International Journal of Scientific & Engineering Research Volume 11, Issue 1, January-2020 1556
ISSN 2229-5518

Ode Solvers In MATLAB For Non-stiff Equations

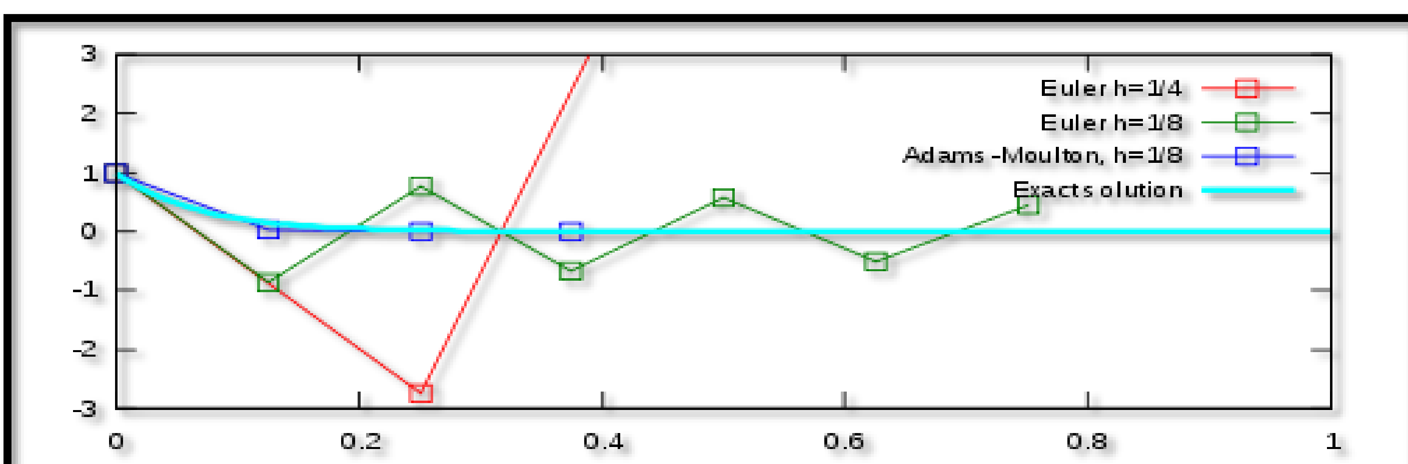
The work done by Raheela Bibi under the supervision of Maa'm Beenish Farhan in the institute of BPGCW. Ordinary differential equation model almost all of our daily life problems. There is an increase in finding ways and methods for solving higher order linear and nonlinear differential equations. Many researchers are trying hard to solve these problems for many years. Firstly, using the analytical approach, when they fail in solving them analytically they try to solve using numerical approaches. Some of the numerical methods are Cauchy Euler's equation, Runge Kutta method etc. Now these are the methods on which different computer solvers are working. One of the basic soft wares used to solve odes is MATLAB. MATLAB was developed in 20th century. MATLAB is one of the computer programming high level languages for technical computing. MATLAB is computer application in which we solve numerical problems for different differential equations. My work is based on detailed study on ode113 and ode45 solvers in MATLAB.

Differential equations are categorized into two basic types:

Stiff Differential Equation

Non-stiff Differential Equation

A **stiff equation** is a differential equation for which certain numerical methods for solving the equation are numerically unstable unless the step size is taken to be extremely small.



Explicit numerical methods exhibiting instability when integrating a stiff ordinary differential equation

Non-stiff problems are often solved by using explicit methods usually with some error control.

The solution to differential equations is classified into three approaches. The **analytic approach** seeks to provide an explicit solution to the differential equation. The **qualitative approach** yields solutions in form of direction fields, solution curves and phase plots. This method may be used to validate an analytic or numeric result. The **numerical method** provides approximate values of the solutions to the differential equation.

The most widely known member of the Runge–Kutta family is generally referred to as "RK4", the "**classic Runge–Kutta method**" or simply as "**the Runge–Kutta method**". The Runge-kutta method order formula is:

$$y(x+h)=y(x)=1/6(F_1+2F_2+2F_3+F_4)$$

where

$F1=hf(x,y)$

$F2=hf(x+h/2 , y+F_1/2)$

$F3=hf(x+h/2,y+F/2)$

$F4=hf(x+h , y+F3)$

Ode45 is a sophisticated built-in MATLAB function that gives very accurate solutions. Ode45 is based on a simultaneous implementation of an explicit fourth and fifth order **Runge-Kutta formula called the Dormand-Prince pair**. It is a one-step solver. This is the first solver to be tried for most problems. It is **designed for non-stiff problems**. Ode45 can use long step size and so the default is to compute solution values at four points equally spaced within the span of each natural step. **The Dormand-Prince pair is an explicit method and a member of the**

Runge-Kutta family of solvers. The method employs function evaluations to calculate fourth and fifth order accurate solutions. **The difference between these solutions is then taken to be the error of the fourth order solution**.

Adams methods are based on the idea of approximating the integrand with a polynomial within the interval **(t_n, t_{n+1})**. Using a **kth** order polynomial results in a **k+1th** order method. There are two types of Adams methods, **the explicit and the implicit types**. The explicit type is called the **Adams-Bash forth (AB) methods** and the implicit type is called the **Adams-Moulton (AM) methods.**

Ode113 is a multi-step variable order method which uses Adams–Bashforth–Moulton predictor correctors of order 1 to 13. It may be more efficient than ode45 at stringent tolerances and when the ODE problem is particularly expensive to evaluate. **It is designed for non-stiff problems.**

Euler numerical method can solve first order first degree differential equation with a given initial value. It is the most basic explicit method for numerical integration of ordinary differential equations and is the simplest **Runge-kutta method**. The Euler method is a first-order method, which means that the **local error (error per step) is proportional to the square of the step size**, and the **global error (error at a given time) is proportional to the step size**. The Euler method often serves as the basis to construct more complex methods, e.g., predictor-corrector method. The basic Euler's equation is:

$$ax^2y''+bxy'+cy=0$$

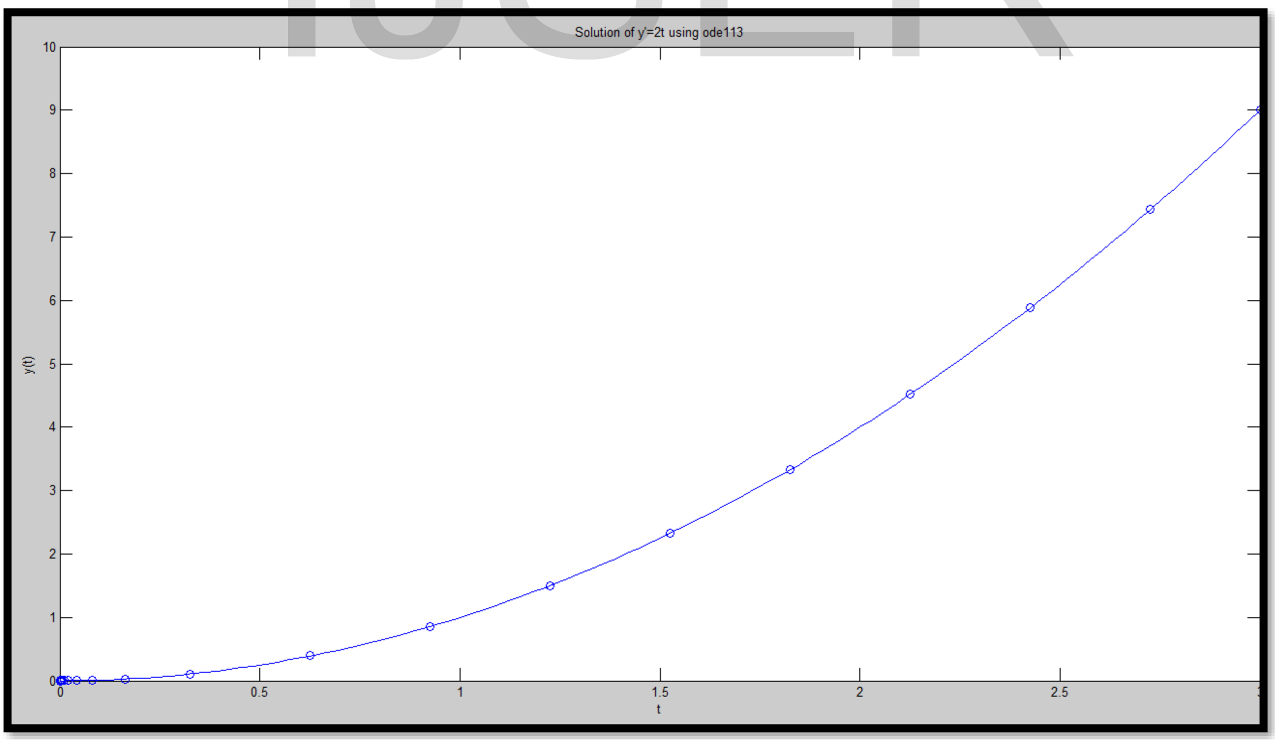
For example:

I consider here is the Euler's method using ode 45 and ode113 solvers on MATLAB analytically:

Code Using Ode 113 solver:

```
>> y = @(t) t.^2;
x = linspace(0,3);
figure
plot(x,y(x))
xlabel('t'), ylabel('y(t)')
hold on
[t,y] = ode113(@(t,y) 2*t, [0 3], 0);
plot(t,y,'o')
xlabel('t'), ylabel('y(t)')
title('Solution of y''=2t using ode113')
>>
```

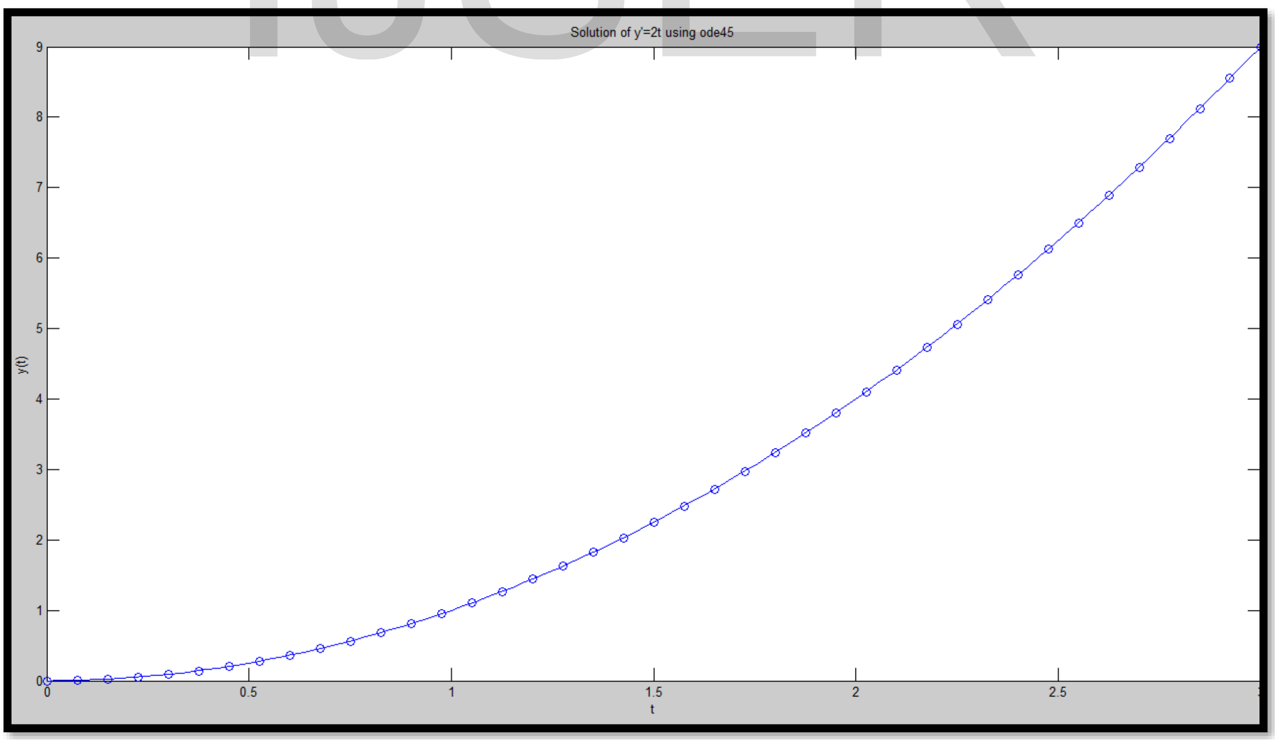

Result:



Code using ode 45 solver:

```
>> y = @(t) t.^2;
x = linspace(0,3);
figure
plot(x,y(x))
xlabel('t'), ylabel('y(t)')
hold on
[t,y] = ode45(@(t,y) 2*t, [0 3], 0);
plot(t,y,'o')
xlabel('t'), ylabel('y(t)')
title('Solution of y''=2t using ode45')
>>
```


Result:



Stats for ode113:

75 successful steps

0 failed attempts

451 function evaluations

Elapsed time is 0.011970 seconds.

Stats for ode45:

159 successful steps

3 failed attempts

322 function evaluations

Elapsed time is 0.018056 seconds.

Keywords:

Stiff differential equations, non-stiff differential equations, Runge-kutta Methods,

Adams-Bash forth-–Moulton predictor correctors, ode45 solver,ode113 solver, Euler's Method

About The Authors:

This abstract is conducted by **Raheela Bibi**. I am a student of **BS-Maths at BPGCW. Maa'm Beenish Farhan** is the supervisor of my this work. She is the **Head of the Mathematics Department in BPGCW**.