1 Solving System of Linear Equations $Ax = b$

1.1 p-Norm and Condition Number

Vector p-Norm: $\boxed{\|\vec{x}\|_p = \sqrt[p]{\sum_i |x_i|^p}}$

$$\text{1-Norm}: \quad \|\vec{x}\|_1 = \sum_i |x_i|$$

$$\infty\text{-Norm}: \quad \|\vec{x}\|_\infty = \max |x_i|$$

- $\|x\|_1 \geq \|x\|_2 \geq \|x\|_\infty$
- $\|x\|_1 \leq \sqrt{n}\, \|x\|_2 \leq \sqrt{n}\, \|x\|_\infty$

Function/Vector Condition Number:

$$\operatorname{cond}\Big(f(x)\Big) = \left|\frac{[f(\hat{x}) - f(x)]/f(x)}{[\hat{x} - x]/x}\right| = \left|\frac{\Delta y/y}{\Delta x/x}\right| = \left|\frac{y' \cdot \Delta x/y}{\Delta x/x}\right| = \left|\frac{x f'(x)}{f(x)}\right|$$

Matrix p-Norm: $\boxed{\|A\|_p = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}}$

$$\text{1-Norm}: \quad \|A\|_1 = \max_j \sum_i |a_{ij}|$$

$$\infty\text{-Norm}: \quad \|A\|_\infty = \max_i \sum_j |a_{ij}|$$

- $\|AB\| \leq \|A\| \cdot \|B\|$
- $\|Ax\| \leq \|A\| \cdot \|x\|$

For p-norms (not necessarily in general)

Matrix Condition Number:

$$\boxed{\operatorname{cond}_p(A) = \|A\|_p \cdot \|A^{-1}\|_p} \qquad (\infty \text{ if singular})$$

$$= \frac{\max_{x \neq 0} \|Ax\|_p / \|x\|_p}{\min_{x \neq 0} \|Ax\|_p / \|x\|_p} = \operatorname{cond}_p(\gamma A) \geq 1$$

- Diagonal, D: $\operatorname{cond}(D) = \frac{\max |d_i|}{\min |d_i|}$
- $\|z\| = \|A^{-1}y\| \leq \|A^{-1}\| \cdot \|y\|$

$\rightarrow \frac{\|z\|}{\|y\|} \leq \max \frac{\|z\|}{\|y\|} \stackrel{?}{=} \|A^{-1}\|$ (optimize)

1.2 Error Bounds and Residuals

Error Bound: $\boxed{\dfrac{\|\hat{x}-x\|}{\|x\|} \lessapprox \operatorname{cond}(A)\ \epsilon_{\text{mach}}}$ $\rightarrow$ A computed solution is expected to lose about $\log_{10}(\operatorname{cond}(A))$ digits, so the input data must be more accurate to these digits and the working precision must carry more than these digits.

$A\hat{x} = b + \Delta b = Ax + A\Delta x$

- $\|b\| \leq \|A\| \cdot \|x\|$
- $\|\Delta x\| \leq \|A^{-1}\| \cdot \|\Delta b\|$

$\rightarrow \boxed{\frac{\|\Delta x\|}{\|x\|} \leq \operatorname{cond}(A)\ \frac{\|\Delta b\|}{\|b\|}}$

$A\hat{x} + r = b$

- $\|\Delta x\| = \|A^{-1}(A\hat{x} - b)\| = \|-A^{-1}r\|$
  $\leq \|A^{-1}\| \cdot \|r\|$

$\rightarrow \boxed{\frac{\|\Delta x\|}{\|\hat{x}\|} \leq \operatorname{cond}(A)\ \frac{\|r\|}{\|A\|\cdot\|\hat{x}\|}}$

$(A + \Delta A)\hat{x} = b$

- $\|\Delta x\| = \|-A^{-1}(\Delta A)\hat{x}\|$
  $\leq \|A^{-1}\| \cdot \|\Delta A\| \cdot \|\hat{x}\|$

$\rightarrow \boxed{\frac{\|\Delta x\|}{\|x\|} \leq \operatorname{cond}(A)\ \frac{\|\Delta A\|}{\|A\|}}$

$(A + \Delta A)\hat{x} = b$

- $\|r\| = \|b - A\hat{x}\| = \|\Delta A \cdot \hat{x}\|$
  $\leq \|\Delta A\| \cdot \|\hat{x}\|$

$\rightarrow \boxed{\frac{\|r\|}{\|A\|\cdot\|\hat{x}\|} \leq \frac{\|\Delta A\|}{\|A\|}}$, $\frac{\|\Delta x\|}{\|x\|} \leq \frac{\|A^{-1}\|\cdot\|r\|}{\|\hat{x}\|} \leq \operatorname{cond}(A)\ \frac{\|\Delta A\|}{\|A\|}$

$$\Big[A(t)x(t) \ = \ b(t)\Big] = \Big[\big(A_0 + \Delta A \cdot t\big)x(t) \ = \ b_0 + \Delta b \cdot t\Big]$$

- $x'(t) = \frac{b'(t) - A'(t)x(t)}{A(t)} = A^{-1}(t)\Big[\Delta b - \Delta A \cdot x(t)\Big]$
- $x(t) = x_0 + x'(0)t + \mathcal{O}(t^2)$

$\rightarrow \boxed{\frac{\|x(t)-x_0\|}{\|x_0\|} \leq \operatorname{cond}(A)\left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|}\right)|t| + \mathcal{O}(t^2)}$

1.3 Gaussian Elimination with LU/PLU/PLDUQ Decomposition

Elementary Elimination Matrices, L_k

$$\begin{pmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & \frac{-a_{k+1}}{a_k} & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \frac{-a_n}{a_k} & 0 & \dots & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_k \\ a_{k+1} \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} a_1 \\ \vdots \\ a_k \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

- a_k is the "pivot"
- is lower triangular
- $\forall i{\neq}j \quad (L_k^{-1})_{ij} = -(L_k)_{ij}$

Ex :

$$\begin{pmatrix} 1 & 0 & \dots \\ -a_1/a_2 & 1 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} a_1 \\ 0 \\ \vdots \end{pmatrix}$$

LU/PLU Factorization (w/ partial pivoting)

$$\boxed{\begin{array}{ll} A = LU & \begin{array}{l}(L \text{ is gen. triang.}) \\ (U \text{ is upp. triang.})\end{array} \\ L = (\dots L_2 P_2 L_1 P_1)^{-1} & \end{array}}$$

$$\{\dots\}b = (\dots L_2 P_2 L_1 P_1)Ax$$

$$L^{-1}b = (P_1^T L_1^{-1} P_2^T L_2^{-1} \dots)^{-1} Ax$$

$$= L^{-1}(LU)x = y$$

$$\boxed{\underset{\text{(forw.-sub.)}}{b = Ly} \;,\; \underset{\text{(back.-sub.)}}{y = Ux}}$$

- Permutation matrix, P_i, rowswaps s.t. $a_k \neq 0$
- P_i rowswaps s.t. a_k is largest s.t. $a_{k+i}/a_k \leq 1$ for numerical stability/ minimize errors
- Pivoting isn't needed if A is diag. dom. $(a_{jj} > \sum_{i,i\neq j} a_{ij})$
- A can be singular

$$\boxed{\begin{array}{ll} A = PLU & \begin{array}{l}(P \text{ is rowswap permu.}) \\ (L \text{ is unit low. triang.}) \\ (U \text{ is upp. triang.})\end{array} \\ P = (\dots P_2 P_1)^{-1} & \end{array}}$$

$$\{\dots\}b = (\dots P_2 P_1)Ax$$

$$P^T b = (P_1^T P_2^T \dots)^{-1} Ax$$

$$= P^T(PLU)x = Ly$$

$$\boxed{P^T b = Ly \;, \quad y = Ux}$$

$$\boxed{P^T A = LDU \qquad \text{(D is diag.)}}$$

- LDU is unique up to D
- LDU is unique if L/U are unit low./upp. diag., resp.

$$\boxed{P^T A Q^T = LDU \qquad \begin{array}{l}\text{(P is permu. for rows)} \\ \text{(Q is permu. for cols.)}\end{array}}$$

- "Complete pivoting" search for largest a_k
- Would be most numerically stable
- Expensive, so not really used

Error Bound: $\frac{\|r\|}{\|A\|\|x\|} \leq \frac{\|\Delta A\|}{\|A\|} \leq \underset{\text{(Wilkinson)}}{\rho\, n^2 \epsilon_{\text{mach}}} \sim \underset{\text{(usually)}}{n\epsilon_{\text{mach}}}$

(growth factor, ρ, is the largest entry at any point during factorization - usually at U - divided by the largest entry of A)

1.4 Gaussian-Jordan with MD Decomposition

Elementary Elimination Matrices, M_k

$$\begin{pmatrix} 1 & \cdots & \frac{-a_1}{a_k} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & \frac{-a_{k+1}}{a_k} & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{-a_n}{a_k} & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_{k-1} \\ a_k \\ a_{k+1} \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ a_k \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

- a_k is the "pivot"
- $\forall i \neq j \quad (M_k^{-1})_{ij} = -(M_k)_{ij}$

MD Factorization (w/ partial pivoting)

$$\boxed{\begin{array}{l} A = MD \quad \begin{array}{l}(M \text{ is elem. elim.}) \\ (D \text{ is diag.})\end{array} \\ M = (\ldots M_2 P_2 M_1 P_1)^{-1} \end{array}}$$

$$\{\ldots\}b = (\ldots M_2 P_2 M_1 P_1) A x$$

$$M^{-1} b = (P_1^T M_1^{-1} P_2^T M_2^{-1} \ldots)^{-1} A x$$

$$= M^{-1}(MD)x = y$$

$$\boxed{M^{-1} b = y \;, \quad y = Dx \;\; \text{(division)}}$$

- Permutation matrix, P_i, rowswaps s.t. $a_k \neq 0$
- P_i rowswaps cannot ensure numerical stability (≤ 1)
- Division is $\mathcal{O}(n)$, so may be useful for parallel comps.
- Can also find A^{-1}

Finding A^{-1}

$$D^{-1} M^{-1} (A|I) = (I|A^{-1})$$

$$= D^{-1} M^{-1} \left[\begin{array}{cc|cc} a_{11} & \cdots & 1 & 0 \\ \vdots & a_{nn} & 0 & 1 \end{array}\right]$$

$$= \left[\begin{array}{cc|cc} 1 & 0 & a'_{11} & \cdots \\ 0 & 1 & \vdots & a'_{nn} \end{array}\right]$$

1.5 Symmetric Matrices

Positive Definite: $\boxed{x^T A x \geq 0}$

Cholesky Factorization for Sym., Pos. Def.: $\boxed{A = LL^T = LDL^T}$

$$\begin{pmatrix} a_{11} & a_{21} & a_{31} & \cdots \\ a_{21} & a_{22} & a_{32} & \cdots \\ a_{31} & a_{32} & a_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 & \cdots \\ l_{21} & l_{22} & 0 & \cdots \\ l_{31} & l_{32} & l_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & l_{31} & \cdots \\ 0 & l_{22} & l_{32} & \cdots \\ 0 & 0 & l_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} = \begin{pmatrix} l_{11}^2 & \cdots & \cdots & \cdots \\ l_{21} l_{11} & l_{21}^2 + l_{22}^2 & \cdots & \cdots \\ l_{31} l_{11} & l_{31} l_{21} + l_{32} l_{22} & l_{31}^2 + l_{32}^2 + l_{33}^2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

- Pivoting not needed
- Well defined (always works)
- Only lower triangle needed for storage
- $A = LDL^T$ is sometimes useful, where D is diag.

Symmetric Indefinite Matrices

- Pivoting Needed : $\boxed{PAP^T = LDL^T}$
- Ideally, D is diag., but if not possible, then D is tridiag. (Aasen) or 1x1/2x2 block diag. (Bunch, Parlett, Kaufmann, etc.)

1.6 Banded Matrices

- Similar to normal Gaussian Elim., but less work since more zeroes
- Pivoting means bandwidth will expand no more than double
- Only $\mathcal{O}(\beta n)$ storage needed

1.7 Rank-1 Update with Sherman-Morrison

$$\tilde{A}\tilde{x} = b = (A - uv^T)\tilde{x}$$

$$\rightarrow \ \tilde{x} = \tilde{A}^{-1}b$$

$$\boxed{\begin{array}{rcl} \tilde{A}^{-1} \ = \ (A - uv^T)^{-1} & = & A^{-1} + \frac{A^{-1}u}{1 - v^T(A^{-1}u)} \ v^T A^{-1} \\ \tilde{A}^{-1}b \ = \qquad \tilde{x} & = & (A^{-1}b) + \frac{A^{-1}u}{1 - v^T(A^{-1}u)} \ v^T(A^{-1}b) \\ & & x + \frac{y}{1 - v^T y} \ v^T x \end{array}}$$

General Woodbury Formula: $\boxed{(A - UV^T)^{-1} = A^{-1} + (A^{-1}U)(I - V^T A^{-1}U)^{-1} \ v^T A^{-1}}$

- U and V are general $n \times k$ matrices
- No guarantee of numerical stability, so caution is needed

1.8 Complexity

$$\begin{array}{rccccc}
\text{Explicit Inversion :} & \begin{array}{rcl} LUA^{-1} & = & I \\ D^{-1}M^{-1}I & = & A^{-1} \end{array} & \rightarrow \ \mathcal{O}(n^3) & , & A^{-1}b = x & \rightarrow \ \mathcal{O}(n^2) \\
\text{Gaussian Elimination :} & A = LU & \rightarrow \ \mathcal{O}(n^3/3) & , & LUx = b & \rightarrow \ \mathcal{O}(n^2) \\
\text{Gaussian-Jordan :} & A = MD & \rightarrow \ \mathcal{O}(n^3/2) & , & MDx = b & \rightarrow \ \mathcal{O}(n) \\
\text{Symmetric :} & \begin{array}{rcl} A & = & LL^T \\ PAP^T & = & LDL^T \end{array} & \rightarrow \ \mathcal{O}(n^3/6) & , & LL^Tx = b & \rightarrow \ \mathcal{O}(n^2) \\
\text{Banded :} & A_\beta = LU & \rightarrow \ \mathcal{O}(\beta^2 n) & , & LUx = b & \rightarrow \ \mathcal{O}(\beta n) \\
\text{Sherman-Woodbury :} & \tilde{A} = A - uv^T & \rightarrow \ \mathcal{O}(n^2) & , & \tilde{x} = \tilde{A}b & \rightarrow \ \mathcal{O}(n^2)
\end{array}$$

1.9 Diagonal Scaling

Ill-conditioned

$$\begin{pmatrix} 1 & 0 \\ 0 & \epsilon \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ \epsilon \end{pmatrix}$$

Well-conditioned

$$\begin{pmatrix} 1 & 0 \\ 0 & 1/\epsilon \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \epsilon \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1/\epsilon \end{pmatrix} \begin{pmatrix} 1 \\ \epsilon \end{pmatrix}$$

- No general way to correct poor scaling

1.10 Iterative Refinement

$$\begin{aligned} r_0 &= b - Ax_0 = A\Delta x_0 \\ r_1 &= b - A(x_0 + \Delta x_0) &= b - Ax_1 = A\Delta x_1 \\ r_2 &= b - A(x_1 + \Delta x_1) &= b - Ax_2 = A\Delta x_2 \end{aligned}$$

$$\boxed{x = x_0 + \lim_{n=0}^{\infty} \Delta x_n} \qquad \text{(terminate when } r_n \text{ is small enough)}$$

- Double storage needed to hold original matrix
- r_n usually must be computed with higher precision than x_n
- Useful for badly scaled systmes, or making unstable systems stable
- If x_n is not accurate, r_n might not need better accuracy

2 Least $\|r\|$ Linear Regression/Fit for $Ax + r = b$

- $A = A_{m\times n}$ $\quad (m>n)$
- $\|r(y = Ax)\|$ is cont. & coer. $\rightarrow$ $\exists \|r(y)\|_{\min}$
- $r(y)$ is strictly convex $\rightarrow$ $y = Ax$ is unique
- $\text{rank}(A) = n$ (full column rank) $\Rightarrow$ $A(x_1 - x_2) = 0$ (unique x)
  $(x_1 - x_2) = 0 \rightarrow x_1 = x_2$

Example - Vandermonde Matrix, A:

$$Ax = \begin{pmatrix} -\vec{f}(t_1)- \\ \vdots \\ -\vec{f}(t_m)- \end{pmatrix} \begin{pmatrix} | \\ \vec{x} \\ | \end{pmatrix} = \begin{pmatrix} y(t_1) \\ \vdots \\ y(t_m) \end{pmatrix} = \begin{pmatrix} | \\ \vec{y} \\ | \end{pmatrix} = (x^T A^T)^T \quad , \quad y(t) = \textstyle\sum_{i=1}^n x_i f_i(t) = \vec{x} \cdot \vec{f}$$

Decompose b :

$$\begin{aligned} b &= Ax + r \\ &= y + r \\ &= Pb + P_\perp b \end{aligned}$$

Projector of A, P

$$\text{Projector}: \quad \underset{\text{(Idempotent)}}{P^2 = P} \quad \rightarrow \quad \underset{\text{(Projector of A)}}{PA = A}$$

$$\underline{\text{Orthogonal Projector}}: \quad P^T = P \ \rightarrow \ P_\perp A = (I - P)A = 0$$

Minimize residual, r:

$$\begin{aligned} \nabla \|r\|_2^2 &= 0 \qquad \left(\tfrac{\partial r^2}{\partial x_i} = 0\right) \\ &= \nabla\left[(b - Ax)^T (b - Ax)\right] \\ &= \nabla\left(b^T b - 2x^T A^T b + x^T A^T A x\right) \\ 0 &= 2A^T A x - 2A^T b \\ &\downarrow \\ A^T A x &= A^T b \quad \text{(Solvable with Cholesky)} \end{aligned}$$

$$\begin{aligned} \|r\|_2^2 &= \|Pr + P_\perp r\|_2^2 = \|b - Ax\|^2 \\ &= \|Pr\|^2 + \|P_\perp r\|^2 \\ &= \cancel{\|Pb - Ax\|_2^2} + \|P_\perp b\|_2^2 \\ &\downarrow \\ Ax &= Pb \\ A^T A x &= A^T P b = (P^T A)^T b \\ A^T A x &= A^T b \quad \text{(System of Normal Equations)} \end{aligned}$$

Cross-Product Matrix of A: $\boxed{A^T A}$

$$\begin{aligned} \text{Symmetric}: &\quad (A^T A)^T = A^T A \\ \text{Pos. Def.}: &\quad \text{rank}(A) = n \\ &\quad \rightarrow \langle x | A^T A x\rangle = x^T A^T A x \\ &\qquad = (Ax)^T (Ax) \\ &\qquad = \|Ax\|^2 \geq 0 \\ \text{Nonsingular}: &\quad A^T A x = 0 \\ &\quad \rightarrow \|Ax\|^2 = 0 = Ax \\ &\quad \rightarrow (x = 0) \end{aligned}$$

System of Normal Equations: $\boxed{A^T A x = A^T b}$

Pseudoinverse, A^+

$$\boxed{\begin{aligned} x &= (A^T A)^{-1} A^T b \\ &\equiv A^+ b \end{aligned}} \quad \rightarrow \quad \boxed{\begin{aligned} A^+ &\equiv (A^T A)^{-1} A^T \\ A^+ A &= I \end{aligned}}$$

Ortho. Proj., P

$$\boxed{\begin{aligned} Ax &= A(A^T A)^{-1} A^T b \\ &= Pb \end{aligned}} \quad \rightarrow \quad \boxed{\begin{aligned} P &= A(A^T A)^{-1} A^T \\ &= AA^+ \end{aligned}}$$

System of Normal Equations Issues:

- Info can be lost forming A^TA, e.g, $A = \begin{pmatrix} 1 & 0 \\ \epsilon & 0 \\ 0 & \epsilon \end{pmatrix} \to A^TA = \begin{pmatrix} 1+\epsilon^2 & 1 \\ 1 & 1+\epsilon^2 \end{pmatrix} \approx \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ (singular)
- System of Normal Equations: $\boxed{\mathrm{cond}(A^TA) = [\mathrm{cond}(A)]^2}$

2.1 Error Bounds and Residuals

Error Bound: $\boxed{\dfrac{\|\Delta x\|}{\|x\|} \lessapprox \mathrm{cond}(A)\ \epsilon_{\text{mach}}}$ $\to$ A computed solution is expected to lose about $\log_{10}(\mathrm{cond}(A))$ digits, so the input data must be more accurate to these digits and the working precision must carry more than these digits.

Norm and Conditioning:

$$\boxed{\|A\| = \max_{x \neq 0} \left(\frac{\|Ax\|}{\|x\|} = \frac{\|AA^+b\|}{\|A^+b\|} \right)}$$

$$\boxed{\mathrm{cond}(A) = \begin{cases} \|A\|_2 \cdot \|A^+\|_2 & \mathrm{rank}(A) = n \\ \infty & \mathrm{rank}(A) < n \end{cases}}$$

$A^TA(x + \Delta x) = A^TA(b + \Delta b)$

- $\|\Delta x\| \ \leq \ \|A^+\| \cdot \|\Delta b\|$

$$\to \boxed{\begin{aligned} \frac{\|\Delta x\|}{\|\hat{x}\|} &\leq \left(\mathrm{cond}(A)\frac{\|b\|}{\|Ax\|}\right)\frac{\|\Delta b\|}{\|b\|} \\ &= \left(\mathrm{cond}(A)\frac{1}{\cos\theta}\right)\frac{\|\Delta b\|}{\|b\|} \end{aligned}}$$

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

- $\cancel{A^TAx} + A^T\Delta Ax + (\Delta A)^TAx + \cancel{(\Delta A)^T\Delta Ax} + A^TA\Delta x + \cancel{A^T\Delta A\Delta x} + \cancel{(\Delta A)^TA\Delta x} + \cancel{(\Delta A)^T\Delta A\Delta x} = \cancel{A^Tb} + (\Delta A)^Tb$
- $\|\Delta x\| = \|(A^TA)^{-1}(\Delta A)^Tr - A^+\Delta Ax\|$
  $\leq \|(A^TA)^{-1}\| \cdot \|\Delta A\| \cdot \|r\| + \|A^+\| \cdot \|\Delta A\| \cdot \|x\|$

- Cond. number is a func. of $\mathrm{cond}(A)$ and b
- $Pb \approx 0$ or $\theta \approx 90^\circ$ is highly sensitive

$$\to \boxed{\begin{aligned} \frac{\|\Delta x\|}{\|\hat{x}\|} &\leq \left([\mathrm{cond}(A)]^2\frac{\|r\|}{\|Ax\|} + \mathrm{cond}(A)\right)\frac{\|\Delta A\|}{\|A\|} \\ &= \left([\mathrm{cond}(A)]^2\tan\theta + \mathrm{cond}(A)\right)\frac{\|\Delta A\|}{\|A\|} \end{aligned}}$$

2.2 Solving $A^TAx = A^Tb$ with an Augmented Matrix

$$\begin{aligned} r + Ax &= b \\ A^Tr \quad &= 0 \end{aligned} \Rightarrow \begin{pmatrix} I & A \\ A^T & 0 \end{pmatrix}\begin{pmatrix} r \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix} \Rightarrow \begin{pmatrix} \alpha I & A \\ A^T & 0 \end{pmatrix}\begin{pmatrix} r/\alpha \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$$

- Solvable with LU Decomp or Symm. Pos. Def. Methods
- α "controls the relative weights of the two subsystems in chooing pivots from either"
- $\alpha = \max a_{ij}/1000$ (rule of thumb)
- MATLAB uses it for large, sparse systems

2.3 QR Decomposition

<u>Motivation</u>: $Q^T A = \binom{R}{0}$ (R is upp. triag.) $\rightarrow$ $Q^T Ax + Q^T r = Q^T b$, $\binom{Rx}{0} + \binom{r'_1}{r'_2} = \binom{b'_1}{b'_2}$ $\rightarrow$ $\|r'\|^2 = \cancel{\|b'_1 - Rx\|^2} + \|b'_2\|^2$ $\downarrow$ $Rx = b'_1\ ,\quad r' = \binom{0}{b'_2}$ (solve with back-sub)

<u>Orthogonal Matrix, Q</u>

$$\boxed{Q^T Q = QQ^T = I}$$

<u>QR Factorization</u>

$$\boxed{A = Q\begin{pmatrix} R \\ 0 \end{pmatrix}}$$

<u>Reduced QR Factorization</u>

$$\boxed{A = Q\begin{pmatrix} R \\ 0 \end{pmatrix} = \begin{pmatrix} Q_\parallel & Q_\perp \end{pmatrix}\begin{pmatrix} R \\ 0 \end{pmatrix} = Q_\parallel R}$$

1. Q^T is a span(A) Plane Rotation through $\mathbb{R}^m$ to span$\big([R\ \ 0]^T\big)$

2-norm Preserved (Q is a rotation/reflection)

- $\|Qv\|^2 = \langle v|Q^T Qv\rangle = \|v\|^2$
  $\|Q^T v\|^2 = \langle v|QQ^T v\rangle = \|v\|^2$

- $Q^T = H_n \dots H_1$
- $H_i^T H_i = H_i H_i^T = I$
- $A = [a_1\ \dots\ a_n]$
- $I_n = [e_1\ \dots\ e_n]$

$$H_1 a_1 = \alpha_1 e_1 \qquad (\|a_1\| = |\alpha_1|)$$

$$H_i \dots H_1 a_i = \textstyle\sum_j^i c_j e_j = H_n \dots H_1 a_i$$
$$(\|a_i\|^2 = |\alpha_1|^2 = \textstyle\sum_j^i c_j^2)$$

$$\langle r|a_i\rangle = 0 \qquad (1 \le i \le n)$$

$$\langle H_i \dots H_1 r|e_j\rangle = 0 \qquad (1 \le j \le i)$$

$\rightarrow$ $Q^T A$ rotates A until the column vectors are aligned with certain axes described above

2. A is a Lin. Sum of $Q_\parallel$'s Orthogonal Column Vectors Given by R

$$\Big\{ Q_\parallel = Q_{m\times n} \ \Big|\ \text{span}(Q_\parallel) = \text{span}(A) \Big\}$$

$\rightarrow$ $Q^+ = (Q^T Q)^{-1} Q^T = Q^T$

$\rightarrow$ $P = Q_\parallel Q_\parallel^T$

$\rightarrow$ $Q_\parallel^T Ax = Q_\parallel^T Pb = \cancel{Q_\parallel^T Q_\parallel} Q_\parallel^T b$
$= Q_\parallel^T b$ (System of Orthogonal Equations?)

$$A = Q_\parallel R = \begin{pmatrix} | & & | \\ \vec{q_1} & \dots & \vec{q_n} \\ | & & | \end{pmatrix}\begin{pmatrix} r_{11} & \dots & r_{1n} \\ 0 & \ddots & \vdots \\ 0 & 0 & r_{nn} \end{pmatrix} = \begin{pmatrix} | & & | \\ \vec{a_1} & \dots & \vec{a_n} \\ | & & | \end{pmatrix}$$

- $\vec{a_j} = \sum_i^j r_{ij} \cdot \vec{q_i}$

$\rightarrow$ R transforms the $Q_\parallel$ column vectors about span(A), an $\mathbb{R}^n$ plane, until they equal the column vectors of A

2.3.1 Householder Transformation/Elementary Reflector, H

$$\begin{aligned} H\vec{a_1} &= \alpha_1 \vec{e_1} \quad \overset{\|a_1\| = |\alpha_1|}{\text{(rotation)}} \\ &= \boxed{\vec{a_1} - 2\hat{v}(\hat{v}\cdot\vec{a_1})} \end{aligned} \quad \rightarrow \quad \boxed{H = I - 2vv^T = I - \frac{2vv^T}{v^Tv}}$$

$[v_\perp \text{ bisects } \theta(a_1, e_1)]$

- $H = H^T = H^{-1}$ (symmetric and orthogonal)

- $\alpha_1 e_1 = a_1 - (2v_1)\dfrac{v_1 \cdot a_1}{v_1 \cdot v_1} \quad \Rightarrow \quad v_1 = (a_1 - \alpha e_1)\dfrac{\cancel{v_1 \cdot v_1}}{\cancel{2v_1 \cdot a_1}}$ (magnitude doesn't matter)

$$\rightarrow \quad \boxed{v_1 = (a_1 - \alpha e_1)}$$

$$\alpha_1 = \pm\|a_1\| \quad \rightarrow \quad \boxed{\alpha_i = -\text{sign}(a_i)\|a_i\|}$$ (avoid "cancellation" in finite-calc. of v above)

$$H_j \ldots H_1 a_i = a_i^j \quad \rightarrow \quad \boxed{v_{j+1} = \begin{pmatrix} 0 \\ \vdots \\ (a_i^j)_i \\ \vdots \\ (a_i^j)_m \end{pmatrix} - \alpha_i e_i}$$

2.3.2 Givens Rotation, G

$$\boxed{G = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}} \rightarrow Gx = G\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \pm\begin{pmatrix} \|a\| \\ 0 \end{pmatrix}$$

(creates 0's one at a time)
(useful for spare matrices)

$$\rightarrow \quad \boxed{c = \frac{a_1}{\sqrt{a_1^2 + a_2^2}} \ , \quad s = \frac{a_2}{\sqrt{a_1^2 + a_2^2}}}$$

Avoid squaring any number $\gg 1$ to prevent overflow/underflow

- $t = \frac{a_2}{a_1} < 1 \quad \rightarrow \quad c = \frac{1}{\sqrt{1+t^2}} \quad , \quad s = c \cdot t$
- $\tau = \frac{a_1}{a_2} < 1 \quad \rightarrow \quad s = \frac{1}{\sqrt{1+\tau^2}} \quad , \quad c = s \cdot \tau$

2.3.3 Gram-Schmidt Orthogonalization

3 Matrix Types

Hermitian:

$$\boldsymbol{H} = \boldsymbol{H}^\dagger$$

Unitary:

$$\boldsymbol{U}\boldsymbol{U}^\dagger = \boldsymbol{I}$$

$H = UDU^{-1}$

- D is real

$U = e^{iH}$

- $U = e^{iH} = U_H e^{iD} (U_H)^{-1}$