NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Module 2: Linear Systems

Learning Objectives

- Learn direct methods
  - Gaussian elimination
  - LU factorization
- Learn iterative methods
  - Jacobi method
  - Gauss-Seidel method
- Learn methods for symmetric positive-definite matrices
  - Gradient descent

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Sources

- Textbook (Chapter 2: Systems of Equations)
- Wiki: Cramer’s rule

  https://en.wikipedia.org/wiki/Cramer%27s_rule

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Outline

§1. Introduction

§2. LU Decomposition

§3. Iterative methods

§4. Symmetric positive-definite matrices

§5. Homework

§6. Summary

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Introduction

- Basic problem:

Given a linear system consisting of a set of n linear equation in n unknowns $x_1, x_2, \cdots, x_n$:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & = & b_2 \\ \qquad\quad \vdots & = & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n & = & b_n \end{cases}$$

how to find the solution?

Re-write the equations compactly as $Ax = b$, where

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

Inverting A gives $x = A^{-1}b$, but there's no explicit formula for A^{-1}.

Application examples

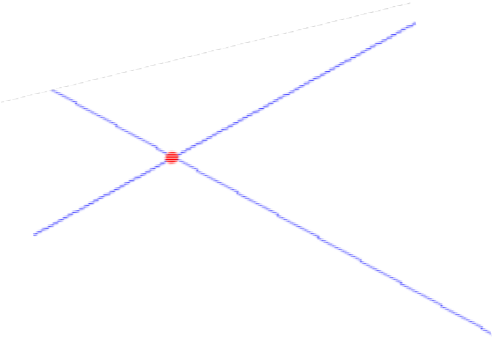
- Find the intersection of two parametric lines on a plane.

  Given two lines $\mathbf{r}_1(t)$ and $\mathbf{r}_2(s)$ defined by

$$\mathbf{r}_1(t) = \begin{pmatrix} x_1(t) \\ y_1(t) \end{pmatrix} = \begin{pmatrix} a_{11}t + m_1 \\ a_{21}t + m_2 \end{pmatrix}, \quad \mathbf{r}_2(s) = \begin{pmatrix} x_2(s) \\ y_2(s) \end{pmatrix} = \begin{pmatrix} a_{12}s + n_1 \\ a_{22}s + n_2 \end{pmatrix},$$

  their interection is the solution of the 2 linear equations

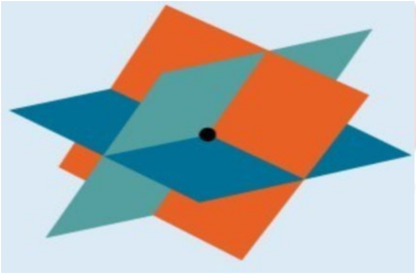
$$\begin{cases} a_{11}t - a_{12}s & = & n_1 - m_1 \\ a_{21}t - a_{22}s & = & n_2 - m_2 \end{cases}$$



- Find the intersection of three planes defined implicitly by $a_i x + b_i y + c_i z + d_i = 0$ for $i = 1, 2, 3$.

  The intersection point is the solution of the following equations:

$$\begin{cases} a_1 x + b_1 y + c_1 z & = & -d_1 \\ a_2 x + b_2 y + c_2 z & = & -d_2 \\ a_3 x + b_3 y + c_3 z & = & -d_3 \end{cases}$$



Cramer's rule

- The solution for a system of linear equations $Ax = b$ is given by

$$x_i = \frac{\det(A_i)}{\det(A)}, \quad i = 1, 2, \cdots, n$$

where A_i is the matrix formed by replacing the i-th column of A by the column vector b.

- Example: given 3 linear equations $\begin{cases} a_1 x + b_1 y + c_1 z & = d_1 \\ a_2 x + b_2 y + c_2 z & = d_2 \\ a_3 x + b_3 y + c_3 z & = d_3 \end{cases}$

the solutions are

$$x = \frac{\begin{vmatrix} d_1 & b_1 & c_1 \\ d_2 & b_2 & c_2 \\ d_3 & b_3 & c_3 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}}, \quad y = \frac{\begin{vmatrix} a_1 & d_1 & c_1 \\ a_2 & d_2 & c_2 \\ a_3 & d_3 & c_3 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}}, \text{ and } z = \frac{\begin{vmatrix} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ a_3 & b_3 & d_3 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}}.$$

- Note that the computation of the determinants is in general nontrivial. So in practice, algorithms for solving linear equations are needed!

Outline

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

2.1 Two basic operations on equations

- Swap one equation from another

$$\begin{cases} x + y &= 3 \\ 3x - 4y &= 2 \end{cases} \Rightarrow \begin{cases} 3x - 4y &= 2 \\ x + y &= 3 \end{cases}$$

- Subtract a multiple of one equation from another

$$\begin{cases} x + y &= 3 \\ 3x - 4y &= 2 \end{cases} \xRightarrow{\text{Eq2 - 3×Eq1}} \begin{cases} x + y &= 3 \\ -7y &= -7 \end{cases}$$

- Applying these operations to a system of linear equations yields an ***equivalent*** system (which has the same solutions).

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Gaussian elimination: example

Consider equations

$$\left\{\begin{array}{lcl} x+y & = & 3 \\ 3x-4y & = & 2 \end{array}\right. \overset{\text{Convert to matrix (tableau) form}}{\Rightarrow} \left[\begin{array}{cc|c} 1 & 1 & 3 \\ 3 & -4 & 2 \end{array}\right]$$

- "Naïve" approach
  - Elimination

$$\left[\begin{array}{cc|c} 1 & 1 & 3 \\ 3 & -4 & 2 \end{array}\right] \overset{\text{row2 – 3× row1}}{\Rightarrow} \left[\begin{array}{cc|c} 1 & 1 & 3 \\ 0 & -7 & -7 \end{array}\right]$$

  - Back-substitution

$$\left\{\begin{array}{rcl} x+y & = & 3 \\ -7y & = & -7 \end{array}\right. \Rightarrow \left\{\begin{array}{rcl} x+y & = & 3 \\ y & = & \frac{-7}{-7}=1 \end{array}\right. \Rightarrow \left\{\begin{array}{lcl} x & = & 3-y=2 \\ y & = & 1 \end{array}\right.$$

Gaussian elimination: example

- Elimination with partial pivoting: before carrying out elimination for column k, the largest $|a_{pk}|$ is located and rows p and k are swapped, which can improve numerical stability.
  - Swapping

$$\left[\begin{array}{cc|c} 1 & 1 & 3 \\ 3 & -4 & 2 \end{array}\right] \xRightarrow{\text{row1} \leftrightarrow \text{row2}} \left[\begin{array}{cc|c} 3 & -4 & 2 \\ 1 & 1 & 3 \end{array}\right]$$

  - Elimination

$$\left[\begin{array}{cc|c} 3 & -4 & 2 \\ 1 & 1 & 3 \end{array}\right] \xRightarrow{\text{row2} - (1/3)\times\text{row1}} \left[\begin{array}{cc|c} 3 & -4 & 2 \\ 0 & 7/3 & 7/3 \end{array}\right]$$

  - Back-substitution

$$\begin{cases} 3x - 4y &= 2 \\ \frac{7}{3}y &= \frac{7}{3} \end{cases} \Rightarrow \begin{cases} 3x - 4y &= 2 \\ y &= 1 \end{cases} \Rightarrow \begin{cases} x &= \frac{2+4y}{3} = 2 \\ y &= 1 \end{cases}$$

Gaussian elimination: general

- Input

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & = & b_2 \\ \vdots & = & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n & = & b_n \end{cases}$$

- Convert to matrix (tableau) form

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right]$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Gaussian elimination: general

- Elimination

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right]$$

$$\Downarrow$$

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22} - \frac{a_{21}}{a_{11}} a_{12} & \cdots & a_{2n} - \frac{a_{21}}{a_{11}} a_{1n} & b_2 - \frac{a_{21}}{a_{11}} b_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n2} - \frac{a_{n1}}{a_{11}} a_{12} & \cdots & a_{nn} - \frac{a_{n1}}{a_{11}} a_{1n} & b_n - \frac{a_{n1}}{a_{11}} b_1 \end{array}\right]$$

$$\Downarrow$$

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & \bar{a}_{22} & \cdots & \bar{a}_{2n} & \bar{b}_2 \\ 0 & 0 & \cdots & \bar{a}_{3n} & \bar{b}_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \bar{a}_{nn} & \bar{b}_n \end{array}\right]$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Gaussian elimination: general

- Back-substitution

$$
\left[\begin{array}{cccc|c}
a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\
0 & \bar{a}_{22} & \cdots & \bar{a}_{2n} & \bar{b}_2 \\
0 & 0 & \cdots & \bar{a}_{3n} & \bar{b}_3 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & \bar{a}_{nn} & \bar{b}_n
\end{array}\right]
$$

$$
\left\{\begin{array}{rcl}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1(n-1)}x_{n-1} + a_{1n}x_n & = & b_1 \\
\bar{a}_{22}x_2 + \cdots + \bar{a}_{2(n-1)}x_{n-1} + \bar{a}_{2n}x_n & = & \bar{b}_2 \\
\vdots & = & \vdots \\
\bar{a}_{(n-1)(n-1)}x_{n-1} + \bar{a}_{(n-1)n}x_n & = & \bar{b}_{n-1} \\
\bar{a}_{nn}x_n & = & \bar{b}_n
\end{array}\right.
$$

$$
\left\{\begin{array}{rcl}
x_n & = & \dfrac{\bar{b}_n}{\bar{a}_{nn}} \\
x_{n-1} & = & \dfrac{\bar{b}_{n-1} - \bar{a}_{(n-1)n}x_n}{\bar{a}_{(n-1)(n-1)}} \\
\vdots & = & \vdots \\
x_2 & = & \dfrac{\bar{b}_2 - \bar{a}_{23}x_3 - \cdots - \bar{a}_{2n}x_n}{\bar{a}_{22}} \\
x_1 & = & \dfrac{b_1 - a_{12}x_2 - a_{13}x_3 - \cdots - a_{1n}x_n}{a_{11}}
\end{array}\right.
$$

Gaussian elimination: complexity

- Elimination

number of operations of $(+, -, \times \text{ or } \div)$

$$
\begin{bmatrix}
0 \\
2n+1 & 0 \\
2n+1 & 2(n-1)+1 & 0 \\
2n+1 & 2(n-1)+1 & 2(n-2)+1 & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots \\
\vdots & \vdots & \vdots \\
2n+1 & 2(n-1)+1 & 2(n-2)+1 & \cdots & 2(3)+1 & 0 \\
2n+1 & 2(n-1)+1 & 2(n-2)+1 & \cdots & 2(3)+1 & 2(2)+1 & 0
\end{bmatrix}
$$

$(n-1)(2n+1)$ $=2n^2-(n+1)$	$(n-2)[2(n-1)+1]$ $=2(n-1)^2-n$	…	$(2-1)[2(2)+1]$ $=2(2)^2-3$

Totally, $\left(\frac{2}{3}n^3+\frac{1}{2}n^2-\frac{7}{6}n\right)$ operations in $(+, -, \times \text{ or } \div)$.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Gaussian elimination: complexity

- Back-substitution

$$
\begin{cases}
x_n = \dfrac{\bar{b}_n}{\bar{a}_{nn}} & \longrightarrow 1 \\
x_{n-1} = \dfrac{\bar{b}_{n-1} - \bar{a}_{(n-1)n} x_n}{\bar{a}_{(n-1)(n-1)}} & \longrightarrow 3 \\
\vdots = \vdots & \quad \vdots \\
x_2 = \dfrac{\bar{b}_2 - \bar{a}_{23} x_3 - \cdots - \bar{a}_{2n} x_n}{\bar{a}_{22}} & \\
x_1 = \dfrac{b_1 - a_{12} x_2 - a_{13} x_3 - \cdots - a_{1n} x_n}{a_{11}} & \longrightarrow 2n-1
\end{cases}
$$

Totally, n^2 operations in terms of $(+, -, \times \text{ or } \div)$.

- To sum up, the complexity of Gaussian elimination method to solve *n* equations in *n* unknowns is O(n^3).

2.2 LU factorization

- Gaussian elimination
  - Elimination: $O(n^3)$
  - Back-substitution: $O(n^2)$
- In engineering, $Ax = b$

Structure matrix (depending on the design)

Loading vector

  - A remains the same
  - b changes for various inputs

- Q: Can we separate the elimination process on A and b?

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Two basic operations revisited

- Row swapping

Row1 ←→ Row2

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} a_{21} & a_{22} & a_{23} \\ a_{11} & a_{12} & a_{13} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Permutation matrix

- Subtract a multiple of one row from another

Lower triangular matrix

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ c & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} - ca_{11} & a_{22} - ca_{12} & a_{23} - ca_{13} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Row2 – c × Row1

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

PA=LU factorization

Problem: find the $PA = LU$ factorization of matrix $A = \begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix}$.

Solution:

Initialize permutation matrix P and lower triangular matrix L to be the indentity matrix.

Carry out Gaussian elimination with partial pivoting.

Update P and L during the process.

(1) swap rows 1 and 2.

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{P} \begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix} \xrightarrow{\text{row1} \leftrightarrow \text{row2}} \begin{bmatrix} 4 & 4 & -4 \\ 2 & 1 & 5 \\ 1 & 3 & 1 \end{bmatrix}$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

PA=LU factorization

(2) eliminate the first column.

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{P} \begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix} \xrightarrow{\text{row1} \leftrightarrow \text{row2}} \begin{bmatrix} 4 & 4 & -4 \\ 2 & 1 & 5 \\ 1 & 3 & 1 \end{bmatrix}$$

row2 – (1/2) ×row1

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{L} \begin{bmatrix} 4 & 4 & -4 \\ 0 & -1 & 7 \\ 1 & 3 & 1 \end{bmatrix}$$

row3 – (1/4) ×row1

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{4} & 0 & 1 \end{bmatrix}}_{L} \begin{bmatrix} 4 & 4 & -4 \\ 0 & -1 & 7 \\ 0 & 2 & 2 \end{bmatrix}$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

PA=LU factorization

(3) swap rows 2 and 3.

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{P} \begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix} \qquad \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}}_{P} \begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix}$$

row2 ←→row3

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{4} & 0 & 1 \end{bmatrix}}_{L} \begin{bmatrix} 4 & 4 & -4 \\ 0 & -1 & 7 \\ 0 & 2 & 2 \end{bmatrix} \Rightarrow \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{bmatrix}}_{L} \begin{bmatrix} 4 & 4 & -4 \\ 0 & 2 & 2 \\ 0 & -1 & 7 \end{bmatrix}$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

PA=LU factorization

(4) eliminate the 2nd column.

$$\begin{array}{cc} P & A \\ \| & \| \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix} \end{array}$$

$$\begin{array}{cc} L & \\ \| & \\ \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{bmatrix} & \begin{bmatrix} 4 & 4 & -4 \\ 0 & 2 & 2 \\ 0 & -1 & 7 \end{bmatrix} \end{array}$$

row3 - (-1/2)×row2

$$\begin{array}{cc} L & U \\ \| & \| \\ \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix} & \begin{bmatrix} 4 & 4 & -4 \\ 0 & 2 & 2 \\ 0 & 0 & 8 \end{bmatrix} \end{array}$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

PA=LU factorization

$$PA = LU$$

P: Permutation matrix
L: lower triangular matrix
U: upper triangular matrix

When P is the identity matrix, PA=LU becomes the common LU factorization.

$$\begin{array}{cc} P & A \\ \| & \| \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix} \end{array}$$

$$\begin{array}{cc} L & \\ \| & \\ \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{bmatrix} & \begin{bmatrix} 4 & 4 & -4 \\ 0 & 2 & 2 \\ 0 & -1 & 7 \end{bmatrix} \end{array}$$

row3 - (-1/2)×row2

$$\begin{array}{cc} L & U \\ \| & \| \\ \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix} & \begin{bmatrix} 4 & 4 & -4 \\ 0 & 2 & 2 \\ 0 & 0 & 8 \end{bmatrix} \end{array}$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Solving equations using LU factorization

Problem: Use the $PA = LU$ factorization to solve equations $Ax = b$, where

$$A = \begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 5 \\ 0 \\ 6 \end{bmatrix}.$$

Solution: $Ax = b$ becomes $PAx = Pb$, and thus $LUx = Pb$.

This can be done by two steps:

(1) solve $Lc = Pb$ for c;

(2) solve $Ux = c$ for x.

1. $Lc = Pb$:

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 5 \\ 0 \\ 6 \end{bmatrix} = \begin{bmatrix} 0 \\ 6 \\ 5 \end{bmatrix} \Rightarrow \begin{aligned} c_1 &= 0 \\ c_2 &= 6 \\ c_3 &= 8. \end{aligned}$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Solving equations using LU factorization

2. $Ux = c$:

$$\begin{bmatrix} 4 & 4 & -4 \\ 0 & 2 & 2 \\ 0 & 0 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 6 \\ 8 \end{bmatrix}$$

$$\begin{aligned} & x_3 = 1 \\ \Rightarrow\ & x_2 = 2 \\ & x_1 = -1. \end{aligned}$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Advantages and disadvantages of Gaussian elimination or LU factorization

- Pros
  - Solve equations in a finite number of steps
  - Provide exact solutions

- Cons
  - Complexity O(n^3)
  - Not efficient for sparse systems of equations
  - Not efficient for simple updates

Outline

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Iterative methods

- Motivation: for **sparse** systems, where the coefficient matrix A has only $O(n)$ nonzero entries, it is often better to use an iterative method.
- Basic idea: start with some initial guess of the solutions $x^{(0)}$, improve the guess iteratively, ensuring that in each iteration
  - the new approximation of the solution $x^{(k+1)}$ is computed efficiently from the previous one
  - the generated sequence $\{x^{(k)}\}$ converges to the correct solution.
- Jacobi method
- Gauss-Seidel method

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

3.1 Jacobi method

Look at the following equations as an example:

$$\begin{cases} 3x_1 + x_2 - x_3 & = & 1 \\ 2x_1 - 5x_2 + 2x_3 & = & 2 \\ x_1 + 6x_2 + 8x_3 & = & 3 \end{cases}$$

Rewrite the equations:

$$\begin{cases} x_1 & = & \dfrac{1 - x_2 + x_3}{3} \\ x_2 & = & \dfrac{2 - 2x_1 - 2x_3}{-5} \\ x_3 & = & \dfrac{3 - x_1 - 6x_2}{8} \end{cases}$$

Method: start with some initial guess and iterate the following:

$$\begin{cases} x_1^{(k+1)} & = & \dfrac{1 - x_2^{(k)} + x_3^{(k)}}{3} \\ x_2^{(k+1)} & = & \dfrac{2 - 2x_1^{(k)} - 2x_3^{(k)}}{-5} \\ x_3^{(k+1)} & = & \dfrac{3 - x_1^{(k)} - 6x_2^{(k)}}{8} \end{cases}$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Jacobi method

- To solve $Ax = b$, the Jacobi method works in an iterative way. In each iterative step, the i-th entry of the next guess $x^{(k+1)}$ is computed from the other entries of the current guess $x^{(k)}$ by solving the i-th linear equation of the system:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(k)} \right), \quad i = 1, \cdots, n$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Jacobi method

- If we rewrite the system matrix $A = L + D + U$, where
  - $L = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 0 \end{bmatrix}$

    A lower triangular matrix with zeros on the diagonal
  - $D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}$

    A diagonal matrix
  - $U = \begin{bmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ 0 & 0 & 0 & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$

    A upper triangular matrix with zeros on the diagonal

Jacobi method

Then the iteration of the Jacobi method for $(L+D+U)x = b$ can be written

$$x^{(k+1)} = D^{-1}\left(b - Lx^{(k)} - Ux^{(k)}\right)$$

and D^{-1} is just the diagonal matrix with entries $\dfrac{1}{a_{ii}}$:

$$D^{-1} = \begin{bmatrix} \frac{1}{a_{11}} & 0 & \cdots & 0 \\ 0 & \frac{1}{a_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{a_{nn}} \end{bmatrix}$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

3.2 Gauss-Seidel method

- Gauss-Seidel method improved the Jacobi method. The basic idea is to use the ***most recent*** values in the iteration.
- For example

$$\begin{cases} x_1 = \dfrac{1 - x_2 + x_3}{3} \\ x_2 = \dfrac{2 - 2x_1 - 2x_3}{-5} \\ x_3 = \dfrac{3 - x_1 - 6x_2}{8} \end{cases}$$

Jacobi

$$\begin{cases} x_1^{(k+1)} = \dfrac{1 - x_2^{(k)} + x_3^{(k)}}{3} \\ x_2^{(k+1)} = \dfrac{2 - 2x_1^{(k)} - 2x_3^{(k)}}{-5} \\ x_3^{(k+1)} = \dfrac{3 - x_1^{(k)} - 6x_2^{(k)}}{8} \end{cases}$$

Gauss-Seidel

$$\begin{cases} x_1^{(k+1)} = \dfrac{1 - x_2^{(k)} + x_3^{(k)}}{3} \\ x_2^{(k+1)} = \dfrac{2 - 2x_1^{(k+1)} - 2x_3^{(k)}}{-5} \\ x_3^{(k+1)} = \dfrac{3 - x_1^{(k+1)} - 6x_2^{(k+1)}}{8} \end{cases}$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Gauss-Seidel method

- The iteration of Gauss-Seidel method:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right), \quad i = 1, \cdots, n$$

- or in matrix form

$$x^{(k+1)} = D^{-1}\left(b - Lx^{(k+1)} - Ux^{(k)}\right)$$

$$\Updownarrow$$

$$x^{(k+1)} = (D+L)^{-1}\left(b - Ux^{(k)}\right)$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Convergence

- Definition (Def 2.9, Chapter 2):

The $n \times n$ matrix $A = (a_{ij})$ is **strictly diagonally dominant** if, for each $1 \leq i \leq n$, $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$. In other words, each main diagonal entry dominates its row in the sense that it is greater in magnitude than the sum of magnitudes of the remainder of the entries in its row. ❒

- Theorem: *If the system matrix is diagonally dominant, both methods (Jacobi and Gauss-Seidel) are guaranteed to converge to the correct (unique) solution for any initial guess.*

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Discussion

- Computational cost
  - The cost for each iteration of both Jacobi and Gauss-Seidel methods is O(n^2) and only O(n) if the matrix is sparse, while Gaussian elimination is O(n^3).
- Jacobi vs Gauss-Seidel
  - Gauss-Seidel often converges faster than Jacobi.
  - Gauss-Seidel is a bit easier to implement: the current value can simply be overwritten the new one since it is not needed later anymore.
- Iterative methods are even suitable for the situation where the current problems are a minor change to a problem with a known solution.

Outline

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Symmetric positive-definite matrices

- If the coefficient matrix of the system is symmetric positive-definite, the Gauss-Seidel method converges. However, there exist even better methods.
  - An $n \times n$ matrix A is called **symmetric** if $A^T = A$.
  - An $n \times n$ matrix A is called **positive-definite** if $x^T A x > 0$ for any nonzero x.

- Basic idea: for symmetric positive-definite matrix A, the solution of linear system $Ax = b$ is the minimizer of the quadratic function $f(x) = \frac{1}{2} x^T A x - x^T b$.

Gradient Descent method

- Begin with some initial guess $x^{(0)}$.
- Iteratively compute the next guess $x^{(k+1)}$ by searching for the minimum of $f(x)$
  - starting at the current guess $x^{(k)}$
  - in the opposite directin of the gradient $\nabla f(x^{(k)})$ of f at $x^{(k)}$:

$$x^{(k+1)} = x^{(k)} + \alpha_k \left(-\nabla f(x^{(k)}) \right)$$

where α_k is chosen such that $f(x^{(k+1)})$ is minimal with respect to α_k.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Gradient Descent method

- How to choose α_k.

Set the derivative of of $f(x^{(k+1)})$ with respect to α_k to zero.

Note that $\nabla f(x) = Ax - b$. Let $r_k = -\nabla f(x^{(k)}) = b - Ax^{(k)}$.

Then $x^{(k+1)} = x^{(k)} + \alpha_k r_k$, and

$$\begin{aligned}\frac{df(x^{(k+1)})}{d\alpha_k} &= (\nabla f(x^{(k+1)}))^T \frac{x^{(k+1)}}{d\alpha_k} = (Ax^{(k+1)} - b)^T r_k \\ &= \left(Ax^{(k)} + \alpha_k A r_k - b\right)^T r_k = \alpha_k (Ar_k)^T r_k - r_k^T r_k = 0\end{aligned}$$

which gives

$$\alpha_k = \frac{r_k^T r_k}{r_k^T A r_k}$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Conjugate gradient descent

- The gradient descent method can be further improved. For example, if we do not necessarily descend in the current gradient direction, but in a particularly chosen direction instead.
  - The main trick is to take the direction that are conjugate to the previous one, which follows the *principle of orthogonality*.

(for your own study. Refer to Chapter 2.6.3)

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Outline

§1. Introduction

§2. LU Decomposition

§3. Iterative methods

§4. Symmetric positive-definite matrices

§5. Homework

§6. Summary

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Homework

- Find the $PA = LU$ factorization of the matrix A, where

$$A = \begin{bmatrix} 10 & 20 & 1 \\ 1 & 2 & 6 \\ 0 & 50 & 1 \end{bmatrix}$$

- Hence solve the equations $Ax = b$ with

$$b = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Outline

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Summary

- Direct methods
  - Gaussian elimination
  - PA=LU factorization
- Iterative methods
  - Jacobi method
  - Gauss-Seidel method
- Gradient Descent method for symmetric positive-definite matrices

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

End

Research project example

- Minimization problem:

$$\begin{aligned} \min_{\text{all rods}} \quad & \alpha \sum_{i,j} F_1^2(P_j^i, P_i^j) + \beta \sum_{\{V_i\}} F_2^2(V_1 V_2 \cdots V_m) \\ + \quad & \gamma \sum_i F_3^2(V_{i-1}, V_i, V_{i+1}) \\ + \quad & \zeta \sum_i F_4^2(V_i, V_{i+1}) + \lambda \sum_i (P_i - \bar{P}_i)^2 \ , \end{aligned}$$

where

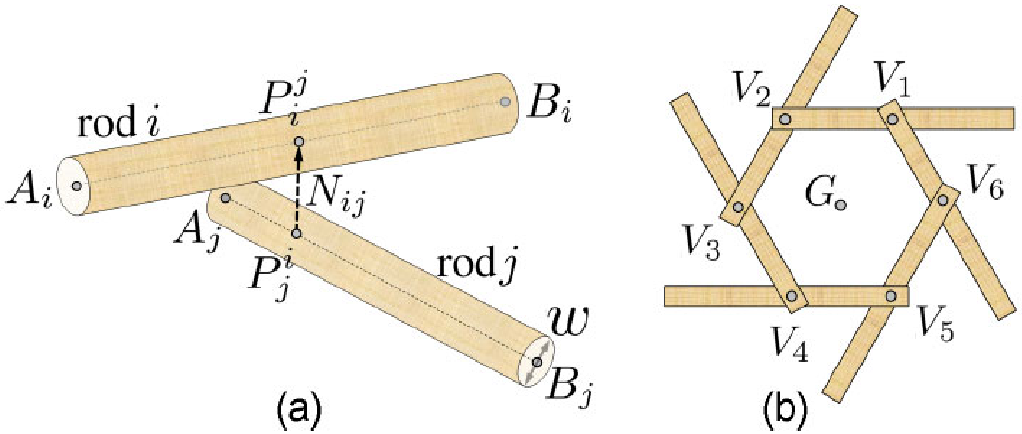
$$F_1(P_i^j, P_j^i) \triangleq P_i^j - P_j^i - wN_{ij} = 0$$

$$F_3(V_{i-1}, V_i, V_{i+1}) \triangleq \frac{(V_{i-1} - V_i) \cdot (V_{i+1} - V_i)}{\|V_{i-1} - V_i\| \|V_{i+1} - V_i\|} - \cos\theta_i = 0$$

…

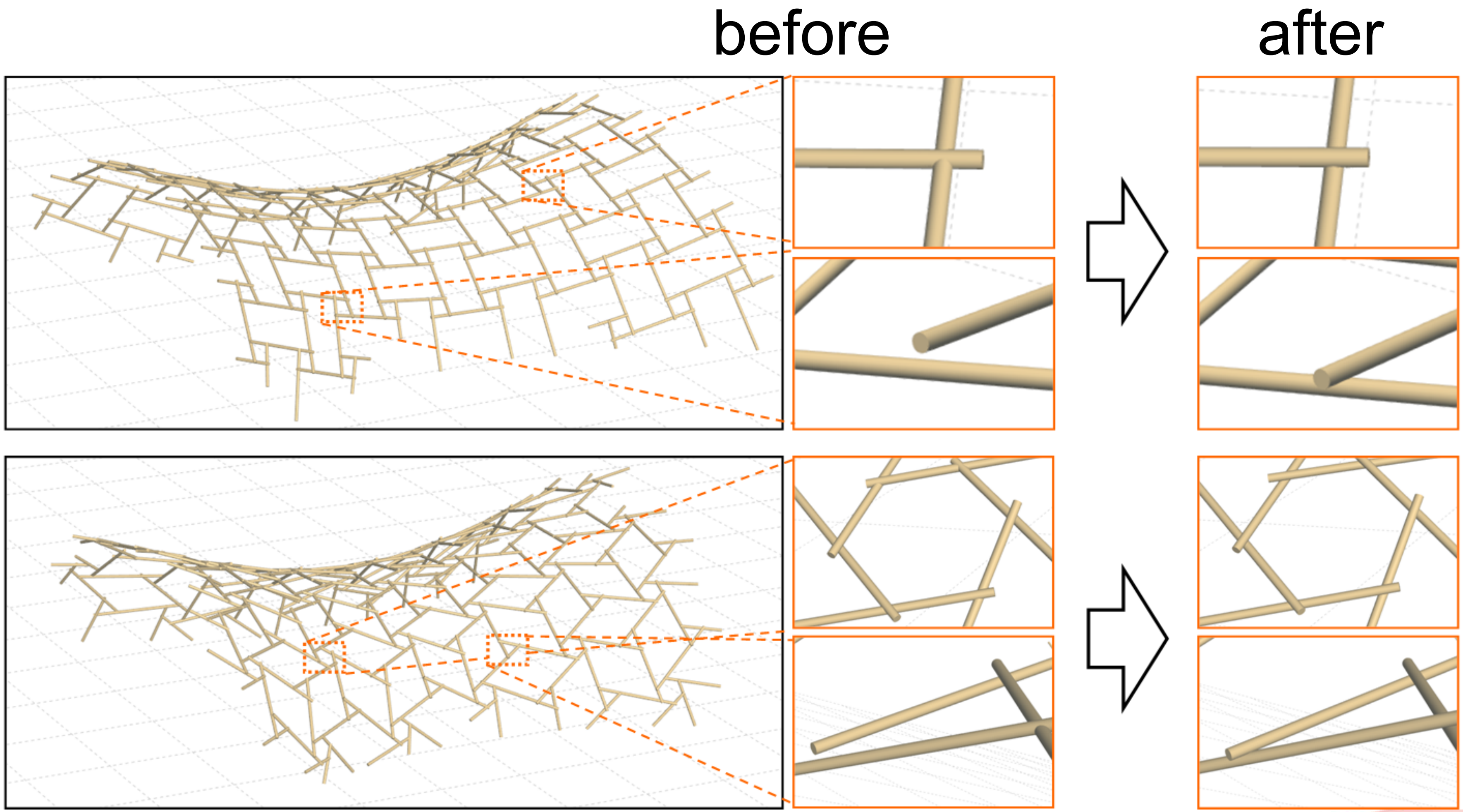
Solution:

- use iteration
- Replace some variables by their current values to linearize the problem.



Research project example

- Design of large Reciprocal Frame structure



One problem: how to ensure rods contacts while preserving the geometric properties of the RF-structure.