**Problem No. :** 01

**Problem Name :** Using trapezoidal and simpson’s rule evaluate the given double integration.

**Theory :**

The trapezium rule works by splitting the area under a curve into a number of trapeziums, which we know the area of. If we want to find the area under a curve between the points x0 and xn, we divide this interval up into smaller intervals, each of which has length h (see diagram above).

Program for Simpson's 1/3 Rule. In numerical analysis, Simpson's 1/3 rule is a method for numerical approximation of definite integrals. Specifically, it is the following approximation: In Simpson's 1/3 Rule, we use parabolas to approximate each part of the curve.We divide. the area into n equal segments of width Δx.

**Source Code :**

#include<iostream>

using namespace std;

const double h = 2.0,k=2.0;

void trapizoidal(int n,double a[10][10])

{

int i,j;

double sum = 0,result;

for(i=0;i<=n;i++)

{

for(j=0;j<=n;j++)

{

if((i==0&&j==0)||(i==n&&j==n))

sum = sum + a[i][j];

else

sum = sum + 2 \* a[i][j];

}

}

result = ((h\*k)/4.0) \* sum;

cout<<"Trapizoidal solution : "<<result<<endl;

}

void simpsons(int n,double a[10][10])

{

int i,j;

double sum = 0,result;

for(i=0;i<=n;i++)

{

for(j=0;j<=n;j++)

{

if((i==0||i==n)&&(j==0||j==n))

sum = sum + a[i][j];

else if(i==j)

sum = sum + 16 \* a[i][j];

else

sum = sum + 4 \* a[i][j];

}

}

result = ((h\*k)/9.0) \* sum;

cout<<"Simpson's solution : "<<result<<endl;

}

int main()

{

int i,j,k;

int n = 4.0 / h;

double x[n+1],y[n+1];

x[0] = 0.0;

y[0] = -2.0;

for(i=1;i<=n;i++)

{

x[i] = x[i-1] + h;

y[i] = y[i-1] + h;

}

cout<<"x\ty\n\n";

for(i=0;i<=n;i++)

cout<<x[i]<<"\t"<<y[i]<<endl;

cout<<endl;

double a[10][10];

for(i=0;i<=n;i++)

{

for(j=0;j<=n;j++)

a[i][j] = x[j]\*x[j] - x[j]\*y[i] + y[i]\*y[i];

}

for(i=0;i<=n;i++)

{

for(j=0;j<=n;j++)

cout<<a[i][j]<<"\t";

cout<<endl;

}

cout<<endl;

while(1)

{

cout<<"1. Trapizoidal\n2. Simpson's\n3. Exit\nPress : ";

cin>>k;

if(k==1)

trapizoidal(n,a);

else if(k==2)

simpsons(n,a);

else

break;

}

return 0;

}

**Output :**

