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:

$$\text{cond}\Big(f(x)\Big) = \left| \frac{[f(\hat{x}) - f(x)]/f(x)}{[\hat{x} - x]/x} \right|$$

$$= \left| \frac{\Delta y/y}{\Delta x/x} \right| = \left| \frac{y' \cdot \Delta x/y}{\Delta x/x} \right|$$

$$= \left| \frac{x f'(x)}{f(x)} \right|$$

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

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

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

$$\left.\begin{array}{l} \bullet\ \|AB\| \leq \|A\| \cdot \|B\| \\ \bullet\ \|Ax\| \leq \|A\| \cdot \|x\| \end{array}\right\} \text{For } p\text{-norms (not necessarily in general)}$$

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: $\boxed{\dfrac{\|\hat{x}-x\|}{\|x\|} \lessapprox \operatorname{cond}(A)\ \epsilon_{\text{mach}}}$ $\rightarrow$ 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

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

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

MD Factorization (w/ partial pivoting)

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

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

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

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

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

- 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}

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

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

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

1.5 Symmetric Matrices

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

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

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

- 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

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

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

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

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

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

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

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

$$\text{Symmetric}: \begin{aligned} A &= LL^T \\ PAP^T &= LDL^T \end{aligned} \ \rightarrow \ \mathcal{O}(n^3/6) \ , \ LL^Tx = b \ \rightarrow \ \mathcal{O}(n^2)$$

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

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

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

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

$$\boxed{x = x_0 + \lim_{n=0}^{\infty} \Delta x_n}$$ (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 systmes, or making unstable systems stable
- If x_n is not accurate, r_n might not need better accuracy

2 Matrix Types

Hermitian:

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

Unitary:

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

$H = UDU^{-1}$

- D is real

$U = e^{iH}$

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