dsolve: dsolve is symbolic solver and it gives solution to ODE as formula.

Example: Given the initial value problem

$$\frac{dy}{dx} = 3x + \frac{y}{2}, y(0) = 1.$$

i. Obtain the exact solution using MATLAB's built-in function '*dsolve*'
ii. Plot the solution in the interval $[0, 1]$

Code:

```
y=dsolve('Dy=3*x+y/2','y(0)=1','x')
x=0:0.1:1;
z=eval(y);
plot(x,z,'*')
```

1. Given the initial value problem

$$\frac{dy}{dx} = sin\, x + y, y(0) = 0.$$

iii. Obtain the exact solution using MATLAB's built-in function '*dsolve*'
iv. Plot the solution in the interval $[0, 5]$

2. Given the initial value problem

$$\frac{dy}{dx} = \mathrm{y} \sin x, \, y(0) = 1.$$

i. Obtain the exact solution using MATLAB's built-in function 'dsolve'
ii. Plot the solution in the interval $[0, 1]$

3. Given the initial value problem

$$\frac{dy}{dx} = y, y(0) = 3.$$

i. Obtain the exact solution using MATLAB's built-in function '*dsolve*'
ii. Plot the solution in the interval $[0, 4]$.

Ode45: ode45 is a numerical solver and it gives solution as list of numbers (y, x).

Example: Given the initial value problem

$$\frac{dy}{dx} = 3x + \frac{y}{2}, y(0) = 1.$$

i. Obtain the exact solution using MATLAB's built-in function 'ode45'
ii. Plot the solution in the interval $[0, 1]$

Code:

```
y0=1;
xspan=[0,1];
f=@(x,y) 3*x+y/2;
[x,y]=ode45(f,xspan,y0);
plot(x,y,'r')
```

1. Given the initial value problem

$$\frac{dy}{dx} = 2y + x^2 , y(0) = 1.$$

   i. Obtain the exact solution using MATLAB's built-in function 'ode45'.
   ii. Plot the solution in the interval $[0, 5]$.

2. Given the initial value problem

$$\frac{dy}{dx} = 2x , y(1) = 4.$$

i. Obtain the exact solution using MATLAB's built-in function 'ode45'
ii. Plot the solution in the interval $[0, 4]$.

3. Given the initial value problem

$$\frac{dy}{dx} = x^2 + y^2 , y(1) = 3.$$

   iii. Obtain the exact solution using MATLAB's built-in function 'ode45'
   iv. Plot the solution in the interval $[0, 3]$.

polyfit: *polyfit(x,y,n)* finds the coeeficients of a polynomial $p(x)$ of degree n that fits the y data by minimizing the sum of the squares of the deviations of the data from the model.

Example*:* Use MATLAB built-in function 'polyfit' to determine a polynomial that fits the set of data points given in table 1

x	16	15	12	10	8
y	11	18	10	20	17

Table-1

Code:

```
x=[ 10   12   16   20   25   30];
y=[29   33   41   53   65   70];
p=polyfit(x,y,1)
x1=linspace(x(1),x(end));
y1=polyval(p,x1);
plot(x,y,'*',x1,y1)
legend('data points','linear curve')
```

1. Use MATLAB built-in function ‘polyfit’ to determine a polynomial that fits the set of data points given in table 1

x	16	15	12	10	8
y	11	18	10	20	17

Table-1

2. Use MATLAB built-in function to determine a polynomial that fits the following a set of data points given in table 1

x	2	4	6	8	10	12
y	58	46	39	26	16	12

Table-1

3. Use MATLAB built-in function ‘polyfit’ to determine a polynomial that fits the set of data points given in table 1

x	1	2	3	4	5
y	5.5	43.1	128	290.7	498.4

Table-1

2. Milne’s Thomson method

```
function[]=Milnes_EC(x0,y0,xn,h)
f=@(x,y) write here given Equation;
ERR=0.000001;
y(1)=y0;
x=x0:h:xn;
n=length(x);
for i=1:3
    k1=h*f(x(i),y(i));
    k2=h*f(x(i)+h/2,y(i)+k1/2);
    k3=h*f(x(i)+h/2,y(i)+k2/2);
    k4=h*f(x(i)+h,y(i)+k3);
```

```
    y(i+1)=y(i)+(1/6)*(k1+2*k2+2*k3+k4);
end
% predictor Formula
for i=4:n-1
    y(i+1)=y(i-3)+(4*h/3)*(2*f(x(i-2),y(i-2))...
    -f(x(i-1),y(i-1))+2*f(x(i),y(i)));
s(i+1,1)=y(i+1);
% correctors  Formula
for j=1:10
s(i+1,j+1)=y(i-1)+(h/3)*(f(x(i-1),y(i-1))...
    +4*f(x(i),y(i))+f(x(i+1),s(i+1,j)));
if abs(s(i+1,j+1)-s(i+1,j)) <ERR
    y(i+1)=s(i+1,j+1);
    break
end
end
end
fprintf('App. value by Milnes method is
y(%f)=%f\n',x(end),y(end))
plot(x,y,'*')
u=dsolve('Dy=Equation,'y(x0)=y0','x');
u=eval(u);
hold on
plot(x,u,'b--')
legend('Milnes Method Solution','Exact solution')
end
```

3. Adam's Method

```
function [x,y]=Adams_method(x0,y0,xn,n)
h=(xn-x0)/n;
f=@(x,y) Given Equation;
x=x0:h:xn;
y=zeros(1,n);
maxit=10;
s(:,:)=zeros(n,maxit);
y(1)=y0;
for i=1:3
    K1=h*f(x(i),y(i));
    K2=h*f(x(i)+0.5*h,y(i)+0.5*K1);
    K3=h*f(x(i)+0.5*h,y(i)+0.5*K2);
    K4=h*f(x(i)+h,y(i)+K3);
    y(i+1)=y(i)+(1/6)*(K1+2*K2+2*K3+K4);
end
%predictor formula

for i=4:n
    y(i+1)=y(i)+(h/24)*(55*f(x(i),y(i))-...
           59*f(x(i-1),y(i-1))+37*f(x(i-2),y(i-2))-...
           9*f(x(i-3),y(i-3)));
     s(i+1,1)=y(i+1);
%corrector formula
    for j=1:maxit
        s(i+1,j+1)=y(i)+(h/24)*(f(x(i-2),y(i-2))-...
        5*f(x(i-1),y(i-1))+19*f(x(i),y(i))...
        +9*f(x(i+1),s(i+1,j)));
```

```
        if (s(i+1,j+1)-s(i+1,j))<10^(-6)
            y(i+1)=s(i+1,j+1);
            break
        end
    end
end
plot(x,y,'*')
hold on
u=eval(dsolve('Dy=Given Equation','y(0)=0','x'));
plot(x,u,'r')
end
```

4. Modified Euler's method

```
function []=EulermodCSB(x1,y1,h)
f=@(x,y) Given equation;
xn=1;
x=x1:h:xn;
n=length(x);
y(1)=y1;
yp(1)=y1;
maxit=10;
for i=2:n
    yp(i)=y(i-1)+h*f(x(i-1),y(i-1));
    s(i,1)=yp(i);
    for j=2:maxit
        s(i,j)=y(i-1)+(h/2)*(f(x(i-1),y(i-1))+f(x(i),s(i,j-
1)));
        if abs(s(i,j)-s(i,j-1))<10^(-6)
            y(i)=s(i,j);
            break
        end
    end
end
fprintf('Appr. sol. y(%f)=%f',x(end),y(end));
plot(x,yp,'m--')
hold on
plot(x,y,'m--')
hold on
plot(x,y,'r--')
hold on
u=dsolve('Dy= Equation,'y(x0)=y0','x');
u=eval(u);
plot(x,u,'b--')
legend('Appr.   sol.   by   Euler','Appr.   sol.   by   Modified
Euler','Exact sol.');
end
```

5. One Dimensional Heat equation

```
function oneD_heat1(t0, tn,x0,xn,h,k,c )
t=t0:k:tn;
```

```
x=x0:h:xn;
m=length(x);
n=length(t);
a=c*k/h^2;
f=@(x) 3*x+1;  %  Change f according with the problem
u=zeros(m,n);
u(:,1)=f(x);
if a> 0.5
    fprintf('The method fails')
    return
end
for  j=1:n-1
    for i=2:m-1
        u(i, j+1)=a*u(i-1,j)+(1-2*a)*u(i,j)+a*u(i+1,j);
    end
end
for j=1:n
    plot(x,u(:,j))
    hold on
end
figure
surf(t,x,u)
xlabel('t')
ylabel('x')
zlabel('u')
end
```

For Example (Applicable for all the above programs)

For the given initial value problem

$$\frac{dy}{dx} = x^2 + y, (Given\ Equation)\ \ y(0) = 1(Initial\ condition)$$

Write the MATLAB function to solve numerically using Euler's modified method

Tutorial Problems: (Manual Calculation)

Euler Modified Method

1. **Manual Calculation**

Solve the following initial value problem using Euler's method

$$\frac{dy}{dx} = \log{(x + y)}\ ,\ \text{subjected to}\ \ y(1) = 2$$

perform three iterations to determine the value of y $x = 1.2$ and $x = 1.4$ with $h = 0.1$

2. **Manual Calculation**

Solve the following initial value problem using Euler's method

$$\frac{dy}{dx} = x + y\ \ ,\ \text{subjected to}\ \ y(0) = 1$$

perform three iterations to determine the value of y at $x = 0.5$ with

$h = 0.1.$

3 **Manual Calculation**

Solve the initial value problem using Euler's method

$$\frac{dy}{dx} = cos\, x + sin\, x\,, \qquad y(0) = 1$$

and determine the values of y at $x = 0.8$ by taking $h = 0.2$.

4 **Manual Calculation**

Solve the given initial value problem using Euler's method

$$\frac{dy}{dx} = -y\,, \text{ subjected to } y(0) = 1$$

determine the value of y at $x = 0.05$ by taking $h = 0.01$.

Runge-Kutta 4th order

1 Manual calculation

Solve the given initial value problem using R-K method

$$\frac{dy}{dx} = x + y^2, \qquad y(0) = 1$$

and determine the value of y at $x = 0.2$ by taking $h = 0.1$.

2 **Manual Calculation**

Solve the given initial value problem R- K method

$$\frac{dy}{dx} = 3x + \frac{y}{2}, \text{ subjected to } y(0) = 1$$

determine the value of y at $x = 0.2$ by taking $h = 0.1$.