MONASH University

MONASH
ENGINEERING
ENG1060

LINEAR SYSTEMS

Edited and Presented by Soon Foo Chong (Joseph)

Slides by Tony Vo

Assisted by Tham Lai Kuan & Christopher Ng

GROUP OF EIGHT AUSTRALIA

HOUSEKEEPING

- Weekly Moodle post
  - Week 11 Moodle announcement
- Lab-related items
  - Lab 8 marks and feedback available on Moodle Grade Book
  - Lab 9 solutions available on Gdrive > Labs
- PASS Sessions

  1) Monday (3:30-5:30pm MYT , 6:30-8:30pm AEDT):
  https://monash.zoom.us/j/89128532133?pwd=VVVOenhDbW5xZ3h6ZFRZR1dieVhldz09
  2) Tuesday (12-2pm MYT , 3-6pm AEDT):
  https://monash.zoom.us/j/85226581851?pwd=d0YxeWVHd0tudnplanFRYWU2ZGJRUT09

HOUSEKEEPING

- Assignment due this Friday (22 Jan 2021, 8pm MYT / 11pm AEDT)
  - Please check your allocated session from assignment-marking schedule: https://drive.google.com/drive/folders/1k3Bb54xrSnxGx95kHe3u-v_pC6-HAHVv?usp=sharing
  - Remember to check your submission files (e.g. working, not blank, data files, etc.)
  - Do not leave submission until the last minute. Aim to upload it by 5pm MYT / 8pm AEDT – zipping, computer, Moodle issues, etc.
  - Use the support channels available

	Group 01 (Tuesday 9am MYT / 12 Noon AEDT)		
	Christopher Ng		
Zoom link			
Zoom ID			
Time	Student ID	First Name	Last Name
9.00am - 9.30am	1234567	abc	def
9.30am - 10.00am			

HOUSEKEEPING

- SETU questionnaire is now open for a limited time
  - Please spend 5-10 minutes to complete the is during the workshop
  - Always seeking feedback and striving for continuous improvement

IN THIS WORKSHOP

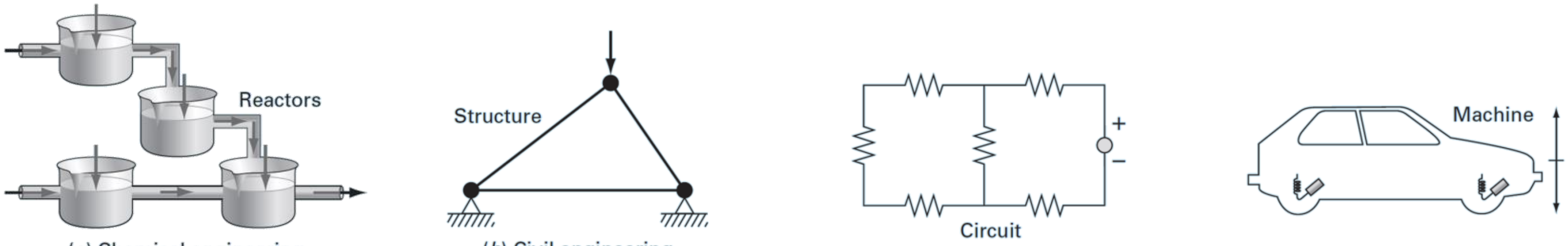
1. Understanding methods to solve a system of linear equations
    a. Naïve Gaussian elimination
    b. Gaussian elimination
    c. Gauss-Jordan elimination

2. Creating function files to solve a system of linear equations
3. Solving a system of linear equations
4. Using inbuilt left division and matrix inversion

RECAP: LINEAR SYSTEMS

- Linear algebraic equations appear everywhere in engineering and science
  - Many fundamental equations based on conservation laws
  - E.g. mass, energy, and momentum

- System of equations can be written in matrix form: $[A][x] = [b]$
  - Matrix A and vector b are known
  - Vector x is unknown



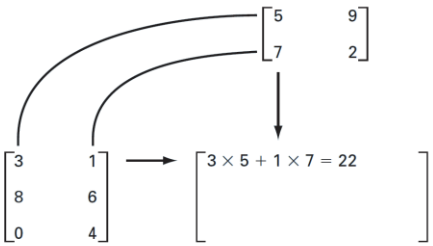
(a) Chemical engineering (b) Civil engineering (c) Electrical engineering (d) Mechanical engineering

RECAP: MATRICES

- Matrix notation: element a_{rc}
  - r is the row and c is the column

- Matrix multiplication $[S] = [A] \times [B]$, each element of $[S]$ is

$$S_{rc} = \sum_{k=1}^{\#\text{cols of A}} a_{rk} b_{kc}$$



Column 3

Row 2

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1c} \\ a_{21} & a_{22} & \boldsymbol{a_{23}} & \cdots & a_{2c} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ a_{r1} & a_{r2} & a_{r3} & \cdots & \boldsymbol{a_{rc}} \end{bmatrix}$$

Row r

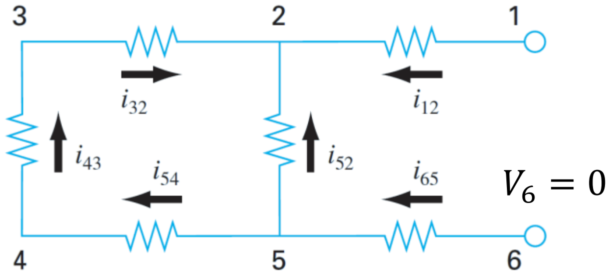
Column c

RECAP: FORMING THE MATRIX EQUATION

- System of equations can be written in matrix form:

$$[A][x] = [b]$$

  - Apply Kirchhoff's current and voltage rules



$$i_{12} + i_{52} + i_{32} = 0$$
$$i_{65} - i_{52} - i_{54} = 0$$
$$i_{43} - i_{32} = 0$$
$$i_{54} - i_{43} = 0$$
$$-i_{54}R_{54} - i_{43}R_{43} - i_{32}R_{32} + i_{52}R_{52} = 0$$
$$-i_{65}R_{65} - i_{52}R_{52} + i_{12}R_{12} - V_1 = 0$$

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & R_{52} & -R_{32} & 0 & -R_{54} & -R_{43} \\ R_{12} & -R_{52} & 0 & -R_{65} & 0 & 0 \end{bmatrix} \begin{bmatrix} i_{12} \\ i_{52} \\ i_{32} \\ i_{65} \\ i_{54} \\ i_{43} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ V_1 \end{bmatrix}$$

RECAP: AUGMENTED MATRIX

- Consider n equations and n unknowns: $[A][x] = [b]$

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n &= b_n
\end{aligned}
$$

- Augmented matrix is defined as Aug = [A b];

$$
\text{Aug} = \left[\begin{array}{ccccc|c}
a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\
a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & b_2 \\
. & . & . & & . & . \\
. & . & . & & . & . \\
. & . & . & & . & . \\
a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & b_n
\end{array}\right]
$$

RECAP: SOLVING MATRICES

- Matrix equation $[A][x] = [b]$ can be solved a variety of ways:

- Matrix inversion
  - $[A]^{-1}[A][x] = [A]^{-1}[b] \rightarrow [x] = [A]^{-1}[b]$
  - x = inv(A)*b

- Elimination of unknowns
  - Gaussian elimination
  - Gauss-Jordan elimination
  - MATLAB's inbuilt left division (x = A\b)
  - Thomas algorithm, Jacobi method, Gauss-Seidel method, etc. (not in ENG1060)

RECAP: GAUSSIAN ELIMINATION

- Gaussian elimination is the most basic scheme for solving the matrix equation $[A][x] = [b]$

- It consists of 2 steps
  a) Forward elimination
  b) Back substitution

- First consider naïve Gaussian elimination
  – Called “naïve” because diagonal elements with zero value are not specially treated
  – This will result in divide-by-zero issues

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right]$$

$$\downarrow$$

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ & a'_{22} & a'_{23} & b'_2 \\ & & a''_{33} & b''_3 \end{array}\right]$$

(*a*) Forward elimination

$$\downarrow$$

$$x_3 = b''_3/a''_{33}$$

$$x_2 = (b'_2 - a'_{23}x_3)/a'_{22}$$

$$x_1 = (b_1 - a_{13}x_3 - a_{12}x_2)/a_{11}$$

(*b*) Back substitution

RECAP: FORWARD ELIMINATION

1) Starting with the first row, add or subtract multiples of that row to eliminate the first coefficient from the second row to the last row

2) Continue this process with the second row to remove the second coefficient from the third row to the last row

3) Continue until a lower triangular matrix is zero

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right]$$

$$\downarrow$$

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ & a'_{22} & a'_{23} & b'_2 \\ & & a''_{33} & b''_3 \end{array}\right]$$

[20 MINS]

ACTIVITY: FORWARD ELIMINATION

NAIVE_GAUSS.M

Process:

1. Start with 1st pivot and zero elements under it
   i. Determine the normalization factor
   ii. Subtract the pivot row*normalization factor from row *r*
2. Move to next pivot and repeat for all pivots

$$\begin{array}{cc} & \xleftrightarrow{\quad n+1 \quad} \\ Aug = & \left[\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & b_3 \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & b_n \end{array}\right] \updownarrow n \end{array}$$

```
for c =
    for r =



    end
end
```

[20 MINS]

ACTIVITY: FORWARD ELIMINATION

NAIVE_GAUSS.M

```
function x = naive_gauss(A, b)
% forward elimination algorithm

for c = 1:n-1
    for r = c+1:n
        factor = Aug(r,c)/Aug(c,c);
        Aug(r,:) = Aug(r,:) - factor*Aug(c,:);


    end
end
```

$$\begin{array}{c} \xleftrightarrow{\quad n+1 \quad} \end{array}$$

$$Aug = \left[\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b'_1 \\ 0 & a'_{22} & a'_{23} & \cdots & a'_{2n} & b'_2 \\ 0 & 0 & a'_{33} & \cdots & a'_{3n} & b'_3 \\ 0 & 0 & a'_{43} & & a'_{4n} & b'_4 \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ 0 & 0 & a'_{n3} & \cdots & a'_{nn} & b'_n \end{array}\right] \updownarrow n$$

iteration c=3,r=4

Col 1: $0 - \text{factor} \times 0$

Col 2: $0 - \text{factor} \times 0$

Col 3: $a'_{43} - \text{factor} \times a'_{33}$

$\cdots$

RECAP: BACK SUBSTITUTION

MONASH University

1) Starting with the last row, solve for the unknown, then substitute that value into the row above it

2) Because of the upper-triangular matrix format, each row will contain only one unknown

3) Continue until all unknowns are determined

$$\left[\begin{array}{ccc:c} a_{11} & a_{12} & a_{13} & b_1 \\ & a'_{22} & a'_{23} & b'_2 \\ & & a''_{33} & b''_3 \end{array}\right]$$

$$\downarrow$$

$$x_3 = b''_3/a''_{33}$$

$$x_2 = (b'_2 - a'_{23}x_3)/a'_{22}$$

$$x_1 = (b_1 - a_{13}x_3 - a_{12}x_2)/a_{11}$$

[20 MINS]

ACTIVITY: BACKWARD SUBSTITUTION

NAIVE_GAUSS.M

$$Aug = \overbrace{\left[\begin{array}{cccccc|c} a_{11} & a_{12} & a_{13} & a_{14} & \cdots & a_{1n} & b'_1 \\ 0 & a'_{22} & a'_{23} & a'_{24} & \cdots & a'_{2n} & b'_2 \\ 0 & 0 & a'_{33} & a'_{34} & \cdots & a'_{3n} & b'_3 \\ 0 & 0 & 0 & a'_{44} & \cdots & \cdot & b'_4 \\ \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdots & a'_{nn} & b'_n \end{array}\right]}^{n+1} \left.\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ x_n \end{bmatrix}\right\} n$$

Process:

1. Solve for unknown in last row
2. Repeat for all rows above

Last row: $a'_{nn} x_n = b'_n$

$$x_n = b'_n / a'_{nn}$$

2^{nd} row: $a'_{22} x_2 + a'_{23} x_3 + \cdots + a'_{2n} x_n = b'_2$

$$x_2 = \frac{b'_2 - a'_{23} x_3 - \cdots - a'_{2n} x_n}{a'_{22}}$$

r^{th} row: $a'_{rr} x_r + a'_{r,r+1} x_{r+1} + \cdots + a'_{rn} x_n = b'_r$

$$x_r = \frac{b'_r - a'_{r,r+1} x_{r+1} - \cdots - a'_{rn} x_n}{a'_{rr}}$$

[20 MINS]

ACTIVITY: BACKWARD SUBSTITUTION

NAIVE_GAUSS.M

n+1

$$
Aug = \left[\begin{array}{cccccc|c}
a_{11} & a_{12} & a_{13} & a_{14} & \cdots & a_{1n} & b'_1 \\
0 & a'_{22} & a'_{23} & a'_{24} & \cdots & a'_{2n} & b'_2 \\
0 & 0 & a'_{33} & a'_{34} & \cdots & a'_{3n} & b'_3 \\
0 & 0 & 0 & a'_{44} & \cdots & \cdot & b'_4 \\
\cdot & \cdot & \cdot & \cdot & & \cdot & \cdot \\
0 & 0 & 0 & 0 & \cdots & a'_{nn} & b'_n
\end{array}\right]
\quad
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ x_n \end{bmatrix}
$$

n

```
function x = naive_gauss(A, b)
% backward substitution algorithm



x(n) = Aug(n,n+1)/Aug(n,n);

for
    x(r) = (Aug(r,n+1) - Aug(r,r+1:n)*x(r+1:n))/Aug(r,r);
end
```

[20 MINS]

ACTIVITY: MATRIX SOLVE

NAIVE_GAUSS.M, MATRIX_SOLVE.M

MATLAB commands:

```
inv(…)
X = …\…
```

- Determine the solutions for the following matrix equations using:
  a) Matrix inversion
  b) MATLAB left division
  c) Naïve Gaussian elimination

Equation set A:

$$\begin{bmatrix} 1 & 4 & 5 \\ 5 & 6 & 7 \\ 8 & 9 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 6 \end{bmatrix}$$

Equation set B:

$$\begin{bmatrix} 1 & 4 & 5 \\ 0 & 0 & 7 \\ 8 & 9 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 6 \end{bmatrix}$$

RECAP: PARTIAL PIVOTING

- Consider the following system of equations which has a zero pivot
  - Not possible to make entries below the pivot equal 0 (forward elimination)

$$2x_1 + 2x_2 + x_4 = 10$$
$$x_3 + 2x_4 = 7$$
$$2x_2 + x_3 + 0.5x_4 = 9$$
$$-x_2 + 3x_3 + 1.5x_4 = 13$$

$$\text{Aug} = \left[\begin{array}{cccc|c} 2 & 2 & 0 & 1 & 10 \\ 0 & 0 & 1 & 2 & 7 \\ 0 & 2 & 1 & 0.5 & 9 \\ 0 & -1 & 3 & 1.5 & 13 \end{array}\right]$$

- Swap rows 2 and 3 to avoid zero pivot
  - Now it is possible to make entries below the pivot equal 0

$$2x_1 + 2x_2 + x_4 = 10$$
$$2x_2 + x_3 + 0.5x_4 = 9$$
$$x_3 + 2x_4 = 7$$
$$-x_2 + 3x_3 + 1.5x_4 = 13$$

$$\text{Aug} = \left[\begin{array}{cccc|c} 2 & 2 & 0 & 1 & 10 \\ 0 & 2 & 1 & 0.5 & 9 \\ 0 & 0 & 1 & 2 & 7 \\ 0 & -1 & 3 & 1.5 & 13 \end{array}\right]$$

SWAPPING ROWS IN MATLAB

- Augmented matrix is defined as Aug = [A b]

- Swapping 2nd and 3rd rows of Aug
  - Fill the 2nd and 3rd rows of Aug with the 3rd and 2nd rows of Aug, respectively
  - Aug([2 3], :) = Aug([3 2], :)

$$\text{Aug} = \left[\begin{array}{cccc|c} 2 & 2 & 0 & 1 & 10 \\ 0 & 0 & 1 & 2 & 7 \\ 0 & 2 & 1 & 0.5 & 9 \\ 0 & -1 & 3 & 1.5 & 13 \end{array}\right] \longrightarrow \text{Aug} = \left[\begin{array}{cccc|c} 2 & 2 & 0 & 1 & 10 \\ 0 & 2 & 1 & 0.5 & 9 \\ 0 & 0 & 1 & 2 & 7 \\ 0 & -1 & 3 & 1.5 & 13 \end{array}\right]$$

[20 MINS]

ACTIVITY: PARTIAL PIVOTING

GAUSS.M

MATLAB commands:

```
Aug([x y],:) = Aug([y x],:)
[M,I] = max(…)
```

function x = gauss(A, b)

Algorithm for partial pivoting

```
for c = 1:n-1



    for r = c+1:n
        <forward elimination code>
    end
end
```

$$Aug = \begin{array}{c} \text{n+1} \\ \left[\begin{array}{cccc|c} 2 & 2 & 0 & 1 & 10 \\ 0 & 0 & 1 & 2 & 7 \\ 0 & 2 & 1 & 0.5 & 9 \\ 0 & -1 & 3 & 1.5 & 13 \end{array}\right] \end{array} \text{n}$$

[20 MINS]

ACTIVITY: MATRIX SOLVE II

NAIVE_GAUSS.M, GAUSS.M, MATRIX_SOLVE2.M

- Determine the solutions for the following matrix equations using:
  a) Matrix inversion
  b) MATLAB left division
  c) Naïve Gaussian elimination
  d) Gaussian elimination with partial pivoting

Equation set A:

$$\begin{bmatrix} 2 & 2 & 0 & 1 \\ 0 & 0 & 1 & 2 \\ 0 & 2 & 1 & 0.5 \\ 0 & -1 & 3 & 1.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 10 \\ 7 \\ 9 \\ 13 \end{bmatrix}$$

Equation set B:

$$\begin{bmatrix} 1 & 4 & 5 \\ 0 & 0 & 7 \\ 8 & 9 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 6 \end{bmatrix}$$

RECAP: GAUSS-JORDAN ELIMINATION

- It consists of 2 steps
  a) Forward elimination
  b) Backward elimination

- Forward elimination identical to Gaussian elimination
  – Produces lower diagonal to be full of zeros

- Backward elimination
  – Starting with the last row, add or subtract multiples of that row to eliminate the last coefficient from all rows above
  – Continue process with other columns
  – Produces upper diagonal to be full of zeros

$$\left[\begin{array}{ccc:c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right]$$

$$\downarrow$$

$$\left[\begin{array}{ccc:c} a_{11} & a_{12} & a_{13} & b_1 \\ & a'_{22} & a'_{23} & b'_2 \\ & & a''_{33} & b''_3 \end{array}\right]$$

$$\downarrow$$

$$\left[\begin{array}{ccc:c} a'_{11} & & & b'_1 \\ & a'_{22} & & b'_2 \\ & & a''_{33} & b''_3 \end{array}\right]$$

RECAP: GAUSS-JORDAN ELIMINATION

- Direct solution
  - The matrix has been reduced to a diagonal form, and the solution can be found directly
  - diag() will extract the diagonal elements of a matrix as a column vector

$$Aug = \left[\begin{array}{ccc:c} a'_{11} & & & b'_1 \\ & a'_{22} & & b'_2 \\ & & a''_{33} & b''_3 \end{array}\right]$$

$$x_3 = b''_3 / a''_{33}$$

$$x_2 = b''_2 / a''_{22}$$

$$x_1 = b''_1 / a''_{11}$$

$$x = Aug(:, n+1) \ ./\mathrm{diag}(Aug)$$

[20 MINS]

ACTIVITY: GAUSS-JORDAN

GAUSS_JORDAN.M

iteration c=3,r=1

function x = gauss_jordan(A, b)

Algorithm for backward elimination

```
for
    for
        factor = Aug(r,c)/Aug(c,c);
        Aug(r,:) = Aug(r,:) - factor*Aug(c,:);



    end
end
x = Aug(:,n+1)./diag(Aug);
```

$$Aug = \overbrace{\left[\begin{array}{cccccc|c} a_{11} & a_{12} & a_{13} & a_{14} & \cdots & a_{1n} & b'_1 \\ 0 & a'_{22} & a'_{23} & a'_{24} & \cdots & a'_{2n} & b'_2 \\ 0 & 0 & a'_{33} & a'_{34} & \cdots & a'_{3n} & b'_3 \\ 0 & 0 & 0 & a'_{44} & \cdots & . & b'_4 \\ . & . & . & . & & . & . \\ 0 & 0 & 0 & 0 & \cdots & a'_{nn} & b'_n \end{array}\right]}^{n+1} \Bigg\} n$$

Col 1: $a'_{11} - \text{factor} \times 0$

Col 2: $a'_{12} - \text{factor} \times 0$

Col 3: $a'_{13} - \text{factor} \times a'_{33}$

Col 4: $0 - \text{factor} \times 0$

Col 5: $0 - \text{factor} \times 0$

Col $(n+1)$: $\mathrm{b}'_1 - \text{factor} \times b'_3$

[20 MINS]

ACTIVITY: MATRIX SOLVE III

NAIVE_GAUSS.M, GAUSS.M, GAUSS_JORDAN.M, MATRIX_SOLVE3.M

- Determine the solutions for the following matrix equations using:
  a) Matrix inversion
  b) MATLAB left division
  c) Naïve Gaussian elimination
  d) Gaussian elimination with partial pivoting
  e) Gauss-Jordan elimination

Equation set A:

$$\begin{bmatrix} 2 & 2 & 0 & 1 \\ 0 & 0 & 1 & 2 \\ 0 & 2 & 1 & 0.5 \\ 0 & -1 & 3 & 1.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 10 \\ 7 \\ 9 \\ 13 \end{bmatrix}$$

Equation set B:

$$\begin{bmatrix} 1 & 4 & 5 \\ 0 & 0 & 7 \\ 8 & 9 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 6 \end{bmatrix}$$

IN THIS WORKSHOP

1. Understanding methods to solve a system of linear equations
   a. Gaussian elimination
   b. Naïve Gaussian elimination
   c. Gauss-Jordan elimination
2. Creating function files to solve a system of linear equations
3. Solving a system of linear equations
4. Using inbuilt left division and matrix inversion

PART B: NUMERICAL METHODS

7. ~~Roots and optimisation~~
8. ~~Curve fitting~~
9. ~~Numerical integration~~
10. ~~Ordinary differential equations~~
11. ~~Linear systems~~
12. Exam information

You can now complete lab 11 (non assessed)!