1. Classical Fourth Order Runge Kutta Method

Script: RK4th.m

```
clear all;
disp('4th order Runge Kutta method');
func=input('Enter the function (dy/dx)=f(x,y)= ');
f=inline(func);
h=input('Enter the value of h: h=');
x0=input('Enter the intial value of x: x0=');
xn=input('Enter the final value of x: xn=');
y0=input('Enter the intial value of y: y0=');
x=x0:h:xn;
n=length(x)-1;
y=zeros(1,length(x));
y(1)=y0;
fprintf('n      x(n)      y(n)      k1          k2          k3          k4        y(n+1)\n');
for i=1:n
    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));
    fprintf('%d    %f    %f    %f    %f    %f    %f    %f\n',i,x(i),y(i),k1,k2,k3,k4,y(i+1));

end

for i=1:n+1
    fprintf('y(%.2f)=y%d=%f\n',x(i),i-1,y(i));
end
```

Command Window:

```
>> RK4th

4th order Runge Kutta method
Enter the function (dy/dx)=f(x,y)= 'x+y'
Enter the value of h: h=0.1
Enter the intial value of x: x0=0
Enter the final value of x: xn=0.3
Enter the intial value of y: y0=1
n    x(n)       y(n)       k1         k2         k3         k4         y(n+1)
1    0.000000   1.000000   0.100000   0.110000   0.110500   0.121050   1.110342
2    0.100000   1.110342   0.121034   0.132086   0.132638   0.144298   1.242805
3    0.200000   1.242805   0.144281   0.156495   0.157105   0.169991   1.399717
y(0.00)=y0=1.000000
y(0.10)=y1=1.110342
y(0.20)=y2=1.242805
y(0.30)=y3=1.399717
>>
```

2. Finite Difference Method

Script:FiniteDifference.m

```
clear all;
disp('Finite Difference method');
fprintf('Consider BVP y''+f(x)y`(x)+g(x)y(x)=r(x), a<=x<=b, y(a)=alpha, y(b)=beta\n');
f=inline(input('Enter value of f(x):f(x)='));
g=inline(input('Enter value of g(x):g(x)='));
r=inline(input('Enter value of r(x):r(x)='));
h=input('Enter the value of h: h=');
x0=input('Enter the intial value of x: x0=');
xn=input('Enter the final value of x: xn=');
y0=input('Enter the intial value of y: y0=');
yn=input('Enter the final value of y: yn=');
x=x0:h:xn;
n=length(x)-1;
y=zeros(1,length(x));
y(1)=y0;
```

```
y(length(x))=yn;
fprintf('The FD scheme is (2-fi h)y(i-1)+(2gi h^2-4)yi+(2+fi h)y(i+1)=2h^2 ri\n');
fprintf('This scheme consists %d system of linear equations with %d unknowns y1,y2,...\n',n-1,n-1);
fprintf('The system in matrix form is\n');
d=zeros(1,n-1);
d1=zeros(1,n-2);
d2=zeros(1,n-2);
for i=1:n-1
    d(i)=2*g(x(i))*h^2-4;
end
for i=1:n-2
    d1(i)=2+f(x(i))*h;
    d2(i)=2-f(x(i+1))*h;
end
A=diag(d)+diag(d1,1)+diag(d2,-1)
X=zeros(n-1,1);
B=zeros(n-1,1);
B(1)=(2*h^2*r(x(1)))-((2-f(x(1))*h)*y0);
B(n-1)=(2*h^2*r(x(1)))+((2+f(x(n-1))*h)*yn);
for i=2:n-2
    B(i)=2*h^2*r(x(i));
end
B
X=inv(A)*B
for i=2:n
    y(i)=X(i-1);
    fprintf('y(%.3f)=y%d=%.4f\n',x(i),i-1,y(i));
end
```

Command Window:

```
>> FiniteDifference
Finite Difference Method
Consider BVP y"+f(x)y`(x)+g(x)y(x)=r(x), a<=x<=b, y(a)=alpha, y(b)=beta
Enter value of f(x):f(x)='0'
Enter value of g(x):g(x)='1'
Enter value of r(x):r(x)='-1'
Enter the value of h: h=.125
Enter the intial value of x: x0=0
Enter the final value of x: xn=1
Enter the intial value of y: y0=0
Enter the final value of y: yn=0
The FD scheme is (2-fi h)y(i-1)+(2gi h^2-4)yi+(2+fi h)y(i+1)=2h^2 ri
This scheme consists 7 system of linear equations with 7 unknowns y1,y2,...
The system in matrix form is

A =

   -3.9688    2.0000         0         0         0         0         0
    2.0000   -3.9688    2.0000         0         0         0         0
         0    2.0000   -3.9688    2.0000         0         0         0
         0         0    2.0000   -3.9688    2.0000         0         0
         0         0         0    2.0000   -3.9688    2.0000         0
         0         0         0         0    2.0000   -3.9688    2.0000
         0         0         0         0         0    2.0000   -3.9688


B =

   -0.0313
   -0.0313
   -0.0313
   -0.0313
   -0.0313
   -0.0313
   -0.0313
```

```
X =

    0.0604
    0.1042
    0.1308
    0.1397
    0.1308
    0.1042
    0.0604

y(0.125)=y1=0.0604
y(0.250)=y2=0.1042
y(0.375)=y3=0.1308
y(0.500)=y4=0.1397
y(0.625)=y5=0.1308
y(0.750)=y6=0.1042
y(0.875)=y7=0.0604
>>
```

3. Trapezoidal rule

Script: Trapezoidal.m

```
clear all;
disp('Trapezoidal method');
func=input('Enter the function y=f(x)=');
f=inline(func);
n=input('Enter the number of sub-intervals: n=');
x0=input('Enter the intial value of x: x0=');
xn=input('Enter the final value of x: xn=');
h=(xn-x0)/n;
x=x0:h:xn
y=zeros(1,length(x));
for i=1:n+1
    y(i)=f(x(i));
end
y
I1=0;
I2=0;
I1=y(1)+y(n+1);
for i=2:n
```

```
I2=I2+2*y(i);
end
I=0.5*h*(I1+I2);
fprintf('Thus the value of the integral is %.4f',I);
```

Command Window:

```
>> Trapezoidal
Trapezoidal method
Enter the function y=f(x)='1/(1+x)'
Enter the number of sub-intervals: n=8
Enter the intial value of x: x0=0
Enter the final value of x: xn=1

x =

         0    0.1250    0.2500    0.3750    0.5000    0.6250    0.7500    0.8750    1.0000


y =

    1.0000    0.8889    0.8000    0.7273    0.6667    0.6154    0.5714    0.5333    0.5000

Thus the value of the integral is 0.6941
>>
```

4. Simpson's 1/3 rule

Script file: Simpsons1by3.m

```
clear all;
disp('Simpsons 1/3 method');
func=input('Enter the function y=f(x)=');
f=inline(func);
n=input('Enter the number of sub-intervals: n=');
x0=input('Enter the intial value of x: x0=');
xn=input('Enter the final value of x: xn=');
h=(xn-x0)/n;
x=x0:h:xn
y=zeros(1,length(x));
for i=1:n+1
    y(i)=f(x(i));
end
y
I1=0;
I2=0;
I1=y(1)+y(n+1);
for i=2:n
    if mod(i-1,2)==0
        I2=I2+2*y(i);
    else
         I2=I2+4*y(i);
    end
 end
I=(h/3)*(I1+I2);
fprintf('Thus the value of the integral is %.4f',I);
```

Command Window:

```
 >> Simpsons1by3
Simpsons 1/3 method
Enter the function y=f(x)='1/(1+x)'
Enter the number of sub-intervals: n=8
Enter the intial value of x: x0=0
Enter the final value of x: xn=1

x =

         0    0.1250    0.2500    0.3750    0.5000    0.6250    0.7500    0.8750    1.0000


y =

    1.0000    0.8889    0.8000    0.7273    0.6667    0.6154    0.5714    0.5333    0.5000

Thus the value of the integral is 0.6932>>
```

5. Simpson’s 3/8 rule

Script file: Simpsons3by8.m

```
clear all;
disp('Simpsons 3/8 method');
func=input('Enter the function y=f(x)=');
f=inline(func);
n=input('Enter the number of sub-intervals: n=');
x0=input('Enter the intial value of x: x0=');
xn=input('Enter the final value of x: xn=');
h=(xn-x0)/n;
x=x0:h:xn
y=zeros(1,length(x));
for i=1:n+1
    y(i)=f(x(i));
end
y
I1=0;
I2=0;
I1=y(1)+y(n+1);
for i=2:n
    if mod(i-1,3)==0
        I2=I2+2*y(i);
    else
         I2=I2+3*y(i);
    end
 end
I=(3/8)*h*(I1+I2);
fprintf('Thus the value of the integral is %.4f',I);
```

Command Window:

```
>> Simpsons3by8
Simpsons 3/8 method
Enter the function y=f(x)='1/(1+x)'
Enter the number of sub-intervals: n=9
Enter the intial value of x: x0=0
Enter the final value of x: xn=1

x =

  Columns 1 through 9

         0    0.1111    0.2222    0.3333    0.4444    0.5556    0.6667    0.7778    0.8889

  Column 10

    1.0000


y =

  Columns 1 through 9

    1.0000    0.9000    0.8182    0.7500    0.6923    0.6429    0.6000    0.5625    0.5294

  Column 10

    0.5000

Thus the value of the integral is 0.6932>>
```