Chapter 9

Gauss Elimination

Associate Prof. Mazen Abualtayef
Civil Engineering Department, The Islamic University of Gaza
Engineering Application for Part II
Read 8.4 PIPE FRICTION

\[ 0 = \frac{1}{\sqrt{f}} + 2.0 \log \left( \frac{\varepsilon}{3.7D} + \frac{2.51}{\text{Re} \sqrt{f}} \right) \]

Bisection method
False position method
**Newton-Raphson method**
Fixed-point iteration method
Introduction

Roots of a single equation: \( f(x) = 0 \)

A general set of equations:
- \( n \) equations,
- \( n \) unknowns.

\[
\begin{align*}
 f_1(x_1, x_2, \ldots x_n) &= 0 \\
 f_2(x_1, x_2, \ldots x_n) &= 0 \\
 \vdots & \quad \vdots \\
 f_n(x_1, x_2, \ldots x_n) &= 0
\end{align*}
\]
Linear Algebraic Equations

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

Nonlinear Equations

\[
\begin{align*}
    a_{11}x_1 + a_{12}x_1x_2 + \cdots + a_{1n}(x_n)^5 &= b_1 \\
    a_{21}(x_1)^3 + a_{22}e^{x_2} + \cdots + a_{2n}(x_2)^3 / x_n &= b_2 \\
    \vdots & \quad \vdots \\
    a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
\end{align*}
\]
Review of Matrices

$$[A] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}_{n \times m}$$

$\text{2}\text{nd row}$

Elements are indicated by $a_{ij}$

$\text{row} \quad \text{column}$

$\text{m}\text{th column}$

Row vector:

$$[R] = \begin{bmatrix} r_1 & r_2 & \cdots & r_n \end{bmatrix}_{1 \times n}$$

Column vector:

$$[C] = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}_{m \times 1}$$

Square matrix:

- $[A]_{nxm}$ is a square matrix if $n=m$.
- A system of $n$ equations with $n$ unknowns has a square coefficient matrix.
Review of Matrices

Special types of square matrices

- **Main (principle) diagonal:**
  \([A]_{n \times n}\) consists of elements \(a_{ii}\); \(i = 1, \ldots, n\)

- **Symmetric matrix:**
  If \(a_{ij} = a_{ji}\) \(\rightarrow [A]_{n \times n}\) is a symmetric matrix

- **Diagonal matrix:**
  \([A]_{n \times n}\) is diagonal if \(a_{ij} = 0\) for all \(i = 1, \ldots, n\); \(j = 1, \ldots, n\) and \(i \neq j\)

- **Identity matrix:**
  \([A]_{n \times n}\) is an identity matrix if it is diagonal matrix with \(a_{ii} = 1\) \(i = 1, \ldots, n\). Shown as \([I]\)
Review of Matrices

• **Upper triangular matrix:**

  \[ [A]_{nxn} \text{ is upper triangular if } a_{ij}=0 \quad i=1,...,n \quad \text{and} \quad j=1,...,n \quad \text{and} \quad i>j \]

  All the elements below the main diagonal are zero

• **Lower triangular matrix:**

  \[ [A]_{nxn} \text{ is lower triangular if } a_{ij}=0 \quad i=1,...,n \quad \text{and} \quad j=1,...,n \quad \text{and} \quad i<j \]

• **Inverse of a matrix:**

  \[ [A]^{-1} \text{ is the inverse of } [A]_{nxn} \quad \text{if} \quad [A]^{-1}[A] = [I] \]

• **Transpose of a matrix:**

  \[ [B] \text{ is the transpose of } [A]_{nxn} \quad \text{if} \quad b_{ij}=a_{ji} \quad \text{Shown as } [A]' \text{ or } [A]^T \]
Special Types of Square Matrices

Symmetric

\[
[A] = \begin{bmatrix}
5 & 1 & 2 & 16 \\
1 & 3 & 7 & 39 \\
2 & 7 & 9 & 6 \\
16 & 39 & 6 & 88
\end{bmatrix}
\]

Diagonal

\[
[D] = \begin{bmatrix}
a_{11} & \quad & \quad & \quad \\
& a_{22} & \quad & \quad \\
& & \ddots & \quad \\
& & & a_{nn}
\end{bmatrix}
\]

Identity

\[
[I] = \begin{bmatrix}
1 & \quad & \quad & \quad \\
& 1 & \quad & \quad \\
& & 1 & \quad \\
& & & 1
\end{bmatrix}
\]

Upper Triangular

\[
[A] = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{22} & \ddots & \cdots & \cdots \\
\vdots & \ddots & \ddots & \ddots \\
a_{nn} & \cdots & \cdots & a_{nn}
\end{bmatrix}
\]

Lower Triangular

\[
[A] = \begin{bmatrix}
a_{11} & a_{21} & a_{22} \\
& \ddots & \ddots \\
& & \ddots & \ddots \\
a_{n1} & \cdots & \cdots & a_{nn}
\end{bmatrix}
\]
Review of Matrices

- **Matrix multiplication:**

\[
[A]_{n \times m} \times [B]_{m \times l} = [C]_{n \times l}
\]

Note: \([A][B] \neq [B][A]\)

\[
c_{ij} = \sum_{k=1}^{r} a_{ik} b_{kj}
\]
Review of Matrices

- **Augmented matrix**: is a special way of showing two matrices together.

For example, \( \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \) augmented with the column vector \( \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \) is \( \begin{bmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \end{bmatrix} \)

- **Determinant of a matrix**: A single number. Determinant of \([A]\) is shown as \(|A|\).
Part 3- Objectives

TABLE PT3.1 Specific study objectives for Part Three.

1. Understand the graphical interpretation of ill-conditioned systems and how it relates to the determinant.
2. Be familiar with terminology: forward elimination, back substitution, pivot equation, and pivot coefficient.
3. Understand the problems of division by zero, round-off error, and ill-conditioning.
4. Know how to compute the determinant using Gauss elimination.
5. Understand the advantages of pivoting; realize the difference between partial and complete pivoting.
6. Know the fundamental difference between Gauss elimination and the Gauss-Jordan method and which is more efficient.
7. Recognize how Gauss elimination can be formulated as an $LU$ decomposition.
8. Know how to incorporate pivoting and matrix inversion into an $LU$ decomposition algorithm.
10. Realize how to use the inverse and matrix norms to evaluate system condition.
11. Understand how banded and symmetric systems can be decomposed and solved efficiently.
12. Understand why the Gauss-Seidel method is particularly well suited for large, sparse systems of equations.
13. Know how to assess diagonal dominance of a system of equations and how it relates to whether the system can be solved with the Gauss-Seidel method.
14. Understand the rationale behind relaxation; know where underrelaxation and overrelaxation are appropriate.
Solving Small Numbers of Equations

There are many ways to solve a system of linear equations:

- Graphical method
- Cramer’s rule
- Method of elimination
- Numerical methods for solving larger number of linear equations:
  - Gauss elimination (Chp.9)
  - LU decompositions and matrix inversion (Chp.10)
1. Graphical Method

- For two equations \((n = 2)\):

\[
\begin{align*}
a_{11}x_1 + a_{12}x_2 &= b_1 \\
a_{21}x_1 + a_{22}x_2 &= b_2
\end{align*}
\]

- Solve both equations for \(x_2\): the intersection of the lines presents the solution.

\[
\begin{align*}
x_2 &= -\left(\frac{a_{11}}{a_{12}}\right)x_1 + \frac{b_1}{a_{12}} \quad \Rightarrow \quad x_2 = (\text{slope})x_1 + \text{intercept} \\
x_2 &= -\left(\frac{a_{21}}{a_{22}}\right)x_1 + \frac{b_2}{a_{22}}
\end{align*}
\]

- For \(n = 3\), each equation will be a plane on a 3D coordinate system. Solution is the point where these planes intersect.

- For \(n > 3\), graphical solution is not practical.
Graphical Method - Example

• Solve:

\[ 3x_1 + 2x_2 = 18 \]
\[ -x_1 + 2x_2 = 2 \]

• Plot \( x_2 \) vs. \( x_1 \), the intersection of the lines presents the solution.

\[ \begin{align*}
3x_1 + 2x_2 & = 18 \\
-x_1 + 2x_2 & = 2
\end{align*} \]
Graphical Method

No solution                   Infinite solution               ill condition
(sensitive to round-off errors)
Graphical Method

Mathematically

• Coefficient matrices of (a) & (b) are **singular**. There is no unique solution for these systems. **Determinants** of the coefficient matrices are **zero** and these matrices cannot be inverted.

• Coefficient matrix of (c) is **almost singular**. Its inverse is difficult to take. This system has a unique solution, which is not easy to determine numerically because of its extreme sensitivity to round-off errors.
2. Determinants and Cramer’s Rule

- Determinant can be illustrated for a set of three equations:

\[
[A]\{x\} = \{B\}
\]

- Where \([A]\) is the coefficient matrix:

\[
A = \begin{bmatrix}
  a_{11} & a_{12} & a_{13} \\
  a_{21} & a_{22} & a_{23} \\
  a_{31} & a_{32} & a_{33}
\end{bmatrix}
\]
# Cramer’s Rule

\[
D = \begin{vmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{vmatrix}
\]

\[
D_{11} = \begin{vmatrix}
a_{22} & a_{23} \\
a_{32} & a_{33}
\end{vmatrix} = a_{22}a_{33} - a_{32}a_{23}
\]

\[
D_{12} = \begin{vmatrix}
a_{21} & a_{23} \\
a_{31} & a_{33}
\end{vmatrix} = a_{21}a_{33} - a_{31}a_{23}
\]

\[
D_{13} = \begin{vmatrix}
a_{21} & a_{22} \\
a_{31} & a_{32}
\end{vmatrix} = a_{21}a_{32} - a_{31}a_{22}
\]

\[
D = a_{11} \begin{vmatrix}
a_{22} & a_{23} \\
a_{32} & a_{33}
\end{vmatrix} - a_{12} \begin{vmatrix}
a_{21} & a_{23} \\
a_{31} & a_{33}
\end{vmatrix} + a_{13} \begin{vmatrix}
a_{21} & a_{22} \\
a_{31} & a_{32}
\end{vmatrix}
\]

\[
x_1 = \frac{\begin{vmatrix}
b_1 & a_{12} & a_{13} \\
b_2 & a_{22} & a_{23} \\
b_3 & a_{32} & a_{33}
\end{vmatrix}}{D}
\]

\[
x_2 = \frac{\begin{vmatrix}
a_{11} & b_1 & a_{13} \\
a_{21} & b_2 & a_{23} \\
a_{31} & b_3 & a_{33}
\end{vmatrix}}{D}
\]

\[
x_3 = \frac{\begin{vmatrix}
a_{11} & a_{12} & b_1 \\
a_{21} & a_{22} & b_2 \\
a_{31} & a_{32} & b_3
\end{vmatrix}}{D}
\]
Cramer’s Rule

• For a singular system $D = 0 \rightarrow$ Solution can not be obtained.

• For large systems Cramer’s rule is not practical because calculating determinants is costly.
3. Method of Elimination

• The basic strategy is to successively solve one of the equations of the set for one of the unknowns and to eliminate that variable from the remaining equations by substitution.

• The elimination of unknowns can be extended to systems with more than two or three equations. However, the method becomes extremely tedious to solve by hand.
Elimination of Unknowns Method

Given a 2x2 set of equations:

\[
\begin{align*}
2.5x_1 + 6.2x_2 &= 3.0 \\
4.8x_1 - 8.6x_2 &= 5.5
\end{align*}
\]

• Multiply the 1\textsuperscript{st} eqn by 8.6 and the 2\textsuperscript{nd} eqn by 6.2 \implies

\[
\begin{align*}
21.50x_1 + 53.32x_2 &= 25.8 \\
29.76x_1 - 53.32x_2 &= 34.1
\end{align*}
\]

• Add these equations \implies

\[
51.26 x_1 + 0 x_2 = 59.9
\]

• Solve for \( x_1 \):

\[
x_1 = \frac{59.9}{51.26} = 1.168552478
\]

• Using the 1\textsuperscript{st} eqn solve for \( x_2 \):

\[
x_2 = \frac{(3.0 - 2.5 \times 1.168552478)}{6.2} = 0.01268045242
\]

• Check if these satisfy the 2\textsuperscript{nd} eqn:

\[
4.8 \times 1.168552478 - 8.6 \times 0.01268045242 = 5.5000000004
\]

(Difference is due to the round-off errors).
9.2 Naive Gauss Elimination Method

• It is a formalized way of the previous elimination technique to large sets of equations by developing a systematic scheme or algorithm to eliminate unknowns and to back substitute.

• As in the case of the solution of two equations, the technique for \( n \) equations consists of two phases:
  1. Forward elimination of unknowns.
  2. Back substitution.
Naive Gauss Elimination Method

• Consider the following system of n equations.

\[ a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \]  
\[ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \]  
\[ \vdots \]  
\[ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \]

Form the augmented matrix of \([A|B]\).

**Step 1 : Forward Elimination:** Reduce the system to an upper triangular system.

1.1- First eliminate \(x_1\) from 2\(^{nd}\) to n\(^{th}\) equations.
   - Multiply the 1\(^{st}\) eqn. by \(\frac{a_{21}}{a_{11}}\) & subtract it from the 2\(^{nd}\) equation. This is the new 2\(^{nd}\) eqn.
   - Multiply the 1\(^{st}\) eqn. by \(\frac{a_{31}}{a_{11}}\) & subtract it from the 3\(^{rd}\) equation. This is the new 3\(^{rd}\) eqn.
   \[ \vdots \]
   - Multiply the 1\(^{st}\) eqn. by \(\frac{a_{n1}}{a_{11}}\) & subtract it from the n\(^{th}\) equation. This is the new n\(^{th}\) eqn.
Naive Gauss Elimination Method (cont’d)

Note:
- In these steps the 1st eqn is the pivot equation and $a_{11}$ is the pivot element.
- Note that a division by zero may occur if the pivot element is zero. Naive-Gauss Elimination does not check for this.

The modified system is

$$\begin{bmatrix}
a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\
0 & a'_{22} & a'_{23} & \cdots & a'_{2n} \\
0 & a'_{32} & a'_{33} & \cdots & a'_{3n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & a'_{n2} & a'_{n3} & \cdots & a'_{nn}
\end{bmatrix} \begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
x_n
\end{bmatrix} = \begin{bmatrix}
b_1 \\
b_2 \\
b_3 \\
\vdots \\
b_n
\end{bmatrix}$$

indicates that the system is modified once.
**Naive Gauss Elimination Method (cont’d)**

1.2- Now eliminate $x_2$ from 3\textsuperscript{rd} to n\textsuperscript{th} equations.

The modified system is $\Rightarrow$

$$
\begin{bmatrix}
    a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\
    0 & a'_{22} & a'_{23} & \cdots & a'_{2n} \\
    0 & 0 & a''_{33} & \cdots & a''_{3n} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & a''_{n3} & \cdots & a''_{nn}
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2 \\
    x_3 \\
    \vdots \\
    x_n
\end{bmatrix}
= 
\begin{bmatrix}
    b_1 \\
    b'_2 \\
    b'_3 \\
    \vdots \\
    b''_n
\end{bmatrix}
$$

Repeat steps (1.1) and (1.2) upto (1.n-1).

we will get this upper triangular system $\Rightarrow$

$$
\begin{bmatrix}
    a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\
    0 & a'_{22} & a'_{23} & \cdots & a'_{2n} \\
    0 & 0 & a''_{33} & \cdots & a''_{3n} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & 0 & 0 & a''_{nn}^{(n-1)}
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2 \\
    x_3 \\
    \vdots \\
    x_n
\end{bmatrix}
= 
\begin{bmatrix}
    b_1 \\
    b'_2 \\
    b'_3 \\
    \vdots \\
    b''_n
\end{bmatrix}
$$
Naive Gauss Elimination Method (cont’d)

**Step 2 : Back substitution**

Find the unknowns starting from the last equation.

1. Last equation involves only $x_n$. Solve for it.

   $$ x_n = \frac{b_{n\downarrow}^{(n-1)}}{a_{nn}^{(n-1)}} $$

2. Use this $x_n$ in the (n-1)th equation and solve for $x_{n-1}$.

   ...  

3. Use all previously calculated $x$ values in the 1st eqn and solve for $x_1$.

   $$ x_i = \frac{b_i^{(i-1)} - \sum_{j=i+1}^{n} a_{ij}^{(i-1)} x_j}{a_{ii}^{(i-1)}} \quad \text{for } i = n-1, n-2, \ldots, 1 $$
Summary of Naive Gauss Elimination Method

\[
\begin{bmatrix}
  a_{11} & a_{12} & a_{13} & | & c_1 \\
  a_{21} & a_{22} & a_{23} & | & c_2 \\
  a_{31} & a_{32} & a_{33} & | & c_3 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
  a_{11} & a_{12} & a_{13} & | & c_1 \\
  a'_{22} & a'_{23} & | & c'_2 \\
  a''_{33} & | & c''_3 \\
\end{bmatrix}
\]

Forward elimination

\[
x_3 = \frac{c''_3}{a''_{33}}
\]

\[
x_2 = \frac{(c'_2 - a'_{23}x_3)}{a'_{22}}
\]

\[
x_1 = \frac{(c_1 - a_{12}x_2 - a_{13}x_3)}{a_{11}}
\]

Back substitution
Pseudo-code of Naive Gauss Elimination Method

(a) Forward Elimination

\begin{align*}
&DO \; k = 1, \; n - 1 \\
&DO \; i = k + 1, \; n \\
&\quad \text{factor} = a_{i,k} / a_{k,k} \\
&\quad DO \; j = k + 1 \; \text{to} \; n \\
&\quad \quad a_{i,j} = a_{i,j} - \text{factor} \cdot a_{k,j} \\
&\quad END \; DO \\
&\quad b_i = b_i - \text{factor} \cdot b_k \\
&\quad END \; DO \\
&END \; DO
\end{align*}

(b) Back substitution

\begin{align*}
&x_n = b_n / a_{n,n} \\
&DO \; i = n - 1, \; 1, \; -1 \\
&\quad \text{sum} = 0 \\
&\quad DO \; j = i + 1, \; n \\
&\quad \quad \text{sum} = \text{sum} + a_{i,j} \cdot x_j \\
&\quad END \; DO \\
&\quad x_i = (b_i - \text{sum}) / a_{i,i} \\
&\quad END \; DO
\end{align*}
Naive Gauss Elimination Method

Example 1

Solve the following system using Naive Gauss Elimination.

\[
\begin{align*}
6x_1 - 2x_2 + 2x_3 + 4x_4 &= 16 \\
12x_1 - 8x_2 + 6x_3 + 10x_4 &= 26 \\
3x_1 - 13x_2 + 9x_3 + 3x_4 &= -19 \\
-6x_1 + 4x_2 + x_3 - 18x_4 &= -34
\end{align*}
\]

**Step 0:** Form the augmented matrix

\[
\begin{align*}
6 & -2 & 2 & 4 & | & 16 \\
12 & -8 & 6 & 10 & | & 26 \\
3 & -13 & 9 & 3 & | & -19 \\
-6 & 4 & 1 & -18 & | & -34
\end{align*}
\]

\[\text{R2} - 2\text{R1} \]

\[\text{R3} - 0.5\text{R1} \]

\[\text{R4} - (-\text{R1}) \]
## Naive Gauss Elimination Method

### Example 1 (cont’d)

### Step 1: Forward elimination

1. **Eliminate \( x_1 \)**  
   \[
   \begin{array}{ccc|c}
   6 & -2 & 2 & 4 | 16 \\
   0 & -4 & 2 & 2 | -6 \\
   0 & -12 & 8 & 1 | -27 & \text{R3} - 3 \text{R2} \\
   0 & 2 & 3 & -14 | -18 & \text{R4} - (-0.5 \text{R2}) \\
   \end{array}
   \]
   (Does not change. Pivot is 6)

2. **Eliminate \( x_2 \)**  
   \[
   \begin{array}{ccc|c}
   6 & -2 & 2 & 4 | 16 \\
   0 & -4 & 2 & 2 | -6 & (Does not change. Pivot is -4) \\
   0 & 0 & 2 & -5 | -9 \\
   0 & 0 & 4 & -13 | -21 & \text{R4} - 2 \text{R3} \\
   \end{array}
   \]

3. **Eliminate \( x_3 \)**  
   \[
   \begin{array}{ccc|c}
   6 & -2 & 2 & 4 | 16 & (Does not change.) \\
   0 & -4 & 2 & 2 | -6 & (Does not change.) \\
   0 & 0 & 2 & -5 | -9 & (Does not change. Pivot is 2) \\
   0 & 0 & 0 & -3 | -3 \\
   \end{array}
   \]
Naive Gauss Elimination Method

Example 1 (cont’d)

Step 2: Back substitution

Find \( x_4 \)
\[ x_4 = \frac{-3}{-3} = 1 \]

Find \( x_3 \)
\[ x_3 = \frac{-9+5*1}{2} = -2 \]

Find \( x_2 \)
\[ x_2 = \frac{-6-2*(-2)-2*1}{-4} = 1 \]

Find \( x_1 \)
\[ x_1 = \frac{16+2*1-2*(-2)-4*1}{6} = 3 \]
Naive Gauss Elimination Method Example 2

(Using 6 Significant Figures)

\[
\begin{align*}
3.0 \ x_1 - 0.1 \ x_2 - 0.2 \ x_3 &= 7.85 \\
0.1 \ x_1 + 7.0 \ x_2 - 0.3 \ x_3 &= -19.3 \\
0.3 \ x_1 - 0.2 \ x_2 + 10.0 \ x_3 &= 71.4
\end{align*}
\]

R2− (0.1/3)R1

R3− (0.3/3)R1

Step 1: Forward elimination

\[
\begin{align*}
3.00000 \ x_1 - 0.100000 \ x_2 - 0.200000 \ x_3 &= 7.85000 \\
7.00333 \ x_2 - 0.293333 \ x_3 &= -19.5617 \\
-0.190000 \ x_2 + 10.0200 \ x_3 &= 70.6150
\end{align*}
\]

\[
\begin{align*}
3.00000 \ x_1 - 0.100000 \ x_2 - 0.200000 \ x_3 &= 7.85000 \\
7.00333 \ x_2 - 0.293333 \ x_3 &= -19.5617 \\
10.0120 \ x_3 &= 70.0843
\end{align*}
\]
Step 2: Back substitution

\[ x_3 = 7.00003 \]
\[ x_2 = -2.50000 \]
\[ x_1 = 3.00000 \]

Exact solution:

\[ x_3 = 7.0 \]
\[ x_2 = -2.5 \]
\[ x_1 = 3.0 \]
Pitfalls of Gauss Elimination Methods

1. Division by zero

\begin{align*}
2 \, x_2 + 3 \, x_3 &= 8 \\
4 \, x_1 + 6 \, x_2 + 7 \, x_3 &= -3 \\
2 \, x_1 + x_2 + 6 \, x_3 &= 5
\end{align*}

It is possible that during both elimination and back-substitution phases a division by zero can occur.

2. Round-off errors

In the previous example where up to 6 digits were kept during the calculations and still we end up with close to the real solution.

\[ x_3 = 7.00003, \text{ instead of } x_3 = 7.0 \]
3. **Ill-conditioned systems**

\[
\begin{align*}
  x_1 + 2x_2 &= 10 \\
  1.1x_1 + 2x_2 &= 10.4
\end{align*}
\]

→ \( x_1 = 4.0 \) & \( x_2 = 3.0 \)

\[
\begin{align*}
  x_1 + 2x_2 &= 10 \\
  1.05x_1 + 2x_2 &= 10.4
\end{align*}
\]

→ \( x_1 = 8.0 \) & \( x_2 = 1.0 \)

**Ill conditioned systems** are those where small changes in coefficients result in large change in solution. Alternatively, it happens when two or more equations are nearly identical, resulting a wide ranges of answers to approximately satisfy the equations. Since round off errors can induce small changes in the coefficients, these changes can lead to large solution errors.
Pitfalls of Gauss Elimination (cont’d)

4. Singular systems

- When two equations are identical, we would loose one degree of freedom and be dealing with case of \( n-1 \) equations for \( n \) unknowns.

To check for singularity:

- After getting the forward elimination process and getting the triangle system, then the determinant for such a system is the product of all the diagonal elements. If a zero diagonal element is created, the determinant is Zero then we have a singular system.
- The determinant of a singular system is zero.
Techniques for Improving Solutions

1. **Use of more significant figures** to solve for the round-off error.

2. **Pivoting.** If a pivot element is zero, elimination step leads to division by zero. The same problem may arise, when the pivot element is close to zero. This Problem can be avoided by:
   - Partial pivoting. Switching the rows so that the largest element is the pivot element.
   - Complete pivoting. Searching for the largest element in all rows and columns then switching.

3. **Scaling**
   - Solve problem of ill-conditioned system.
   - Minimize round-off error
Partial Pivoting

Before each row is normalized, find the largest available coefficient in the column below the pivot element. The rows can then be switched so that the largest element is the pivot element.
Use of more significant figures to solve for the round-off error: Example.

Use Gauss Elimination to solve these 2 equations: (keeping only 4 sig. figures)

\[
\begin{align*}
0.0003 \, x_1 + 3.000 \, x_2 &= 2.0001 \\
1.0000 \, x_1 + 1.000 \, x_2 &= 1.000
\end{align*}
\]

\[
\begin{align*}
0.0003 \, x_1 + 3.0000 \, x_2 &= 2.0001 \\
-9999.0 \, x_2 &= -6666.0
\end{align*}
\]

Solve: \(x_2 = 0.6667\) & \(x_1 = 0.0\)

The exact solution is \(x_2 = 2/3\) & \(x_1 = 1/3\)
Use of more significant figures to solve for the round-off error: Example (cont’d).

\[ x_2 = \frac{2}{3} \]

\[ x_1 = \frac{2.0001 - 3(2/3)}{0.0003} \]

<table>
<thead>
<tr>
<th>Significant Figures</th>
<th>( x_2 )</th>
<th>( x_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.667</td>
<td>-3.33</td>
</tr>
<tr>
<td>4</td>
<td>0.6667</td>
<td>0.000</td>
</tr>
<tr>
<td>5</td>
<td>0.66667</td>
<td>0.3000</td>
</tr>
<tr>
<td>6</td>
<td>0.666667</td>
<td>0.33000</td>
</tr>
<tr>
<td>7</td>
<td>0.6666667</td>
<td>0.333000</td>
</tr>
</tbody>
</table>
Pivoting: Example

Now, solving the previous example using the partial pivoting technique:

\[ 1.0000 \times x_1 + 1.0000 \times x_2 = 1.000 \]
\[ 0.0003 \times x_1 + 3.0000 \times x_2 = 2.0001 \]

The pivot is 1.0

\[ 1.0000 \times x_1 + 1.0000 \times x_2 = 1.000 \]
\[ 2.9997 \times x_2 = 1.9998 \]
\[ x_2 = 0.6667 \quad \text{&} \quad x_1 = 0.3333 \]

Checking the effect of the # of significant digits:

<table>
<thead>
<tr>
<th># of sig</th>
<th>x₂</th>
<th>x₁</th>
<th>Ea% in x₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.6667</td>
<td>0.3333</td>
<td>0.01</td>
</tr>
<tr>
<td>5</td>
<td>0.66667</td>
<td>0.33333</td>
<td>0.001</td>
</tr>
</tbody>
</table>
Scaling: Example

- Solve the following equations using naïve gauss elimination: (keeping only 3 sig. figures)
  
  \[ 2 \, x_1 + 100,000 \, x_2 = 100,000 \]
  \[ x_1 + x_2 = 2.0 \]

- Forward elimination:
  
  \[ 2 \, x_1 + 100,000 \, x_2 = 100,000 \]
  \[ -50,000 \, x_2 = -50,000 \]
  
  Solve \( x_2 = 1.00 \) & \( x_1 = 0.00 \)

- The exact solution is \( x_1 = 1.00002 \) & \( x_2 = 0.99998 \)
Scaling: Example (cont’d)

B) Using the scaling algorithm to solve:

\[ 2 \times_1 + 100,000 \times_2 = 100,000 \]
\[ \times_1 + \quad \times_2 = 2.0 \]

Scaling the first equation by dividing by 100,000:

\[ 0.00002 \times_1 + \quad \times_2 = 1.0 \]
\[ \times_1 + \quad \times_2 = 2.0 \]

Rows are pivoted:

\[ \times_1 + \quad \times_2 = 2.0 \]
\[ 0.00002 \times_1 + \quad \times_2 = 1.0 \]

Forward elimination yield:

\[ \times_1 + \quad \times_2 = 2.0 \]
\[ \times_2 = 1.00 \]

Solve: \[ \times_2 = 1.00 \quad & \quad \times_1 = 1.00 \]

The exact solution is \[ \times_1 = 1.00002 \quad & \quad \times_2 = 0.99998 \]
Scaling: Example (cont’d)

C) The scaled coefficient indicate that pivoting is necessary. We therefore pivot but retain the original coefficient to give:

\[ x_1 + x_2 = 2.0 \]
\[ 2x_1 + 100,000x_2 = 100,000 \]

Forward elimination yields:

\[ x_1 + x_2 = 2.0 \]
\[ 100,000x_2 = 100,000 \]

Solve: \( x_2 = 1.00 \) \& \( x_1 = 1.00 \)

Thus, scaling was useful in determining whether pivoting was necessary, but the equation themselves did not require scaling to arrive at a correct result.
The determinate can be simply evaluated at the end of the forward elimination step, when the program employs partial pivoting:

\[ D = a_{11} a_{22} a_{33} \cdots a_{nn}^{(n-1)} (-1)^p \]

where:

\( p \) represents the number of times that rows are pivoted
Example: Gauss Elimination

\[ \begin{align*}
2x_1 + 2x_2 + 3x_3 + 2x_4 &= 0 \\
4x_1 - 3x_2 + x_4 &= -2 \\
6x_1 + x_2 - 6x_3 - 5x_4 &= 6
\end{align*} \]

a) Forward Elimination

\[
\begin{bmatrix}
0 & 2 & 0 & 1 & 6 \\
2 & 2 & 3 & 2 & 14 \\
4 & -3 & 0 & 1 & 6 \\
6 & 1 & -6 & -5 & 6
\end{bmatrix}
\]

\[
\begin{bmatrix}
6 & 1 & -6 & -5 & 6 \\
2 & 2 & 3 & 2 & -2 \\
4 & -3 & 0 & 1 & -7 \\
0 & 2 & 0 & 1 & 0
\end{bmatrix}
\]

\[ R_1 \leftrightarrow R_4 \]
Example: Gauss Elimination (cont’d)

\[
\begin{bmatrix}
6 & 1 & -6 & -5 & | & 6 \\
2 & 2 & 3 & 2 & | & -2 \\
4 & -3 & 0 & 1 & | & -7 \\
0 & 2 & 0 & 1 & | & 0 \\
\end{bmatrix}
\]

R2 \rightarrow 0.33333 \cdot R1

\[
\begin{bmatrix}
6 & 1 & -6 & -5 & | & 6 \\
0 & 1.6667 & 5 & 3.6667 & | & -4 \\
0 & -3.6667 & 4 & 4.3333 & | & -11 \\
0 & 2 & 0 & 1 & | & 0 \\
\end{bmatrix}
\]

R2 \leftrightarrow R3

\[
\begin{bmatrix}
6 & 1 & -6 & -5 & | & 6 \\
0 & -3.6667 & 4 & 4.3333 & | & -11 \\
0 & 1.6667 & 5 & 3.6667 & | & -4 \\
0 & 2 & 0 & 1 & | & 0 \\
\end{bmatrix}
\]
Example: Gauss Elimination (cont’d)

\[
\begin{bmatrix}
6 & 1 & -6 & -5 & 6 \\
0 & -3.6667 & 4 & 4.3333 & -11 \\
0 & 1.6667 & 5 & 3.6667 & -4 \\
0 & 2 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
6 \\
-11 \\
-4 \\
0
\end{bmatrix}
R_3 + 0.45455 \cdot R_2
R_4 + 0.54545 \cdot R_2
\]

\[
\begin{bmatrix}
6 & 1 & -6 & -5 & 6 \\
0 & -3.6667 & 4 & 4.3333 & -11 \\
0 & 0 & 6.8182 & 5.6364 & -9.0001 \\
0 & 0 & 2.1818 & 3.3636 & -5.9999
\end{bmatrix}
\begin{bmatrix}
6 \\
-11 \\
-9.0001 \\
-5.9999
\end{bmatrix}
R_4 - 0.32000 \cdot R_3
\]
Example: Gauss Elimination (cont’d)

\[
\begin{bmatrix}
6 & 1 & -6 & -5 & | & 6 \\
0 & -3.6667 & 4 & 4.3333 & | & -11 \\
0 & 0 & 6.8182 & 5.6364 & | & -9.0001 \\
0 & 0 & 0 & 1.5600 & | & -3.1199
\end{bmatrix}
\]

b) Back Substitution

\[
x_4 = \frac{-3.1199}{1.5600} = -1.9999
\]

\[
x_3 = \frac{-9.0001 - 5.6364(-1.9999)}{6.8182} = 0.33325
\]

\[
x_2 = \frac{-11 - 4.3333(-1.9999) - 4(0.33325)}{-3.6667} = 1.0000
\]

\[
x_1 = \frac{6 + 5(-1.9999) + 6(0.33325) - 1(1.0000)}{6} = -0.50000
\]
Gauss-Jordan Elimination

It is a variation of Gauss elimination. The major differences are:

– When an unknown is eliminated, it is eliminated from all other equations rather than just the subsequent ones.
– All rows are normalized by dividing them by their pivot elements.
– Elimination step results in an identity matrix.
– It is not necessary to employ back substitution to obtain solution.
### Gauss-Jordan Elimination - Example

\[
\begin{bmatrix}
0 & 2 & 0 & 1 & \, 0 \\
2 & 2 & 3 & 2 & \, -2 \\
4 & -3 & 0 & 1 & \, -7 \\
6 & 1 & -6 & -5 & \, 6 \\
\end{bmatrix}
\rightarrow
\begin{bmatrix}
1 & 0.16667 & -1 & -0.83335 & \, 1 \\
2 & 2 & 3 & 2 & \, -2 \\
4 & -3 & 0 & 1 & \, -7 \\
0 & 2 & 0 & 0 & \, 1 \\
\end{bmatrix}
\]
\[
\begin{bmatrix}
1 & 0.16667 & -1 & -0.83335 & 1 \\
0 & 1.6667 & 5 & 3.6667 & -2 \\
0 & -3.6667 & 4 & 4.3334 & -7 \\
0 & 2 & 0 & 1 & 0
\end{bmatrix}
\]

Dividing the 2\textsuperscript{nd} row by 1.6667 and reducing the second column. (operating above the diagonal as well as below) gives:

\[
\begin{bmatrix}
1 & 0 & -1.5 & -1.2000 & 1.4000 \\
0 & 1 & 2.9999 & 2.2000 & -2.4000 \\
0 & 0 & 15.000 & 12.400 & -19.800 \\
0 & 0 & -5.9998 & -3.4000 & 4.8000
\end{bmatrix}
\]

Divide the 3\textsuperscript{rd} row by 15.000 and make the elements in the 3\textsuperscript{rd} Column zero.
Divide the 4\textsuperscript{th} row by 1.5599 and create zero above the diagonal in the fourth column.

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & -0.49999 \\
0 & 1 & 0 & 0 & 1.0001 \\
0 & 0 & 1 & 0 & -0.33326 \\
0 & 0 & 0 & 1 & -1.9999
\end{bmatrix}
\]

Note: Gauss-Jordan method requires almost 50\% more operations than Gauss elimination; therefore it is not recommended to use it.