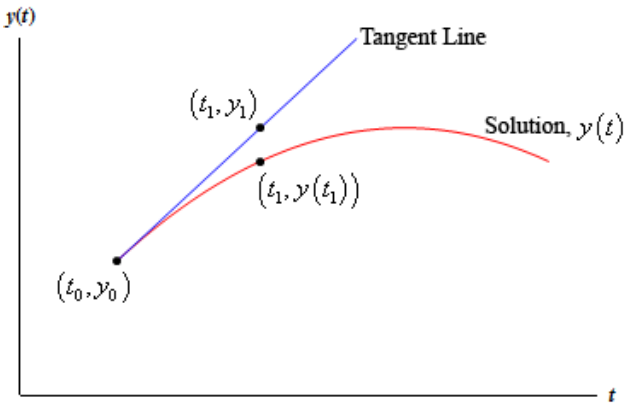
Numerical Solution of Ordinary Differential Equations

Introduction

This type of equations involves the function of the independent variable and its derivatives. Example of ODE can be dy/dx = x + y.

Prerequisites knowledge of ordinary differential equation and how to find their exact solutions using methods like calculating particular integral and integration factor etc.

Whole idea is concentrated on tangent to problem curve.



Runga-Kutta Method(of 4th Code)

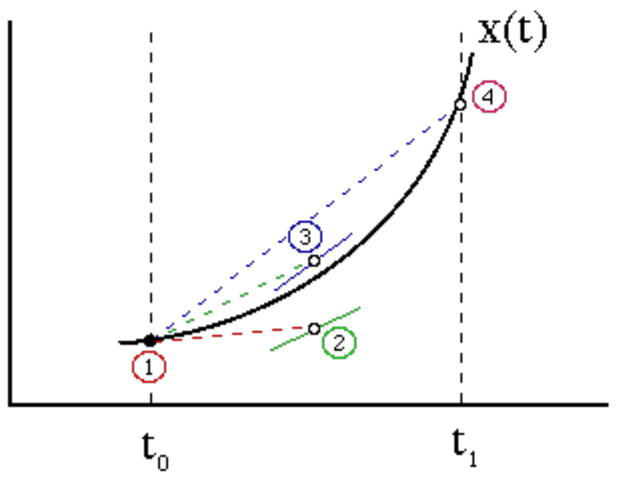
posted Dec 28, 2017, 11:01 AM by Atul Rana **[updated 9 minutes ago]**

Description

This method is one of the most widely used method and again it is also an iterative method which includes the routine of well known Euler's Method.

Suppose dy/dx = f(x,y) , y = y_0 when x = x_0 to find y for x = x_1 = x_0 + h
we have to compute
$k_1 = hf(x_0, y_0)$
$k_2 = hf(x_0 + h/2, y_0 + k_1/2)$
$k_3 = hf(x_0 + h/2, y_0 + k_2/2)$
$k_4 = hf(x_0 + h/2, y_0 + k_3)$
$k = 1/6(k_1 + 2k_2 + 2k_3 + k_4)$, Now the value of y_1 of y for $x_1 = x_0 + k$ is given by $y_1 = y_0 + k$
similarly we have to repeat for next value of x , x_2 ,x_3 ... and so on

In General for x_n
$k_1 = hf(x_{n-1}, y_{n-1})$
$k_2 = hf(x_{n-1} + h/2, y_{n-1} + k_1/2)$
$k_3 = hf(x_{n-1} + h/2, y_{n-1} + k_2/2)$
$k_4 = hf(x_{n-1} + h/2, y_{n-1} + k_3)$
$k = 1/6(k_1 + 2k_2 + 2k_3 + k_4)$, $y_n = y_{n-1} + k$



- **Programming implementation of RK4 in C++**

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

double f(double x,double y){
    return x*y + 1;
}
double f1(double x, double y){
    return 3*x + y/2;
}
int main(){
    //Suppose dy/dx = f(x,y) and y = y0 when x=x0
    double x0,y0;
    double xn,yn;
    double k1,k2,k3,k4, k;
    cout<<"Provide initial (x0,y0) whose differential equ is defined:";
     cin>>x0>>y0;
     cout<<"Enter the value of x for which you need to find out the y=?:";
     cin>>xn;

     double h = (xn-x0)/5;
     yn = y0;
     for(double xi = x0 ; xi < xn ; xi += h){
         k1 = h*f(xi , yn);
         k2 = h*f(xi + h/2, yn + k1/2);
         k3 = h*f(xi + h/2, yn + k2/2);
         k4 = h*f(xi + h/2, yn + k3);

         k = (k1 + 2*k2 + 2*k3 + k4)/6;

         yn = yn+k;
     }

     printf("Value of y at x=%f is %0.4f when h: %f\n",xn , yn, h);
    return 0;
}
```

- **Code Execution of RK4 for dy/dx = xy +1 , y(0)=1 at x = 0.1**

```
Provide initial (x0,y0) whose differential equ is defined:0 1
Enter the value of x for which you need to find out the y=?:0.1
Value of y at x=0.100000 is 1.1052 when h: 0.020000
```

Note: check the value of h, calculate the accuracy of the method and compare with other methods of finding the solution of the ODE.

Modified Euler's Method

posted Dec 28, 2017, 11:01 AM by Atul Rana **[updated 5 minutes ago]**

Description

Modified Euler's Method is just the extended version of Euler's Method to increase the accuracy of the approximate solution.

It's recommended for you to go through Euler's Method.

The approximate value of y_1 of y for x = x_1 = x_0 +h by Euler's method is given by

$y_1 = y_0 + hf(x_0, y_0)$

so in edition to attend more accuracy we repeat the below formula,

$y_1^{(1)} = y_0 + h/2[\ f(x_0, y_0) + f(x_1, y_1)\ ]$

$y_1^{(2)} = y_0 + h/2[\ f(x_0, y_0) + f(x_1, y_1^{(1)})]$

...

in pan and paper process continues untill two value coincides now we have value of y_1

$y_2 = y_1 + hf(x_1, y_1)$

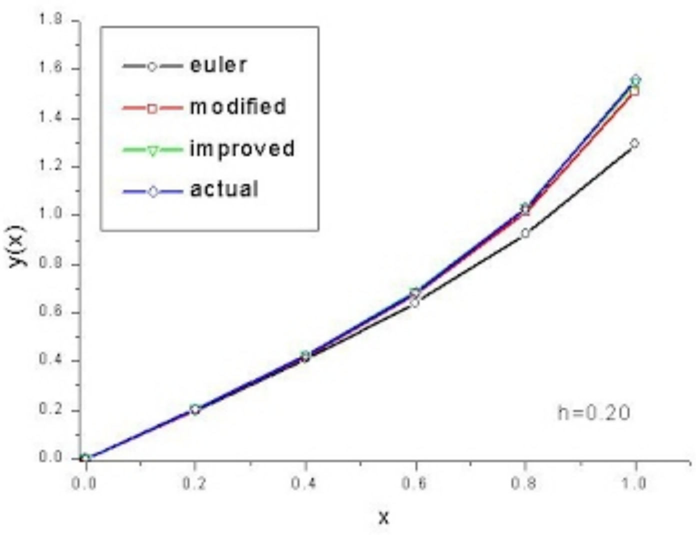
repeat the above process for y2

$y_2^{(1)} = y_1 + h/2[f(x_1, y_1) + f(x_2, y_2)]$

$y_2^{(2)} = y_1 + h/2[f(x_1, y_1) + f(x_2, y_2^{(1)})]$

...

- **Correctness**



- **Programming implementation of modified Euler's Method in C++**

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

double f(double x,double y){
    return x+y;  // dy/dx = x + y
}
double f1(double x,double y){
    return log2(x+y);
}
int main(){
    // COde for solving Ordinary differential equations
    //     dy/dx = f(x,y)
    //Define your equation

    //Given y(x0) = y0
    //Need to find y(xn) = yn
    double x0,y0;
    double xn,yn;

     cout<<"Provide initial (x0,y0) whose differential equ is defined:";
     cin>>x0>>y0;
     cout<<"Enter the value of x for which you need to find out the y=?:";
     cin>>xn;

     double h = (xn-x0)/5;
     yn = y0;
     for(double xi = x0;xi < xn; xi += h){
         cout<<xi<<" "<<yn<<endl;
         double YnPrevious = yn;

         yn = yn + h*f(xi , yn);
         for(int i=0;i<5;i++){
             yn = YnPrevious + h*( f(xi,YnPrevious) + f(xi+h,yn))/2;
         }
     //  Yn = Yn-1 + h*f(Xn-1,Yn-1)
     }

     printf("Value of y at x=%f is %0.4f when h: %f\n",xn , yn, h);
    return 0;
}
```

- **Code Execution for dy/dx = x + y ,y(0) =1 ,for x=0.3**

```
Provide initial (x0,y0) whose differential equ is defined:0 1
Enter the value of x for which you need to find out the y=?:0.3
0 1
0.06 1.06371
0.12 1.13507
0.18 1.21456
0.24 1.30268
Value of y at x=0.300000 is 1.4000 when h: 0.060000
```

Note: In code, h is defined constant depending upon interval size, but you can change as h cause accuracy difference, try out the code for other functions.

Euler's Method

posted Dec 28, 2017, 11:00 AM by Atul Rana **[updated 8 minutes ago]**

Description

Consider a defferential equation dy/dx = f(x,y) with initial condition y = y_0 when x = x_0, suppose we want to find y for x = x_n.

First we divide the interval [x_0,x_n] into n equal parts. x_0, x_1 = x_0 +h , x_2 = x_0 + 2h, .. , x_n where h is small. Using the equation of tangent at point (x_0, y_0)

$y - y_0 = dy/dx * (x - x_0)$ where $dy/dx = f(x_0,y_0)$

In the small interval [x_0, x_1] or [x_0, x_0+h] we approximate the solution curve by the tangent, So the approximate value of y_1 at point y_1 can be given by

$y_1 - y_0 = f(x_0, y_0)(x_0 + h - x_0)$

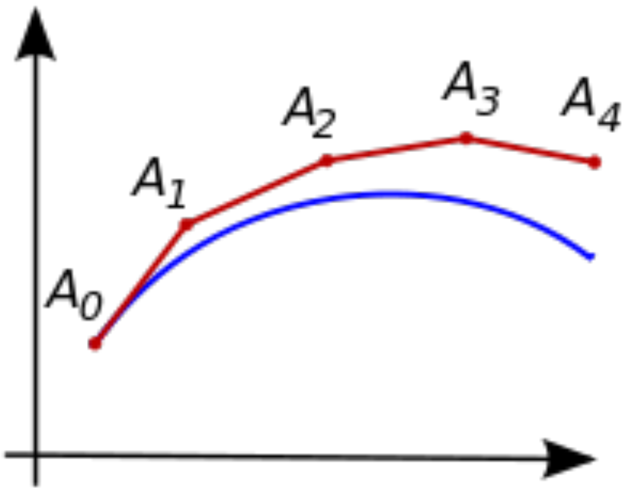
$y_1 = y_0 + hf(x_0, y_0)$

Similarly

$y_2 = y_1 + hf(x_1, y_1)$

...

$y_n = y_{n-1} + hf(x_{n-1}, y_{n-1})$



In the Euler's method we approximate the solution curve by tangent in each subinterval by a sequence of short lines and therefore unless h is very small the error is bound to be quite significant.

- **Programming implementation of Euler's Method in C++**

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

double f(double x,double y){
    return x+y;  // dy/dx = x + y
}
double f1(double x,double y){
    return log2(x+y);
}
int main(){
    // COde for solving Ordinary differential equations
    //     dy/dx = f(x,y)
    //Define your equation

    //Given y(x0) = y0
    //Need to find y(xn) = yn
    double x0,y0;
    double xn,yn;

     cout<<"Provide initial (x0,y0) whose differential equ is defined:";
     cin>>x0>>y0;
     cout<<"Enter the value of x for which you need to find out the y=?:";
     cin>>xn;

     double h = (xn-x0)/5;
     yn = y0;
     for(double xi = x0;xi < xn; xi += h){
         cout<<yn<<endl;
         yn = yn + h*f(xi , yn);
     //  Yn = Yn-1 + h*f(Xn-1,Yn-1)
     }

     printf("Value of y at x=%f is %0.4f when h: %f\n",xn , yn, h);
    return 0;
}
```

- **Code execution for dy/dx = x + y ,y(0) = 1 ,finding y for x = 0.1**

```
Provide initial (x0,y0) whose differential equ is defined:0 1
Enter the value of x for which you need to find out the y=?:0.1
1
1.02
1.0408
1.06242
1.08486
```

```
Value of y at x=0.100000 is 1.1082 when h: 0.020000
```

Note: In code their h is defined constant, change it if needed as accuracy completely dependent on the value of h.

1-3 of 3 ‹ ›