

MATLAB CODES

1.Lagrange Method

EX: Find y at x= (1.1) → c

X=(1,2,3,4) y=(1,8,27,64)

```
function [ pn ] = lagrange( x,y,c )
n=length(y);
pn=0;
for i= 1:n
    L(i)=1;
    for j=1:n
        if i~=j
            L(i)=L(i) * ((c-x(j)) / (x(i)-x(j)));
        end
    end
    pn=pn+L(i)*y(i);
end

end
```

Solve:

```
>> [pn]=lagrange([1 2 3 4], [1 8 27 64],1.1)
```

```
pn =1.3310
```

2. Forward Method

EX: Find y at $x = (1, 1) \rightarrow c$

$X = (1, 2, 3, 4)$ $y = (1, 8, 27, 64)$

```
function [ Pn ] = myforward( x, y, c )
n=length(y);
h=x(2)-x(1);
s=(c-x(1))/h;
table(:,1)=y;
Pn=y(1);
ss=1;
for j=2:n
    for i=1:n-(j-1)
        table(i,j)=table(i+1,j-1)-table(i,j-1);
    end
    ss=ss*(s-(j-2));
    Pn=Pn+ss*table(1,j)/factorial(j-1);
end
end
```

Solve:

```
>> [ Pn ] = myforward( [1 2 3 4], [1 8 27 64], 1.1 )
Pn =      1.3310
```

3.Backward Method

EX: Find y at x= (1.1) \rightarrow c

X=(1,2,3,4) y=(1,8,27,64)

```
function [ pm ] =mybackward( x,y,c )
n=length(x);
for i=1:n
diff(i,1)=y(i);
end
for j=2:n
    for i=n:-1:j
        diff(i,j)=diff(i,j-1)-diff(i-1,j-1);
    end
end
answer=y(n);
h=x(n)-x(n-1);
s=(c-x(n))/h;
for i=1:n-1
    term=1;
    for j=1:i
        term=term*(s+j-1)/j;
    end
    answer=answer+term*diff(n,i+1);
end
pm=answer;
```

Solve:

```
>> [pm]=mybackward([1 2 3 4], [1 8 27 64], 1.1)
```

```
pm = 1.3310
```

4. Trapezoid Method

EX:

$$\int_0^5 \frac{dx}{4x+5} , \quad n = 10$$

$$x_0 = 0, x_n = 5, f(x) = \frac{1}{4x+5}$$

$$h = \frac{x_n - x_0}{n} = \frac{5 - 0}{10} = \frac{1}{2}$$

```
function [ I ] = mytrapezoid( x0,xn,n )
f=@(x) 1/(4*x+5);
h=(xn-x0)/n;
s=0;
for i=1:n-1
    x=x0+i*h;
    s=s+f(x);
end
I=(h/2)*(f(x0)+f(xn)+2*s);

end
```

Solve:

```
>> [I]=mytrapezoid(0,5,10)
```

```
I = 0.4055
```

5.Simpson Method

EX:

$$\int_0^5 \frac{dx}{4x+5} , n = 10$$

$$x0 = 0, xn = 5, f(x) = \frac{1}{4x+5}$$

$$h = \frac{xn - x0}{n} = \frac{5 - 0}{10} = \frac{1}{2}$$

```
function [ I ] = mysimpson( x0, xn, n )
f=@(x) 1/(4*x+5);
h=(xn-x0)/n;
s1=0;
for i=1:2:n-1
    x=x0+i*h;
    s1=s1+f(x);
end
s2=0;
for i=2:2:n-2
    x=x0+i*h;
    s2=s2+f(x);
end
I=(h/3)*(f(x0)+f(xn)+4*s1+2*s2);
end
```

Solve:

```
>> [I]=mysimpson(0,5,10)
```

```
I = 0.4025
```

To find exact and error values:

```
>> syms x
>> exact=int('1/(4*x+5)',x,0,5)

exact =

log(5)/4

>> exact=vpa(int('1/(4*x+5)',x,0,5))

exact =

0.40235947810852509365018983330655

>> error=abs(exact-I)

error =

0.00016127041303102174553549489220359
```

6.Euler Method

EX:

$$\dot{y} = x^2y - 1, y(0) = 1, \text{ find } y(1)$$

$$x_{i+1} = x_i + h$$

$$y_{i+1} = y_i + hf(x_i, y_i)$$

```
function [ y ] = Euler( x0,y0,xn,n )
f=@(x,y)x^2*y-1;
h=(xn-x0)/n;
x(1)=x0;
y(1)=y0;
for i=1:n
    x(i+1)=x(i)+h;
    y(i+1)=y(i)+h*f(x(i),y(i));
end

end
```

Solve:

```
>> [y]=Euler(0,1,1,10)
y =
Columns 1 through 10
1.0000    0.9000    0.8009    0.7041    0.6104    0.5202
0.4332    0.3488    0.2659    0.1829
Column 11
0.0977
```

7. Modified Euler

EX:

$$\dot{y} = x^2 y - 1, y(0) = 1, \text{ find } y(1)$$

$$x_{i+1} = x_i + h$$

$$y_{i+1} = y_i + \frac{h(\dot{y}_i + \dot{y}_{i+1})}{2}$$

$$\dot{y}_i = f(x_i, y_i), \quad \dot{y}_{i+1} = f(x_{i+1}, y_{i+1})$$

```

function [ y ] = modifiedEuler( x0, y0, xn, n )
f=@(x,y)x^2*y-1;
h=(xn-x0)/n;
x(1)=x0;
y(1)=y0;
for i=1:n
    x(i+1)=x(i)+h;
    ynew=y(i)+h*f(x(i),y(i));
    y(i+1)=y(i)+(h/2)*(f(x(i),y(i))+f(x(i+1),ynew));
end

end

```

Solve:

```
[y]=modifiedEuler(0,1,1,10)
```

```
y =
```

```
Columns 1 through 10
```

1.0000	0.9004	0.8025	0.7073	0.6154	0.5269
0.4414	0.3581	0.2757	0.1923		

```
Column 11
```

```
0.1055
```

8. Runge – Kutta Method

EX:

$$\dot{y} = x^2y - 1, y(0) = 1, \text{ find } y(1)$$

$$x_{i+1} = x_i + h$$

$$k_1 = hf(x_i, y_i)$$

$$k_2 = hf(x_i + h, y_i + k_1)$$

$$y_{i+1} = y_i + \frac{k_1 + k_2}{2}$$

```
function [ y ] = RungeKutta( x0,y0,xn,n )
f=@(x,y)x^2y-1;
h=(xn-x0)/n;
x(1)=x0;
y(1)=y0;
for i=1:n
    x(i+1)=x(i)+h;
    k1=h*f(x(i),y(i));
    k2=h*f(x(i)+h,y(i)+k1);
    y(i+1)=y(i)+(k1+k2)/2;
end
end
```

Solve:

```
>> [y]=RungeKutta(0,1,1,10)
y =
Columns 1 through 10
1.0000    0.9004    0.8025    0.7073    0.6154    0.5269
0.4414    0.3581    0.2757    0.1923
Column 11
0.1055
```

Find exact and error values:

```
>> syms yy(x)
>> exact=dsolve(diff(yy)==x^2*yy-1, 'yy(0)==1')

exact =
exp(x^3/3) + exp(x^3/3)*((3^(1/3)*igamma(1/3, x^3/3))/3 -
(2*pi*3^(5/6))/(9*gamma(2/3)))

>> exact=subs(dsolve(diff(yy)==x^2*yy-1, 'yy(0)==1'),x,1);
>> exact=vpa(subs(dsolve(diff(yy)==x^2*yy-1, 'yy(0)==1'),x,1))

exact =
0.10603386871334848150807731915644

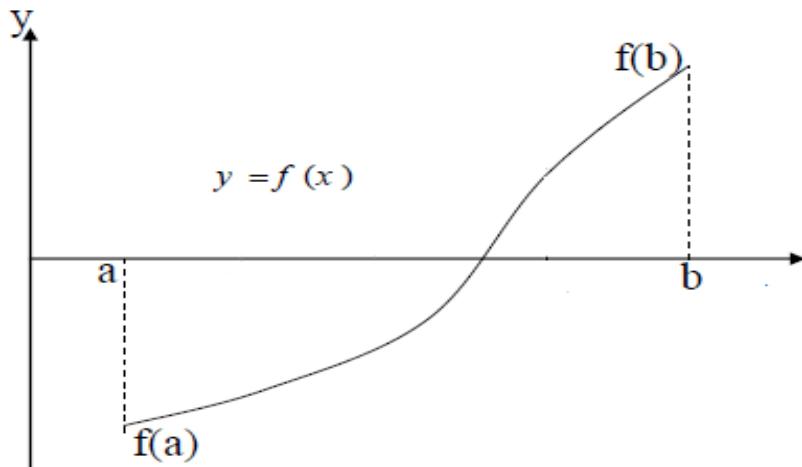
error=abs(exact-y(n+1)) → from example n+1 = 11
```

9. Bisection Method

EX:

$$f(x) = x^6 - x - 1$$

$$[a, b] = [1, 2], \quad tol = 0.0001$$



```
function [ c ] = Bisection( a,b,tol )
f=@(x)x^6-x-1;
c=(a+b)/2;
while(abs(f(c))>tol)
    if(f(a)*f(c)<0)
        b=c;
    else
        a=c;
    end
    c=(a+b)/2;
end
```

Solve:

```
> [ c ]=Bisection( 1,2,0.0001 )

c =
1.1347
```

Find exact value:

```
>> syms x  
>> exact=solve('x^6-x-1')  
  
exact =  
  
RootOf(z^6 - z - 1, z) [1]  
RootOf(z^6 - z - 1, z) [2]  
RootOf(z^6 - z - 1, z) [3]  
RootOf(z^6 - z - 1, z) [4]  
RootOf(z^6 - z - 1, z) [5]  
RootOf(z^6 - z - 1, z) [6]  
  
>> exact=vpa(solve('x^6-x-1'))  
  
exact =  
-0.77808959867860109788068230965929  
  
1.1347241384015194926054460545065  
- 0.62937242847031484088867026312539 +  
0.73575595299977645860977302724461i  
0.4510551586088556435262883907018 -  
1.0023645715871650194295456241605i  
- 0.62937242847031484088867026312539 -  
0.73575595299977645860977302724461i  
0.4510551586088556435262883907018 +  
1.0023645715871650194295456241605i
```

10. Newton – Raphson Method

EX:

$$f(x) = x^6 - x - 1$$

$$x_1 = 1 , \quad tol = 0.0001$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

```
function [ x2 ] = NewtonRaphson( x1,tol )
f=@(x)x^6-x-1;
fd=@(x) 6*x^5-1;
x2=x1-f(x1)/fd(x1);
while abs(f(x2))>tol
    x1=x2;
    x2=x1-f(x1)/fd(x1);
end
end
```

Solve:

```
>> [ x2 ] = NewtonRaphson( 1,0.0001 )

x2 =
1.1347
```

11. Solve system of linear equations:

EX

$$8x - 3y + 2z = 20$$

$$4x + 11y - z = 33$$

$$6x + 3y + 12z = 35$$

find x, y and z

In command window

```
>> syms x y z
>> [x,y,z]=solve('8*x-3*y+2*z==20','4*x+11*y-
z==33','6*x+3*y+12*z==35')

x =
3421/1134

y =
1126/567

z =
517/567
```