Chapter 2 Direct Methods for Linear Systems

sec 2.1 Gaussian Elimination

Goal: 1. Linear systems: some theory

2. Triangular systems: forward / backward substitution

3. Gaussian Elimination

1. Linear systems:

$$\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n = b_n \end{cases} \Rightarrow A\vec{x} = \vec{b}, \quad \text{where } \begin{array}{l} A = (a_{ij}) \in \mathbb{R}^{n\times n} \\ \vec{x} = (x_1, \cdots, x_n)^T \\ \vec{b} = (b_1, \cdots, b_n)^T \end{array}$$

<u>Theorem:</u> For any $n\times n$ matrix A, the following statements are equivalent:

(1) $A\vec{x} = \vec{b}$ has a unique solution for each $\vec{b} \in \mathbb{R}^n$

(2) $A\vec{x} = \vec{0}$ has only the trivial solution $\vec{x} = \vec{0}$

(3) A is nonsingular / invertible, i.e. A^{-1} exists

(4) $\det(A) \neq 0$

(5) The rows (or columns) of A are linearly independent

2. Triangular systems:

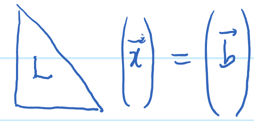
- Consider $R\vec{x} = \vec{b}$, where R is upper triangular and $\det(R) \neq 0$.

$$\begin{cases} r_{11}x_1 + r_{12}x_2 + \cdots + r_{1n}x_n = b_1 \\ \qquad\quad r_{22}x_2 + \cdots + r_{2n}x_n = b_2 \\ \qquad\qquad\qquad \ddots \\ \qquad\qquad\qquad\qquad r_{nn}x_n = b_n \end{cases} \qquad \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ & \ddots & \vdots \\ & & r_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

solve by "backward Substitution":

$$\left.\begin{array}{l} x_n = b_n / r_{nn} \\ x_{n-1} = (b_{n-1} - r_{n-1,n}x_n) / r_{n-1,n-1} \\ \quad\vdots \\ x_1 = \left(b_1 - \sum_{j=2}^{n} r_{1j}x_j\right) / r_{11} \end{array}\right\} \# \text{ of FLOP operations} \sim O(n^2)$$

- If we consider $L\vec{x} = \vec{b}$, where L is lower triangular, we solve by "forward substitution":



$$\left.\begin{aligned} x_1 &= b_1 / l_{11} \\ x_2 &= (b_2 - l_{21} \cdot x_1)/l_{22} \\ &\ \vdots \\ x_n &= \Big(b_n - \sum_{j=1}^{n-1} l_{nj} x_j\Big)/l_{nn} \end{aligned}\right\} \ \text{\# of FLOP operations} \sim O(n^2)$$

3. Gaussian Elimination (without pivoting)

$$\text{Full system} \xrightarrow{\text{Elimination}} \text{Upper Triangular system} \xrightarrow{\text{backward substitution}} \text{solution}$$

Elimination:

three elementary operations:	notation
1. multiply eqn. E_i by a constant λ	$(\lambda E_i) \to E_i$
2. add λE_j to E_i	$(\lambda E_j + E_i) \to E_i$
3. interchange eqn. E_i and E_j	$E_i \longleftrightarrow E_j$

eg.
$$\begin{cases} x_1 - x_2 + x_3 = -2 \\ 2x_1 + x_2 = -7 \\ x_1 + 2x_2 + 3x_3 = 7 \end{cases}$$

$$\left(\begin{array}{ccc|c} 1 & -1 & 1 & -2 \\ 2 & 1 & 0 & -7 \\ 1 & 2 & 3 & 7 \end{array}\right) \xrightarrow[(E_3 - E_1) \to E_3]{(E_2 - 2E_1) \to E_2} \left(\begin{array}{ccc|c} 1 & -1 & 1 & -2 \\ 0 & 3 & -2 & -3 \\ 0 & 3 & 2 & 9 \end{array}\right) \xrightarrow{(E_3 - E_2) \to E_2} \left(\begin{array}{ccc|c} 1 & -1 & 1 & -2 \\ 0 & 3 & -2 & -3 \\ 0 & 0 & 4 & 12 \end{array}\right)$$

Using Backward substitution,
$$\begin{aligned} x_3 &= \tfrac{12}{4} = 3 \\ x_2 &= (-3 + 2x_3)/3 = 1 \\ x_1 &= (-2 - x_3 + x_2)/1 = -4 \end{aligned}$$

In general,
$$\begin{cases} a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n = b_1 & (E_1) \\ a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n = b_2 & (E_2) \\ \quad \vdots & \ \vdots \\ a_{n1} x_1 + a_{n2} x_2 + \cdots + a_{nn} x_n = b_n & (E_n) \end{cases}$$

$$\left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right) \xrightarrow[i = 2, \cdots, n]{(E_i - \frac{a_{i1}}{a_{11}} E_1) \to E_i} \left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{array}\right)$$

repeating this process, $A = A^{(1)} \rightarrow A^{(2)} \longrightarrow \cdots \longrightarrow A^{(n)} = R$, upper triangular

$$\left.\begin{aligned} i^{th}\text{ row} \longrightarrow a_{i,:}^{(2)} &= a_{i,:}^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} a_{1,:}^{(1)}, \\ b_i^{(2)} &= b_i^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} \cdot b_1^{(1)} \end{aligned}\right\} \quad i = 2, \cdots, n$$

In general, for $k = 1, \cdots, n-1$: (eliminating k^{th} column below diagonal)

$$\left.\begin{aligned} a_{i,:}^{(k+1)} &= a_{i,:}^{(k)} - \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}} a_{k,:}^{(k)} \\ b_i^{(k+1)} &= b_i^{(k)} - \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}} b_k^{(k)} \end{aligned}\right\} \quad i = k+1, \cdots, n$$

- This procedure will fail if $a_{kk}^{(k)} = 0$. But we can interchange E_k and E_p where $a_{pk}^{(k)} \neq 0$.
- If $a_{p,k}^{(k)} = 0$ for all $p = k, \cdots, n$, the system does not have a unique solution.

eg.
$$\begin{cases} x_1 + x_2 + x_3 = 4 \\ 2x_1 + 2x_2 + x_3 = 4 \\ x_1 + x_2 + 2x_3 = 6 \end{cases}$$

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 4 \\ 2 & 2 & 1 & 4 \\ 1 & 1 & 2 & 6 \end{array}\right) \xrightarrow[(E_3 - E_1) \rightarrow (E_1)]{(E_2 - 2E_1) \rightarrow (E_1)} \left(\begin{array}{ccc|c} 1 & 1 & 1 & 4 \\ 0 & 0 & -1 & 4 \\ 0 & 0 & 1 & 2 \end{array}\right) \Rightarrow \begin{cases} x_1 + x_2 + x_3 = 4 \\ \left.\begin{aligned} -x_3 &= 4 \\ x_3 &= 2 \end{aligned}\right\} \text{ no solution!} \end{cases}$$

eg.
$$\begin{cases} x_1 + x_2 + x_3 = 4 \\ 2x_1 + 2x_2 + x_3 = 6 \\ x_1 + x_2 + 2x_3 = 6 \end{cases}$$

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 4 \\ 2 & 2 & 1 & 6 \\ 1 & 1 & 2 & 6 \end{array}\right) \xrightarrow[(E_3 - E_1) \rightarrow (E_3)]{(E_2 - 2E_1) \rightarrow (E_2)} \left(\begin{array}{ccc|c} 1 & 1 & 1 & 4 \\ 0 & 0 & -1 & -2 \\ 0 & 0 & 1 & 2 \end{array}\right) \Rightarrow \begin{cases} x_1 + x_2 + x_3 = 4 \\ -x_3 = -2 \\ x_3 = 2 \end{cases}$$

$\Rightarrow x_3 = 2$. $x_1 + x_2 = 2$ $\Rightarrow$ x_1 = any number. $x_2 = 2 - x_1$, $x_3 = 2$. infinitely many solutions.

Algorithm (Gaussian Elimination without Pivoting)

To solve the linear system

$$\begin{cases} a_{11} x_1 + \cdots + a_{1n} x_n = a_{1,n+1} & (E_1) \\ \vdots \\ a_{n1} x_1 + \cdots + a_{nn} x_n = a_{n,n+1} & (E_n) \end{cases}$$

Inputs : $A = (a_{ij})$, $i = 1, \cdots, n$, $j = 1, \cdots, n+1$

Outputs : $x_1, \cdots, x_n$

elimination

step 1 : For $k = 1, \cdots, n-1$, do step 2

step 2 : For $i = k+1, \cdots n$, do step 3 — 4

step 3 : set $m_i = a_{ik}/a_{kk}$

step 4 : perform $A(i, k+1:n+1) = A(i, k+1:n+1) - m_i * A(k, k+1:n+1)$

backward substitution

step 5 : Set $x_n = a_{n,n+1} / a_{nn}$

step 6 : For $i = n-1, \cdots, 1$, set $x_i = \left(a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij} x_j \right) / a_{ii}$

step 7 : Output $x_1, \cdots, x_n$

Operation Counts :

identities : $\sum_{k=1}^{n} k = \frac{n(n+1)}{2} = \frac{n^2}{2} + \frac{n}{2} \sim \frac{n^2}{2}$

$\sum_{k=1}^{n} k^2 = n(n+1)(2n+1)/6 \sim \frac{n^3}{3}$

① elimination :

multiplications : $\sum_{k=1}^{n-1} \sum_{i=k+1}^{n} (1 + (n+1-k)) = \sum_{k=1}^{n-1} (n-k)(n-k+2)$

$\overset{j=n-k}{=} \sum_{j=1}^{n-1} j(j+2) = \sum_{j=1}^{n-1} j^2 + 2\sum_{j=1}^{n-1} j \sim \frac{n^3}{3}$

addition : $\sum_{k=1}^{n-1} \sum_{i=k+1}^{n} (n+1-k) = \sum_{k=1}^{n-1} (n-k)(n-k+1) \sim \sum_{j=1}^{n-1} j^2 \sim \frac{n^3}{3}$

$\sim \frac{2}{3} n^3$

② substitution :

multiplications : $1 + \sum_{i=1}^{n-1} \left(1 + \sum_{j=i+1}^{n} 1\right) = 1 + \sum_{i=1}^{n-1} (1 + (n-i)) \sim \sum_{j=1}^{n} j \sim \frac{n^2}{2}$

additions : $\sum_{i=1}^{n-1} (n-i) \overset{j=n-i}{=} \sum_{j=1}^{n-1} j = \frac{n(n-1)}{2} \sim \frac{n^2}{2}$

$\sim n^2$

$\Rightarrow$ Total number of FLOP operations $\sim \frac{2}{3} n^3$

The computation time increases with n in proporation to n^3