigma^TU^T\}$

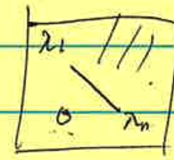
Vy !

11/04/2024.

Problem: $Ax = \lambda x$

① Orthogonal iteration.

$Z = A Q_k$, $Q_{k+1} R_{k+1} = Z$ $\xrightarrow[\text{to}]{\text{Converges}}$ $A = Q T Q^H$

Note $T_k = Q_k^H A Q_k$ → progressively upper triangular &  $= T_\infty$

② QR iteration.

$T_k = U_{k+1} R_{k+1}$

$T_{k+1} = R_{k+1} \cdot U_{k+1}$ $\xrightarrow[\text{Converges}]{}$ $T_\infty =$ 

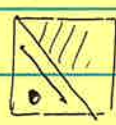
$= \underbrace{U_{k+1}^H T_k}_{= R_{k+1}} \cdot U_{k+1}$

∵ U_k is orthogonal basis.

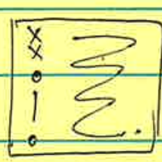
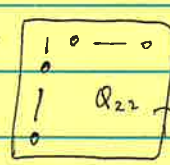
How to accelerate.

① Upper Hessenberg — cost: $O(n^3)$. ② shift.

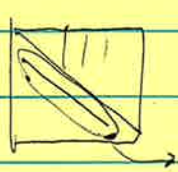
Select Q_0.

$Q_0^H A Q_0 = H =$ 

Then, QR iteration.

$Q_1^H A =$  $Q_1 =$  → Householder Transformation.

⇒ $Q_1^H A Q_1 =$ 

 → Helps us prevent collapse.

• How to QR factorize?

Given's transform (row 1, 2). → This is to find QR factorization.

$Q_{n-1}^T \cdots Q_1^T A = R$

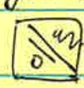
$\Rightarrow A = Q_1 \cdots Q_{n-1} R.$

$$\begin{pmatrix} \times & \times & & \\ \times & \times & & \\ 0 & & \ddots & \end{pmatrix} Q_2 \cdots Q_{n-1} = \begin{pmatrix} \times & \times & \times & \\ \times & \times & \times & \\ & \times & \times & \\ 0 & & & \ddots \end{pmatrix} Q_3 \cdots Q_{n-1}$$

$$= \begin{pmatrix} \ddots & & \\ \ddots & \ddots & \\ 0 & & \ddots \end{pmatrix} \text{ : upper-Hessenberg.}$$

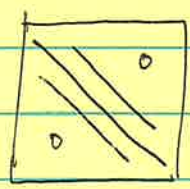
• Non-symmetric vs Symmetric.

$Q^T A Q = H$



If A is sym, H is sym → $H^T = H$ → How ??

⇒ H upper Hessenberg and symmetric, means H is sym. tri-diagonal.



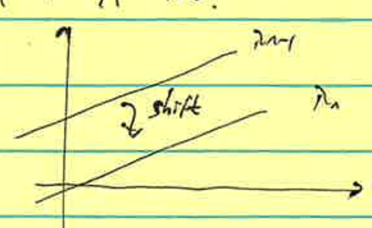
⇒ $QR = H$, cost of each Givens is $O(1)$

QR cost is $O(n)$ vs $O(n^2)$ for gen upp. hessen.

• Shifting

$\Lambda(A) = \Lambda(A_{11}) \cup \Lambda(A_{22})$

$A \to A - \lambda I.$



$$\left| \frac{\lambda_n - \lambda}{\lambda_{n-1} - \lambda} \right|^k \text{ convergence}$$

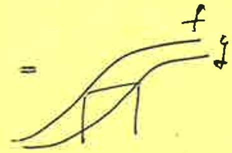
shift

$\lambda = [T_k]_{nn}$ → Converge!

→ $120 \times 5 \times 12$.

→ $600 \times 12 \rightarrow 600 \times 12$.

$b = Ax$.

$\frac{1}{\ell_i} =$ 

$y = g^{-1}(f(x))$.

11/06/2024.

• QR iteration + Convergence.

1) $A - \lambda I$

2) $\left|\frac{\lambda_n - \lambda}{\lambda_{n-1} - \lambda}\right|^k$ is convergence rate.

3).

Algorithm.

$T_k = A$

while not converged.

$\mu = [T_k]_{nn}$

$U_k R_k = T_k - \mu I$. (shift)

$T_{k+1} = R_k U_k + \mu I$ (go back).

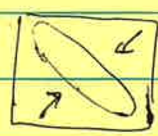
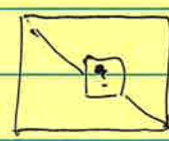
$U_k R_k = T_k - \mu I$.

$$T_{k+1} = \underbrace{U_k^T (T_k - \mu I)}_{R_k} U_k + \mu I$$

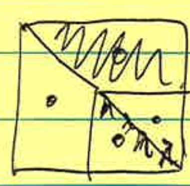
$$= U_k^T T_k U_k - \cancel{\mu I} \cdot \cancel{U_k^T U_k} + \cancel{\mu I} = U_k^T T_k U_k$$

Examples

Unsymmetric : $\left|\frac{\lambda_n}{\lambda_{n-1}}\right| << 1$ → decays very fast. 

symmetric :  if) $\lambda_k \sim \lambda_{k+1} \Rightarrow$ 

what if $\lambda_l = \lambda_{l+1}$ → no convergence.



↓

Schur not unique → anything makes sense in eigen space.

Vacancy formation test.

A : dense matrix

A - sparse : $O(1)$:

Goal : sparse, large $\rightarrow$ dense small.

$x \rightarrow Ax$

$\sum_j a_{ij} x_j = \sum_{\substack{j \\ a_{ij} \neq 0}} a_{ij} x_j$ } store all (i, j, a_{ij}) s.t. $a_{ij} \neq 0$; space : $O(n) = O(nnz)$.

$\Rightarrow \sum_{\substack{j \\ a_{ij} \neq 0}} a_{ij} x_j$ time $O(1)$ per i $\rightarrow$ total time $\boxed{O(nnz)}$

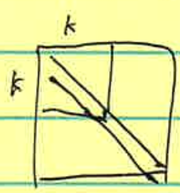
- For sparse matrices,

$A = QHQ^T$ (Householder)

$A = QHQ^T$ (G, S)

$AQ = QH$.

How to $q_1 \sim q_k$ subspace.



Algorithm

choose random q_1

$AQ = QH$

$q_1 \sim q_j$: known

$\rightarrow q_{j+1} = ?$

$\rightarrow Aq_j = q_1 h_{1j} + q_2 h_{2j} + \cdots + q_{j+1} h_{j+1\, j}$ — ②

(stops here ∵ upper Hessenberg)

h_{ij} : proj of Aq_j onto q_i

$\Rightarrow h_{ij} = q_i^T A q_j$ — ①

①, ② $\Rightarrow r_j = Aq_j - h_{1j} q_1 \cdots - h_{jj} q_j$

where $h_{j+1,j} \, q_{j+1} = r_j$

$\Rightarrow h_{j+1,j} = \|r_j\|_2$ so that $\underline{q_{j+1} = r_j / \|r_j\|_2}$.

$\therefore [q_1, Aq_1, \cdots A^{k-1} q_1] = [q_1, \cdots, q_k] \cdot [\text{upper triangular}]$

upper triangular. $\sim R$.

$\sim$(G, S)

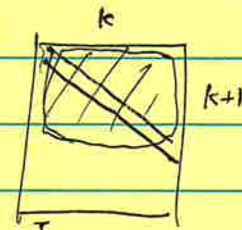
Multiply A → Make orthogonal ↺ $\Rightarrow \{q_1 \ \ Aq_1 \ \cdots \ A^{k-1} q_1\}$

⇒ Krylov subspace.

$AQ = QH$

$Q_k = [q_1 \cdots q_k]$

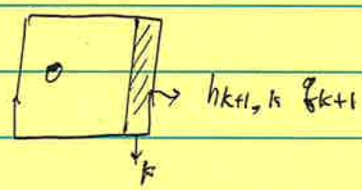
$\Rightarrow \quad AQ_k = Q\underbrace{H[:, :k]}_{k \text{ columns.}} = Q$



$= \underset{[k \times k]}{Q_k H_k} + \underbrace{h_{k+1,k}\, q_{k+1} \underset{(0 \cdots 0\, 1)}{e_k^T}}_{\text{Add to the last } (k^{th}) \text{ column.}}$

$\Rightarrow \quad AQ_k = Q_k H_k + h_{k+1,k}\, q_{k+1}\, e_k^T$

$\parallel$



Then, $AQ_k \approx Q_k H_k \rightarrow H_k = Q_k^T A Q_k$

"spaned by eig. vectors."

If $h_{k+1,k} = 0$, Q_k is stable subspace ($\because AQ_k = Q_k H_k$)

Then, $\boxed{\text{eig vals of } H_k = \text{eig. vals of } A}$

11/11/2024.

- A is sparse. $\rightarrow$ $O(1)$ non-zero per row.

$A = QHQ^T \Leftrightarrow AQ = QH$

$$\left\{ \begin{aligned} & Aq_j = h_{1j}\, q_1 + h_{2j}\, q_2 + \cdots + h_{jj}\, q_j + h_{j+1,j}\, q_{j+1}. \\ & \text{where } h_{ij} = q_i^T A q_j \\ & h_{j+1,j}\, q_{j+1} = A q_j - \sum_{i=1}^{j} h_{ij}\, q_i \\ & h_{j+1,j} = \|r_j\|_2 \rightarrow q_{j+1} = r_j / \|r_j\|_2. \end{aligned} \right.$$

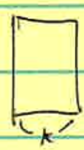
$\rightarrow q_1 \rightarrow Aq_1 \rightarrow q_2 \rightarrow Aq_2 \rightarrow q_3 \rightarrow \cdots \rightarrow$ Krylov subspace. $= K_k$

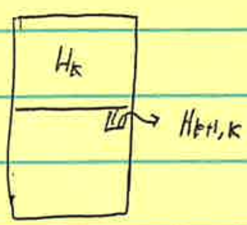
$= \text{span}\{q_1, Aq_1, \cdots, A^{k-1}q_1\}$.

$\therefore K_k = \text{span}\{q_1, \cdots, q_k\}$

$\rightarrow$ When K becomes complete, converges after all iterations..

- $AQ = QH$

Q_k :  $\Rightarrow AQ_k = QH[:, 0:k]$, $H_k = H[:k, :k]$



$AQ_k = Q_k H_k + h_{k+1,k}\, q_{k+1}\, e_k^T$

If q_1 belongs to 'stable subspace' of dimm k.

then, $h_{k+1,k} = 0$ (no more basis needed).

Assume that $h_{k+1,k}$ is small $\equiv AQ_k \approx Q_k \underbrace{H_k}_{k \times k}$.

Ritz eigenvalue.

$H_k = X_k \Lambda_k X_k^{-1} \Rightarrow AQ_k \approx Q_k X_k \Lambda_k X_k^{-1}$

$A(Q_k X_k) \cong (Q_k X_k)\Lambda_k$ eig vecs, eig. vals.

$H_k = Q_k^T A Q_k \rightarrow$ approximate Ritz (eg.)

$$\left.\begin{array}{l} ① \ k \text{ sparse mat. vec. products + operations.} \\ ② \text{ Dense calculation on small } k\times k \text{ matrix} \end{array}\right\} \quad O(k^3)$$

- Conjugate Gradients $(Ax = b)$.

A is sym. pos. def. (SPD) : k steps $\rightarrow Q_k$, $x_k = Q_k \cdot y$.

$$A Q_k y \approx b.$$

method : $A Q_k y \approx b \rightarrow Q_k^T (A Q_k y) = Q_k^T b.$

$$\downarrow$$

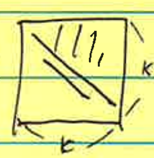
$$\underbrace{H_k}_{k\times k} y = Q_k^T b.$$

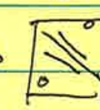
$\rightarrow$ CG solution $\Rightarrow x_k = Q_k y.$

How to choose q_1 in CG ?

$$q_1 = b / \|b\|_2 \text{, then } Q_k^T b = \begin{bmatrix} - q_1 - \\ - q_2 - \\ \vdots \\ - q_k - \end{bmatrix} \begin{bmatrix} | \\ b \\ | \end{bmatrix} = \begin{pmatrix} \|b\|_2 \\ \\ 0 \\ \\ \end{pmatrix} = \|b\|_2 \cdot \vec{e}_1.$$

- Arnoldi Process — applies to general matrix A.

$Q_k, H_k =$  , $H_k = Q_k^T A Q_k$. if A is symmetric, H is symm. tridiag.

$\Rightarrow$ "Lanczos Algorithm" 

$$H_k = \begin{bmatrix} \alpha_1 & \beta_1 & & 0 \\ \beta_1 & \alpha_2 & \beta_3 & \\ & \beta_3 & \alpha_3 & \\ 0 & & & \ddots \end{bmatrix} \Rightarrow A q_k = \beta_{k-1} q_{k-1} + \alpha_k q_k + \beta_{k} q_{k+1}$$

already computed.

$q_1 \rightarrow q_2 \rightarrow \cdots \rightarrow q_k$

$\alpha_k = q_k^T A q_k.$

$\Rightarrow$ $\boxed{\text{Lanczos algorithm}}$

Loop over k

$\alpha_k = q_k^T A q_k$

$r_k = A q_k - \beta_{k-1} q_{k-1} - \alpha_k q_k \rightarrow \beta_k = \|r_k\|_2 \rightarrow q_{k+1} = \dfrac{r_k}{\beta_k}.$

11/13/2024

- Arnoldi / Lanczos.

H_k : $k \times k$, $\underbrace{\Lambda(H_k)}_{k \text{ extremal eig. vals of } A} \cong \Lambda(A)$.

Lanczos : $A = QTQ^H$, $\alpha_k = q_k^T A q_k$.

$$\beta_{k+1} q_{k+1} = A q_k - \beta_{k-1} q_{k-1} - \alpha_k \cdot q_k$$

$\Rightarrow$ $\text{span}\{q_1 \sim q_k\} = \underbrace{\text{span}\{q_1, \cdots, A^{k-1} q_1\}}_{\text{Krylov subspace}}$ $\Rightarrow$ H_k / T_k.

Ex) In C.G., $x_k = Q_k \cdot y$ $\Rightarrow$ $T_k y = Q_k^T b = \|b\|_2 e_1$

- How to solve $Ax = b$ (A is sparse)

$x_1 \Rightarrow \cdots \rightarrow x_k = x$. , only use products (no ~~factorization~~)

① Conjugate Gradient.

$x_k = Q_k \cdot y$ in Krylov subspace. $\rightarrow$ next time

② Splitting Methods.

$Ax = b$ $\rightarrow$ $(A - M)x = b - Mx$ $\Rightarrow$ $Mx = b - (A - M)x$.

$\Rightarrow$ $M x^{k+1} = b - (A - M) x^k$

$x^{k+1} = M^{-1} b - M^{-1}(A - M) x^k = M^{-1} b - M^{-1} A + x^k$

$\Rightarrow x^{k+1} \doteq x_k + M^{-1}(b - A x^k)$.

$\Rightarrow$ $M x^{(k+1)} = b - (A - M) x^{(k)}$

where $A = M - N$

$\Rightarrow$ $\boxed{M x^{(k+1)} = b - N x^{(k)}}$ —— ①

$Mx = b - (A - M)x$ —— ②

$e^{(k)} = x^{(k)} - x$ —— ③

$M e^{(k+1)} = 0 + N e^{(k)}$

$\Rightarrow$ $e^{(k+1)} = M^{-1} \cdot N \cdot e^{(k)}$

$e^{(k)} = (M^{-1} N)^k e^{(0)}$

Therefore, $e^{(k)} = \underbrace{(M^{-1}N)^k}_{\to \text{ we can use largest eig value.}} e^{(0)}$

* Convergence.

$M^{-1}N = X \Lambda X^{-1} \to (M^{-1}N)^k = X \Lambda^k X^{-k} \equiv |\lambda_i| < 1 \quad \forall i$

$\Rightarrow \underbrace{\rho(A)}_{\text{spectral radius} = \max|\lambda_i|} < 1$

Denote $M^{-1}N = G$, $A = M - N$, $N = M - A$, $G = I - M^{-1}A$.

$\Rightarrow \rho(G) < 1$ for convergence.

Ex) Jacobi

$M = \text{diag}(A)$, $A = D - \overset{\text{lower}}{L} - \underset{\text{upper}}{U}$

$\Rightarrow D x^{(k+1)} = b + N x^{(k)} = \underbrace{(L+U)}_{\text{off-diag}} x^{(k)} + b$

$\Rightarrow G = M^{-1}N = D^{-1}(L+U)$.

For convergence, $\rho(G) < 1 \Rightarrow A = \begin{pmatrix} D & -U \\ -L & \end{pmatrix} \Rightarrow$ "Large Diagonal Entries".

Thm) If A is strictly row / col diagonally dominant, Jacobi converges.

Row diag. dom : $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$

Col diag dom. : $|a_{ii}| > \sum_{k \neq i} |a_{ki}|$

- Gauss - Siedel.

$A = D - L - U$

$M = D - L$ more accurate. (lower solve).

$\Rightarrow (D-L)\, x^{(k+1)} = b + U x^{(k)}$.

Convergence, $\rho\left((D-L)^{-1} U \right) < 1 \Leftarrow$ A is strictly $\binom{\text{row}}{\text{col}}$ diag dominant. G.S. converges.

- Thm: If A is SPD, then G.S. converges.

(# itr. G.S) $\approx \frac{1}{2}$ (# itr J.)

Easy to parallelize.

$D x^{(k+1)} = b + (L+U)\, x^{(k)}$: J

$(D-L)\, x^{(k+1)} = b + U x^{(k)}$. : G.S.

lower $\triangle$ → sequential → parallelize X

11/18/2024.

Conjugate Gradient.

- $Ax = b.$
- $K_k = \text{span}\{b, Ab, \cdots, A^{k-1}b\}.$

  $x_k = Q_k \cdot y.$

  $T_k y = \|b\|_2 e_1$
- $\text{span}(\underbrace{q_1, \cdots q_k}_{\text{Lanczos}}) = K_k.$

$\rightarrow p_1, \cdots, p_k$ s.t. $\text{span}\{p_1 \sim p_k\} = K_k.$

$X = P\mu.$

$X_k = P_k \mu_k.$

$X_k = \mu_1 p_1 + \cdots + \mu_k p_k$

$X_{k+1} = X_k + \underbrace{\mu_{k+1} \cdot p_{k+1}}_{?}.$ $\rightarrow$ use $x = P\mu$ and $Ax = b.$

$\Rightarrow AP\mu = b \Rightarrow P^T(AP\mu) = P^Tb \Rightarrow \boxed{(P^TAP)}\mu = P^Tb.$

p_i^Tb easy to calculate.

$\rightarrow$ choose P s.t. $\boxed{P^TAP = \text{diagonal matrix.}} = D.$

Then, $d_i \mu_i = p_i^Tb \Rightarrow \mu_i = p_i^Tb / d_i$

$\Rightarrow X_{k+1} = X_k + \mu_{k+1} \cdot p_{k+1}.$

- How P^TAP diagonal?

$\langle p_i, p_j \rangle = p_i^T A p_j$ use A is sym. pos. def. $\Leftrightarrow A = Q\Lambda Q^T$

$$\Rightarrow p_i^T A p_j = p_i^T Q\Lambda Q^T p_j = \boxed{(\Lambda^{1/2} \underbrace{Q^T p_i}_{})^T (\Lambda^{1/2} \underbrace{Q^T p_j}_{}) \quad \text{Transformation.}}$$

$$\langle p_i, p_i \rangle = p_i^T A p_i \; > 0 \quad p_i \neq 0$$
$$= 0 \quad p_i = 0.$$

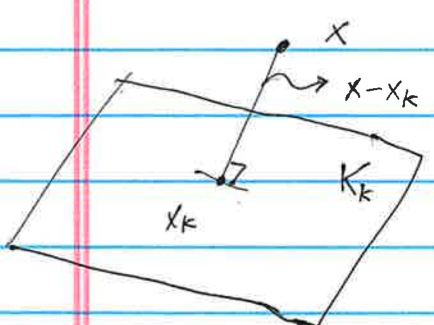
$$\|z\|_A^2 = z^T A z = \|\Lambda^{1/2} Q^T z\|_2^2$$

- Lanczos $\rightarrow q_i$, $Q^T Q = I$
  CG $\rightarrow p_i$, $P^T A P =$ diagonal.

$$\Rightarrow x = \underbrace{\mu_1 p_1 + \mu_2 p_2 + \cdots + \mu_k p_k}_{\text{CG} \quad K_k} + \cdots + \mu_n p_n.$$

$$x - x_k = \mu_{k+1} p_{k+1} + \cdots + \mu_n p_n$$

$\Rightarrow$ $x - x_k \perp_A K_k \rightsquigarrow$ Least Squares!



$\Rightarrow$ $x_k = \operatorname{argmin} \| P_k y - x \|_A$

$$\underline{r_k = b - A x_k}$$
residuals.

- $\operatorname{span}\{p_1 \sim p_k) = K_k = \operatorname{span}(r_0, \cdots, r_{k-1})$
  $r_k \in K_{k+1}$, $\operatorname{span}\{p_1 \sim p_k\} = K_k. = ?$
  $\Rightarrow q_1 = b / \|b\|_2 \rightsquigarrow K_1 = \operatorname{span}(b)$
  $b \in K_{k+1}$, $A x_k \in K_{k+1}$, $r_k \in K_{k+1}$. //

- Patterns

$P^T R = ?$ $\quad p_i^T r_j \quad i<j \to 0 \quad \Rightarrow r_k \perp K_k . \to$ Lower Triangular

$\Rightarrow r_k \perp K_k$ and $r_k \in K_{k+1}$

$R = [r_0, r_1, \ldots, r_{k-1}]$

$\mathrm{span}(r_0 \sim r_{k-1}) = \mathrm{span}(p_1 \sim p_k)$

$R = P\underbrace{U}_{\text{upper-triangular matrix}}$

Lanczos $Aq_k \to q_{k+1}$

CG $r_k \to p_{k+1}$

$R = PU.$

bi-diagonal matrix.

$R = PU.$

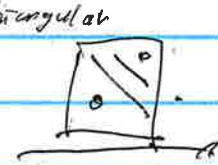
$P^T A R = (P^T A P)U = P.U$

$= (AP)^T R = (PH)^T R = H^T P^T R = H^T L$

"Lower Hessenberg"

Upper Hessenberg + Upper triangular

= Bi-diagonal



$R = PU \to r_{k-1} = U_{k\,k-1} P_k + U_{k-1\,k-1} P_{k-1}$

$r_k = U_{k+1,\,k+1} \cdot P_{k+1} + U_{k\,k+1} P_k$

$Q^T A Q$

- Conjugate Gradient.

- $Ax = b$., A is SPD, $x_0 = 0$, $K_k = \text{span}\{p_1 \sim p_k\}$. $x = P\mu$.

  ↳ $AP\mu = b$, $\underbrace{P^T A P}_{\text{Choose } D}\mu = P^T b \Rightarrow D\mu = P^T b$.

  $\mu_k = p_k^T b / d_k$

- Note : $\langle y, z\rangle_A = y^T A z$, $A = Q\Lambda Q^T = (\Lambda^{1/2} Q^T y)^T (\Lambda^{1/2} Q^T z)$

  Then, $\|z\|_A^2 = z^T A z \Rightarrow p_i^T A p_j = 0$ if $i \neq j$

  $\mu^{(k)} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_k \end{pmatrix}$, $P_k = (p_1, \cdots, p_k)$

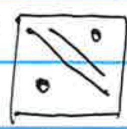
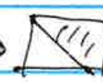
- Efficient to do residuals $r_k = b - A\underbrace{x_k}_{\in K_k}$ // q_{k+1}

  where $b = \|b\|_2 q_1 \in K_1 \to r_{k+1} \in K_{k+1}$ and

  $r_k \perp K_k \Rightarrow p_\ell^T r_k = 0$ (if $\ell \leq k$).

- $P^T R = L$ (lower triangle) $\Rightarrow \ell_{ij} = p_i^T r_{j-1} = 0$ if $i < j$

  $r_k \in K_{k+1} \Rightarrow R = PU$ (QR fact.)

  $R = PU = P$  $\Rightarrow P^T A R = P^T A P U = DU \to$ 

- $x_{k+1} = x_k + \mu_{k+1} \cdot p_{k+1}$ ①

  $r_k = b - A x_k$.

  $\Rightarrow -A \cdot ① \Rightarrow -A x_{k+1} = -A x_k - \mu_{k+1} \cdot A p_{k+1}$

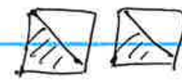
  $r_{k+1} = r_k - \mu_{k+1} A p_{k+1}$

  iterate.

Now, $\mu_k = p_k^T b / d_k$

$u_{k,k+1} = p_k^T A r_k / d_k$

Start from $r_k^T r_l = 0 \ (k \neq l)$. prove that $R^T R$ is diagonal.

① $R = PU$, $R^T R = (PU)^T R = U^T P R = U^T L =$ 

$\Rightarrow$ $R^T R$ is lower-triangular matrix.

② since $R^T R$ is symmetric + L.T $\Rightarrow$ $\underline{\text{diagonal matrix}}$.

- p_k A conjugate
- $r_k \perp K_k$
- $r_k \perp r_l$ if $k \neq l$.

① $p_k^T b = p_k^T \cdot (b - A x_{k-1})$ $\because$ p_k is A-orthogonal to K_{k-1}

$= p_k^T r_{k-1}$

$\Rightarrow$ $p_k = r_{k-1} - u_{k-1,k} \cdot r_{k-1}$ (recall this).

$\Rightarrow$ $p_k^T r_{k-1} = r_{k-1}^T r_{k-1} - u_{k-1,k} \underbrace{p_{k-1}^T r_{k-1}}_{0} = \| r_{k-1} \|_2^2$

$\Rightarrow$ $\mu_k = \dfrac{\| r_{k-1} \|_2^2}{d_k}$ //

② $u_{k,k+1} = \dfrac{p_k^T A r_k}{d_k}$, $p_k^T A r_k = (A p_k)^T r_k = \dfrac{-1}{\mu_k} (r_k - r_{k-1})^T r_k = \underline{-\dfrac{\| r_k \|_2^2}{\mu_k}}$

$\left((r_k - r_{k-1}) = -\mu_k A p_k \right)$

$\Rightarrow$ $u_{k,k+1} = \dfrac{-\| r_k \|_2^2}{d_k \cdot \mu_k} = - \dfrac{\| r_k \|_2^2}{\| r_{k-1} \|_2^2} = -\tau_k$

- Algorithm (CG).

$x_0 = 0, \quad r_0 = b, \quad p_1 = b$

loop k. $k=1$

$\mu_k = \dfrac{\|r_{k-1}\|_2^2}{p_k^T A p_k}$

$x_k = x_{k-1} + \mu_k p_k$

$r_k = r_{k-1} - \mu_k A p_k$

$\tau_k = \|r_k\|_2^2 / \|r_{k-1}\|_2^2$

$p_{k+1} = r_k + \tau_k p_k$

Note : $A p_k$: sparse mat-vec., only when A is SPD.

In Lanczos, $T_k y = \|b\|_2 e_1$, $x_k = Q_k y$ | $r_k \perp K_k$ in CG

$T_k = Q_k^T A Q_k$

$\Lambda(T_k) = \Lambda(A)$

$r_k = b - A x_k$

$\quad = b - A Q_k y$

$Q_k^T r_k = 0$

$\Rightarrow Q_k^T b = Q_k^T A Q_k y = T_k y$ where $Q_k^T b = \|b\|_2 e_1$

- Summary

$x = P\mu$ where $P^T A P = D$. (P is A orthogonal)

$\|z\|_A^2 = z^T A z$

$r_k = b - A x_k$. when $R = PU$

$P^T R = L$

$\Rightarrow K_k = \text{span}\{b, Ab, \cdots, A^{k-1} b\}$

$\quad = \text{span}\{q_1 \sim q_k\}$

$\quad = \text{span}\{p_1 \sim p_k\}$

$\quad = \text{span}\{r_0 \sim r_{k-1}\}$

optimality of C.G.

CG minimizes A-norm of error

$\min_y \|P_k y - x\|_A$

Note : $r_k = b - A x_k \in K_{k+1}$, $x_k \in K_k$ $\Rightarrow r_{k-1} \in K_k$

$x_k \in K_k$

$y \in K_k \Rightarrow Ay \in K_{k+1}$

$\rightarrow$ ① p_k are A orthogonal

② r_k are orthogonal each other

③ $r_k \perp K_k$