I: Basic Matlab functions, Matrix and simultaneous equations

1: Elementary Functions

- Use ^ for power
- Use sqrt() for root
- exp() for exponential
- asin for sin^-1
- log for ln
- log10 and log2 for log_{10} and log_2
- nthroot(x,n) for the nth real root of x
- abs for absolute value
- complex(a,b) for construct the complex number a+bi
- Real() and imag() for the real part or the imaginary part
- pi for π

2: The operations of the matrix and simultaneous equations

- Use the 'space' or ',' to type the elements in a line, and use ';' to change the line
- To create a 3 $\times$ 3 identity matrix:

```
x=[1 0 0;0 1 0;0 0 1];
```

- Use the ' to find the transpose of the matrix
- Use '+,-,*' to calculate
- If $AX = B$, X=$A^{-1} * B$=inv(A)*B=A\B
- Use A(a,b) to extract the element in row(a),col(b)
- Use A(1,:) to extract all the first line of A, while use A(:,1) to extract all the first column

II: Plot data and functions,read data from excel data sheet and do integration

1: Create m-file and plot data

1.1: Create m-file

- Use the matlab and click the 'New'
- Use editor like vscode to create '.m' files

1.2: Plot Data of dots and functions

- Plot data from several dots:

```
x=[1,2,3,4,5,6,7,8];%define x
y=[2,9,28,65,126,219,343,513];%define y
plot(x,y,'b--') %plot('x','y','colour'+'symbol')
```

- Colour specification:
  - y for yellow;
  - r for red;
  - g for green;
  - b for blue;
  - w for white;
  - k for black;
- Typical symbols:
  - 'o' for circle;
  - 'x' for x;
  - '-' for doted line;
  - 's' for square;
  - 'd' for diamond;
- Define a function and plot a function:
  - Use the @ method:

    ```
    y=@(x)1+2*x+x.^3; % Add '.' when express power or fraction
    y([1,2,3]) % Output the value of y
    Z=@(x,y)x.^2+y.^2 % Two variable function
    Z([1,3],[2,4]) %Output the Z
    ```

  - Plot a function:

```
% Method 1;
x=[1:100]; % The continuous number between 1~100
y=1./x+2*sin(x);
plot(x,y,'r-');
% Method 2;
figure(1);% Create a figure windows, '1' just a number
y=@(x)1./x+2*sin(x); %Define a function
fplot(y,[1,100],'r') % The second part is the range of x-axis
```

- The way to plot two functions in one image:

```
plot(x,y,'b-');
hold on;
plot(v,t,'d-')
```

- Add axis labels,legend and title:

```
xlabel('Times(s)');
ylabel('Distance(m)');
legend('Measured data','Fitted'); % Follow the order of the plot
title('Analysis of Distance vs Time') % Add these things after all the plot
```

2: Do the integration using dots and functions

```
% The integration of several dots
x=[1,2,3];
y=[2,4,6];
plot(x,y,'r*');
Z=trapz(y)
% The integration of a function
syms x; % define the variable x
yy=x.^2-x+4;% Define a function
z2=int(yy,x,1,8);% $\int_1^8(x^2-x+4)dx$
double(z2) %Find the answer rounded to two points
```

- Read data from excel sheet

```
% Firstly, add the target file to the matlab folder or path
Data=xlsread('filename.xlsx',1,'A1:C10'); % '1' for the sheet 1, 'A1:C10' for the range
x=Data(:,1); % Extract the first column
y=Data(:,2); % Extract the second column
plot(x,y,'r-')
```

III: Fit the data with a polynomial function and define the polynomial function using polyval

1: Fit the data with a polynomial function

```
x=[1,2,3];
y=[2,4,6];
P1=polyfit(x,y,2);% '2' refer to 2nd order function
y_fitted=p(1)*x.^2+p(2)*t+p(3);% The power decrease but the coefficient increase
% The other way is to use @ method
y_fitted=@(x)polyval(p,x);
figure(2); %open a image windows
plot(x,y,'r*');
hold;
fplot(y_fitted,[1,10],'b-')
```

2: Solve polynomial equations

- The polynomial equations are like this: $2x^3 + x^2 + 5 = 0$

```
p=roots([2,1,1,5]) %the snip of the lacked coefficient
p=p(image(p)==0) %find the real roots
```

3: Roots of arbitrary functions

```
f1=@(x)x.^2+4*sin(x)-16; %define the function
fplot(f1,[-6,6]);
grid on
% plot the function and find the approx roots
x1=fzero(f1,-4);
x2=fzero(f1,4);
```

IV: Differential, partial differential and solving differential equations

1: Differential and partial differential

- Differential:

```
syms x;
y=x.^2+x; %use the function
p=diff(y)
```

- Partial Differential:

```
syms x y;
z=x.^2*y.^2+x*y;
diff(z,x);
diff(z,y);
```

2: Use dsolve to solve differential equations

- First order:

```
syms x;
d=dsolve('Dy=sin(x)+3*x',x);
% Get a G.S
d1=dsolve('Dy=sin(x)+3*x','y(0)=1',x);
% Get the P.S
```

- Second order:

```
syms x;
s=desolve('D2y=2*x-y','y(0)=0','Dy(0)=0',x) %Put the limits in the middle
```

3: Use ode23 to solve ODEs

- y is a single variable of x:

```
f=@(x,y)sin(x)+y;
ode23(f,[0,2],1);
% [0,2] for the range of the solutions and '1' for y(0)=1
% This command could plot the solutions
[x,y]=ode23(f,[0,2],1);
% Print the solutions
```

- y is the vector variable of x:

  such as $\frac{d}{dx}\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} x \\ x+2 \\ x-4 \end{pmatrix}$

```
f=@(x,y)[x;x+2;x-4];
ode23(f,[0,5],[1;0;-1])
% [0,5] for the range and [1;0;-1] for the initial numbers
[x,y]=ode23(f,[0,5],[1;0;-1])
```

V: Input,output,for-loop and if-loop

1: Display and Input

```
disp('Please enter the input');
a=input('Please enter the number:')
```

2: For-loop

```
x=2;
for n=1:2 % No need to add ';'
% '1' for the first corner number and '2' for circulate times
x=sqrt(1+2*x);% The limit situations
end
```

3: If-loop

```
if(a1==0)&&(a2==0) % No need to add the ';'
disp('Nothing');
elseif(a1==0)&&(a2==1)
disp('Right');
else
disp('Not valid');
end
```