Numerical Methods in Engineering - LW2

Athanasios A. Markou

PhD, University Lecturer
Aalto University
School of Engineering
Department of Civil Engineering

January 17, 2024

System of Linear Algebraic Equations

SOLVE the simultaneous **Equations**: $[\mathbf{A}][\mathbf{x}] = [\mathbf{b}]$

$$A_{11}x_1 + A_{12}x_2 + A_{13}x_3 + ... + A_{1n}x_n = b_1$$
$$A_{21}x_1 + A_{22}x_2 + A_{23}x_3 + ... + A_{2n}x_n = b_2$$
$$A_{31}x_1 + A_{32}x_2 + A_{33}x_3 + ... + A_{3n}x_n = b_3$$
$$.$$
$$.$$
$$.$$
$$A_{n1}x_1 + A_{n2}x_2 + A_{n3}x_3 + ... + A_{nn}x_n = b_n$$

SOLVE the simultaneous **Equations**: $[\mathbf{A}][\mathbf{x}] = [\mathbf{b}]$

In MATRIX NOTATION the system can be written as, [2]:

$$\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

SOLVE the simultaneous **Equations**: $[\mathbf{A}][\mathbf{x}] = [\mathbf{b}]$

- The solution of a system of linear algebraic equations with many variables is common problem in, [1]:
  1. **Engineering**
  2. Science
  3. Economics
  4. Business
  5. Statistics
  6. etc
- **Linear systems** include, [2]:
  1. **structures**
  2. elastic solids
  3. heat flow
  4. leakage of fluid
  5. electromagnetic fields
  6. electric circuits
- Solve n algebraic equations with n **unknowns**

SOLVE the simultaneous **Equations**: $[\mathbf{A}][\mathbf{x}] = [\mathbf{b}]$

- **Arguably** the **most important topic** in the course, [2].
- **Numerical analysis** of any sort includes **simultaneous equations**, [2].
- '*I personally believe that many more ppl need* **linear algebra** *than calculus.* **Isaac Newton** *might not agree! But he is* **not** *teaching mathematics in the 21^{st} century (and maybe he wasn't a great teacher, but we will give him the benefit of the doubt). Certainly the* **laws of physics** *are well expressed by* **differential equations**. **Newton** *needed calculus - quite right. But the scope of science and engineering and management (and* **life**) *is now so much* **wider**, *and* **linear algebra** *has moved into a central space.*' ∼ Gilbert Strang, [4].
- A great book in **linear algebra**: *Linear Algebra and its applications*, by G. Strang, [4].

Introduction

SOLVE the simultaneous **Equations**: $[\mathbf{A}][\mathbf{x}] = [\mathbf{b}]$

- Numerical analysis of any sort involves the solution of simultaneous equations, [2].
- The variables in vector $\mathbf{x}$ are **dependent** on each other.
- The **equations** we need to solve from physical problems are usually very large and they require **huge** amount of computational resources, [2].
- By using special properties of **coefficient matrix A** (sparseness= most elements are zero) storage requirements can be **reduced** , [2].
- To this end, there are **many algorithms** that DEAL with solution of large systems, [2].

Linear Algebraic Systems - Notation

A linear algebraic system has the following **FORM**, [2]:

$$
\begin{aligned}
A_{11}x_1 + A_{12}x_2 + A_{13}x_3 + \ldots + A_{1n}x_n &= b_1 \\
A_{21}x_1 + A_{22}x_2 + A_{23}x_3 + \ldots + A_{2n}x_n &= b_2 \\
A_{31}x_1 + A_{32}x_2 + A_{33}x_3 + \ldots + A_{3n}x_n &= b_3 \\
&. \\
&. \\
&. \\
A_{n1}x_1 + A_{n2}x_2 + A_{n3}x_3 + \ldots + A_{nn}x_n &= b_n
\end{aligned}
$$

- **In** MATRIX NOTATION the system can be written as, [2]:

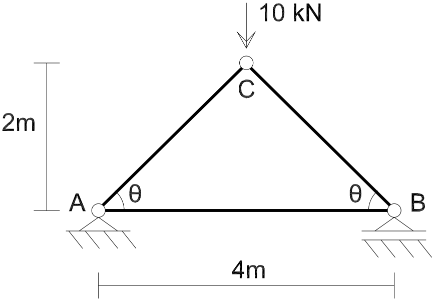
$$\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$[\mathbf{A}][\mathbf{x}] = [\mathbf{b}]$$

- A useful representation for `computational purposes`, is the **augmented coefficient matrix** by joining the coefficient matrix **A** and the constant vector **b**, [2]:

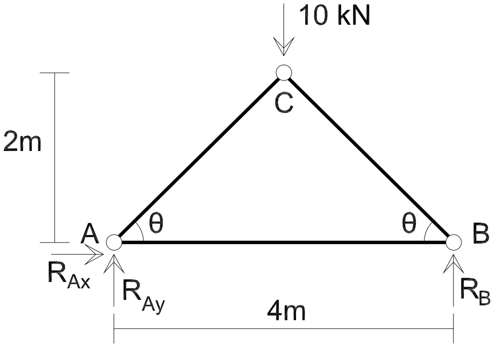
$$[\mathbf{A}|\mathbf{b}] = \left[\begin{array}{cccc|c} A_{11} & A_{12} & \dots & A_{1n} & b_1 \\ A_{21} & A_{22} & \dots & A_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nn} & b_n \end{array}\right]$$

Linear Algebraic Systems - Example



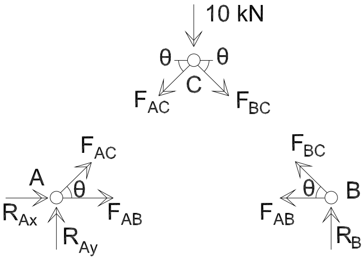
A *statically determinate* (equations of static equilibrium are enough to define the internal forces and reactions) truss (structure with hinged-hinged elements, called **bars** that can only carry axial loads, either compressive or tensile) is given as an example of linear systems.

Linear Algebraic Systems - Example 1



Free Body Diagram

Linear Algebraic Systems - Example 1

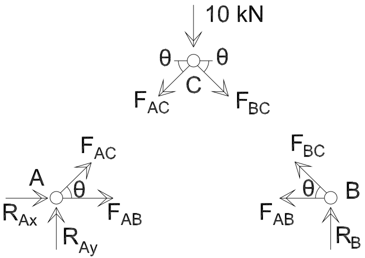


All internal forces of the truss are noted positive (tensile, because the force is going away from the node). In case the calculated force is negative, then the force is compressive. Equilibrium along horizontal axis x and vertical axis y for node A:

$$F_{AC}\cos(\theta) + F_{AB} + R_{Ax} = 0$$

$$R_{Ay} + F_{AC}\sin(\theta) = 0$$

Linear Algebraic Systems - Example 1



All internal forces of the truss are noted positive (tensile, because the force is going away from the node). In case the calculated force is negative, then the force is compressive. Equilibrium along horizontal axis x and vertical axis y for node A and B:

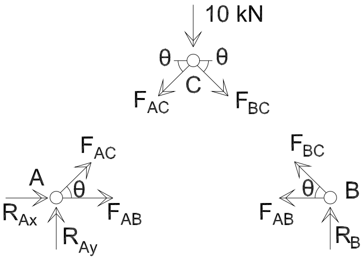
$$F_{AC}\cos(\theta) + F_{AB} + R_{Ax} = 0$$

$$R_{Ay} + F_{AC}\sin(\theta) = 0$$

$$F_{BC}\cos(\theta) + F_{AB} = 0$$

$$F_{BC}\sin(\theta) + R_B = 0$$

Linear Algebraic Systems - Example 1



All internal forces of the truss are noted positive (tensile, because the force is going away from the node). In case the calculated force is negative, then the force is compressive. Finally, equilibrium along horizontal axis x and vertical axis y for all nodes:

$$F_{AC}\cos(\theta) + F_{AB} + R_{Ax} = 0$$

$$R_{Ay} + F_{AC}\sin(\theta) = 0$$

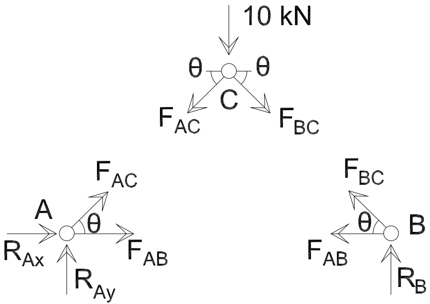
$$F_{BC}\cos(\theta) + F_{AB} = 0$$

$$F_{BC}\sin(\theta) + R_B = 0$$

$$-F_{AC}\cos(\theta) + F_{BC}\cos(\theta) = 0$$

$$-F_{AC}\sin(\theta) - F_{BC}\sin(\theta) = 10$$

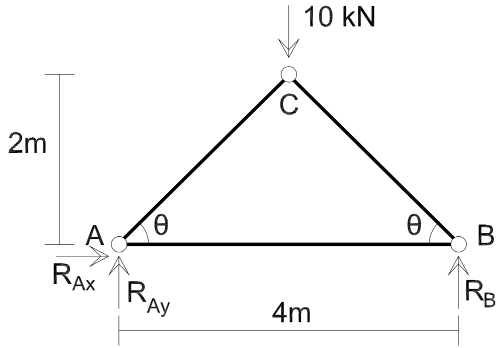
Linear Algebraic Systems - Example 1



$$\mathbf{Ax} = \mathbf{b} \Leftrightarrow$$

$$\begin{bmatrix} 1 & cos(\theta) & 0 & 0 & 1 & 0 \\ 0 & sin(\theta) & 0 & 1 & 0 & 0 \\ 1 & 0 & cos(\theta) & 0 & 0 & 0 \\ 0 & 0 & sin(\theta) & 0 & 0 & 1 \\ 0 & -cos(\theta) & cos(\theta) & 0 & 0 & 0 \\ 0 & -sin(\theta) & -sin(\theta) & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} F_{AB} \\ F_{AC} \\ F_{BC} \\ R_{Ay} \\ R_{Ax} \\ R_B \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix} \Leftrightarrow \begin{bmatrix} F_{AB} \\ F_{AC} \\ F_{BC} \\ R_{Ay} \\ R_{Ax} \\ R_B \end{bmatrix} = \begin{bmatrix} 5 \\ -7.07 \\ -7.07 \\ 5 \\ 0 \\ 5 \end{bmatrix}$$

Linear Algebraic Systems - Example 1



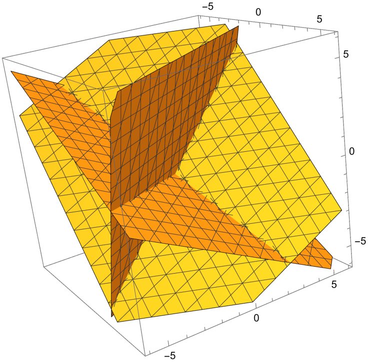
Check with the equilibrium equations of the **structure**:

$$\sum M_A = 0 \Leftrightarrow 10 * 2 - R_B * 4 = 0 \Leftrightarrow 10 * 2 - 5 * 4 = 0$$

$$\sum F_y = 0 \Leftrightarrow R_{Ay} + R_B = 10 \Leftrightarrow 5 + 5 = 10$$

$$\sum F_x = 0 \Leftrightarrow R_{Ax} = 0 \Leftrightarrow 0 = 0$$

Linear Algebraic Systems - Example 2



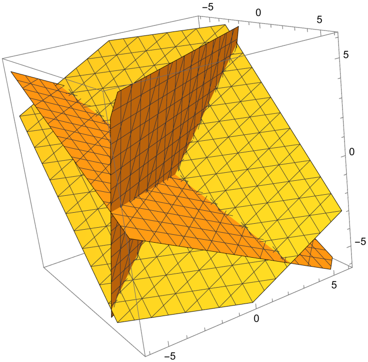
Find the intersection between 3 planes:

$$3x + 2y - z = 1$$

$$2x - 2y + 4z = -2$$

$$-x + \frac{y}{2} - z = 0$$

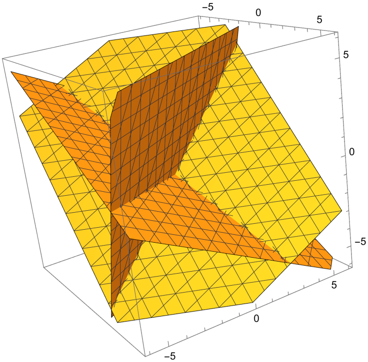
Linear Algebraic Systems - Example 2



Find the intersection between 3 planes:

$$\begin{bmatrix} 3 & 2 & -1 \\ 2 & -2 & 4 \\ -1 & \frac{1}{2} & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix}$$

Linear Algebraic Systems - Example 2



Find the intersection between 3 planes:

$$\begin{bmatrix} 3 & 2 & -1 \\ 2 & -2 & 4 \\ -1 & \frac{1}{2} & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix} \Leftrightarrow \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ -2 \end{bmatrix}$$

- The system has **a** UNIQUE SOLUTION if the determinant of coefficient matrix **A** is different than zero, namely **NONSINGULAR**, [2]:

$$det\left(\mathbf{A}\right) = \left|\mathbf{A}\right| \neq 0$$

- 'The rows and columns of a **NONSINGULAR** matrix are *linearly independent* (no row or column is linear combination of row or column)', [2].
- In the case of SINGULAR matrix the solutions of the system are either infinite or zero, [2].

Linear Algebraic Systems - Uniqueness of solution

- Take the equations:

$$3x + y = 5; \quad 6x + 2y = 10$$

- The second equation can be obtained by multiplying with 2 the first one, so the `coefficient matrix` is singular ($det(\mathbf{A}) = 0$, $[\mathbf{A}] = \begin{bmatrix} 3 & 1 \\ 6 & 2 \end{bmatrix}$). In this case, there are infinite solutions, [2].
- Take the equations:

$$3x + y = 5; \quad 6x + 2y = 0$$

- The second equation contradicts (divide by 2, $3x + y = 0$) the first one, so there are **no solutions**, because `ANY solution` that satisfies the first one, DOES NOT satisfy the second one, [2]. The `coefficient matrix` is singular ($det(\mathbf{A}) = 0$).

Linear Algebraic Systems - Uniqueness of solution

- Check out yourselves in MatLab:
  - Define matrix $[\mathbf{A}] = \begin{bmatrix} 3 & 1 \\ 6 & 2 \end{bmatrix}$
  - Check its determinant `det(A)`

Linear Algebraic Systems - Conditioning

- What happens when the `coefficient matrix` is ALMOST **singular**?
- There is a need for a reference against which the determinant can be measured, [2].
- To this end, we use the NORM of a matrix. If the determinant is small compared to the norm, then the matrix is singular, [2]:

$$|[\mathbf{A}]| << \|[\mathbf{A}]\|$$

- The *Euclidean norm* can be defined as, [2]:

$$\|[\mathbf{A}]\|_e = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2}$$

- The **condition number** can be used to define the conditioning of a matrix, [2]:

$$cond(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|$$

if the number is close to $\mathbf{1}$, the matrix is `well conditioned`. It reaches infinity for singular matrix.

Linear Algebraic Systems - Conditioning

- Condition number is computationally expensive for **LARGE** matrices, [2].
- It is sufficient to estimate `conditioning` by comparing the determinant with the magnitudes of the matrix elements, [2].
- Take the equations:

$$2x + y = 3; \quad 2x + 1.001y = 0$$

  that have the solution $x = 1501.5$ and $y = -3000$. But the determinant $\det(\mathbf{A}) = 0.002$ is much smaller compared with the *coefficients*, [2].
- By changing the second equation to $2x + 1.002y = 0$ and this results in the solution: $x = 751.5$ and $y = -1500$. Only by changing 0.1% in the coefficient of y the solution **changes** 100%.

Linear Algebraic Systems - Error and Residual

- **Numerical solution** is not an exact solution, even though **direct methods** (Gauss, Gauss-Jordan, LU decomposition) can be *exact*, [1].
- **Numerical solution** is prone to round-off errors.
- More susceptible are the large systems and with ill-conditioned systems.
- Iterative methods are `approximate` by their own nature, [1].
- We need **measures** to quantify the **error** of a **numerical solution**

Linear Algebraic Systems - Error and Residual

- $[\mathbf{A}][\mathbf{x}] = [\mathbf{b}]$ is set of n equations.
- $[\mathbf{x_{NS}}]$ is the **numerical solution (approximate)**.
- $[\mathbf{x_{TS}}]$ is the **true solution (exact)**.
- $[\mathbf{e}]$ is the **true error**.
- $[\mathbf{e}] = [\mathbf{x_{TS}}] - [\mathbf{x_{NS}}]$.
- The **true error** $[\mathbf{e}]$ in general **cannot** be calculated because the `exact solution` is `not always` available, [1].
- An alternative measure is $[\mathbf{r}]$, which is the **residual**.
- $[\mathbf{r}] = [\mathbf{A}][\mathbf{x_{TS}}] - [\mathbf{A}][\mathbf{x_{NS}}] = [\mathbf{b}] - [\mathbf{A}][\mathbf{x_{NS}}]$.
- The residual $[\mathbf{r}]$ measures how well the equations $[\mathbf{A}][\mathbf{x}] = [\mathbf{b}]$ are **satisfied** when $[\mathbf{x}]$ is replaced with $[\mathbf{x_{TS}}]$, [1].

Linear Algebraic Systems - Error and Residual - Example

The exact solution of the system

$$\begin{aligned} 1.02x_1 + 0.98x_2 &= 2 \\ 0.98x_1 + 1.02x_2 &= 2 \end{aligned}$$

is $x_1 = x_2 = 1$.
Find the true error and the residual for the approximate solutions:

1. $x_1 = 1.02$, $x_2 = 1.02$
2. $x_1 = 2$, $x_2 = 0$, [1]

Linear Algebraic Systems - Error and Residual - Example

The exact solution of the system

$$\begin{aligned} 1.02x_1 + 0.98x_2 &= 2 \\ 0.98x_1 + 1.02x_2 &= 2 \end{aligned}$$

is $x_1 = x_2 = 1$.

Find the true error and the residual for the approximate solutions:

1. $x_1 = 1.02$, $x_2 = 1.02$
2. $x_1 = 2$, $x_2 = 0$, [1]

Solution

$$[\mathbf{A}][\mathbf{X}] = [\mathbf{b}] \Leftrightarrow \begin{bmatrix} 1.02 & 0.98 \\ 0.98 & 1.02 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

1. For $x_1 = 1.02$, $x_2 = 1.02$ we get:

$$[\mathbf{e}] = [\mathbf{x_{TS}}] - [\mathbf{x_{NS}}] = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1.02 \\ 1.02 \end{bmatrix} = \begin{bmatrix} 0.02 \\ 0.02 \end{bmatrix}$$

$$[\mathbf{r}] = [\mathbf{b}] - [\mathbf{A}][\mathbf{x_{NS}}] = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 1.02 & 0.98 \\ 0.98 & 1.02 \end{bmatrix} \begin{bmatrix} 1.02 \\ 1.02 \end{bmatrix} = - \begin{bmatrix} 0.04 \\ 0.04 \end{bmatrix}$$

Linear Algebraic Systems - Error and Residual - Example

The exact solution of the system

$$\begin{aligned} 1.02x_1 + 0.98x_2 &= 2 \\ 0.98x_1 + 1.02x_2 &= 2 \end{aligned}$$

is $x_1 = x_2 = 1$.

Find the true error and the residual for the approximate solutions:

1. $x_1 = 1.02$, $x_2 = 1.02$
2. $x_1 = 2$, $x_2 = 0$

Solution

2. For $x_1 = 2$, $x_2 = 0$ we get:

$$[\mathbf{e}] = [\mathbf{x_{TS}}] - [\mathbf{x_{NS}}] = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$[\mathbf{r}] = [\mathbf{b}] - [\mathbf{A}][\mathbf{x_{NS}}] = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 1.02 & 0.98 \\ 0.98 & 1.02 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.04 \\ 0.04 \end{bmatrix}$$

In this case the error [**e**] is large, but the residual [**r**] is small. The residual depends on the "magnitude" of the matrix.

Linear Algebraic Systems - Norms and Condition number

- A **norm** is a **real number** assigned to a matrix or vector with the following properties, [1]:
- (i) A **norm** of a vector or matrix $\|[\mathbf{A}]\|$ is a positive quantity. It is zero **only** if the object $[\mathbf{A}]$ itself is zero. In general $\|[\mathbf{A}]\| \geq 0$ unless $[\mathbf{A}] = [\mathbf{0}]$ where $\|[\mathbf{A}]\| = 0$. All vectors or matrices have a positive norm apart from the zero vector or matrix, [1].
- (ii) For all numbers β, $\|\beta[\mathbf{A}]\| = |\beta|\|[\mathbf{A}]\|$, which means that two vectors or matrices $[-\mathbf{A}]$ and $[\mathbf{A}]$, have the same 'magnitude', [1]. Also the magnitude of $\|[5\mathbf{A}]\|$ is 5 times the magnitude of $\|[\mathbf{A}]\|$.
- (iii) For matrices and vectors $\|[\mathbf{A}][\mathbf{x}]\| \leq \|[\mathbf{A}]\|\|[\mathbf{x}]\|$, which means that the **norm** of a product of two matrices is less or equal to the product of the norms of the same matrices, [1].
- (iv) For any matrices or vectors $[\mathbf{A}], [\mathbf{x}]$: $\|[\mathbf{A}]+[\mathbf{x}]\| \leq \|[\mathbf{A}]\|+\|[\mathbf{x}]\|$. It states that the sum of the lengths of two sides of a triangle are always larger than the length of the third side (**triangular inequality**), [1].

Linear Algebraic Systems - Vector Norms

- For a given vector $[\mathbf{v}]$ with n elements the **infinity norm** is:

$$||[\mathbf{v}]||_\infty = max|v_i|; \quad 1 \leq i \leq n$$

The infinity vector norm is a number equal to the element of the vector with the maximum absolute value.

- The **1-norm** of a vector is:

$$||[\mathbf{v}]||_1 = \sum_{i=1}^{n} |v_i|; \quad n = \text{number of elements}$$

The vector **1-norm** is equal to the sum of all the absolute values of its elements.

- The **Euclidean 2-norm** of a vector is:

$$||[\mathbf{v}]||_2 = \left(\sum_{i=1}^{n} v_i^2 \right)^{1/2}; \quad n = \text{number of elements}$$

The vector **Euclidean 2-norm** is the square root of the sum its squared elements.

Linear Algebraic Systems - Matrix Norms

- For a given matrix $[\mathbf{A}]$ with n elements the infinity norm is:

  $$\|[\mathbf{A}]\|_\infty = max \sum_{j=1}^{n} |A_{ij}|; \quad 1 \le i \le n$$

  The **infinity matrix norm** is the value of the largest sum of the absolute values in each row of the matrix.
- The **1-norm** of a matrix is:

  $$\|[\mathbf{A}]\|_1 = max \sum_{i=1}^{n} |A_{ij}|; \quad 1 \le j \le n$$

  The matrix **1-norm** is the value of the largest sum of the absolute values in each column of the matrix.
- The **2-norm** of a matrix is:

  $$\|[\mathbf{A}]\|_2 = max \left(\frac{\|[\mathbf{A}][\mathbf{v}]\|}{\|[\mathbf{v}]\|} \right)$$

  where $[\mathbf{v}]$ is the eigenvector of the matrix $[\mathbf{A}]$ corresponding to an eigenvalue λ.

- For a given $m \times n$ matrix $[\mathbf{A}]$ the **Euclidean norm** is:

$$\|[\mathbf{A}]\|_e = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2}$$

The **Euclidean matrix norm** is different from the matrix **2-norm**.

Linear Algebraic Systems - Matrix Norms - Example

- Find the infinity norm and the 1-norm of the matrix **A**:

$$\mathbf{A} = \begin{bmatrix} 9 & -2 & 3 & 2 \\ 2 & 8 & -2 & 3 \\ -3 & 2 & 11 & -4 \\ -2 & 3 & 2 & 10 \end{bmatrix}$$

- Matrix infinity norm:

$$\|\mathbf{A}\|_\infty = max \sum_{j=1}^{4} |A_{ij}| = max \ of \ |9| + |-2| + |3| + |2|, |2| + |8| + |-2| + |3|,$$

$$|-3|+|2|+|11|+|-4|, |-2|+|3|+|2|+|10| = max[16, 15, 20, 17] = 20; \quad 1 \leq i \leq n$$

- Matrix 1-norm:

$$\|\mathbf{A}\|_1 = max \sum_{i=1}^{4} |A_{ij}| = max \ of \ |9| + |2| + |-3| + |-2|, |-2| + |8| + |2| + |3|,$$

$$|3|+|-2|+|11|+|2|, |2|+|3|+|-4|+|10| = max[16, 15, 18, 19] = 19; \quad 1 \leq j \leq n$$

Linear Algebraic Systems - Matrix Norms - Example

- Find the infinity norm and the 1-norm of the matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} -1 & -2 & -3 \\ 5 & 6 & -2 \\ 3 & 1 & -14 \end{bmatrix}$$

- Matrix infinity norm:

$$\|\mathbf{A}\|_\infty = max \sum_{j=1}^{4} |A_{ij}| =$$

$$max[|-1| + |-2| + |-3|, |5| + |6| + |-2|, |3| + |1| + |-14|] =$$

$$max[6, 13, 18] = 18; \quad 1 \leq i \leq n$$

- Matrix 1-norm:

$$\|\mathbf{A}\|_1 = max \sum_{i=1}^{4} |A_{ij}| =$$

$$max[|-1| + |5| + |3|, |-2| + |6| + |1|, |-3| + |-2| + |-14|] =$$

$$max[9, 9, 19] = 19; \quad 1 \leq j \leq n$$

Linear Algebraic Systems - Norms to define errors

- Residual in terms of error $[\mathbf{r}] = [\mathbf{A}][\mathbf{x_{TS}}] - [\mathbf{A}][\mathbf{x_{NS}}]$
- $[\mathbf{r}] = [\mathbf{A}]\left([\mathbf{x_{TS}}] - [\mathbf{x_{NS}}]\right) = [\mathbf{A}][\mathbf{e}]$
- $[\mathbf{e}] = [\mathbf{A}]^{-1}[\mathbf{r}]$
- $\|[\mathbf{e}]\| = \|[\mathbf{A}]^{-1}[\mathbf{r}]\| \leq \|[\mathbf{A}]^{-1}\|\|[\mathbf{r}]\|$
- $[\mathbf{r}] = [\mathbf{A}][\mathbf{e}]$
- $\|[\mathbf{r}]\| = \|[\mathbf{A}][\mathbf{e}]\| \leq \|[\mathbf{A}]\|\|[\mathbf{e}]\|$
- $\frac{\|[\mathbf{r}]\|}{\|[\mathbf{A}]\|} \leq \|[\mathbf{e}]\| = \|[\mathbf{A}]^{-1}[\mathbf{r}]\| \leq \|[\mathbf{A}]^{-1}\|\|[\mathbf{r}]\|$
- True relative error $\frac{\|[\mathbf{e}]\|}{\|[\mathbf{x_{TS}}]\|}$
- Relative residual $\frac{\|[\mathbf{r}]\|}{\|[\mathbf{b}]\|}$
- $\frac{\|[\mathbf{r}]\|}{\|[\mathbf{A}]\|} \leq \|[\mathbf{e}]\| \leq \|[\mathbf{A}]^{-1}\|\|[\mathbf{r}]\|$
- $\frac{1}{\|[\mathbf{x_{TS}}]\|}\frac{\|[\mathbf{r}]\|}{\|[\mathbf{A}]\|} \leq \frac{\|[\mathbf{e}]\|}{\|[\mathbf{x_{TS}}]\|} \leq \frac{1}{\|[\mathbf{x_{TS}}]\|}\|[\mathbf{A}]^{-1}\|\|[\mathbf{r}]\|$
- $\frac{1}{\|[\mathbf{A}]\|}\frac{\|[\mathbf{b}]\|}{\|[\mathbf{x_{TS}}]\|}\frac{\|[\mathbf{r}]\|}{\|[\mathbf{b}]\|} \leq \frac{\|[\mathbf{e}]\|}{\|[\mathbf{x_{TS}}]\|} \leq \|[\mathbf{A}]^{-1}\|\frac{\|[\mathbf{b}]\|}{\|[\mathbf{x_{TS}}]\|}\frac{\|[\mathbf{r}]\|}{\|[\mathbf{b}]\|}$

Linear Algebraic Systems - Norms to define errors

- $\frac{1}{\|[\mathbf{A}]\|}\frac{\|[\mathbf{b}]\|}{\|[\mathbf{x_{TS}}]\|}\frac{\|[\mathbf{r}]\|}{\|[\mathbf{b}]\|} \leq \frac{\|[\mathbf{e}]\|}{\|[\mathbf{x_{TS}}]\|} \leq \|[\mathbf{A}]^{-1}\|\frac{\|[\mathbf{b}]\|}{\|[\mathbf{x_{TS}}]\|}\frac{\|[\mathbf{r}]\|}{\|[\mathbf{b}]\|}$
- $[\mathbf{b}] = [\mathbf{A}][\mathbf{x_{TS}}]$
- $\|[\mathbf{b}]\| \leq \|[\mathbf{A}]\|\|[\mathbf{x_{TS}}]\|$ or $\frac{\|[\mathbf{b}]\|}{\|[\mathbf{x_{TS}}]\|} \leq \|[\mathbf{A}]\|$
- In the right hand side of first equation replace $\frac{\|[\mathbf{b}]\|}{\|[\mathbf{x_{TS}}]\|}$ with $\|[\mathbf{A}]\|$
- $[\mathbf{x_{TS}}] = [\mathbf{A}]^{-1}[\mathbf{b}]$
- $\|[\mathbf{x_{TS}}]\| \leq \|[\mathbf{A}^{-1}]\|\|[\mathbf{b}]\|$ or $\frac{1}{\|[\mathbf{A}]^{-1}\|} \leq \frac{\|[\mathbf{b}]\|}{\|[\mathbf{x_{TS}}]\|}$
- In the left hand side of first equation replace $\frac{\|[\mathbf{b}]\|}{\|[\mathbf{x_{TS}}]\|}$ with $\frac{1}{\|[\mathbf{A}]^{-1}\|}$
- $\frac{1}{\|[\mathbf{A}]\|\|[\mathbf{A}]^{-1}\|}\frac{\|[\mathbf{r}]\|}{\|[\mathbf{b}]\|} \leq \frac{\|[\mathbf{e}]\|}{\|[\mathbf{x_{TS}}]\|} \leq \|[\mathbf{A}]^{-1}\|\|[\mathbf{A}]\|\frac{\|[\mathbf{r}]\|}{\|[\mathbf{b}]\|}$
- The true relative error $\frac{\|[\mathbf{e}]\|}{\|[\mathbf{x_{TS}}]\|}$ (unknown) is bounded between $\frac{1}{\|[\mathbf{A}]\|\|[\mathbf{A}]^{-1}\|}$ times the relative residual $\frac{\|[\mathbf{r}]\|}{\|[\mathbf{b}]\|}$ (lower bound) and $\|[\mathbf{A}]^{-1}\|\|[\mathbf{A}]\|$ times the relative residual $\frac{\|[\mathbf{r}]\|}{\|[\mathbf{b}]\|}$ (upper bound), [1].

Linear Algebraic Systems - Condition number

- $$\boxed{\frac{1}{\|[\mathbf{A}]\|\|[\mathbf{A}]^{-1}\|}\frac{\|[\mathbf{r}]\|}{\|[\mathbf{b}]\|} \leq \frac{\|[\mathbf{e}]\|}{\|[\mathbf{x_{TS}}]\|} \leq \|[\mathbf{A}]^{-1}\|\|[\mathbf{A}]\|\frac{\|[\mathbf{r}]\|}{\|[\mathbf{b}]\|}}$$
- The number $\|[\mathbf{A}]^{-1}\|\|[\mathbf{A}]\|$ is called the condition number of matrix $[\mathbf{A}]$, $Cond([\mathbf{A}]) = \|[\mathbf{A}]^{-1}\|\|[\mathbf{A}]\|$
- Note that the properties of the inverse matrix $[\mathbf{A}]^{-1}$ are $[\mathbf{A}]^{-1}[\mathbf{A}] = [\mathbf{A}][\mathbf{A}]^{-1} = [\mathbf{I}]$, where $[\mathbf{I}]$ is the identity matrix, where $[\mathbf{A}]$ is a square matrix.
- The condition number of the identity matrix is 1. For all matrices the condition number is larger than 1.
- If the condition number is almost equal to 1, it means that the relative error and the relative residual are of the same order, [1].
- If the condition number is larger than 1, then a small relative residual does not imply necessarily a small relative error, [1].
- The condition number depends on the type of matrix **norm** that is used.
- The `inverse` of a matrix needs to be known to calculate the condition number, [1].

Linear Algebraic Systems - Condition number - Example

Use the 2-norm to evaluate the error bounds for the relative error, by using as [**A**] the Hilbert matrix and as [**b**] ones, [3]. See MatLab file.

Note that Hilbert matrix is defined as follows:

$$H_n = \frac{1}{i + j - 1}; \quad i, j = 1, 2, 3, ...n$$

where n is the number of rows or columns of the square matrix H_n, for example for $n = 3$:

$$H_3 = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}$$

Linear Algebraic Systems - Ill-Conditioned Systems

- In an ill-conditioned system the coefficient matrix has usually condition number significantly larger than 1, [1].
- Consider the following equations:

$$6 \cdot x_1 - 2 \cdot x_2 = 10; \quad 11.5 \cdot x_1 + 3.85 \cdot x_2 = 17$$

- The *solution* is $x_1 = 45$ and $x_2 = 130$.
- If a small change is made in A_{22} from 3.85 to 3.84.

$$6 \cdot x_1 - 2 \cdot x_2 = 10; \quad 11.5 \cdot x_1 + 3.84 \cdot x_2 = 17$$

- The `solution` is $x_1 = 110$ and $x_2 = 325$.
- $$[A] = \begin{bmatrix} 6 & -2 \\ 11.5 & -3.85 \end{bmatrix}; \quad [A]^{-1} = \begin{bmatrix} 38.5 & -20 \\ 115 & -60 \end{bmatrix}$$
- Infinity norm $Cond([\mathbf{A}]) = \|[\mathbf{A}]^{-1}\| \|[\mathbf{A}]\| = 2686.25$
- 1-norm $Cond([\mathbf{A}]) = \|[\mathbf{A}]^{-1}\| \|[\mathbf{A}]\| = 2686.25$
- 2-norm $Cond([\mathbf{A}]) = \|[\mathbf{A}]^{-1}\| \|[\mathbf{A}]\| = 1870.7$
- All condition numbers are much larger than 1. The system is most likely `ill-conditioned`.

Linear Algebraic Systems - Continuous Systems

- For discrete systems like the case of the truss, the analysis leads **directly** to the formulation of the linear system, [2].
- For continuous systems a 'discretization' is required, [2].
- Continuous systems are described by differential equations, [2].
- Methods such as: Finite Difference Method, Finite Element Method, Boundary Element Method use **approximations** to achieve the formulation of linear algebraic systems, [2].
- Equation solving **Algorithms** which can handle the solution of linear systems with minimal computational effort, [2].

Linear Algebraic Systems - Methods

- Two classes of methods for solving linear algebraic equations:
  1. Direct
  2. Iterative
- Direct methods transform initial equations to equivalent equations to be solved *easier*.
- Three operations are needed to transform the equations to equivalent ones:
  1. exchanging two equations
  2. multiplying an equation with a non zero constant
  3. multiplying an equation with a non zero constant and then subtracting from another equation
- Iterative methods use a guess for $\mathbf{x}$ and they **iterate** until they go closer to the solution (*convergence achieved*).
- Iterative methods are less efficient, but they have **advantages** for large and sparse coefficient equations, [2].

Linear Algebraic Systems - Direct Methods Overview

Gauss Elimination	$\mathbf{Ax} = \mathbf{b}$	$\mathbf{Ux} = \mathbf{c}$
LU decomposition	$\mathbf{Ax} = \mathbf{b}$	$\mathbf{LUx} = \mathbf{b}$
Gauss-Jordan Elimination	$\mathbf{Ax} = \mathbf{b}$	$\mathbf{Ix} = \mathbf{b}$

where $\mathbf{I}$ - identity matrix, $\mathbf{U}$ - upper triangular matrix, $\mathbf{L}$ - lower triangular matrix

- Upper Triangular Matrix

$$\mathbf{U} = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

- Lower Triangular Matrix

$$\mathbf{L} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix}$$

Linear Algebraic Systems - Direct Methods Overview

Triangular matrices are **important** in linear algebra, due to the fact that simplify calculations, [2]. Consider the equation, $\mathbf{L}\mathbf{x} = \mathbf{c}$:

$$
\begin{gathered}
L_{11}x_1 = c_1 \Leftrightarrow x_1 = c_1 / L_{11} \\
L_{21}x_1 + L_{22}x_2 = c_2 \Leftrightarrow x_2 = \left(c_2 - L_{21}x_1\right) / L_{22} \\
L_{31}x_1 + L_{32}x_2 + L_{33}x_3 = c_3 \Leftrightarrow x_3 = \left(c_3 - L_{31}x_1 - L_{32}x_2\right) / L_{33}
\end{gathered}
$$

- The process is called forward substitution.
- In the same way $\mathbf{U}\mathbf{x} = \mathbf{c}$ (in Gauss Elimination) can be solved by backward substitution.

Gauss Elimination Method

Gauss Elimination Method

- Most famous method to solve simultaneous equations, [2].
- Two parts:
  1. elimination phase (transform to $\mathbf{Ux} = \mathbf{c}$)
  2. solution phase (solve by back substitution $\mathbf{Ux} = \mathbf{c}$)
- Take the equations:

$$
\begin{aligned}
4x_1 - 2x_2 + x_3 &= 11 && (\alpha) \\
-2x_1 + 4x_2 - 2x_3 &= -16 && (\beta) \\
x_1 - 2x_2 + 4x_3 &= 17 && (\gamma)
\end{aligned}
$$

- Elimination phase:
  $Eq.(i) \leftarrow Eq.(i) - \lambda Eq.(j)$
  where eq.(j) is the pivot equation.
  In the current case: $Eq.(\beta) \leftarrow Eq.(\beta) - \lambda Eq.(\alpha)$

Gauss Elimination Method

- Take the equations:

$$
\begin{aligned}
4x_1 - 2x_2 + x_3 &= 11 && (\alpha)\\
-2x_1 + 4x_2 - 2x_3 &= -16 && (\beta)\\
x_1 - 2x_2 + 4x_3 &= 17 && (\gamma)
\end{aligned}
$$

- Elimination phase:
  $Eq.(\beta) \leftarrow Eq.(\beta) - (-0.5)Eq.(\alpha)$
  $Eq.(\gamma) \leftarrow Eq.(\gamma) - (0.25)Eq.(\alpha)$
- Note that $Eq.(\alpha)$ is the pivot equation.
- So the equations become:

$$
\begin{aligned}
4x_1 - 2x_2 + x_3 &= 11 && (\alpha')\\
3x_2 - 1.5x_3 &= -10.5 && (\beta')\\
-1.5x_2 + 3.75x_3 &= 14.25 && (\gamma')
\end{aligned}
$$

Gauss Elimination Method

-

$$
\begin{aligned}
4x_1 - 2x_2 + x_3 &= 11 & (\alpha') \\
3x_2 - 1.5x_3 &= -10.5 & (\beta') \\
-1.5x_2 + 3.75x_3 &= 14.25 & (\gamma')
\end{aligned}
$$

- Elimination phase:
  $Eq.(\gamma') \leftarrow Eq.(\gamma') - (-0.5)Eq.(\beta')$
- Note that $Eq.(\beta')$ is the pivot equation.
- So the equations become:

$$
\begin{aligned}
4x_1 - 2x_2 + x_3 &= 11 & (\alpha'') \\
3x_2 - 1.5x_3 &= -10.5 & (\beta'') \\
3x_3 &= 9 & (\gamma'')
\end{aligned}
$$

Gauss Elimination Method

-

$$\begin{aligned} 4x_1 - 2x_2 + x_3 &= 11 && (\alpha'') \\ 3x_2 - 1.5x_3 &= -10.5 && (\beta'') \\ 3x_3 &= 9 && (\gamma'') \end{aligned}$$

- So the equations become:

$$\left[\begin{array}{ccc|c} 4 & -2 & 1 & 11 \\ 0 & 3 & -1.5 & -10.5 \\ 0 & 0 & 3 & 9 \end{array}\right]$$

- The **determinant** of the modified coefficient matrix **remains unchanged** with regards to the original coefficient matrix.
- The **determinant** of a triangular matrix $(\mathbf{L}, \mathbf{U})$ is equal to the **product of its diagonals**:
  $|\mathbf{A}| = |\mathbf{U}| = U_{11} * U_{22} * U_{33} * \ldots * U_{nn}$

Gauss Elimination Method

Augmented coefficient matrix:

$$\left[\begin{array}{ccc|c} 4 & -2 & 1 & 11 \\ 0 & 3 & -1.5 & -10.5 \\ 0 & 0 & 3 & 9 \end{array}\right]$$

- **Back substitution** (solution) phase

$$\begin{aligned} x_3 &= 9/3 && = 3 \\ x_2 &= \left(-10.5 + 1.5x_3\right)/3 = \left(-10.5 + 1.5 * 3\right)/3 && = -2 \\ x_1 &= \left(11 + 2x_2 - x_3\right)/4 = \left(11 + 2(-2) - 3\right)/4 && = 1 \end{aligned}$$

Gauss Elimination Method - example

- Take the equations:

$$
\begin{aligned}
2x_1 - 4x_2 + x_3 &= 4 && (\alpha) \\
6x_1 + 2x_2 - x_3 &= 10 && (\beta) \\
-2x_1 + 6x_2 - 2x_3 &= -6 && (\gamma)
\end{aligned}
$$

- Elimination phase:
  $Eq.(\beta) \leftarrow Eq.(\beta) - (6/2)Eq.(\alpha)$
  $Eq.(\gamma) \leftarrow Eq.(\gamma) - (-2/2)Eq.(\alpha)$
- Note that $Eq.(\alpha)$ is the pivot equation.
- So the equations become:

$$
\begin{aligned}
2x_1 - 4x_2 + x_3 &= 4 && (\alpha') \\
14x_2 - 4x_3 &= -2 && (\beta') \\
2x_2 - x_3 &= -2 && (\gamma')
\end{aligned}
$$

Gauss Elimination Method - example

- Take the equations:

$$\begin{aligned} 2x_1 - 4x_2 + x_3 &= 4 && (\alpha') \\ 14x_2 - 4x_3 &= -2 && (\beta') \\ 2x_2 - x_3 &= -2 && (\gamma') \end{aligned}$$

- Elimination phase:
  $Eq.(\gamma') \leftarrow Eq.(\gamma') - (2/14)Eq.(\beta')$
- Note that $Eq.(\beta')$ is the pivot equation.
- So the equations become:

$$\begin{aligned} 2x_1 - 4x_2 + x_3 &= 4 && (\alpha'') \\ 14x_2 - 4x_3 &= -2 && (\beta'') \\ -3/7x_3 &= -12/7 && (\gamma'') \end{aligned}$$

Gauss Elimination Method

- The equations took a triangluar form:

$$
\begin{aligned}
2x_1 - 4x_2 + x_3 &= 4 && (\alpha'') \\
14x_2 - 4x_3 &= -2 && (\beta'') \\
-3/7x_3 &= -12/7 && (\gamma'')
\end{aligned}
$$

- So the equations become:

$$
\begin{aligned}
x_3 = 12/3 = 4 &\Rightarrow x_1 = 4 \\
14x_2 - 4 * 4 = -2 &\Rightarrow x_2 = 1 \\
2x_1 - 4 * 1 + 4 = 4 &\Rightarrow x_1 = 2
\end{aligned}
$$

Gauss Elimination Method - Algorithm

Elimination phase:

$$
\left[\begin{array}{cccccc|c}
A_{11} & A_{12} & \dots & A_{1k} & \dots & A_{1n} & b_1 \\
0 & A_{22} & \dots & A_{2k} & \dots & A_{2n} & b_2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \dots & A_{kk} & \dots & A_{kn} & b_k \\
\hline
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \dots & A_{ik} & \dots & A_{in} & b_i \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \dots & A_{nk} & \dots & A_{nn} & b_n
\end{array}\right]
\begin{array}{l}
\\
\\
\\
\leftarrow \textit{pivot row} \\
\\
\\
\leftarrow \textit{row being transformed} \\
\\
\\
\end{array}
$$

The i_{th} row is to be transformed
A_{ik} is to be eliminated
Multiply the pivot row by $\lambda = A_{ik}/A_{kk}$ and subtract from i_{th} row
The changes in i_{th} row:
$A_{ij} \leftarrow A_{ij} - \lambda A_{kj} \quad j = k, k+1, ..., n$

$b_i \leftarrow b_i - \lambda b_k$

Gauss Elimination Method - Algorithm

Elimination phase:
To transform the coefficient matrix to upper triangular form k and i must range $k = 1, 2, ...n-1$ (selects pivot row) and $i = k+1, k+2, ..., n$ (the row to be transformed)

$$\left[\begin{array}{cccccc|c}
A_{11} & A_{12} & \dots & A_{1k} & \dots & A_{1n} & b_1 \\
0 & A_{22} & \dots & A_{2k} & \dots & A_{2n} & b_2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \dots & A_{kk} & \dots & A_{kn} & b_k \\
\hline
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \dots & A_{ik} & \dots & A_{in} & b_i \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \dots & A_{nk} & \dots & A_{nn} & b_n
\end{array}\right]
\begin{array}{l}
\\ \\ \\ \leftarrow \textit{pivot row} \\ \\ \leftarrow \textit{row being transformed} \\ \\ \\
\end{array}$$

The index i can start from $k+1$ and not k. The coefficient A_{ik} is not replaced by zero because the solution never access the lower triangular part of a table.

Gauss Elimination Method - Algorithm

Back substitution phase:

$$[\mathbf{A}|\mathbf{b}] = \left[\begin{array}{ccccc|c} A_{11} & A_{12} & A_{13} & \dots & A_{1n} & b_1 \\ 0 & A_{22} & A_{23} & \dots & A_{2n} & b_2 \\ 0 & 0 & A_{33} & \dots & A_{3n} & b_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & A_{nn} & b_n \end{array}\right]$$

The last equation is solved first: $A_{nn}x_n = b_n$ and therefore $x_n = b_n/A_{nn}$

For back substitution $x_n, x_{n-1}, .., x_{k+1}$ have been calculated in this order. To define x_k of the k_{th} equation:

$$A_{k,k}x_k + A_{k,k+1}x_{k+1} + ... + A_{kn}x_n = b_k$$

The solution is $x_k = \left(b_k - \sum_{j=k+1}^{n} A_{kj}x_j\right)\frac{1}{A_{kk}} \quad , k = n-1, n-2, ..., 1$

Gauss Elimination Method - Summary

From [1]:

Intial set of equations

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

Step 1

$$\begin{bmatrix} \boxed{a_{11}} & a_{12} & a_{13} & a_{14} \\ \cancel{a_{21}} & a'_{22} & a'_{23} & a'_{24} \\ \cancel{a_{31}} & a'_{32} & a'_{33} & a'_{34} \\ \cancel{a_{41}} & a'_{42} & a'_{43} & a'_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix}$$

Step 2

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & \boxed{a'_{22}} & a'_{23} & a'_{24} \\ 0 & \cancel{a'_{32}} & a''_{33} & a''_{34} \\ 0 & \cancel{a'_{42}} & a''_{43} & a''_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \\ b''_4 \end{bmatrix}$$

Step 3

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & \boxed{a''_{33}} & a''_{34} \\ 0 & 0 & \cancel{a''_{43}} & a'''_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \\ b'''_4 \end{bmatrix}$$

Equations in upper triangular form

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & 0 & a'''_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \\ b'''_4 \end{bmatrix}$$

Pivot Element

Pivot Row

Gauss Elimination Method - Time efficiency

- Gauss *elimination phase* contains approximately, [2]:
  $n^3/3$ operations (multiplications and divisions)
  where n is the number of the equations.
- *Back substitution phase* contains approximately, [2]:
  $n^2/2$ operations (multiplications and divisions)
  where n is the number of the equations.
- Computational time goes to the elimination phase
- The time increases rapidly with number of equations, [2].

LU Decomposition Method

LU Decomposition Methods

- Every matrix $\mathbf{A}$ can be written as a product of two triangular matrices, lower $\mathbf{L}$ and upper $\mathbf{U}$, [2]:

$$\mathbf{A} = \mathbf{LU}; \quad \mathbf{L} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix}; \quad \mathbf{U} = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

- The process of computing the lower $\mathbf{L}$ and upper $\mathbf{U}$ matrices is called LU Decomposition, [2].
- The combinations of LU decompositions are **infinite**.
- Constraints are required to distinguish between the different types of LU decompositions:

Name	Constraints
Doolittle	$L_{ii} = 1, \quad i = 1, 2, ..., n$
Crout	$U_{ii} = 1, \quad i = 1, 2, ..., n$
Choleski	$\mathbf{L} = \mathbf{U}^T$

LU Decomposition Methods

- After decomposition of $\mathbf{A}$, the equation $\mathbf{Ax} = \mathbf{b}$ can easily be solved, [2].
- The equation $\mathbf{Ax} = \mathbf{b}$ can be rewritten as $\mathbf{LUx} = \mathbf{b}$, [2].
- Using the $\mathbf{Ux} = \mathbf{y}$ the equation can be written as:

$$\mathbf{Ly} = \mathbf{b}$$

  which can be solved by forward substitution.
- Then the equation

$$\mathbf{Ux} = \mathbf{y}$$

  will provide the unknown $\mathbf{x}$ by backward substitution.
- *'The advantage of LU decomposition method compared to Gauss elimination method is that once the matrix* $\mathbf{A}$ *has been decomposed, the equation equation* $\mathbf{Ax} = \mathbf{b}$*, can be solved for many constant vectors* $\mathbf{b}$*'*, [2].
- Note that forward and backward substitution processes are less time consuming, compared to the decomposition phase, [2].

Doolittle's LU decomposition

LU Decomposition Method - Doolittle's decomposition

- Doolittle's decomposition is closely related to Gauss elimination, [2]. Assume that there is a matrix **A** and exist the lower and upper triangular matrices:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix}; \quad \mathbf{U} = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

such that $\mathbf{A} = \mathbf{LU}$:

$$\mathbf{A} = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ U_{11}L_{21} & U_{12}L_{21} + U_{22} & U_{13}L_{21} + U_{23} \\ U_{11}L_{31} & U_{12}L_{31} + U_{22}L_{32} & U_{13}L_{31} + U_{23}L_{32} + U_{33} \end{bmatrix}$$

LU Decomposition Method - Doolittle's decomposition

- By applying Gauss elimination:
  row 2 $\leftarrow$ row 2 - $L_{21} x$ row 1 (eliminates A_{21})
  row 3 $\leftarrow$ row 3 - $L_{31} x$ row 1 (eliminates A_{31})

$$\mathbf{A}' = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & U_{22} L_{32} & U_{23} L_{32} + U_{33} \end{bmatrix}$$

- By applying again Gauss elimination:
  row 3 $\leftarrow$ row 3 - $L_{32} x$ row 2 (eliminates A_{32})

$$\mathbf{A}'' = \mathbf{U} = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

- The **U** is identical to the upper triangular matrix of the Gauss elimination
- The off-diagonal elements of **L** are the pivot equation's multipliers.

LU Decomposition Method - Doolittle's decomposition

- The form of the coefficient matrix is:

$$[\mathbf{L} \setminus \mathbf{U}] = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ L_{21} & U_{22} & U_{23} \\ L_{31} & L_{32} & U_{33} \end{bmatrix}$$

- The difference between Gauss elimination and the **Doolittle's** decomposition is that in the later the multipliers of the pivot equations are stored in the lower triangular portion of $\mathbf{A}$.
- The number of operations in $[\mathbf{L} \setminus \mathbf{U}]$ decomposition is the same with Gauss elimination, namely $n^3/3$.

Doolittle's decomposition - Solution phase

- In order to solve $\mathbf{Ly} = \mathbf{b}$ forward substitution is required:

$$\begin{gathered} y_1 = b_1 \\ L_{21} y_1 + y_2 = b_2 \\ \vdots \\ L_{k1} y_1 + L_{k2} y_2 + \ldots + L_{k,k-1} y_{k-1} + y_k = b_k \end{gathered}$$

- For the k^{th} equation we have:

$$y_k = b_k - \sum_{j=1}^{k-1} L_{kj} y_j, \quad k = 2, 3, \ldots, n$$

- MatLab returns an empty vector for $h : p$ when $h > p$ as an index as well as a number
- The back substitution phase for $\mathbf{Ux} = \mathbf{y}$ is identical to the Gauss elimination method.

Doolittle's decomposition - Example

Solve $\mathbf{Ax} = \mathbf{b}$ by using Doolittle's method:

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 1 \\ 1 & 6 & -1 \\ 2 & -1 & 2 \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} 7 \\ 13 \\ 5 \end{bmatrix}$$

Doolittle's decomposition - Example, solution

Gauss elimination method:

row 2 $\leftarrow$ row 2 - 1x row 1 (eliminates A_{21})

row 3 $\leftarrow$ row 3 - 2x row 1 (eliminates A_{31})

$$\mathbf{A}' = \begin{bmatrix} 1 & 4 & 1 \\ [1] & 2 & -2 \\ [2] & -9 & 0 \end{bmatrix}$$

The second Gauss elimination:

row 3 $\leftarrow$ row 3 - $(-4.5)x$ row 2 (eliminates A_{32})

$$\mathbf{A}'' = [\mathbf{L} \setminus \mathbf{U}] = \begin{bmatrix} 1 & 4 & 1 \\ [1] & 2 & -2 \\ [2] & [-4.5] & -9 \end{bmatrix}$$

Doolittle's decomposition - Example, solution

The decomposition is complete:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & -4.5 & 1 \end{bmatrix}; \quad \mathbf{U} = \begin{bmatrix} 1 & 4 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & -9 \end{bmatrix}$$

The solution of $\mathbf{Ly} = \mathbf{b}$:

$$[\mathbf{L}|\mathbf{b}] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 7 \\ 1 & 1 & 0 & 13 \\ 2 & -4.5 & 1 & 5 \end{array}\right]$$

The solution is given by forward substitution:

$$\begin{aligned} y_1 & & = 7 \\ y_2 &= 13 - y_1 = 13 - 7 & = 6 \\ y_3 &= 5 - 2 * y_1 + 4.5 * y_2 = 5 - 2 * 7 + 4.5 * 6 & = 18 \end{aligned}$$

Doolittle's decomposition - Example, solution

The solution of $\mathbf{Ux} = \mathbf{y}$:

$$[\mathbf{U}|\mathbf{y}] = \left[\begin{array}{ccr|r} 1 & 4 & 1 & 7 \\ 0 & 2 & -2 & 6 \\ 0 & 0 & -9 & 18 \end{array}\right]$$

The solution is given by backward substitution:

$$\begin{aligned} x_3 & & = 18/-9 = -2 \\ x_2 &= (6 + 2 * x_3)/2 = (6 + 2 * (-2))/2 & = 1 \\ x_1 &= 7 - 4 * x_2 - x_3 = 7 - 4 * 1 - (-2) & = 5 \end{aligned}$$

Finally:

$$\mathbf{x} = \begin{bmatrix} 5 \\ 1 \\ -2 \end{bmatrix}$$

Choleski's LU Decomposition

LU Decomposition - Choleski's method

- Choleski's decomposition $\mathbf{A} = \mathbf{L}\mathbf{L}^\mathbf{T}$ has two limitations, [2]:
  1. can be applied only to symmetric matrices
  2. the decomposition process includes square roots of elements of $\mathbf{A}$, which can be avoided only if the matrix $\mathbf{A}$ is positive definite. *'In linear algebra, a symmetric $n \times n$ real matrix $\mathbf{M}$ is said to be positive definite if the scalar $\mathbf{z}^T\mathbf{M}\mathbf{z}$ is strictly positive for every non-zero column vector $\mathbf{z}$ of n real numbers. Here $\mathbf{z}^T$ denotes the transpose of $\mathbf{z}$.'*, [5]. The determinant of a positive definite matrix is always positive. In other words, a positive definite matrix is nonsingular. Furthermore, a nonsingular matrix $\mathbf{A}$ is the one that has matrix inverse $\mathbf{A}^{-1}$. MatLab:Determine Whether Matrix Is Symmetric Positive Definite
- It contains $n^3/6$ plus n square root operations, [2].
- It is **half** the number of operations needed for LU decomposition, [2].
- It uses the advantage of symmetry.

LU Decomposition - Choleski's method

- The decomposition is based on:

$$\mathbf{A} = \mathbf{L}\mathbf{L}^{\mathbf{T}}$$

where:

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

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} L_{11}^2 & L_{11}L_{21} & L_{11}L_{31} \\ L_{11}L_{21} & L_{21}^2 + L_{22}^2 & L_{21}L_{31} + L_{22}L_{32} \\ L_{11}L_{31} & L_{21}L_{31} + L_{22}L_{32} & L_{31}^2 + L_{32}^2 + L_{33}^2 \end{bmatrix}$$

- Six equations can be obtained by equating matrices **A** and $\mathbf{L}\mathbf{L}^{\mathbf{T}}$:

$$\begin{aligned}
A_{11} &= L_{11}^2 & L_{11} &= \sqrt{A_{11}} \\
A_{21} &= L_{11}L_{21} & L_{21} &= A_{21}/\sqrt{A_{11}} \\
A_{31} &= L_{11}L_{31} & L_{31} &= A_{31}/\sqrt{A_{11}} \\
A_{22} &= L_{21}^2 + L_{22}^2 & L_{22} &= \sqrt{A_{22} - A_{21}^2/A_{11}} \\
A_{32} &= L_{21}L_{31} + L_{22}L_{32} & L_{32} &= \left(A_{32} - \left(A_{21}A_{31}\right)/A_{11}\right)/\sqrt{A_{22} - A_{21}^2/A_{11}} \\
A_{33} &= L_{31}^2 + L_{32}^2 + L_{33}^2 & L_{33} &= \sqrt{A_{33} - L_{31}^2 - L_{32}^2}
\end{aligned}$$

LU Decomposition - Choleski's method

- For the diagonal terms:

$$L_{jj} = \sqrt{A_{jj} - \sum_{k=1}^{j-1} L_{jk}^2}, \quad j = 1, 2, 3, .., n$$

  note that in programming j will run from 1 to n, because the $\sum_1^0 = 0$

- For the non-diagonal terms:

$$L_{ij} = \left(A_{ij} - \sum_{k=1}^{j-1} L_{ik} L_{jk} \right) / L_{jj}, \quad j = 1, 2, .., n-1, i = j+1, j+2, .., n$$

Choleski's method - Example

- Compute the Choleski's decomposition of the matrix:

$$\mathbf{A} = \begin{bmatrix} 4 & -2 & 2 \\ -2 & 2 & -4 \\ 2 & -4 & 11 \end{bmatrix}$$

- Using the equations derived before for Choleski's method for 3×3 matrix:

$$\mathbf{L} = \begin{bmatrix} \sqrt{4} & 0 & 0 \\ -2/\sqrt{4} & \sqrt{2-(-2)^2/4} & 0 \\ 2/\sqrt{4} & \frac{-4-(-2\cdot 2)/4}{\sqrt{2-(-2)^2/4}} & \sqrt{11-1^2-(-3)^2} \end{bmatrix}$$

- Finally,

$$\mathbf{L} = \begin{bmatrix} 2 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & -3 & 1 \end{bmatrix}$$

Gauss-Jordan Elimination

Gauss-Jordan Elimination

- Is the Gauss elimination to the limits, [2].
- In Gauss elimination only the equations **below the pivot** line are transformed.
- In Gauss-Jordan method, the elimination is `also` carried out on the equations **above** the pivot equation.
- Main **disadvantage** of Gauss-Jordan method is that it involves about $n^3/2$ operations.
- The operations are 1.5 times the number required in Gauss elimination
- In MatLab use `rref([A|b])` with input the augmented form of the coefficient matrix to find the result with Gauss-Jordan method. Try to solve the last example with this function.

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 1 \\ 1 & 6 & -1 \\ 2 & -1 & 2 \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} 7 \\ 13 \\ 5 \end{bmatrix}$$

Gauss-Jordan Elimination - Procedure

Schematic illustration of Gauss-Jordan procedure, [1]:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \\ a_{41} & a_{42} & a_{43} & a_{44} & b_4 \end{bmatrix} \xrightarrow{\text{Gauss–Jordan procedure}} \begin{bmatrix} 1 & 0 & 0 & 0 & b'_1 \\ 0 & 1 & 0 & 0 & b'_2 \\ 0 & 0 & 1 & 0 & b'_3 \\ 0 & 0 & 0 & 1 & b'_4 \end{bmatrix}$$

Gauss-Jordan Elimination - Example

Solve the following equations using Gauss-Jordan elimination method, [1]:

$$\begin{aligned} 4x_1 - 2x_2 - 3x_3 + 6x_4 &= 12 \\ -6x_1 + 7x_2 + 6.5x_3 - 6x_4 &= -6.5 \\ x_1 + 7.5x_2 + 6.25x_3 + 5.5x_4 &= 16 \\ -12x_1 + 22x_2 + 15.5x_3 - x_4 &= 17 \end{aligned}$$

Gauss-Jordan Elimination - Example

In matrix form, [1]:

$$\begin{bmatrix} 4 & -2 & -3 & 6 \\ -6 & 7 & 6.5 & -6 \\ 1 & 7.5 & 6.25 & 5.5 \\ -12 & 22 & 15.5 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ -6.5 \\ 16 \\ 17 \end{bmatrix}$$

Gauss-Jordan Elimination - Example

The augmented matrix (coefficient matrix along with the right hand side), [1]:

$$\begin{bmatrix} 4 & -2 & -3 & 6 & 12 \\ -6 & 7 & 6.5 & -6 & -6.5 \\ 1 & 7.5 & 6.25 & 5.5 & 16 \\ -12 & 22 & 15.5 & -1 & 17 \end{bmatrix}$$

Gauss-Jordan Elimination - Example

The first pivoting row is the first row, and the first element in this row is the pivot element. We normalize it by dividing it with the pivot element, [1]:

$$\begin{bmatrix} \frac{4}{4} & \frac{-2}{4} & \frac{-3}{4} & \frac{6}{4} & \frac{12}{4} \\ -6 & 7 & 6.5 & -6 & -6.5 \\ 1 & 7.5 & 6.25 & 5.5 & 16 \\ -12 & 22 & 15.5 & -1 & 17 \end{bmatrix} = \begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 & 3 \\ -6 & 7 & 6.5 & -6 & -6.5 \\ 1 & 7.5 & 6.25 & 5.5 & 16 \\ -12 & 22 & 15.5 & -1 & 17 \end{bmatrix}$$

Gauss-Jordan Elimination - Example

The first elements in rows 2, 3 and 4 are eliminated, [1]:

$$\begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 & 3 \\ -6 & 7 & 6.5 & -6 & -6.5 \\ 1 & 7.5 & 6.25 & 5.5 & 16 \\ -12 & 22 & 15.5 & -1 & 17 \end{bmatrix} \begin{matrix} \\ \longleftarrow -(-6)\begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 & 3 \end{bmatrix} \\ \longleftarrow -(1)\begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 & 3 \end{bmatrix} \\ \longleftarrow -(-12)\begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 & 3 \end{bmatrix} \end{matrix} = \begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 & 3 \\ 0 & 4 & 2 & 3 & 11.5 \\ 0 & 8 & 7 & 4 & 13 \\ 0 & 16 & 6.5 & 17 & 53 \end{bmatrix}$$

Gauss-Jordan Elimination - Example

Now the pivot row is the second and the pivot element is the second. By normalizing the pivot row with the pivot element, we get, [1]:

$$\begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 & 3 \\ 0 & \frac{4}{4} & \frac{2}{4} & \frac{3}{4} & \frac{11.5}{4} \\ 0 & 8 & 7 & 4 & 13 \\ 0 & 16 & 6.5 & 17 & 53 \end{bmatrix} = \begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 & 3 \\ 0 & 1 & 0.5 & 0.75 & 2.875 \\ 0 & 8 & 7 & 4 & 13 \\ 0 & 16 & 6.5 & 17 & 53 \end{bmatrix}$$

Gauss-Jordan Elimination - Example

All the elements in the second column are eliminated:

$$\begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 & 3 \\ 0 & 1 & 0.5 & 0.75 & 2.875 \\ 0 & 8 & 7 & 4 & 13 \\ 0 & 16 & 6.5 & 17 & 53 \end{bmatrix} \begin{array}{l} \longleftarrow \; -(-0.5)\begin{bmatrix}0 & 1 & 0.5 & 0.75 & 2.875\end{bmatrix} \\ \\ \longleftarrow \; -(8)\begin{bmatrix}0 & 1 & 0.5 & 0.75 & 2.875\end{bmatrix} \\ \longleftarrow \; -(16)\begin{bmatrix}0 & 1 & 0.5 & 0.75 & 2.875\end{bmatrix} \end{array} = \begin{bmatrix} 1 & 0 & -0.5 & 1.875 & 4.4375 \\ 0 & 1 & 0.5 & 0.75 & 2.875 \\ 0 & 0 & 3 & -2 & -10 \\ 0 & 0 & -1.5 & 5 & 7 \end{bmatrix}$$

Gauss-Jordan Elimination - Example

The new pivot row is now row 3 and the pivot element is the third element, [1]:

$$\begin{bmatrix} 1 & 0 & -0.5 & 1.875 & 4.4375 \\ 0 & 1 & 0.5 & 0.75 & 2.875 \\ 0 & 0 & \frac{3}{3} & \frac{-2}{3} & \frac{-10}{3} \\ 0 & 0 & -1.5 & 5 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -0.5 & 1.875 & 4.4375 \\ 0 & 1 & 0.5 & 0.75 & 2.875 \\ 0 & 0 & 1 & -0.667 & -3.333 \\ 0 & 0 & -1.5 & 5 & 7 \end{bmatrix}$$

Gauss-Jordan Elimination - Example

Now all the elements in 3rd column are eliminated, [1]:

$$\begin{bmatrix} 1 & 0 & -0.5 & 1.875 & 4.4375 \\ 0 & 1 & 0.5 & 0.75 & 2.875 \\ 0 & 0 & 1 & -0.667 & -3.333 \\ 0 & 0 & -1.5 & 5 & 7 \end{bmatrix} \begin{matrix} \longleftarrow & -(-0.5)\begin{bmatrix} 0 & 0 & 1 & -0.667 & -3.333 \end{bmatrix} \\ \longleftarrow & -(0.5)\begin{bmatrix} 0 & 0 & 1 & -0.667 & -3.333 \end{bmatrix} \\ & \\ \longleftarrow & -(-1.5)\begin{bmatrix} 0 & 0 & 1 & -0.667 & -3.333 \end{bmatrix} \end{matrix} = \begin{bmatrix} 1 & 0 & 0 & 1.5417 & 2.7708 \\ 0 & 1 & 0 & 1.0833 & 4.5417 \\ 0 & 0 & 1 & -0.667 & -3.333 \\ 0 & 0 & 0 & 4 & 2 \end{bmatrix}$$

Gauss-Jordan Elimination - Example

Now the pivot row is the row 4 and 4th element is the pivot element. We normalize the row with the 4th element, [1]:

$$\begin{bmatrix} 1 & 0 & 0 & 1.5417 & 2.7708 \\ 0 & 1 & 0 & 1.0833 & 4.5417 \\ 0 & 0 & 1 & -0.667 & -3.333 \\ 0 & 0 & 0 & \frac{4}{4} & \frac{2}{4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1.5417 & 2.7708 \\ 0 & 1 & 0 & 1.0833 & 4.5417 \\ 0 & 0 & 1 & -0.667 & -3.333 \\ 0 & 0 & 0 & 1 & 0.5 \end{bmatrix}$$

Gauss-Jordan Elimination - Example

All the element in column 4 are eliminated, [1]:

$$\begin{bmatrix}1 & 0 & 0 & 1.5417 & 2.7708\\ 0 & 1 & 0 & 1.0833 & 4.5417\\ 0 & 0 & 1 & -0.667 & -3.333\\ 0 & 0 & 0 & 1 & 0.5\end{bmatrix} \begin{matrix}\longleftarrow & -(1.5417)\begin{bmatrix}0 & 0 & 0 & 1 & 0.5\end{bmatrix}\\ \longleftarrow & -(1.0833)\begin{bmatrix}0 & 0 & 0 & 1 & 0.5\end{bmatrix}\\ \longleftarrow & -(-0.667)\begin{bmatrix}0 & 0 & 0 & 1 & 0.5\end{bmatrix}\\ & \end{matrix} = \begin{bmatrix}1 & 0 & 0 & 0 & 2\\ 0 & 1 & 0 & 0 & 4\\ 0 & 0 & 1 & 0 & -3\\ 0 & 0 & 0 & 1 & 0.5\end{bmatrix}$$

Gauss-Jordan Elimination - Example

The solution is, [1]:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ -3 \\ 0.5 \end{bmatrix}$$

Gauss-Jordan Elimination - Inverse Matrix

The Gauss-Jordan elimination method can be used to derive the inverse of a matrix, as follows for a 4×4 matrix, [1]:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 1 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 & 1 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} & 0 & 0 & 1 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} & 0 & 0 & 0 & 1 \end{bmatrix}$$

Apply Gauss-Jordan Elimination Method

$$\begin{bmatrix} 1 & 0 & 0 & 0 & a'_{11} & a'_{12} & a'_{13} & a'_{14} \\ 0 & 1 & 0 & 0 & a'_{21} & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & 1 & 0 & a'_{31} & a'_{32} & a'_{33} & a'_{34} \\ 0 & 0 & 0 & 1 & a'_{41} & a'_{42} & a'_{43} & a'_{44} \end{bmatrix}$$

Pivoting

Pivoting

- The order of the equations has a significant effect on the solution, [2]. Consider the equations:

$$\begin{aligned} 2x_1 - x_2 &= 1 \\ -x_1 + 2x_2 - x_3 &= 0 \\ -x_2 + x_3 &= 0 \end{aligned}$$

- The corresponding augmented coefficient matrix is:

$$[\mathbf{A}|\mathbf{b}] = \left[\begin{array}{rrr|r} 2 & -1 & 0 & 1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 1 & 0 \end{array}\right]$$

- From the above equations we can get the right answer, namely $x_1 = x_2 = x_3 = 1$ by Gauss elimination or LU decomposition. If we now exchange the first and third equations:

$$[\mathbf{A}|\mathbf{b}] = \left[\begin{array}{rrr|r} 0 & -1 & 1 & 0 \\ -1 & 2 & -1 & 0 \\ 2 & -1 & 0 & 1 \end{array}\right]$$

- The Gauss elimination fails due to zero pivot element ($A_{11} = 0$).

Pivoting

- It is essential to reorder the equations during elimination phase, [2]. The reorder is also required in the case of a pivot element very small (ϵ) compared to other elements in the pivot row.

$$[\mathbf{A}|\mathbf{b}] = \left[\begin{array}{rrr|r} \epsilon & -1 & 1 & 0 \\ -1 & 2 & -1 & 0 \\ 2 & -1 & 0 & 1 \end{array}\right]$$

- After the first elimination the **augmented coefficient matrix** becomes:

$$\left[\mathbf{A}'|\mathbf{b}'\right] = \left[\begin{array}{lrr|r} \epsilon & -1 & 1 & 0 \\ 0 & 2-1/\epsilon & -1+1/\epsilon & 0 \\ 0 & -1+2/\epsilon & -2/\epsilon & 1 \end{array}\right]$$

- Because ϵ is very small the terms $1/\epsilon$ becomes huge:

$$\left[\mathbf{A}'|\mathbf{b}'\right] = \left[\begin{array}{lrr|r} \epsilon & -1 & 1 & 0 \\ 0 & -1/\epsilon & 1/\epsilon & 0 \\ 0 & 2/\epsilon & -2/\epsilon & 1 \end{array}\right]$$

The second and third equations contradict each other and the solution fails. Problem would disappear if before elimination, lines one and two or one and three would interchange. The difficulty can be avoided by pivoting.

Pivoting - Diagonal Dominance

- An nxn matrix **A** is diagonally dominant if each diagonal element is larger than the sum of the other elements in the same row, [2]

$$|A_{ii}| > \sum_{j=1,\ j\neq i}^{n} |A_{ij}| \ (j = 1, 2, ..., n)$$

- Example: the matrix is not **diagonally dominant**. Try to make it dominant.

$$\begin{bmatrix} -2 & 4 & -1 \\ 1 & -1 & 3 \\ 4 & -2 & 1 \end{bmatrix}$$

- After rearranging the lines the matrix becomes diagonally dominant:

$$\begin{bmatrix} 4 & -2 & 1 \\ -2 & 4 & -1 \\ 1 & -1 & 3 \end{bmatrix}$$

- In order to avoid pivoting we need to arrange the coefficient matrix in diagonally dominant order, [2].

Pivoting - Diagonal Dominance

- Pivoting is aiming at improving diagonal dominance, [2]

$$s_i = max|A_{ij}|, \quad i = 1, 2, ..., n$$

s_i is the scale factor of row i, namely the absolute value of the largest element in the i^{th} row of $\mathbf{A}$.

- The relative size of any element A_{ij} can be defined as:

$$r_{ij} = \frac{|A_{ij}|}{s_i}$$

- When we are in the elimination phase, we do not **automatically** accept A_{kk} as pivot, but we look for a *better* one. The best choice, after looking on k_{th} line, is element A_{pk} with the largest relative size:

$$r_{pk} = \max_{j \geq k} r_{jk}$$

if we find such an element we interchange the lines, [2]. The interchange must take also place in the scale factor $\mathbf{s}$.

Pivoting - Example

- Implement Gauss elimination with scaled row pivoting to solve the problem $\mathbf{Ax} = \mathbf{b}$:

$$\mathbf{A} = \begin{bmatrix} 2 & -2 & 6 \\ -2 & 4 & 3 \\ -1 & 8 & 4 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 16 \\ 0 \\ -1 \end{bmatrix}$$

- Augmented coefficient matrix and scale factor array:

$$[\mathbf{A}|\mathbf{b}] = \left[\begin{array}{ccc|c} 2 & -2 & 6 & 16 \\ -2 & 4 & 3 & 0 \\ -1 & 8 & 4 & -1 \end{array}\right] \quad \mathbf{s} = \begin{bmatrix} 6 \\ 4 \\ 8 \end{bmatrix}$$

- Relative sizes in order to determine the pivot element:

$$\begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \end{bmatrix} = \begin{bmatrix} |A_{11}|/s_1 \\ |A_{21}|/s_2 \\ |A_{31}|/s_3 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/2 \\ 1/8 \end{bmatrix}$$

Pivoting - Example

- Since r_{21} is the biggest element, A_{21} becomes the pivot:

$$[\mathbf{A}|\mathbf{b}] = \left[\begin{array}{ccc|c} -2 & 4 & 3 & 0 \\ 2 & -2 & 6 & 16 \\ -1 & 8 & 4 & -1 \end{array}\right] \qquad \mathbf{s} = \begin{bmatrix} 4 \\ 8 \\ 6 \end{bmatrix}$$

- Gauss elimination is carried out:

$$[\mathbf{A}'|\mathbf{b}'] = \left[\begin{array}{ccc|c} -2 & 4 & 3 & 0 \\ 0 & 2 & 9 & 16 \\ 0 & 6 & 5/2 & -1 \end{array}\right] \qquad \mathbf{s} = \begin{bmatrix} 4 \\ 6 \\ 8 \end{bmatrix}$$

- Potential pivot elements:

$$\begin{bmatrix} * \\ r_{22} \\ r_{32} \end{bmatrix} = \begin{bmatrix} * \\ |A_{22}|/s_2 \\ |A_{32}|/s_3 \end{bmatrix} = \begin{bmatrix} * \\ 1/3 \\ 3/4 \end{bmatrix}$$

Pivoting - Example

- Since r_{32} is the biggest element, A_{32} becomes the :

$$[\mathbf{A}'|\mathbf{b}'] = \left[\begin{array}{ccc|c} -2 & 4 & 3 & 0 \\ 0 & 6 & 5/2 & -1 \\ 0 & 2 & 9 & 16 \end{array}\right] \quad \mathbf{s} = \begin{bmatrix} 4 \\ 8 \\ 6 \end{bmatrix}$$

- The second Gauss elimination yields:

$$[\mathbf{A}''|\mathbf{b}''] = [\mathbf{U}|\mathbf{c}] = \left[\begin{array}{ccc|c} -2 & 4 & 3 & 0 \\ 0 & 6 & 5/2 & -1 \\ 0 & 0 & 49/6 & 49/3 \end{array}\right]$$

- $\mathbf{U}$ is the matrix that should come form the LU decomposition phase. By back substitution of $\mathbf{Ux} = \mathbf{c}$:

$$\mathbf{x} = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}$$

Pivoting - max abs

Another way to use pivoting is to check all the candidate pivot elements from different rows and chose the one with the maximum absolute value and the rows are interchanged.

Iterative Methods

Iterative methods

- Iterative methods start with a guess of the solution $\mathbf{x}$
- They improve the solution until change in $\mathbf{x}$ becomes small (convergence)
- Slower than the direct methods
- Advantages of iterative methods:
  1. Store only nonzero elements of coefficient matrix
  2. Self-correcting (round off errors in one cycle are corrected in the next ones)
- They do not always converge
- They converge if the coefficient matrix is diagonally dominant
- Initial guess affects only number of iterations and not if convergence takes place

Gauss-Seidel Iterative Method

Gauss-Seidel Method

- The equations $\mathbf{Ax} = \mathbf{b}$ can be written in scalar form:

$$\sum_{j=1}^{n} A_{ij} x_j = b_i, \quad i = 1, 2, 3..., n$$

- Extracting the term x_i from the summation:

$$A_{ii} x_i + \sum_{j=1,\ j \neq i}^{n} A_{ij} x_j = b_i, \quad i = 1, 2, 3..., n$$

- Solving for x_i:

$$x_i = \frac{1}{A_{ii}} \left(b_i - \sum_{j=1,\ j \neq i}^{n} A_{ij} x_j \right), \quad i = 1, 2, 3..., n$$

- The iterative scheme is:

$$x_i \leftarrow \frac{1}{A_{ii}} \left(b_i - \sum_{j=1,\ j \neq i}^{n} A_{ij} x_j \right), \quad i = 1, 2, 3..., n$$

Gauss-Seidel Method

- Start by choosing a vector **x**
- The following equation computes the elements of **x**, using the latest available values of x_j

$$x_i \leftarrow \frac{1}{A_{ii}} \left(b_i - \sum_{j=1,\ j\neq i}^{n} A_{ij} x_j \right), \quad i = 1, 2, 3..., n$$

- The process is repeated until the changes in **x** are small (convergence has been achieved).
- Convergence can be improved by a technique called relaxation
- Take a new values as a weighted average of its previous value and the value predicted in the equation above:

$$x_i \leftarrow \frac{\omega}{A_{ii}} \left(b_i - \sum_{j=1,\ j\neq i}^{n} A_{ij} x_j \right) + (1-\omega)\, x_i, \quad i = 1, 2, 3..., n$$

where ω is the relaxation factor.

Gauss-Seidel Method

- $$x_i \leftarrow \frac{\omega}{A_{ii}} \left(b_i - \sum_{j=1,\ j\neq i}^{n} A_{ij} x_j \right) + (1-\omega)\, x_i, \quad i = 1, 2, 3..., n$$

  If $\omega = 1$ no relaxation
  If $\omega < 1$ underrelaxation (interpolation)
  If $\omega > 1$ overrelaxation (extrapolation)
- Optimal value for ω can be obtained by calculating $\Delta x^{(k)} = |\mathbf{x}^{(k-1)} - \mathbf{x}^{(k)}|$, k iterations ($k \geq 5$) implemented without relaxation ($\omega = 1$)
- An approximation of optimal value of ω:

$$\omega_{opt} \approx \frac{2}{1 + \sqrt{1 - \left(\Delta x^{(k+p)} / \Delta x^{(k)}\right)^{1/p}}}$$

where p is an integer.

Gauss-Seidel Method

- Gauss-Seidel algorithm with relaxation:
  1. Carry out k iterations with $\omega = 1$ $(k = 10)$
  2. Record $\Delta x^{(k)}$
  3. Perform additional p iterations
  4. Record $\Delta x^{(k+p)}$
  5. Compute ω_{opt}
  6. Perform all subsequent iterations with $\omega = \omega_{opt}$

Gauss-Seidel Method - Example 1

- Solve the equations with Gauss-Seidel method without relaxation:
$$\begin{bmatrix} 4 & -1 & 1 \\ -1 & 4 & -2 \\ 1 & -2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 12 \\ -1 \\ 5 \end{bmatrix}$$

- $$\begin{aligned} x_1 &= \tfrac{1}{4}(12 + x_2 - x_3) \\ x_2 &= \tfrac{1}{4}(-1 + x_1 - 2x_3) \\ x_3 &= \tfrac{1}{4}(5 - x_1 - 2x_2) \end{aligned}$$

Gauss-Seidel Method - Example 1

- By choosing $x_1 = x_2 = x_3 = 0$

$$
\begin{aligned}
x_1 &= \tfrac{1}{4}(12 + 0 - 0) = 3 \\
x_2 &= \tfrac{1}{4}(-1 + 3 - 2(0)) = 0.5 \\
x_3 &= \tfrac{1}{4}(5 - 3 - 2(0.5)) = 0.75
\end{aligned}
$$

- Second iteration

$$
\begin{aligned}
x_1 &= \tfrac{1}{4}(12 + 0.5 - 0.75) = 2.9375 \\
x_2 &= \tfrac{1}{4}(-1 + 2.9375 - 2(0.75)) = 0.85938 \\
x_3 &= \tfrac{1}{4}(5 - 2.9375 - 2(0.85938)) = 0.94531
\end{aligned}
$$

- Third iteration

$$
\begin{aligned}
x_1 &= \tfrac{1}{4}(12 + 0.85938 - 0.94531) = 2.97852 \\
x_2 &= \tfrac{1}{4}(-1 + 2.97852 - 2(0.94531)) = 0.96729 \\
x_3 &= \tfrac{1}{4}(5 - 2.97852 - 2(0.96729)) = 0.98902
\end{aligned}
$$

- After five iterations, the results converge $x_1 = 3, x_2 = x_3 = 1$ within five decimal places.

MatLab functions

MatLab functions

Some MatLab functions for linear algebra:

- Left division $\backslash$ calculates $[\mathbf{A}][\mathbf{x}] = [\mathbf{b}]$
- Right division $/$ calculates $[\mathbf{x}][\mathbf{A}] = [\mathbf{b}]$
- `lu` function calculates LU decomposition
- `rref` function uses Gauss-Jordan elimination
- `inv` for inverse of matrix
- `det` for determinant
- `norm` for norms
- `cond` for condition number

MatLab functions

The operators $\backslash$ and $/$ in MatLab when solving $[\mathbf{A}][\mathbf{x}] = [\mathbf{b}]$ are operating as follows, [3]:

- if $[\mathbf{A}]$ is triangular the system is solved by backward or forward substitution only,
- else if $[\mathbf{A}]$ is positive definite or Hermitian matrix, Cholesky decompoition is implemented,
- else if $[\mathbf{A}]$ is square matrix, general LU decomposition is applied,
- else if $[\mathbf{A}]$ is full nonsquare matrix QR decomposition is applied ($[\mathbf{A}] = [\mathbf{Q}][\mathbf{R}]$, where $[\mathbf{R}]$ is upper triangluar matrix and $[\mathbf{Q}]$ is orthogonal matrix, namely $[\mathbf{Q}]^T = [\mathbf{Q}]^{-1}$),
- else if $[\mathbf{A}]$ is a sparse nonsquare matrix then minimum degree preordering is applied and then sparse Gaussian elimination is applied.

References

Amos Gilat and Vish Subramaniam.
Numerical Methods for Engineers and Scientists, An Introduction with Applications Using MATLAB.
Wiley, Danvers, Massachusetts, 2014.

Jan Kiusalaas.
Numerical Methods in Engineering with MATLAB.
Cambridge University Press, Cambridge, United Kingdom, 2016.

G. Lindfield and J. Penny.
Numerical Methods: Using MATLAB.
Matlab examples. Elsevier Science, 2012.

Gilbert Strang.
Linear algebra and its applications.
Thomson, Brooks/Cole, Belmont, CA, 2006.

Wikipedia.
Definiteness of a matrix.
https://en.wikipedia.org/wiki/Definiteness_of_a_matrix.