Title: Trapezoidal and simpson's Rules for Numerical Integration

OBJECTIVE: To solve numerical integrals using Trapezoidal and simpson's rules

Theory:

1. About Trapezoidal and simpson's rule & their types

Let, $y_0, y_1, y_2, \ldots, y_n$ be the rules of $y=f(x)$ corresponding to $x = x_0, x_1, x_2, \ldots, x_n$ which are equally spaced with step-size h.

Then by newton's forward interpolation formula

$$\int_{x_0}^{x_n} y\,dx = h\left[n y_0 + \frac{n^2}{2}\Delta y_0 + \frac{1}{2}\left(\frac{n^3}{3} - \frac{n^2}{2}\right)\Delta^2 y_0 + \frac{1}{6}\left(\frac{n^4}{4} - n^3 + n^2\right)\Delta^3 y_0\right.$$

$$\left. + \frac{1}{24}\left(\frac{n^5}{5} - \frac{3n^4}{2} + \frac{11n^3}{3} - 3n^2\right)\Delta^4 y_0 + \ldots\right] \quad 2$$

By putting n=1 in above eqn newton's - cotequandrature formula and assuming that there are only two paired value of x and y or that the interpolating polynomial is linear,

$$\int_{x_0}^{x_1} y\,dx = h\left[y_0 + \frac{\Delta y_0}{2}\right]$$ as higher order differences do not exist.

$$\text{or, } \int_{x_0}^{x_1} y\,dx = h\left[y_0 + \frac{1}{2}(y_1 - y_0)\right]$$

$$\text{or, } \int_{x_0}^{x_1} y\,dx = \frac{h}{2}(y_0 + y_1)$$

simpson's one-third Rule

By putting n=2 in Newton-cote's quadrature formula i.e, assuming that there are only three paired values of x and y or that the interpolating polynomial is a second degree polynomial

$$\int_{x_0}^{x_2} y\,dx = h\left[2y_0 + 2\Delta y_0 + \frac{1}{2}\left(\frac{8}{3}-2\right)\Delta^2 y_0\right]$$

as $\Delta^3 y_0$ & higher order difference do not exist.

$$\int_{x_0}^{x_2} y\,dx = h\left[2y_0 + 2(y_1 - y_0) + \frac{1}{3}(y_2 - 2y_1 - y_0)\right]$$

since $\Delta^2 y_0 = y_2 - 2y_1 + y_0$

$$\int_{x_0}^{x_2} y\,dx = \frac{1}{3}(y_0 - 4y_1 + y_2)$$

Algorithms

Simple Trapezoidal Rule

1. Start
2. Declare the variable
3. Input the lower limit & upper limit of integration, say a and b respectively.
4. compute the Integration as $I = h/2 * (f(a) + f(b))$ where, $h = b - a$ & $f(a)$, $f(b)$ are functional value for given function.
5. Display the result as I.
6. Stop

Simpson's 1/3 Rule

1. Start
2. Declare the variable
3. Input the lower limit & upper limit of integration, say a and b respectively
4. compute the Integration as:
   $I = h/3 * [f(a) + 4 * (f(a+b)/2) + f(b)]$;
   where, $h = (b-a)/2$ and $f(a)$, $f(b)$ and $f(a+b)$ are functional value for given function.
5. Display the result as I.
6. Stop

composite Trapezoidal Rule

1. start
2. Declare the variable.
3. Input the lower limit and upper limit of integration, say a and b respectively.
4. compute the no. of strip required, say n.
5. compute the width of the strip as: h=(b-a)/n.
6. compute the integration as:

$$2 * \sum_{i=1}^{n-1} f(a+i*h)$$

$$I = h/2 * [f(a)+f(b)]$$

7. Display the result as I
8. Stop

composite simpson's 1/3 Rule

1. start
2. Declare the variable
3. Input the lower limit and upper limit integration, say a and b respectively.
4. compute the no. of strip required, say n.
5. compute the width of the strip as: h=(b-a)/n.
6. compute the integration as

$$I = h/3 * [f(a) + 4*\{f(a+h)/2) + f(a+3h) + \} + 2*\{f(a+2h) + f(a+4h) + \dots\} + f(b)]$$

7. where, h=(b-a)/3 and f(a), f(b), f(a+h) & f(a+2h) are functional value for a given function.
8. Display the result as I.
9. stop

Simpson's 3/8 Rule

1. Start
2. Declare the variable
3. Input the lower limit & upper limit of integration, say a and b respectively.
4. ~~Input the no. of strip required say n.~~
5. Compute the integration as

   $I = 3h/8 * [f(a) + 3 * f(a+h) + 3 * f(a+2h) + f(b)];$

   where, $h = (b-a)/3$ and $f(a)$, $f(b)$, $f(a+h)$ and $f(a+2h)$ are functional value for given function.
6. Display the result as I.
7. Stop

Composite Simpson's 3/8 Rule

1. Start
2. Declare the variable
3. Input the lower limit & upper limit of integration say a and b respectively.
4. Input the no. of strip required, say n.
5. Compute the integration as

   $I = 3h/8 * [f(a) + 3 * \{f(a+h) + (a+2h) + f(a+4h) + \ldots + f(a+(n-1)h\} + 2\{f(a+3h) + f(a+6h) + \ldots f(b)\}];$

   where, $h = (b-a)/n$ and $f(a)$, $f(b)$, $f(a+h)$ and $f(a+2h)$ are functional value for given function.
6. Display the result as I
7. Stop

1. Source code

1. simple trapezoidal Rule

```
#include <stdio.h>
#include <math.h>
float f(float x)
{
    return (1+x*x*x);
}
int main()
{
    float a,b,h,I;
    printf("enter the lower and upper number a and b");
    scanf("%f%f", &a,&b);
    h=(b-a);
    I= h/2 *(f(a)+f(b));
    printf("i= %f", i);
    return(0);
}
```

output

```
enter the lower and upper number a and b: 1
                                           2
              I = 5.500000
```

2. ~~Algorithm~~ (composite Trapezoidal Rule

```
#include <stdio.h>
#include <math.h>
float f(float x)
{
    return (1+x*x*x);
}
int main()
{
    float a,b,h,I,n,sum=0; int i;
    printf("Enter the lower and upper variable a and b");
    Scanf("%f%f", &a,&b);
    printf(" enter the n");
    Scanf("%f", &n);
    h=(b-a)/n;
```

```
for (i=1; i<=n-1; i++)
{
  sum = sum + f(a+i*h);
}
  I = h/2 * (f(a) + f(b) + 2* sum);
  printf("I = %f", I);
  return(0);
}
```

output:

```
enter lower and upper variable a and b: 1
                                        2
                    enter n : 4
                    I = 4.796875
```

3. simpson's 1/3 Rule

```
#include<stdio.h>
#include<math.h>
float f(float x)
{
  return (1+2*x*x);
}
int main()
{
   int i;
   float a, b, h, I;
   printf("\n enter lower and upper variable a and b");
   scanf("%f%f", &a, &b);
   h = (b-a)/2;
   I = (h/3) * (f(a) + 4 * f((a+b)/2) + f(b));
   printf("I = %f", I);
   return (0);
}
```

output:

```
enter lower and upper variable a and b: 1
                                        5
                    I = 160
```

4. <u>composite simpson's 1/3 Rule</u>

```
#include<stdio.h>
#include <math.h>
 float f(float x)
 {
    return (1+x*x*x);
 }
   ~~float~~
int main()
   {
       float a, b, h, I, sume = 0, sumo = 0;
       int i, n;
      printf(" enter the lower and upper variables");
      scanf("%f %f", &a, &b);
      printf("enter the no.of variables");
      scanf("%d", &n);
      h=(b-a)/n;
      ~~while~~ i=2;
        while(i<=n-1)
          {
            sum = sume+f(a+i*h);
            i= i+2;
          }
          i=1;
          while(i<=n-1)
          {
            sumo = sumo+f(a+i*h);
            i= i+2;
          }
          I =(h/3)*(f(a)+4*sumo+2*sume+f(b));
          printf("%f", I);
        retun 0;
   }
```

<u>Output</u>:

enter the lower and upper number a and b?

1

2

Enter the no.of strips : 4

I = 160

5) simpson's 3/8 Rule

```
#include<stdio.h>
# include<math.h>
float f(floatx)
{
  return(1+2*x*x*x);
}
int main()
{
  float a,b,h,I;
  int i;
  printf("enter the lower and upper number a and b");
  scanf("%f%f", &a,&b);
  h=(b-a)/3;
  I= (3*h/8 )*(f(a)+3*f(a+h)+3*f(a+2*h)+f(b));
  printf(" I=%f",I);
  return(0);
}
```

output: Enter the lower and upper number: a and b:

1

2

I = 4.75

6. composite simpson's 3/8 Rule

```
#include<stdio.h>
# include<math.h>
float f(float x){
return (1+2*x*x*x);
}
 float  a, b, h, I, sum=0,n,x;
  int i;
  printf(" Enter the lower and upper variable a and b");
  scanf(" %f%f", &a,&b);
  printf(" enter the value of n);
  scanf ("%f", &n)
  h=(b-a)/n;
  for (i=1; i<n; i++)
  {
   x = a+i*h;
   if (i%3== 0)
   {
     sum= sum+2*f(x);
   }
```

```
else {
    sum = sum + 3* f(x);
  }
  I = (3*h/8)*(f(a)+f(b)+sum);
  printf("I = %f", I);
  return 0;
}
```

Output: Enter the ~~lower~~ lower and upper variable a and b:

1

2

Enter the value of n: 4

I = 6.815918

DISCUSSION AND CONCLUSION:

From this lab, we learnt more about simple and composite trapezoidal and simpson's rule. we were more clear about ~~the~~ simpson's rule and trapezoidal rules formula. ~~using C-prog~~