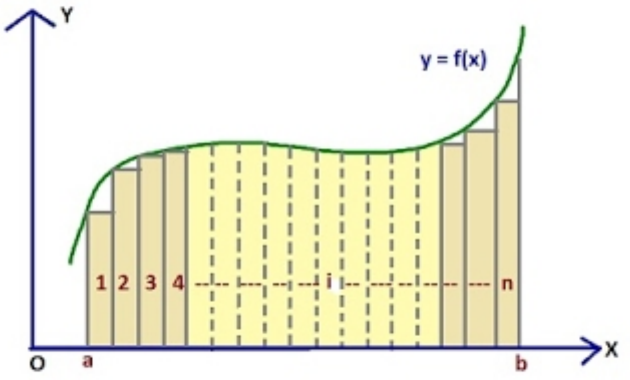
Numerical Integration

Introduction

Need prerequisite knowledge of **Newtons Forward Interpretation formula.**
Forward interpretation formula is a statistical method of finding the value function f(x) which lies near the beginning of the table(we have data of y=f(x) for x_0, x_1, x_2, .. x_i which is equally spaced).



Suppose **I** is an integration of f(x) in the limit (a,b). Let us divide the interval (a,b) into n equal parts of width **h** so that
[a= x_0, x_1] ,[x_1, x_2] , [x_2, x_3], ... [x_{n-1}, x_n = b]
by using Newtons Forward Interpretation formula in integration of f(x) we get the equation called **Newtons General Quadrature.**

Recommended reading
Link: http://www.nptel.ac.in/courses/122104018/node119.html

Weddle's Rule

posted Dec 28, 2017, 10:52 AM by Atul Rana **[updated 3 hours ago]**

Description

Integration of *f(x)* in range (a,b), first we divide the interval into n equally spaced intervals just like Airthematic Progression.

$x_0 = a$
$x_1 = x_0 + h$
$x_2 = x_0 + 2h$ or $x_1 + h$
...
...
$x_n = b$ or $x_0 + nh$ or $x_{n-1} + h$

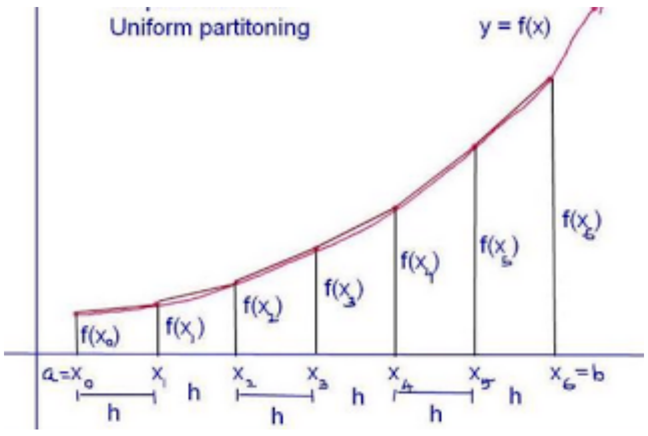
Values of *f(x)* can be shown as.

x	f(x)
x_0	y_0
x_1	y_1
x_2	y_2
x_3	y_3
...	...
x_n	y_n

Now using the formula of **Newtons General Quadrature Method,** if we integrate the function as below interval size.

[x_0 , $x_0 + 6h$]
[$x_0 + 6h$, $x_0 + 12h$]
....
....
[$x_0 + (n-6)h$, $x_0 + nh$]   and by adding all terms we get the integration of f(x) in range (a,b).

Finally, after adding formula of Weddle's Rule is for n=6 looks like:




```
I = 3h/10[ y0 +  5*y1 + y2 + 6*y3 + y4 + 5*y5 + y6]
```

- **Programming implementation of Weddle's Rule in C++**

```
#include<iostream>
#include<math.h>
using namespace std;

double f(double x){
    return 1/(1  + x*x);
}

int main(){
    // Define your function
    double a,b,h;
    cout<<"Give Limit of integration (a,b):";
    cin>>a>>b;
    int n;
    //do{
        cout<<"Define value of h:";
        cin>>h;
        n = ceil((b-a)/h);
    //}while(n%6 != 0);

    double integration=0;
    for(int j=0; j<=n;j++){
        if(j%2 == 0)
            integration += f(a + j*h);
        else if(j%3 == 0)
            integration += 6*f(a + j*h);
        else
            integration += 5*f(a + j*h);
    }

    integration = (3*h*integration)/10;
    cout<<"The value of Integration of f(x) limit (a,b) is:"<<integration<<endl;
    return 0;
}
```

- **Code execution looks like:**

```
Give Limit of integration (a,b):0 1
Define value of h:0.1666667
The value of Integration of f(x) limit (a,b) is:0.7854
```

Note: Method is applied when the number of subintervals is multiple of 6 or 6, the above code is only for n=6, try out code as per your requirements, and do experiments.

Simpson's 3/8th Rule

posted Dec 28, 2017, 10:51 AM by Atul Rana **[updated 3 hours ago]**

Description

Integration of *f(x)* in range (a,b), first we divide the interval into n equally spaced intervals just like Airthematic Progression.

$x_0 = a$

$x_1 = x_0 + h$

$x_2 = x_0 + 2h$ or $x_1 + h$

...

...

$x_n = b$ or $x_0 + nh$ or $x_{n-1} + h$

Values of *f(x)* can be shown as.

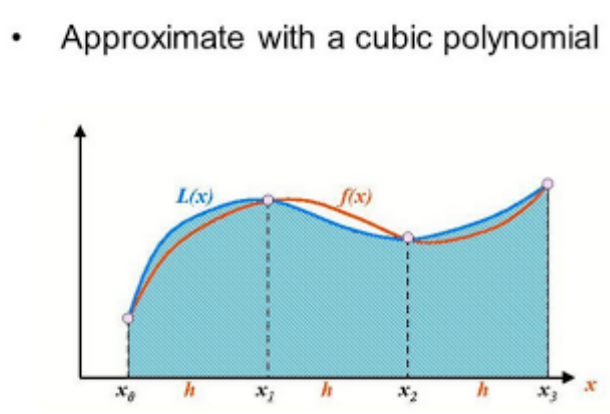
x	f(x)

x_0	y_0
x_1	y_1
x_2	y_2
x_3	y_3
...	...
x_n	y_n

Now using the formula of **Newtons General Quadrature Method**, if we integrate the function as below interval sizes.

$[x_0 , x_0 + 3h]$
$[x_0 + 3h, x_0 + 6h]$
....
....
$[x_0 + (n-3)h, x_0 + nh]$ and by adding them all we get the intergration of *f(x)* in range (a,b).

Finally, after adding formula of Simpson's 3/8th Rule looks like:



$$I = 3h/8[\ y_0 + 2\ (y_3 + y_6 + y_9 + \dots\ ) + 3(y_2 + y_4 + y_5 + y_7 + \dots) + y_n]$$

- **Programming implementation of Simpson's3/8th Rule in C++**

```
#include<iostream>
#include<math.h>
using namespace std;

double f(double x){
    return 1/(1  + x*x);
}

int main(){
    // Define your function
    double a,b,h;
    cout<<"Give Limit of integration (a,b):";
    cin>>a>>b;
    int n;
    //do{
        cout<<"Define value of h:";
        cin>>h;
        n = ceil((b-a)/h);
    //}while(n%3 != 0);

    double integration=0;
    for(int j=0; j<=n;j++){
        if(j==0 || j==n)
            integration += f(a + j*h);
        else if(j%3 == 0)
            integration += 2*f(a + j*h);
        else
            integration += 3*f(a + j*h);
    }

    integration = (3*h*integration)/8;
    cout<<"The value of Integration of f(x) limit (a,b) is:"<<integration<<endl;

    return 0;
}
```

- Code execution looks like:

```
Give Limit of integration (a,b):0 1
Define value of h:0.1666667
The value of Integration of f(x) limit (a,b) is:0.785396
```

Note: Simpson's 3/8th rule is only applied when the number of subintervals is multiple of 3.
Recommend to try out the code on other functions with different interval size h, test the accuracy of the method.

Simpson's 1/3rd Rule

posted Dec 28, 2017, 10:50 AM by Atul Rana   **[updated 3 hours ago]**

Description

Integration of *f(x)* in range (a,b), first we divide the interval into n equally spaced intervals just like Airthematic Progression.

$x_0 = a$

$x_1 = x_0 + h$

$x_2 = x_0 + 2h$ or $x_1 + h$

...

...

$x_n = b$ or $x_0 + nh$ or $x_{n-1} + h$

Values of *f(x)* can be shown as.

x	f(x)
x_0	y_0
x_1	y_1
x_2	y_2
x_3	y_3
...	...
x_n	y_n

Now using the formula of **Newtons General Quadrature Method**, if we integrate the function as below interval sizes.

$[x_0 , x_0 + 2h]$

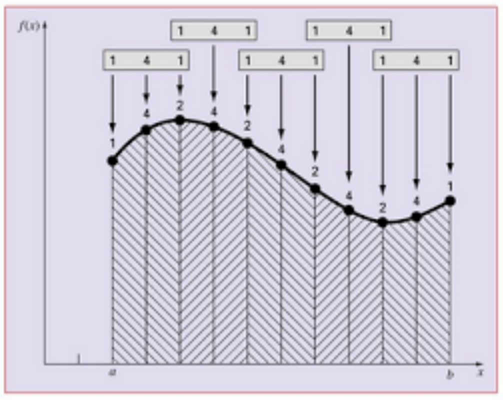
$[x_0 + 2h, x_0 + 4h]$

....

....

$[x_0 + (n-2)h, x_0 + nh]$ and by adding them all we get the intergration of *f(x)* in range (a,b).

Finally, after adding formula of Simpson's 1/3rd Rule is looks like:



➢ **Applicable only if the number of segments is <u>even</u>**

$$I = h/3[\ y_0 + 4(y_1 + y_3 + y_5 + .. + y_{n-1}) + 2(\ y_2 + y_4 + ... + y_{n-2}) + y_n]$$

- **Programming implementation of Simpson's 1/3 rd Rule:**

```
#include<iostream>
#include<math.h>
using namespace std;

double f(double x){
    return 1/(1  + x*x);
}

int main(){

    // Define your function
    double a,b,h;
    cout<<"Give Limit of integration (a,b):";
    cin>>a>>b;
    cout<<"Define value of h:";
    cin>>h;

    int n = ceil((b-a)/h);

    double integration=0;
    for(int j=0; j<=n;j++){
        if(j==0 || j==n)
            integration += f(a + j*h);
        else if(j%2 == 1)
            integration += 4*f(a + j*h);
        else
            integration += 2*f(a + j*h);
    }

    integration = (h*integration)/3;
    cout<<"The value of Integration of f(x) limit (a,b) is:"<<integration<<endl;

    return 0;
}
```

- Execution of above code:

```
Give Limit of integration (a,b):0 1
Define value of h:0.25
The value of Integration of f(x) limit (a,b) is:0.785392
```

Note: Simpson's 1/3rd rule applied only when the number of subintervals is even. Recommend to try out the code on other function with different interval size h, test the accuracy of the method.

Trapezoidal Rule

posted Dec 28, 2017, 10:42 AM by Atul Rana **[updated 3 hours ago]**

Description

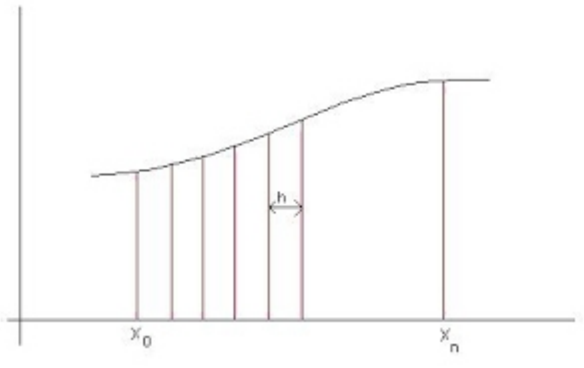
Integration of *f(x)* in range (a,b), first we divide the interval into n equally spaced intervals just like Airthematic Progression with difference **h**.

$x_0 = a$
$x_1 = x_0 + h$
$x_2 = x_0 + 2h$ or $x_1 + h$
...
...
$x_n = b$ or $x_0 + nh$ or $x_{n-1} + h$

Values of *f(x)* can be shown as.

x	f(x)
x_0	y_0
x_1	y_1
x_2	y_2
x_3	y_3
...	...
x_n	y_n

Now using the formula of *Newtons General Quadrature Method*, if we integrate the function as below interval sizes.
$[x_0 , x_0 + h]$
$[x_0 + h, x_0 + 2h]$
....
....
$[x_0 + (n-1)h, x_0 + nh]$ and by adding them all, we get the integration of f(x) in range (a,b).
Finally after adding, formula of Trapezoidal Rule is looks like:



$$I = h/2[\ y_0 + 2\ (y_1 + y_2 + y_3 + \dots + y_{n-1}) + y_n\ ]$$

- **Programming implementation of Trapezoidal Rule in C++**

```
#include<iostream>
using namespace std;

double f(double x){
    return 1/(1  + x*x);
}

int main(){

    // Define your function
    double a,b,h;
    cout<<"Give Limit of integration (a,b):";
    cin>>a>>b;
    cout<<"Define value of h:";
    cin>>h;

    double integration=0;
    for(double i=a; i<=b; i+= h){
        if(i==a || i==b)
            integration += f(i);
        else
```

```
            integration += 2*f(i);
    }

    integration = (h*integration)/2;
    cout<<"The value of Integration of f(x) limit (a,b) is:"<<integration<<endl;

    return 0;
}
```

- Execution of above code:

```
Give Limit of integration (a,b):0 1
Define value of h:0.2
The value of Integration of f(x) limit (a,b) is:0.783732
```

Note: Trapezoidal Rule is very simple and good approximate method to find integration with good accuracy(increasing n or decreasing h some time have a bad effect on result accuracy).

Recommend to try out the code on other function with different interval size h, test the accuracy of the method.