

1 Solving System of Linear Equations $Ax = b$

1.1 p -Norm and Condition Number

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

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

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

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

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

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

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

- $\|AB\| \leq \|A\| \cdot \|B\|$
 - $\|Ax\| \leq \|A\| \cdot \|x\|$
- For p -norms (not necessarily in general)

Function/Vector Condition Number:

$$\begin{aligned} \text{cond}(f(x)) &= \left| \frac{[f(\hat{x}) - f(x)]/f(x)}{[\hat{x} - x]/x} \right| \\ &= \left| \frac{\Delta y/y}{\Delta x/x} \right| = \left| \frac{y' \cdot \Delta x/y}{\Delta x/x} \right| \\ &= \left| \frac{x f'(x)}{f(x)} \right| \end{aligned}$$

Matrix Condition Number:

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

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

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

1.2 Error Bounds and Residuals

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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)

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

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

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

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

- 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. ($a_{jj} > \sum_{i \neq j} a_{ij}$)
- A can be singular

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

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

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

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

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

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

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

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

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

(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

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

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

MD Factorization (w/ partial pivoting)

$$\boxed{A = MD \quad \begin{array}{l} (M \text{ is elem. elim.}) \\ (D \text{ is diag.}) \end{array}}$$

$$M = (\dots M_2 P_2 M_1 P_1)^{-1}$$

$$\{\dots\}b = (\dots M_2 P_2 M_1 P_1)Ax$$

$$M^{-1}b = (P_1^T M_1^{-1} P_2^T M_2^{-1} \dots)^{-1} Ax$$

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

$$\boxed{M^{-1}b = y, \quad y = Dx}$$

(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}

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

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

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

1.5 Symmetric Matrices

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

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

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

- 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

$$\begin{array}{l} \tilde{A}\tilde{x} = b = (A - uv^T)\tilde{x} \\ \rightarrow \tilde{x} = \tilde{A}^{-1}b \end{array} \quad \left| \quad \begin{array}{l} \tilde{A}^{-1} = (A - uv^T)^{-1} = A^{-1} + \frac{A^{-1}u}{1 - v^T(A^{-1}u)} v^T A^{-1} \\ \tilde{A}^{-1}b = \quad \quad \quad \tilde{x} = (A^{-1}b) + \frac{A^{-1}u}{1 - v^T(A^{-1}u)} v^T(A^{-1}b) \\ \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad x + \frac{y}{1 - v^T y} v^T x \end{array} \right.$$

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

Explicit Inversion : $\begin{matrix} LUA^{-1} = I \\ D^{-1}M^{-1}I = A^{-1} \end{matrix} \rightarrow \mathcal{O}(n^3) \quad , \quad A^{-1}b = x \rightarrow \mathcal{O}(n^2)$

Gaussian Elimination : $A = LU \rightarrow \mathcal{O}(n^3/3) \quad , \quad LUx = b \rightarrow \mathcal{O}(n^2)$

Gaussian-Jordan : $A = MD \rightarrow \mathcal{O}(n^3/2) \quad , \quad MDx = b \rightarrow \mathcal{O}(n)$

Symmetric : $\begin{matrix} A = LL^T \\ PAP^T = LDL^T \end{matrix} \rightarrow \mathcal{O}(n^3/6) \quad , \quad LL^T x = b \rightarrow \mathcal{O}(n^2)$

Banded : $A_\beta = LU \rightarrow \mathcal{O}(\beta^2 n) \quad , \quad LUx = b \rightarrow \mathcal{O}(\beta n)$

Sherman-Woodbury : $\tilde{A} = A - uv^T \rightarrow \mathcal{O}(n^2) \quad , \quad \tilde{x} = \tilde{A}b \rightarrow \mathcal{O}(n^2)$

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

$$r_0 = b - Ax_0 = A\Delta x_0$$

$$r_1 = b - A(x_0 + \Delta x_0) = b - Ax_1 = A\Delta x_1$$

$$r_2 = b - A(x_1 + \Delta x_1) = b - Ax_2 = A\Delta x_2$$

$x = x_0 + \lim_{n=0}^{\infty} \Delta x_n$
--

 (terminate when r_n 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 systems, 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 $Ax + r = b$

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

Example - Vandermonde Matrix, A :

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

Decompose b :

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

Projector of A , P

$$\text{Projector : } P^2 = P \rightarrow PA = A$$

(Idempotent) (Projector of A)

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

Minimize residual, r :

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

$$\begin{aligned} \|r\|_2^2 &= \|Pr + P_{\perp} r\|_2^2 = \|b - Ax\|_2^2 \\ &= \|Pr\|_2^2 + \|P_{\perp} r\|_2^2 \\ &= \cancel{\|Pb - Ax\|_2^2} + \|P_{\perp} b\|_2^2 \\ &\downarrow \end{aligned}$$

$$Ax = Pb$$

$$A^T A x = A^T P b = (P^T A)^T b$$

$$A^T A x = A^T b \quad (\text{System of Normal Equations})$$

Cross-Product Matrix of A : $A^T A$

$$\text{Symmetric : } (A^T A)^T = A^T A$$

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

$$\begin{aligned} \text{Nonsingular : } A^T A x &= 0 \\ &\rightarrow \|Ax\|^2 = 0 = Ax \\ &\rightarrow (x = 0) \end{aligned}$$

System of Normal Equations: $A^T A x = A^T b$

Pseudoinverse, A^+

$$\boxed{x = (A^T A)^{-1} A^T b \equiv A^+ b} \rightarrow \boxed{A^+ \equiv (A^T A)^{-1} A^T, A^+ A = I}$$

Ortho. Proj., P

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

System of Normal Equations Issues:

- Info can be lost forming $A^T A$, e.g, $A = \begin{pmatrix} 1 & 0 \\ \epsilon & 0 \\ 0 & \epsilon \end{pmatrix} \rightarrow A^T A = \begin{pmatrix} 1+\epsilon^2 & 1 \\ 1 & 1+\epsilon^2 \end{pmatrix} \approx \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ (singular)

- System of Normal Equations: $\boxed{\text{cond}(A^T A) = [\text{cond}(A)]^2}$

2.1 Error Bounds and Residuals

Error Bound: $\boxed{\frac{\|\Delta x\|}{\|x\|} \lesssim \text{cond}(A) \epsilon_{\text{mach}}}$ \rightarrow A computed solution is expected to lose about $\log_{10}(\text{cond}(A))$ 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:

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

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

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

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

$$\bullet \quad \|\Delta x\| \leq \|A^+\| \cdot \|\Delta b\|$$

$$\bullet \quad \cancel{A^T Ax} + A^T \Delta Ax + (\Delta A)^T Ax + \cancel{(\Delta A)^T \Delta Ax} = \cancel{A^T b} + (\Delta A)^T b \\ + A^T A \Delta x + \cancel{A^T \Delta A \Delta x} + \cancel{(\Delta A)^T A \Delta x} + \cancel{(\Delta A)^T \Delta A \Delta x}$$

$$\rightarrow \boxed{\frac{\|\Delta x\|}{\|\hat{x}\|} \leq \left(\text{cond}(A) \frac{\|b\|}{\|Ax\|} \right) \frac{\|\Delta b\|}{\|b\|} \\ = \left(\text{cond}(A) \frac{1}{\cos \theta} \right) \frac{\|\Delta b\|}{\|b\|}}$$

$$\bullet \quad \|\Delta x\| = \|(A^T A)^{-1} (\Delta A)^T r - A^+ \Delta Ax\| \\ \leq \|(A^T A)^{-1}\| \cdot \|\Delta A\| \cdot \|r\| + \|A^+\| \cdot \|\Delta A\| \cdot \|x\|$$

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

$$\rightarrow \boxed{\frac{\|\Delta x\|}{\|\hat{x}\|} \leq \left([\text{cond}(A)]^2 \frac{\|r\|}{\|Ax\|} + \text{cond}(A) \right) \frac{\|\Delta A\|}{\|A\|} \\ = \left([\text{cond}(A)]^2 \tan \theta + \text{cond}(A) \right) \frac{\|\Delta A\|}{\|A\|}}$$

2.2 Solving $A^T Ax = A^T b$ with an Augmented Matrix

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

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

2.3 QR Decomposition

Motivation: $Q^T A = \begin{pmatrix} R \\ 0 \end{pmatrix} \rightarrow Q^T A x + Q^T r = Q^T b \rightarrow \begin{pmatrix} R x \\ 0 \end{pmatrix} + \begin{pmatrix} r'_1 \\ r'_2 \end{pmatrix} = \begin{pmatrix} b'_1 \\ b'_2 \end{pmatrix} \rightarrow \begin{matrix} \|r'\|^2 = \cancel{\|b'_1 - R x\|^2} + \|b'_2\|^2 \\ \downarrow \\ R x = b'_1, \quad r' = \begin{pmatrix} 0 \\ b'_2 \end{pmatrix} \quad (\text{solve with back-sub}) \end{matrix}$

Orthogonal Matrix, Q

$$Q^T Q = Q Q^T = I$$

QR Factorization

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

Reduced QR Factorization

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

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

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

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

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

$$H_1 a_1 = \alpha e_1 \quad (\|a_1\| = |\alpha|)$$

$$H_i \dots H_1 a_i = \sum_j^i c_j e_j = H_n \dots H_1 a_i$$

$$(\|a_i\|^2 = |\alpha|^2 = \sum_j^i c_j^2)$$

$$\langle r | a_i \rangle = 0 \quad (1 \leq i \leq n)$$

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

→ $Q^T A$ 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

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

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

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

$$\rightarrow Q_{\parallel}^T A x = Q_{\parallel}^T P b = \cancel{Q_{\parallel}^T Q_{\parallel}} Q_{\parallel}^T b$$

$$= Q_{\parallel}^T b \quad (\text{System of Orthogonal Equations?})$$

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

- $\vec{a}_j = \sum_i^j r_{ij} \cdot \vec{q}_i$

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

2.3.1 Householder Transformation/Elementary Reflector

$$H = I - (2\hat{v})\hat{v}^T = I - \frac{(2v)v^T}{v^T v}$$

- $H = H^T = H^{-1}$ (symmetric and orthogonal)
- $\alpha e_1 = Ha = a - (2v)\frac{v \cdot a}{v \cdot v} \rightarrow v = (a - \alpha e_1)\frac{v \cdot v}{2v \cdot a} \rightarrow v = (a - \alpha e_1)$
- $\alpha = \pm \|a\| \Rightarrow$ Choose $\alpha = -\text{sign}(a_1)\|a\|$
- Just use v to transform a - don't find H

$$\bullet \quad v_i = \begin{pmatrix} 0 \\ \vdots \\ a_i \\ \vdots \\ a_m \end{pmatrix} - \alpha e_i = \begin{pmatrix} 0 \\ \vdots \\ a_i - \alpha \\ \vdots \\ a_m \end{pmatrix}$$

3 Matrix Types

Hermitian:

$$H = H^\dagger$$

Unitary:

$$UU^\dagger = I$$

$$H = UDU^{-1}$$

- D is real

$$U = e^{iH}$$

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