Lecture Module - Systems of Linear Equations

ME3001 - Mechanical Engineering Analysis

Mechanical Engineering

Tennessee Technological University

Module 3 - Systems of Linear Equations

Module 3 - Systems of Linear Equations

- Topic 1 - Linear Systems Review
- Topic 2 - Matrix Multiplication
- Topic 3 - Existence of Solutions

Topic 1 - Linear Systems Review

- What is a Linear Equation ?
- General Form of A Linear System
- The Geometrical Explanation
- Examples in MATLAB

What is a Linear Equation ?

What is a Linear Equation

- "A linear equation is an algebraic equation in which each term is either a constant or the product of a constant and a single variable" - Wikipedia
- slope intercept form
- does not contain

What is a System of Linear Equations?

- multiple linear equations with...
- also known as...

General Form of A Linear System

The general system of linear equations is shown with variables $x_{1,2,..,n}$, coefficients $a_{11,12,..,nm}$, and knowns $b_{1,2,\ldots,m}$

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2$$

$$.$$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m$$

The equations are cast into matrix form of the system.

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ & . & & \\ & . & & \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ . \\ . \\ b_m \end{bmatrix}$$

General Form of A Linear System

To verify the matrix form $[A]\{x\} = \{b\}$ is correct, use matric multiplication and the result will match the individual equations.

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ & . & & \\ & . & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ . \\ . \\ b_m \end{bmatrix}$$

The Geometrical Explanation

Consider the intersection of two Lines on the XY plane (2D).

- Write an equation for each line. $ax + by = c$
- Organize the equations.

The Geometrical Explanation

Consider the intersection of two Lines on the XY plane (2D).

- Cast the system into matrix form.

- Solve the system. What exactly does this mean?
  -
  -
  -

The Geometrical Explanation

Repeat the exercise, and now consider the intersection of three planes in space (3D). What does the solution represent?

- Write an equation for each plane. $ax + by + cz = d$
- Organize the equations.

The Geometrical Explanation

- Cast the system into matrix form.

- Solve the system. What exactly does this mean?
  -
  -
  -

Examples in MATLAB

Topic 2 - Matrix Multiplication

- Motivation
- Multiplication of Conformible Matrices
- Generalized Description of Multiplication
- Exercise in MATLAB

Motivation

- Why do we need to multiply matrices?

- Why do we need to use a computer?

Motivation

Applications of Matrix Multiplication:

-
-
-

Motivation

Multiplication of Conformible Matrices

Consider 2 conformable matrices F and G with elements f_{ij} and g_{ij}. Matrix Multiplication gives the product matrix E with elements e_{ij}.

$$E = F \times G \qquad e_{ij} = \Sigma_{k=1}^{n} f_{ik} \times g_{kj}$$

$$\begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \end{bmatrix} \times \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \\ g_{31} & g_{32} \end{bmatrix}$$

Multiplication of Conformible Matrices

Generalized Description of Multiplication

$$e_{ij} = \Sigma_{k=1}^{n} f_{ik} \times g_{kj}$$

$$\begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} = \left[\begin{array}{c|c} f_{11}g_{11} + f_{12}g_{21} + f_{13}g_{31} & f_{11}g_{12} + f_{12}g_{22} + f_{13}g_{32} \\ f_{21}g_{11} + f_{22}g_{21} + f_{23}g_{31} & f_{21}g_{12} + f_{22}g_{22} + f_{23}g_{32} \end{array}\right]$$

- What does that equation above mean?
- How can we write a *General Solution Technique* using the equation?

Generalized Description of Multiplication

Exercise in MATLAB

A Programming Exercise - Matrix Multiplication

Exercise in MATLAB

Topic 3 - Existence of Solutions

- Techniques for Solving Linear Systems
- Homogeneous and Inhomogeneous Systems
- Solution Existence Cases in 2D
- Numerical Error and System Condition

Techniques for Solving Linear Systems

There are many different techniques for solving linear systems. This is not an exhaustive list.

- Kramer's Method
- Gaussian Elimination
- Gauss-Seidel Method
- Jacobi Method

Techniques for Solving Linear Systems

Homogeneous and Inhomogeneous Systems

Not all problems can be solved with this type of technique!

- non-homogeneous system is one in which ...

- most of the time the system will be non-homogeneous

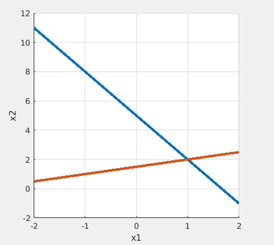
- a non-homogeneous system has a proper solution if and only if

$$rank(A) = rank([A|b]) = n$$

Homogeneous and Inhomogeneous Systems

Solution Existence Cases in 2D

Normal Case - 2 Equations - 2 Unknowns - 1 Solution

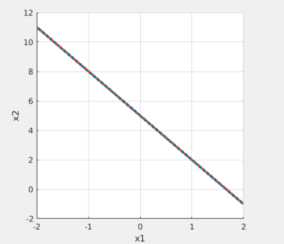


$$3x_1 + x_2 = 5$$

$$x_1 - 2x_2 = -3$$

Solution Existence Cases in 2D

Abnormal Case - 2 Equations - 2 Unknowns - ∞ Solutions

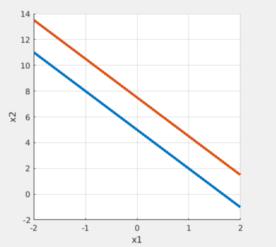


$$3x_1 + x_2 = 5$$

$$6x_1 + 2x_2 = 10$$

Solution Existence Cases in 2D

Abnormal Case - 2 Equations - 2 Unknowns - 0 Solutions

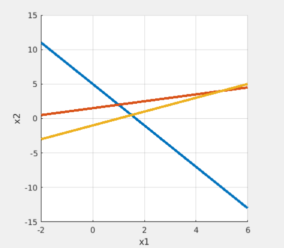


$$3x_1 + x_2 = 5$$

$$6x_1 + 2x_2 = 15$$

Solution Existence Cases in 2D

Abnormal Case - 3 Equations - 2 Unknowns - 0 Solutions



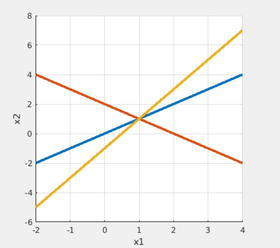
$$3x_1 + x_2 = 5$$

$$x_1 - 2x_2 = -3$$

$$x_1 - x_2 = 1$$

Solution Existence Cases in 2D

Abnormal Case - 3 Equations - 2 Unknowns - 1 Solution



$$-x_1 + x_2 = 0$$

$$x_1 + x_2 = 2$$

$$-2x_1 + x_2 = -1$$

Solution Existence Cases in 2D

What happened to the summary table?

Numerical Error and System Condition

We want our answer to have as little error as possible.

What causes error in the numerical methods?

"In software engineering and mathematics, numerical error is the combined effect of two kinds of error in a calculation. The first is caused by the finite precision of computations involving floating-point or integer values. The second usually called truncation error is the difference between the exact mathematical solution and the approximate solution obtained when simplifications are made to the mathematical equations to make them more amenable to calculation."-wikipedia

Numerical Error and System Condition

Major Causes of Error

- **floating point computations**
- **truncation and solution simplification**
- **system condition**
- **lack of sleep...**

The System Condition can cause problems!

- An ill-conditioned system can cause error.
- A system is ill-conditioned if small changes in the coefficients on the either side of the equation create large variations in the solution.

Numerical Error and System Condition

Consider this simple 2x2 example. The solution will have huge variations if $k \approx 1$.

$$x_1 - x_2 = 5$$

$$kx_1 - x_2 = 4$$

When $k = 0.99$, this gives a solution $(x_1, x_2) = (100, 95)$

$$x_1 - x_2 = 5$$

$$(0.99)x_1 - x_2 = 4$$

When $k = 1.01$, this gives a solution $(x_1, x_2) = (-100, 105)$

$$x_1 - x_2 = 5$$

$$(1.01)x_1 - x_2 = 4$$

Topic 3 - Gaussian Elimination

- Various Row-Reduction Methods
- Gaussian Elimination Technique
- A Generalized Algorithm
-

Various Row-Reduction Methods

The Gaussian Elimination method has many variations. You may have used a different version in linear algebra, but that is fine. This method in generalized so that is can be automated easily with a computer program.

Various Row-Reduction Methods

Gaussian Elimination Technique

The Gaussian Elimination consists of two main steps. Some variations of the method combine the two steps into a single procedure.

1. Forward Elimination of Unknowns
2. Backwards Substitution

Gaussian Elimination Technique

Step 1: Forward Elimination of Unknowns

- Eliminate x_1 from equations 2 to n
  - Eliminate x_1 from equation 2
    - define the eliminating factor f_{21} as a_{21}/a_{11}
    - redefine a_{21} as $a_{21} - a_{11} * f_{21}$
    - redefine a_{22} as $a_{22} - a_{12} * f_{21}$

      . . .
    - redifine a_{2n} as $a_{2n} - a_{1n} * factor$
  - Eliminate x_1 from equation 3
    - define the eliminating factor f_{31} as a_{31}/a_{11}
    - redefine a_{31} as $a_{31} - a_{11} * f_{31}$
    - redefine a_{32} as $a_{32} - a_{12} * f_{31}$

      . . .
    - redefine a_{3n} as $a_{3n} - a_{1n} * f_{31}$

Gaussian Elimination Technique

- Eliminate x_2 from equations 3 to n
  - Eliminate x_2 from equation 3
    - define the eliminating factor f_{32} as a_{32}/a_{22}
    - redefine a_{32} as $a_{32} - a_{22} * f_{32}$
    - redefine a_{33} as $a_{33} - a_{23} * f_{32}$

      . . .
    - redefine a_{3n} as $a_{3n} - a_{2n} * f_{32}$

  . . .
- Eliminate x_{n-1} from equation n
  - define the eliminating factor $f_{n,n-1}$ as $a_{n,n-1}/a_{n-1,n-1}$
  - redefine $a_{n,n-1}$ as $a_{n,n-1} - a_{n-1,n-1} * f_{n,n-1}$

Gaussian Elimination Technique

Step 2: Backwards Subsitution

- Solve Equations n through 1
  - Solve for x_n as $\frac{b_n}{a_{n,n}}$
  - Solve for x_{n-1} as $\frac{b_{n-1}-(a_{n-1,n}x_n)}{a_{n-1,n-1}}$
  - Solve for x_{n-2} as $\frac{b_{n-2}-(a_{n-2,n-1}x_{n-1})-(a_{n-2,n}x_n)}{a_{n-2,n-2}}$

    .
    .
    .
  - Solve for x_1 as $\frac{b_1-(a_{12}x_2)-\ldots-(a_{1,n-1}x_{n-1})-(a_{1,n}x_n)}{a_{1,1}}$

A Generalized Algorithm

Step 1: Forward Elimination

for k *from* 1 *to* n-1

for i *from* k+1 *to* n

fact$= a_{i,k}/a_{k,k}$

for j *from* k *to* n

$a_{i,j} = a_{i,j} - fact \times a_{k,j}$

end

$b_i = b_i - fact \times b_k$

end

end

Step 2: Backwards Substitution

$x_n = b_n/a_{n,n}$

for i *from* n-1 *to* 1

$x_i = (b_i - \sum_{j=i+1}^{n} (a_{i,j}x_j))/a_{i,i}$

end