Lab 1 2/B 13/10/18

Task 1:  
clc

clear all;

a=5;

b=10;

sum=a+b

Task 2:

clc

clear all

a=[ 1 2 3

4 5 6

7 8 9];

b=[ 9 8 7

6 5 4

3 2 1];

sum= a+b

Task 3:

clc

clear all;

sum=0;

for i=1:1:50

sum=sum+i;

end

sum

sum\_odd=0;

for i=1:2:50

sum\_odd=sum\_odd+i;

end

sum\_odd

sum\_even=0;

for i=0:2:50

sum\_even=sum\_even+i;

end

sum\_even

Another approach:

clc

clear all;

for i=1:1:50

a(i)=i;

end

sum\_all=sum(a)

for i=1:2:50

b(i)=i;

end

sum\_odd=sum(b)

for i=2:2:50

c(i)=i;

end

sum\_even=sum(c)

Task 4:

clc

clear all

a=[ 16 12 20 15];

b= sort(a,'descend') (for ascending order b=sort(a))

average=(b(1,1)+b(1,2)+b(1,3)+b(1,4))/4

Task 5:

clc

clear all

m=5;

disp('The number triangle from 1 to 5 is given below: ')

for i=1:m

for j=1:m-i

fprintf(' ');

end

for k=1:i

fprintf('%d ',i);

end

fprintf('\n');

end

Task 6:

clc

clear all

a=[ 14 12 20 19

15 11 18 8

13 9 7 15

17 14 20 16];

b=sort(a,2) %row wise sorting . For column wise sorting b=sort(a)

average=[b(1,1)+(b(1,2)+b(1,3)+b(1,4))/4

b(2,1)+(b(2,2)+b(2,3)+b(2,4))/4

b(3,1)+(b(3,2)+b(3,3)+b(3,4))/4

b(4,1)+(b(4,2)+b(4,3)+b(4,4))/4]

rounded\_average= round(average)

Task 6:

clc

clear all

a=[1 5 3

9 4 2

8 7 5];

sum\_total=a(1,1)+a(1,2)+a(1,3)+a(2,1)+a(2,2)+a(2,3)+a(3,1)+a(3,2)+a(3,3)

sum\_diagonal=a(1,1)+a(2,2)+a(3,3)

Lab 2 3/B 30/10/18

Task 1:

clc

clear all;

t=0:.001:20;

y=sin(t);

plot(t,y);

grid on;

Task 2:

clc

clear all;

t=0:.001:20;

y=cos(t);

plot(t,y);

grid on;

Task 3:

clc

clear all;

t=0:.35:20;

y=tan(t);

plot(t,y);

grid on;

Task 4:

clc

clear all;

t=0:.20:20;

y=cot(t);

plot(t,y);

grid on;

Task 5:

clc

clear all;

t=0:0.45:20;

subplot(2,2,1);

y=sin(t);

plot(t,y);

title('SINE');

subplot(2,2,2);

y=cos(t);

plot(t,y);

title('COSINE');

subplot(2,2,3);

y=tan(t);

plot(t,y);

title('TANGENT');

subplot(2,2,4);

y=cot(t);

plot(t,y);

title('COTANGENT');

Lab 3 6/11/18 4/B

clc

clear all;

f=@(x)((x^3)-x-1);

a=1;

b=2;

display('IT No. a b x f(x) ');

display('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

for i=1:1:100

x=(a+b)/2;

fprintf(' %d %.3f %.3f %.3f %.3f\n',i,a,b,x,f(x));

if(abs(f(x))<.001)

break;

end

if(f(x)<0)

a=x;

else

b=x;

end

end

Another approach:

clc

clear all

a=1;

b=2;

disp('it\_no a b x f(x) ')

disp('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')

for i=1:1:50

x=(a+b)/2;

f=(x.^3)-x-1;

if abs(f)<=.01

break

end

if f<0

a=x;

elseif f>0

b=x;

end

fprintf(' %d %.3f %.3f %.3f %.3f\n',i,a,b,x,f)

end

fprintf('\n\nRequired root = %.3f',x)

Lab 4 5/B

clc

clear all;

f=@(x)((x^3)-x-1);

a=1;

b=2;

display('IT No. a b x f(x) ');

display('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

for i=1:1:10

x=(a\*f(b)-b\*f(a))/(f(b)-f(a));

fprintf(' %d %.3f %.3f %.3f %.3f\n',i,a,b,x,f(x));

if(abs(f(x))<.001)

break;

end

if(f(x)<0)

a=x;

else

b=x;

end

end

Lab 5 6/B

Newton-Raphson Method:

clc

clear all;

x=1;

f=@(x)(power(x,3)-x-1);

df=@(x)(3\*power(x,2)-1);

display('IT No. a b x f(x) ');

display('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

for i=1:1:50

xn=x-((f(x))/(df(x)));

fprintf(' %d %.3f %.3f %.3f\n',i,x,xn,f(x));

x=xn;

if(abs(f(x))<=.001)

break;

end

end

Lab 6 8/B

Task 1

Secant Method:

clc

clear all

xold=0;

x=1;

f=@(x)(exp(-x)-x);

display('IT No. xold x xnew f(x) ');

display('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

for i=1:1:100

xnew=(xold\*f(x)-x\*f(xold))/(f(x)-f(xold));

fprintf(' %d %.3f %.3f %.3f %.3f\n',i,xold,x,xnew,f(x));

x=xnew;

if(abs(f(x))<.001)

break;

end

end

Task 2:

clc

clear all

x1 = 'Enter value of X:' ;

x = input(x1);

y1 = 'Enter value of Y:' ;

y = input(y1);

disp(' IT No. u v du/dx du/dy dv/dx dv/dy xi+1 yi+1');

disp('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

for i=1:1:100

u=x\*x-y\*y-3;

v=x\*x+y\*y-13;

dux=2\*x;

duy=-2\*y;

dvx=2\*x;

dvy=2\*y;

xi1=x-((u\*dvy-v\*duy)/(dux\*dvy-duy\*dvx));

yi1=y-((v\*dux-u\*dvx)/(dux\*dvy-duy\*dvx));

disp([i',u',v',dux',duy',dvx',dvy',xi1',yi1']);

if(abs(u)<0.001) && (abs(v)<.001)

fprintf('The roots are: %f & %f',xi1,yi1)

break;

end

x=xi1;

y=yi1;

end

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* TBS END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* BA Session start()\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Lab 7 9/B 15/1/19

clc;

clear all;

x=0:.08:5;

y=1./(1+x);

z=@(x) 1./(1+x);

calc\_area=trapz(x,y);

exact\_area=integral(z,0,5);

error=(calc\_area-exact\_area)/exact\_area;

error\_percent=abs(error)\*100;

fprintf("error percent= %.4f%%\n",error\_percent);

Another approach:

clc;

clear all;

a=0;

b=5;

n=100;

h=(b-a)/n;

f=@(x)(1/(1+x));

z=@(x)(1./(1+x));

c=(f(a)+f(b));

for i=1:1:n-1

c=c+2\*f(a+(i\*h));

end

calc\_area=(h/2)\*c;

exact\_area=integral(z,0,5);

error=(calc\_area-exact\_area)/calc\_area\*1;

error\_percent=abs(error)\*100;

fprintf("error percent= %.4f%%\n",error\_percent);

Lab 8 10/B 22/1/19

clc;

clear all;

f=@(x)x\*sin(x);

a=0;

b=pi/2;

n=18;

h=(b-a)./n;

s=f(a)+f(b);

for i=1:2:n-1

s=s+4\*f(a+i\*h)

end

for i=2:2:n-2

s=s+2\*f(a+i\*h)

end

I=(h./3)\*s;

plot(f(a),f(b));

s=0:0.01:20;

y=s.\*sin(s);

plot(s,y);

Lab 9 11/B 29/1/19

clear, clc, close all

% Work with values around center c

c = pi/2;

x = -4 : .1 : 6;

y = cos(x);

% Plot the goal

plot(x, y, 'g', 'Linewidth', 3)

title('Study of Taylor series')

xlabel('x')

ylabel('cos(x) with different number of terms')

axis([-4 6 -3 3])

grid on

hold on

% Consider 4 terms in the series

smp = taylor\_cosine(c, x, 4);

plot(x, smp, 'ro')

% Consider 6 terms

smp = taylor\_cosine(c, x, 6);

plot(x, smp, 'b-.')

% Consider 10 terms

smp = taylor\_cosine(c, x, 10);

plot(x, smp, 'k', 'linewidth', 3)

% Label the calculated lines

legend('cos(x)', '4-term series', ...

'6 terms', '10 terms')

The results are:

We see that all of the Taylor expansions work well when we are close to pi/2 (x approx. 1.57).

More terms approximate better a larger portion of the cosine curve.

Lab 10 12/B 5/2/19

Task 1:

clc

clear all;

x=0;

y=1;

f=@(x,y)((y-x)/(y+x));

h=0.002;

j=0;

display('IT.NO. X Y');

display('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

for i=0:h:0.1

j=j+1;

x=x+h;

y=y+h\*f(x,y);

fprintf("%d %f %f\n",j,x,y);

if(x>.1)

break;

end

end

Task 2:

clc

clear all;

x=0;

y=1;

f=@(x,y)((y-x)/(y+x));

h=0.002;

j=0;

display('IT.NO. X Y');

display('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

for i=0:h:0.1

j=j+1;

x=x+h;

y=y+(h/2)\*(f(x,y)+f(x+h,y+h\*f(x,y)));

fprintf("%d %f %f\n",j,x,y);

if(x>.1)

break;

end

end

Task 3:

clc

clear all;

x=0;

y=1;

f=@(x,y)((y-x)/(y+x));

h=0.002;

k1=h\*f(x,y);

k2=h\*f(x+(h/2),y+(k1/2));

k3=h\*f(x+(h/2),y+(k2/2));

k4=h\*f(x+h,y+k3);

dy=(1/6)\*(k1+(2\*k2)+(2\*k3)+k4);

y=y+dy;

y

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The End\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*