INTRODUCTION TO MATLAB® WORKSHOP FOR CHEMICAL ENGINEERING STUDENTS

- Instructor: Sina Ghanbari
- Kimia Scientific Group - Chemical & Petroleum Department
- Sharif University of Technology

- March 2024

INTRODUCTION

- MATLAB is a matrix-based tool for numerical computations. It's very powerful and easy to use.
- Both programming language and interactive environment!
- Lots of available toolboxes

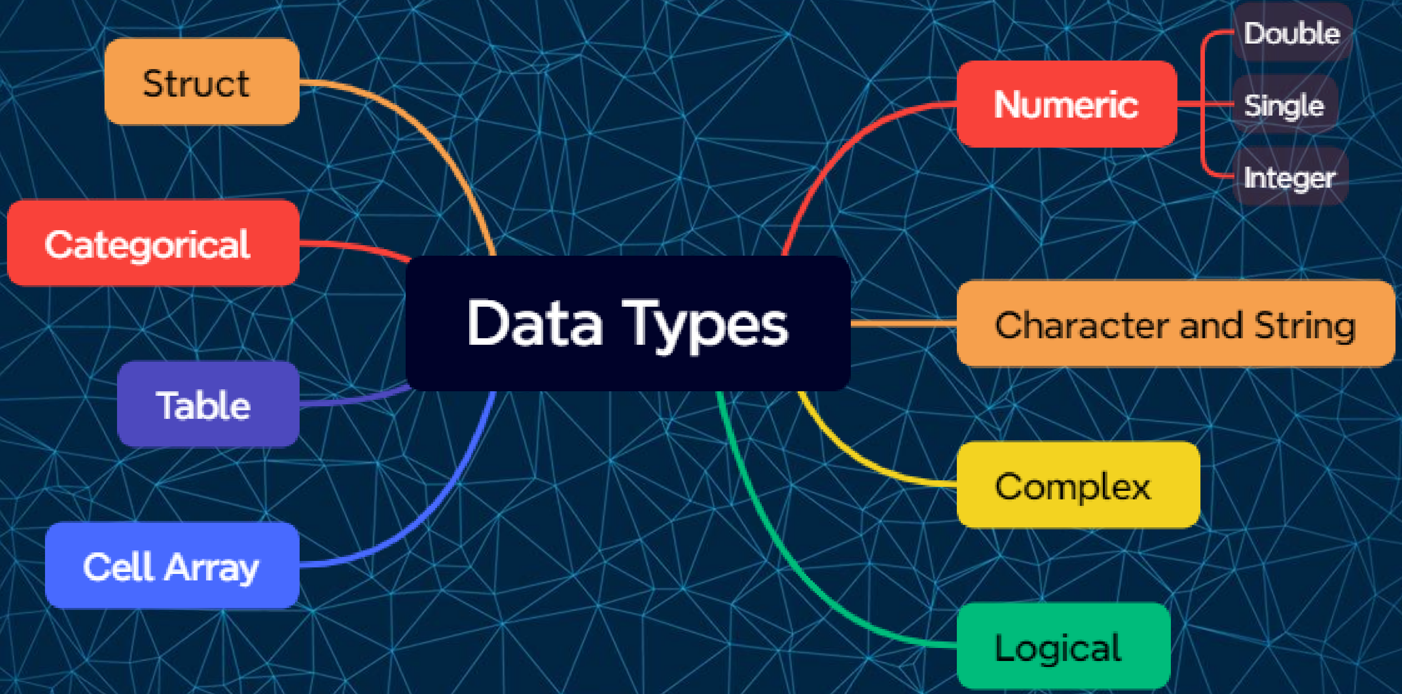
ADVANTAGES AND DISADVANTAGES

Advantages

- User-Friendly
- Rich Functionality
- Powerful Visualization
- Community and Resources

Disadvantages

- Cost (Excludes Toolboxes)
- Resource Intensive
- Lack of Speed
- Not Ideal for Large-Scale Projects


Data Types
Struct
Categorical
Table
Cell Array
Numeric
Double
Single
Integer
Character and String
Complex
Logical

BASIC COMMANDS

BASIC COMMANDS FOR WORKSPACE

who, whos	current workspace vars.
clear all	clear workspace vars.
close all	close all figures
clc	clear screen
clf	clear figure
class(x)	show the data type of ‘x’
disp(x)	Show ‘x’ to user
input(x)	**Input ‘x’ from user**

BASIC OPERATION COMMANDS

%	used to denote a comment
%%	used to divide your code into several sections
;	suppresses display of value (when placed at end of a statement)
...	continues the statement on next line
eps	machine epsilon
inf	machine infinity
realmin	Smallest positive floating-point number
Realmax	Largest positive floating-point number
naN	not-a number, e.g., 0/0.

MATHEMATICAL COMMANDS

- Mathematical functions: **sqrt(x)**, **exp(x)**, **cos(x)**, **sin(x)**, **log(x), log10(x), log2(x), asin(x),acos(x), sec(x), sinh(x), cosh(x),** etc.
- Operations: **+ - * / ^ '**
- Constants: **pi**, **exp(1)**,

LOGICAL CONDITIONS

- **==, <, >, <=, >=, ~=** (not equal), **~** (not)
- **&** (element-wise logical and), **|** (or)
- **find('condition')** – Return indices of A's elements that satisfies the condition.

NOTE FOR NAMING M-FILES

- ✓ M-file names must start with an alphabetic character, may contain any alphanumeric characters or underscores, and must be no longer than the maximum allowed M-file name length(63 character).
- ✓ Never use blank space in the file name.
- ✓ Use (_) instead of space or (-)

NUMERIC & MATRICES

VECTORS & MATRICES

- **v = [-4 8 0 2.5 -1.5];** % length 5 row vector.
- **v = v';** % transposes v.
- **v(1);** % first element of v.
- **v= 4:-1:2;** % same as v=[4 3 2];
- **a=1:3; b=2:3; c=[a b];** → c = [1 2 3 2 3];

- **x = linspace(-pi,pi,10);** % creates 10 linearly-spaced elements from –pi to pi.
- **logspace** is similar.
- **A = [1 2 3; 4 5 6];** % creates 2x3 matrix
- **A(1,2)** % the element in row 1, column 2.
- **A(:,2)** % the second column.
- **A(2,:)** % the second row.

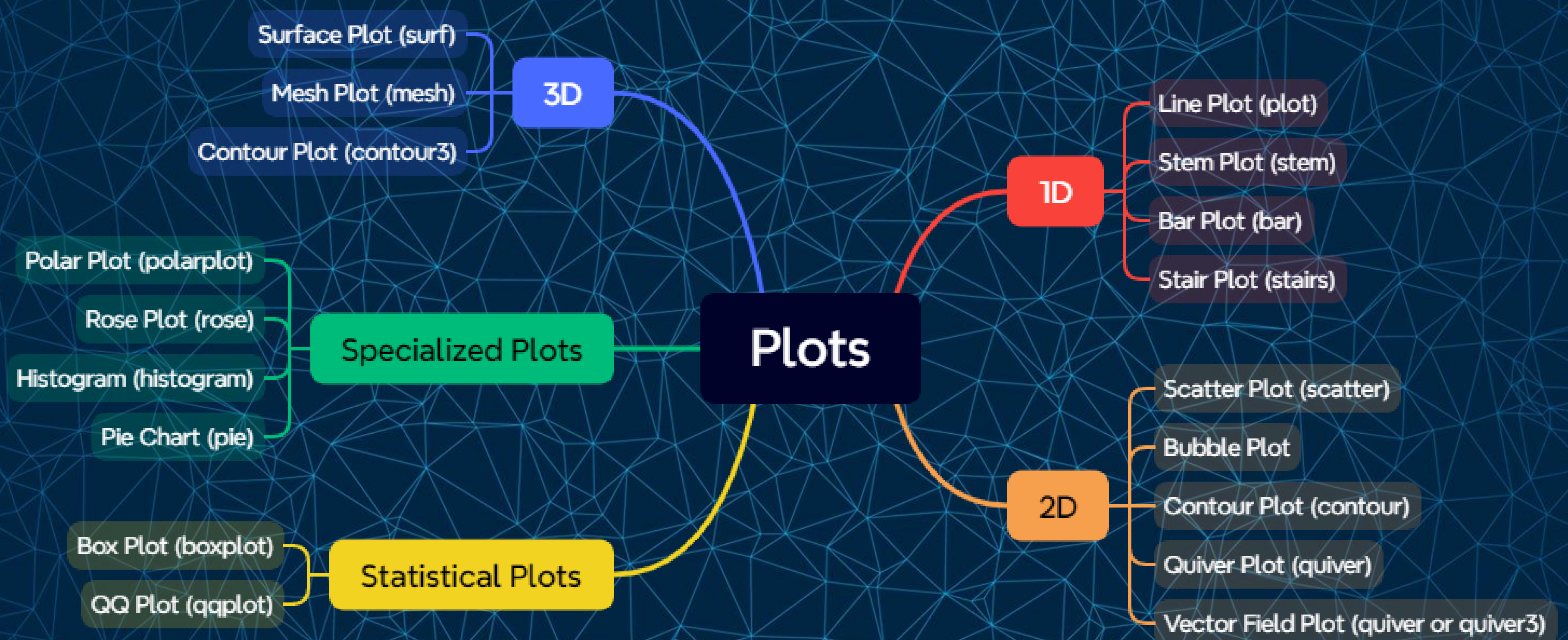
- **A+B, A-B, 2*A,** % matrix addition, matrix subtraction, scalar multiplication
- **A.*B** % element-by-element multiple
- **A./B** % element-by-element div.
- **A’** % transpose of A (complex-conjugate transpose)
- **dot(A,B)** % dot product of A & B
- **A*B** % cross product of A & B
- **det(A)** % determinant of A
- **inv(A)** % inverse matrix of A

- **diag(v)** % change a vector v to a diagonal matrix.
- **diag(A)** % get diagonal of A.
- **eye(n)** % identity matrix of size n.
- **zeros(m,n)** % m-by-n zero matrix.
- **ones(m,n)** % m*n matrix with all ones.
- **Randi([a, b], m,n)** % Create a m*n matrix with random variables from a to b

MORE MATRICES/VECTOR OPERATION

• **length(v)**	% determine length of vector.
• **size(A)**	% determine size of matrix.
• **rank(A)**	% determine rank of matrix.
• **find(A)**	% determine indices of non-zero elements
• **sum(A)**	% determine sum of elements
• **max(A)**	% determine maximum element
• **min(A)**	% determine minimum element
• **mean(A)**	% determine mean of elements
• **sort(A)**	% sort element from minimum to maximum value

PLOTTING

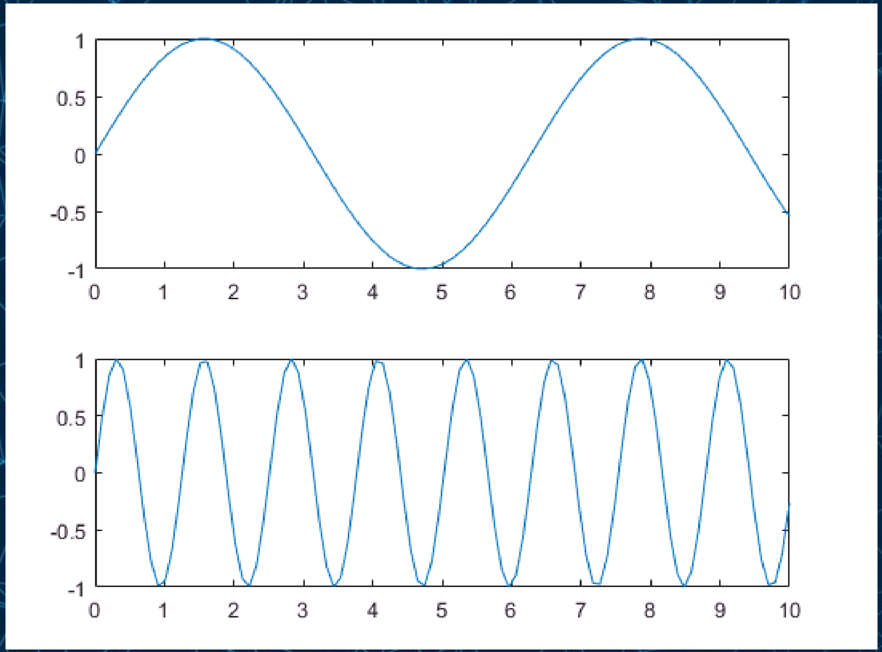

Plots
1D
Line Plot (plot)
Stem Plot (stem)
Bar Plot (bar)
Stair Plot (stairs)
2D
Scatter Plot (scatter)
Bubble Plot
Contour Plot (contour)
Quiver Plot (quiver)
Vector Field Plot (quiver or quiver3)
3D
Surface Plot (surf)
Mesh Plot (mesh)
Contour Plot (contour3)
Specialized Plots
Polar Plot (polarplot)
Rose Plot (rose)
Histogram (histogram)
Pie Chart (pie)
Statistical Plots
Box Plot (boxplot)
QQ Plot (qqplot)

PLOTTING COMMON SYNTAX

• figure	% new figure window
• grid on	% Turn on gridlines
• xlabel(‘’)	% add label to axis x
• ylabel(‘’)	% add label to axis y
• xlim([a,b])	% set limits to axis x
• ylim([c,d])	% set limits to axis y
• title(‘name’.’Fontsize’,22)	% title of figure
• hold on	% retains current figure when adding new stuff
• hold off	% restores to default (no hold on)
• loglog(x,y)	% plot y & x on log scale
• text(x,y,’text’)	% place text at position x,y

MERGE MULTIPLE PLOTS INTO ONE FIGURE

```
subplot(2,1,1);
 x = linspace(0,10);
 y1 = sin(x);
plot(x,y1)
subplot(2,1,2);
 y2 = sin(5*x);
 plot(x,y2)
```



EXTRA

Character color	Character symbol	character line style
b blue	. point	- solid
g green	o circle	: dotted
r red	x x-mark	-. dashdot
c cyan	+ plus	-- dashed
m magenta	* star	
y yellow	s square	
k black	d diamond	
	v triangle (down)	
	^ triangle (up)	
	< triangle (left)	
	> triangle (right)	
	p pentagram	
	h hexagram	

CONTROL FLOW

LOOPS - FOR

We use **"for"** to repeat a particular command number of times!

A = matrix that defined by user

```
for i = A
    Statement
end
```

LOOPS - WHILE

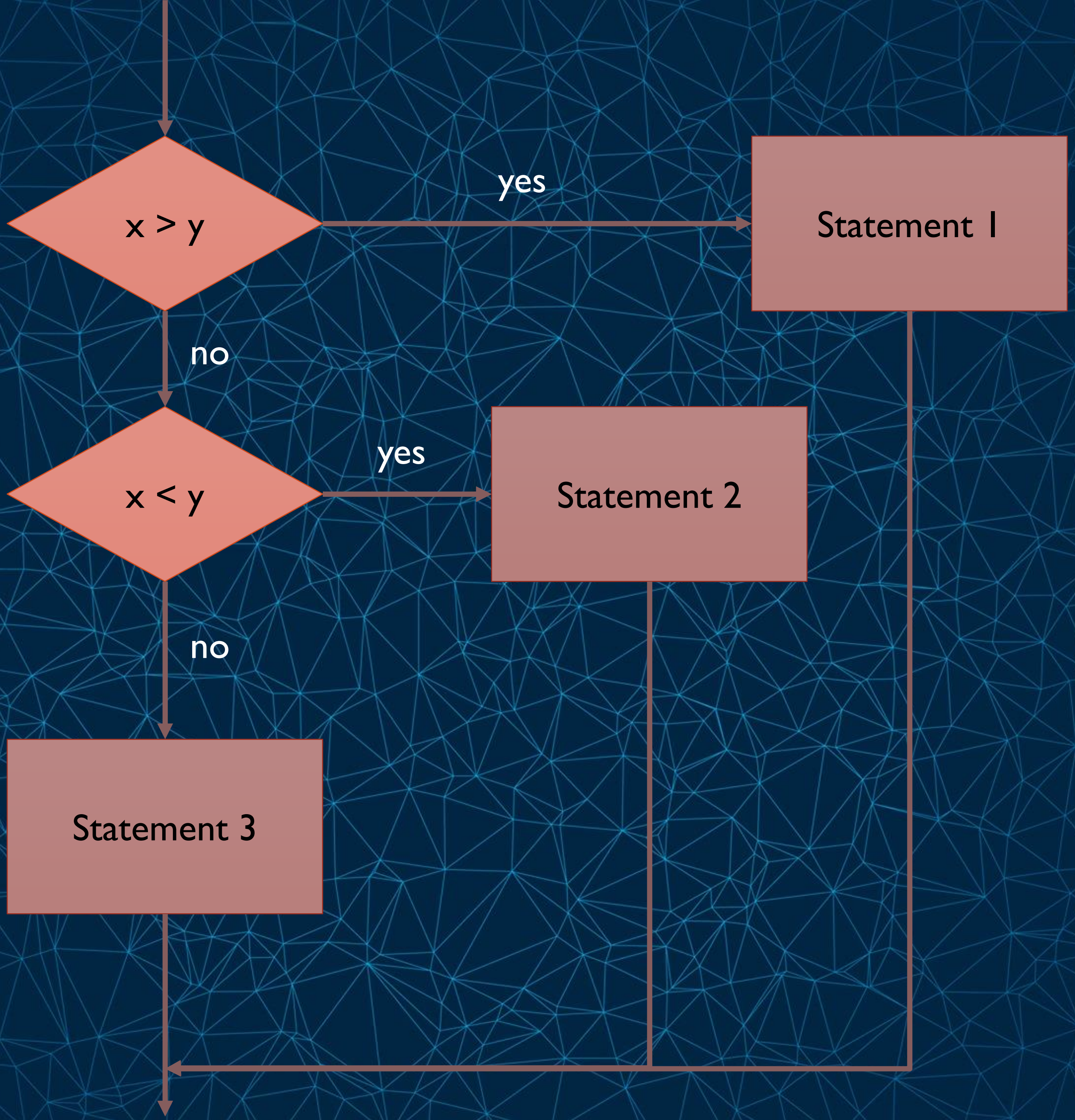
We use **“while”** to repeat a particular command number of times, even infinite!

A = Number that defined by user

while I < A

    Statement

end


x > y
yes
Statement 1
no
x < y
yes
Statement 2
no
Statement 3

CONDITIONAL STATEMENTS
IF/ELSE/ELSEIF

A = number that defined by user

B = another number that defined by user

If i = A

    Statement 1

elseif i = B

    Statement 2

else

    Statement 3

end

CONDITIONAL STATEMENTS SWITCH/CASE/OTHERWISE

```
switch i
    case A
        statement 1
    case B
        statement 2
    otherwise
        statement 3
end
```

BREAK/CONTINUE

- **break** – terminates execution of for and while loops. For nested loops, it exits the innermost loop only.
- **continue -** passes control to the next iteration of a for or while loop

MathWorks

EXAMPLE1: BMI CALCULATOR

Write a program that gets height and weight from user, then calculates his BMI and shows his fat region.

<18.5	18.5< < 24.9	25< < 29.9	30< < 34.9	35< < 39.9	40<
Under Weight	Normal Weight	Overweight	Obese (Class1)	Obese (Class2)	Obese (Class3)

FUNCTIONS

FUNCTION FILE

```
function [output] = function_name(input variables)
statement
end
```

SYMBOLIC FUNCTION

syms x,y

f = function equation

subs(f,{x,y},{value of x, value of y})

EXAMPLE2 : ANTOINE EQUATION

Chemical Engineers uses Antoine equation to calculate saturated pressure using given temperature and vice versa. Antoine equation is written as follows:

$$\log(P) = A - \frac{B}{T + C} \qquad (Pressure\ form)$$

$$T = \frac{B}{A - \log(P)} - C \qquad (Temperature\ form)$$

- Write a function that gets saturated pressure or temperature from user and returns saturated pressure or temperature. Then, Calculate the saturate pressure and temperature for a certain substance in $80\ ℃$ and 500 mmHg.

A	B	C
8.043	1582.27	239.726

COMMON NUMERICAL AND ANALYTICAL CALCULATIONS WITH MATLAB

POLYNOMIALS

- How to define a polynomial in MATLAB?

$f(x) = ax^n + bx^{n-1} + cx^{n-2} + \cdots + d$ => coeff = [a, b, c, …, d]

- polyval(coeff,m) assign "m" instead of "x"
- roots(coeff) Solve f(x) = 0
- polyfit(x,y,n) Fit a "n^{th}" order polynomial

EXAMPLE 3

The following table shows measurements of reaction temperature versus time. Determine the 1st – order, 2nd – order, 4th – order, and 8th-order polynomials to represent this data and reaction temperature when t = 4.26 hr.

t (hour)	1	2	3	4	5	6	7	8
T (℃)	50.8	54.4	55.1	57.6	61.2	59.5	54.6	53.5

EXAMPLE 4

The van der Waals equation can be rearranged as:

$$Pv^3 - (bP + RT)v^2 + av - ab = 0$$

In this equation, v is represented as specific volume, R = 0.082054 lit.atm/(mol.K) , a = 3.592 & b = 0.04267 (for CO_2).

- Find the specific volume of CO_2 when P = 12 atm ,T = 315.6 K

DERIVATIVE AND INTEGRATION

• diff(f)	Calculate the analytical derivative of *f*
• Int(f)	Calculate the analytical integration of *f*
• trapz(x,y)	Calculate the numerical integration (trapezoidal method)

EXAMPLE 5

In a chemical process, the concentration of a reactant A, C_A, as a function of time t is given by the function:

$$C_A(t) = \frac{2t^3 - 5t^2 + 3t}{t^2 + 2t + 1}$$

- Calculate the rate of change of C_A with respect to time t. This represents the rate of consumption or production of reactant A. Calculate this rate at t=4.5 min.
- Determine the total amount of reactant A consumed or produced over a given time interval by computing the definite integral of $C_A(t)$ over that interval. Calculate this value from t=0 min to t= 5 min.

EXAMPLE 6

Total mass of a bar with inhomogeneous density is obtained from this equation:

$$m = \int_0^L \rho(x)A(x)\,dx$$

that m is represented as mass of bar, ρ(x) as density, A(x) as Area , L as total length of bar. This table contains the data for a 10 m bar. Calculate the total mass of bar (kg).

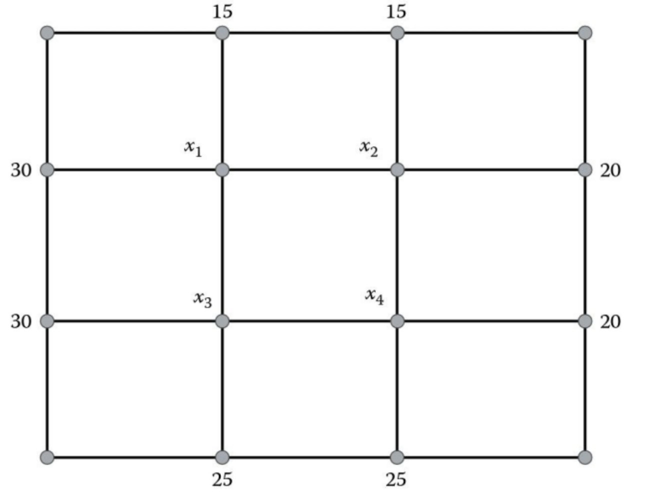
x, m	0	2	3	4	6	8	10
ρ , gr/cm^3	4.00	3.95	3.89	3.8	3.6	3.41	3.3
A, cm^2	100	103	106	110	120	133	150

EXAMPLE 7: SYSTEM OF LINEAR EQUATIONS

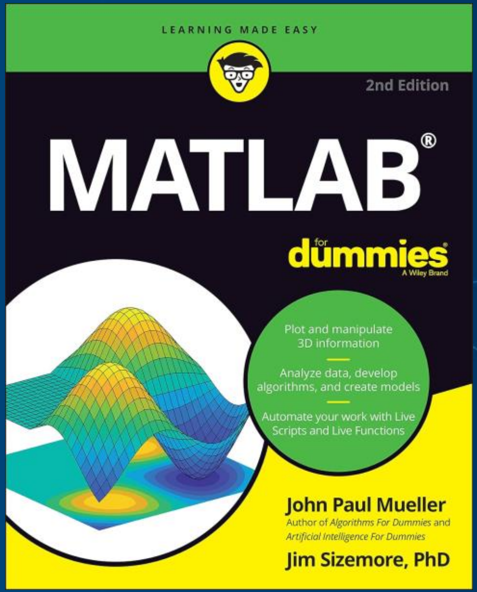
The following figure shows a flat square plate with its side held at constant temperature. Find the temperature at each node x1, x2,x3,x4. Each dot represent a node, and the temperature at each node is assumed to be given by the average temperature of adjacent nodes.

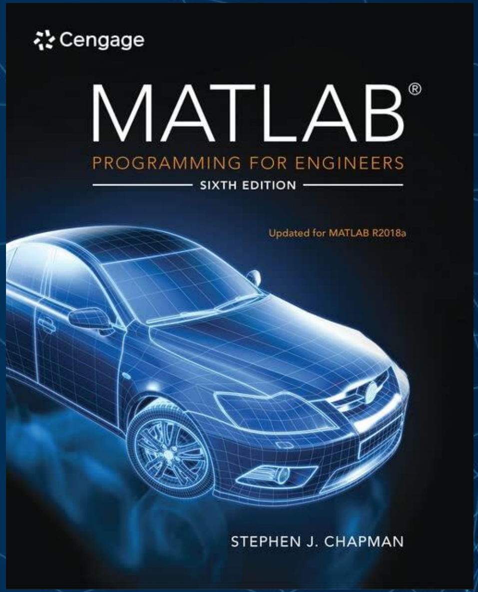
- Illustrate the temperature profile of plate as heat map and 3D plot. (Assume the temperature of corner nodes is equal to 25℃)

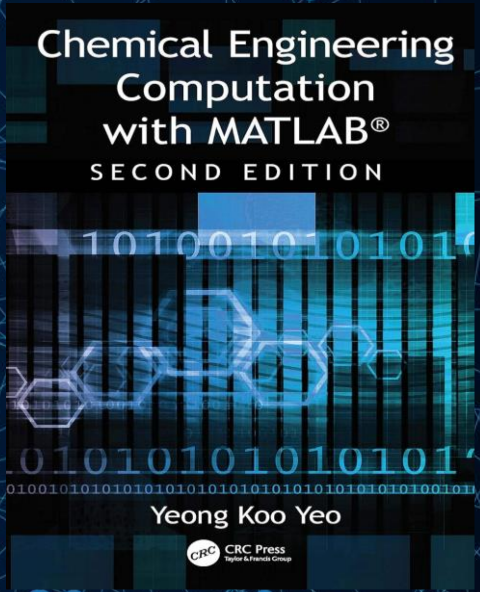
$$\left.\begin{aligned} x_1 &= \frac{1}{4}(30 + 15 + x_2 + x_3) \\ x_2 &= \frac{1}{4}(x_1 + 15 + 20 + x_4) \\ x_3 &= \frac{1}{4}(30 + x_1 + x_4 + 25) \\ x_4 &= \frac{1}{4}(x_3 + x_2 + 20 + 25) \end{aligned}\right\} \Rightarrow \left\{\begin{aligned} 4x_1 - x_2 - x_3 &= 45 \\ -x_1 + 4x_2 - x_4 &= 35 \\ -x_1 + 4x_3 - x_4 &= 55 \\ -x_2 - x_3 + 4x_4 &= 45 \end{aligned}\right.$$



USEFUL RESOURCES







END OF PRESENTATION!

Thanks for your attention. 😳