Q1

Table

Description automatically generated

**Q2**

Text

Description automatically generated

**Q3Graphical user interface, text, application

Description automatically generated**

A picture containing table

Description automatically generated

**Q4**

f(x)=e^(-2366936\*|x^3|+54321\*x^4)

C1>100\*C2

Tol=1e-3

a=-1111

b=1111

Text

Description automatically generatedA picture containing text

Description automatically generatedText

Description automatically generated with medium confidence

**Matlabcode**

Note that the function recursion\_simpson will be used to construct adaptive\_simpson

function [S,C2]=recursion\_simpson(f,S,nodes,y,C2,tol);

h=(nodes(1,end)-nodes(1))/2;

nnodes=[nodes(1)+h/2 nodes(2)+h/2];

ny=feval(f,nnodes);

S1=(y(1)+4\*ny(1)+y(2))\*h/6;

S2=(y(2)+4\*ny(2)+y(3))\*h/6;

temp=S1+S2;

C2=C2+2;

if abs(temp-S)<tol

S=16/15\*(S1+S2)-S/15;

else

tol=tol/2;

[S1,C21]=recursion\_simpson(f,S1,[nodes(1),nodes(1)+h/2,nodes(2)],[y(1),ny(1),y(2)],C2,tol);

[S2,C22]=recursion\_simpson(f,S2,[nodes(2),nodes(2)+h/2,nodes(end)],[y(2),ny(2),y(3)],C2,tol);

S=S1+S2;

C2=C21+C22;

end

end

function [S,C2]=adaptive\_simpson(f,a,b,tol)

C2=0;

h=b-a;

nodes=[a,a+h/2,b];

y=f(nodes);

S=(y(1,1)+4\*y(1,2)+y(1,3))\*h/6;

tol=tol\*15;

[S,C2]=recursion\_simpson(f,S,nodes,y,C2,tol);

C2=C2+3

end

function [SC,C1]=composite\_simpson(f,a,b,tol,I);

n=0;

error=88888;

while error>tol

n=n+2;

x=linspace(a,b,2\*n+1);

y=f(x);

SC=((b-a)/n/6)\*(y(1)+y(2\*n+1)+2\*sum(y(3:2:2\*n-1))+4\*sum(y(2:2:2\*n)));

error=abs(SC-I);

end

C1=2\*n+1;

end

tol=1e-3;

f = @(x)exp(-2366936.\*(abs(x.^3)+54321.\*x.^4));

I = quad(f,-1111,1111,1e-5);

[A2 C2]=adaptive\_simpson(f,-1111,1111,tol)

[A1 C1]=composite\_simpson(f,-1111,1111,tol,I)

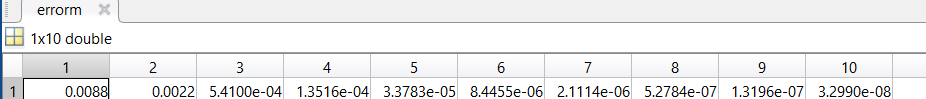
ERROR1=abs(A1-I)

ERROR2=abs(A2-I)

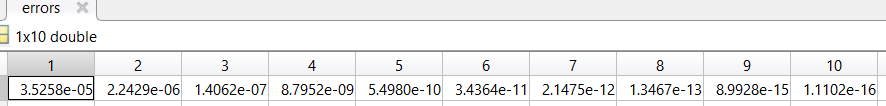
**Q5**

**(a)**

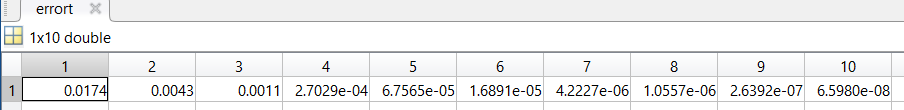
Erf1 of midpoint rule

****

Erf1 simpson rule

****

Erf1 of trapezoid rule

****

**Error plotChart, line chart

Description automatically generated**

**(b)**

Text, letter

Description automatically generatedGraphical user interface, application, chat or text message

Description automatically generated

**Matlab code**

function erft=composit\_trapezoid(f,b,h)

n=b/h;

i=1:n;

erft=sum(f(i.\*h)).\*h+f(0)\*h/2-f(b)\*h/2;

end

function erfm=composit\_midpoint(f,b,h)

n=b/h;

i = 1:n;

erfm=sum(f((i-0.5).\*h))\*h;

end

function erfs=composit\_simpson(f,b,h)

n=b/h;

ti=linspace(0,b,2\*n+1);

y=f(ti);

erfs=(h/6)\*(y(1)+y(2\*n+1)+2\*sum(y(3:2:2\*n-1))+4\*sum(y(2:2:2\*n)));

end

f = @(t)exp(-t.^2)\*2/sqrt(pi);

I=integral(f,0,1);

for i = 1:10

h(i)=1/2^i;

errort(i) = abs(composit\_trapezoid(f,1,h(i))-I);

errorm(i) = abs(composit\_midpoint(f,1,h(i))-I);

errors(i) = abs(composit\_simpson(f,1,h(i))-I);

end

loglog(h,errort);

hold on

loglog(h,errorm);

loglog(h,errors);

legend('composit\_trapezoid','composit\_midpoint','composit\_simpson');

xlabel('h');

ylabel('error');

hold off

t=[ones(10,1),log(h)']\(log(errort))';

ct=exp(t(1));

kt=t(2);

m=[ones(10,1),log(h)']\(log(errorm))';

cm=exp(m(1));

km=m(2);

s=[ones(10,1),log(h)']\(log(errors))';

cs=exp(s(1));

ks=s(2);

fprintf('trapezoid %.2e\*h^%.2f\n', ct, kt);

fprintf('midpoint %.2e\*h^%.2f\n', cm, km);

fprintf('simpson %.2e\*h^%.2f\n', cs, ks);

**Q6**

Text

Description automatically generated

nbodydata=importdata('nbody.dat');

l=length(nbodydata);

c=zeros(5,5);

x = zeros(l,5);

y = zeros(l,5);

fprintf(' a b c d e\n')

for i=0:4

x(:,i+1)=nbodydata(:,2+3\*i);

y(:,i+1)=nbodydata(:,3+3\*i);

c(:,i+1)=[y(:,i+1).^2,y(:,i+1).\*x(:,i+1),x(:,i+1),y(:,i+1),ones(l,1)]\x(:,i+1).^2;

end

fprintf('Jupiter %f %f %f %f %f\nSaturn %f %f %f %f %f\nUranus %f %f %f %f %f\nNeptune %f %f %f %f %f\nPluto %f %f %f %f %f\n',c(1,1),c(2,1),c(3,1),c(4,1),c(5,1),c(1,2),c(2,2),c(3,2),c(4,2),c(5,2),c(1,3),c(2,3),c(3,3),c(4,3),c(5,3),c(1,4),c(2,4),c(3,4),c(4,4),c(5,4),c(1,5),c(2,5),c(3,5),c(4,5),c(5,5));