Exercise 2:
Modelling and Analysis Lab

MAYANK AHUJA(SC18B037)
Department of Aerospace Engineering, IIST
(Dated: September 28, 2020)

This report aims to understand different types of numerical integration algorithms used in commercial packages and learn about different implicit and explicit methods to solve first order ODE and system of ODE's. Also to understand the advantages and disadvantages of different numerical integration algorithms in terms of time and accuracy.

Several different techniques to solve the first order ODE, system of ODE's were studied. Some algorithms used in ABAQUS, ADAMS were learnt and their advantages and disadvantages were studied in terms of the compiling time as well as the accuracy of the solution and finally their efficiency was compared for different examples. To do this and prove the above arguments Python codes were written using the numpy and scipy libraries and the plots for numerical and analytical solution were plotted for better understanding. Also different implicit and explicit algorithms were studied.

In this report there are as much as ten different implicit and explicit algorithms which were studied and analysed using the comparison between analytical solution and numerical integration solution for each case. Techniques including Euler's, Range kutta methods, two step and four step Adams Bashforth method were used for solving ODE. Also the Van Der Pol oscillator method was used in solving system of ODE's. In all the above methods some random ODE's were taken to make the comparison between different solutions. Also some techniques for solving second order ODE's were studied

I. EULER'S METHOD

The Euler's method is based on the Taylor series expansion of a function f such as-

$$y = f(a + x) \tag{1}$$

Then according to the taylor series we get-

$$f(a + h) = f(a) + hf'(a) + \frac{h^2}{2!}f''(a) + ... \tag{2}$$

The approximation used with Euler's method is to take only the first two terms of the Taylor series:

$$f(a + h) = f(a) + hf'(a) \tag{3}$$

If

$$f(a + h) = y_{i+i} \tag{4}$$

and

$$f(a) = y_i \tag{5}$$

then:

$$y_{i+i} = y_i + hy_i' \tag{6}$$

I.1. Euler's Forward method

As from the above approach considering step length as

$$h = x_{i+1} - x_i \tag{7}$$

Therefore,

$$y_{i+i} = y_i + hf(x_i, y_i) \tag{8}$$

Implementing this algorithm on the following equation

$$\frac{dy}{dx} = 3(1 + x) - y \tag{9}$$

Given the initial conditions that x=1 at y=4, for the range x=1 to x=2 with intervals of 0.2. We have plotted the numerical solution as well as exact solution as shown below.

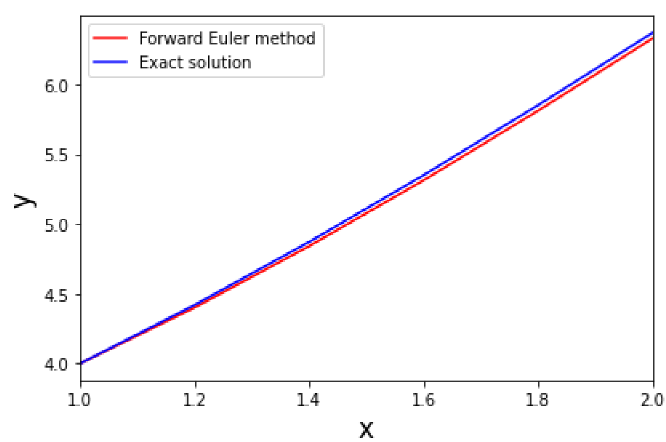

FIG. 1. Step length = 0.2

Now Implementing the same algorithm again but this time reducing the step size to 0.1. The error should decrease by increasing the number of iterations. But this

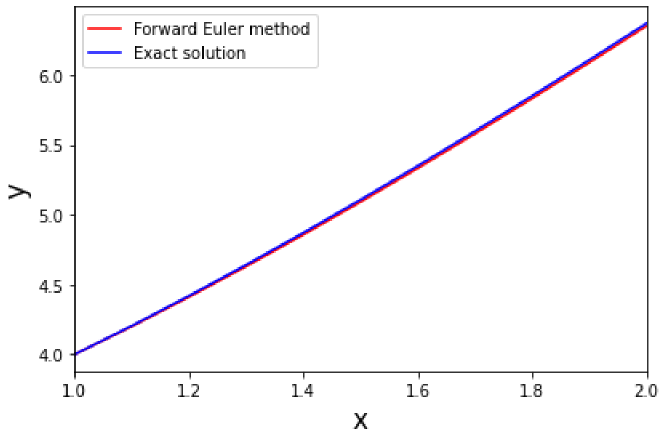

FIG. 2. Step length = 0.1

method takes more iterations for more accurate values and thus the time complexity of this code is the drawback of this algorithm.

Also another potential for improvement in this algorithm is by using the midpoint method. To predict the value of y at the midpoint of the interval:

$$y_{i+i} = y_i + hf(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}) \tag{10}$$

After applying this algorithm we get a much better solution without even reducing the step length. The forward

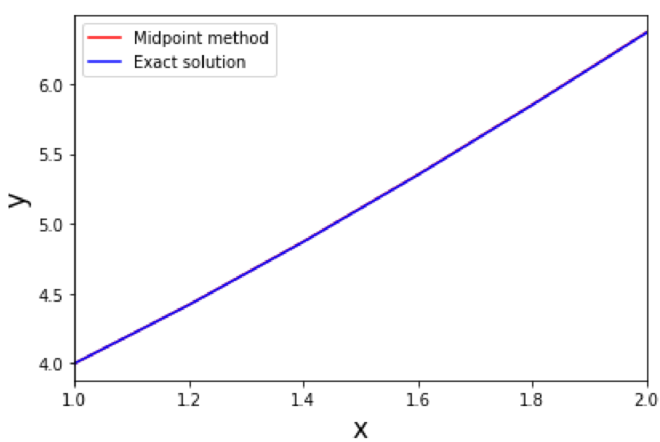

FIG. 3. Step length = 0.2

method we discussed above was an explicit method which is generally easier to implement but it's drawback arises from the limitation of step length to ensure numerical stability.

I.2. Euler's Backward Method

In this (implicit) we calculate the approximations using

$$y_{i+i} = y_i + hf(x_i, y_i) \tag{11}$$

The solution we get here is:

$$y_{i+i} = (\frac{1}{1+ah})^{i+1} y_0 \tag{12}$$

Implementing this algorithm on the following equation

$$\frac{dy}{dx} = 3(1+x) - y \tag{13}$$

Given the initial conditions that x=1 at y=4, for the range x=1 to x=2 with intervals of 0.2. We have plotted the numerical solution as well as exact solution as shown below. Now Implementing the same algorithm again but

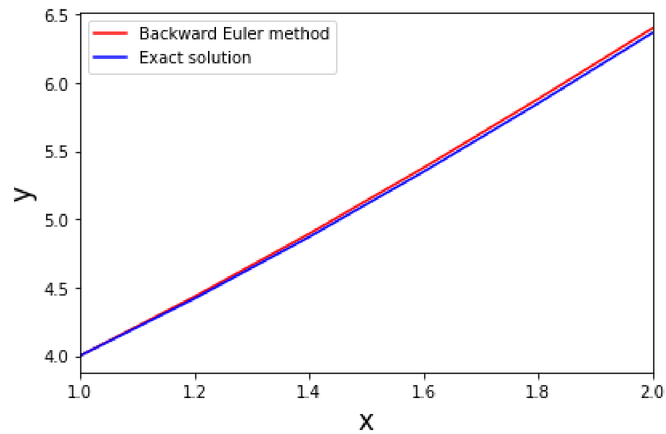

FIG. 4. Step length = 0.2

this time reducing the step size to 0.1. The error should decrease by increasing the number of iterations.

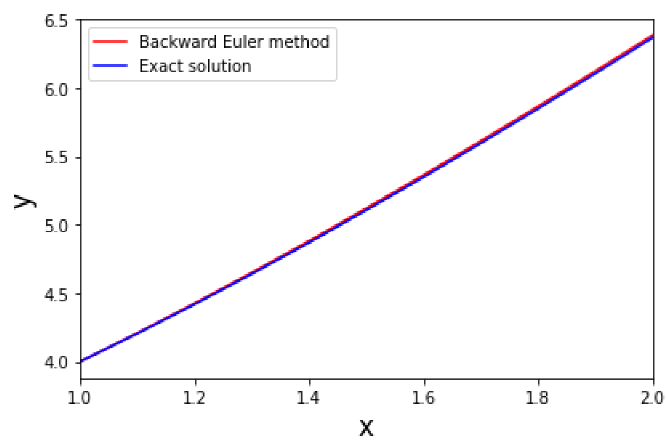

FIG. 5. Step length = 0.1

II. HEUN'S METHOD

The idea is to construct an algorithm to solve the IVP ODE such as:

$$y'(x) = f(x, y(x)) \tag{14}$$

over

$$[x_0, x_1] and y(x_0) = y_0. \tag{15}$$

In this Heun's method this equation is solved by derivation from integration and finally we get the equation:

$$y(x_1) = y(x_0) + \frac{h}{2}(f(x_0, y(x_0)) + f(x_1, y(x_1))) \tag{16}$$

Implementing this algorithm on the following equation

$$\frac{dy}{dx} = 3(1 + x) - y \tag{17}$$

Given the initial conditions that x=1 at y=4, for the range x=1 to x=2 with intervals of 0.2. We have plotted the numerical solution as well as exact solution as shown below. So in this we can see that step length of 0.4 in

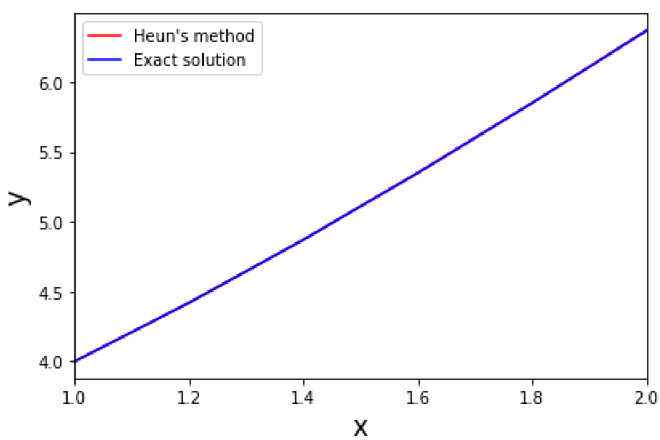

FIG. 6. Step length = 0.2

Heun's method is as good as step length of 0.2 in Euler's Forward method.

Now Implementing the same algorithm again but this time reducing the step size to 0.4. The error should increase by decreasing the number of iterations. Also we can solve the system of ODE's using this algorithm. Now solving a system of ODE's using this method. The ODE's we are solving are:

$$\frac{dx}{dt} = \frac{1 + 5x^2}{1 + x^2 + y} - x \tag{18}$$

and

$$\frac{dy}{dt} = 0.1(4x + y_0 - y) \tag{19}$$

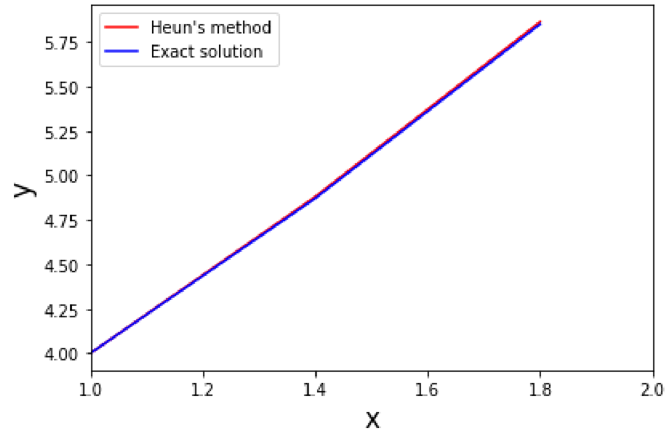

FIG. 7. Step length = 0.4

After solving the above system of ODE's, the plots we get are:

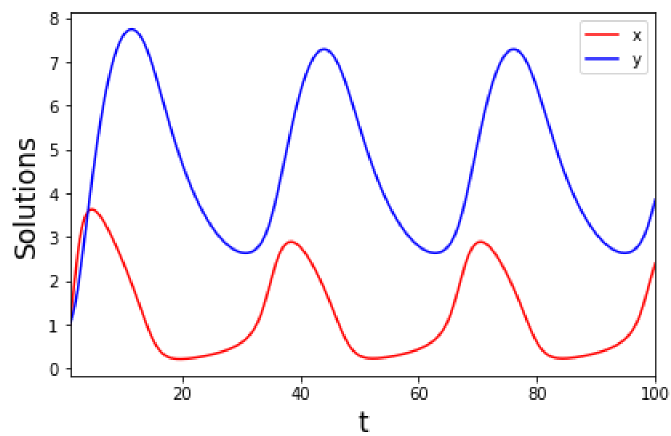

FIG. 8. Step length = 0.1

III. RANGE KUTTA METHODS

One of the most powerful predictor-corrector algorithms of all—one which is so accurate, that most computer packages designed to find numerical solutions for differential equations will use it by default— is the fourth order Runge-Kutta method. There are second order, third order and fourth order Range-Kutta methods, but we are going to discuss only about the third and fourth order Range-Kutta methods in this report as it is the most used in most of the commercial packages as well as it is the most accurate as far as our calculations are concerned.

III.1. Third order Range-Kutta Methods

The classical third-order Runge-Kutta method is given as:

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 4k_2 + k_3) \tag{20}$$

where

$$k_1 = f(x_i, y_i) \tag{21}$$

$$k_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1) \tag{22}$$

$$k_3 = f(x_i + h, y_i + 2hk_2 - hk_1) \tag{23}$$

Implementing this algorithm on the following equation

$$\frac{dy}{dx} = e^{-2x} - 2y \tag{24}$$

Given the initial conditions that x=1 at y=4, for the range x=1 to x=2 with intervals of 0.2. We have plotted the numerical solution as well as exact solution as shown below.

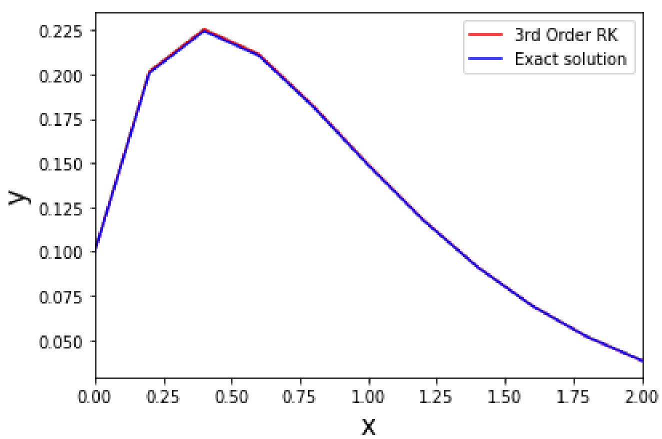

FIG. 9. Step length = 0.2

III.2. Fourth Order Range-Kutta Methods

The classical fourth-order Runge-Kutta method is given as:

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 3k_3 + 4k_4) \tag{25}$$

where

$$k_1 = f(x_i, y_i) \tag{26}$$

$$k_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1) \tag{27}$$

$$k_3 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_2) \tag{28}$$

$$k_4 = f(x_i + h, y_i + hk_3) \tag{29}$$

Implementing this algorithm on the following equation

$$\frac{dy}{dx} = e^{-2x} - 2y \tag{30}$$

Given the initial conditions that x=1 at y=4, for the range x=1 to x=2 with intervals of 0.2. We have plotted the numerical solution as well as exact solution as shown below. After executing the code in this method we get

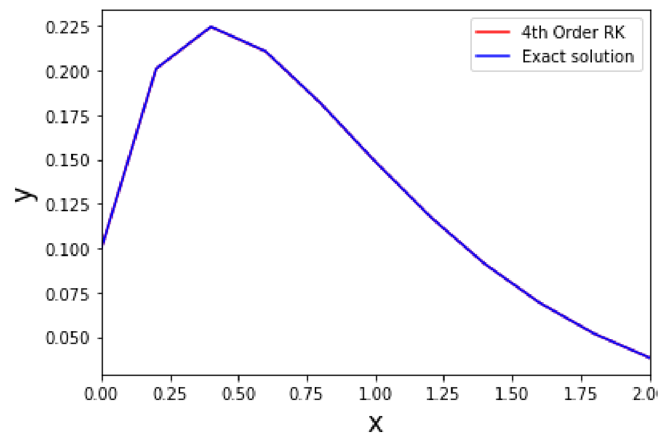

FIG. 10. Step length = 0.2

maximum difference as 7.524347035092749e-05. So this method is the most effective algorithm for numerical integral solutions.

Now solving a system of ODE's using this method. The ODE's we are solving are:

$$\frac{dy_1}{dt} = y_2 \tag{31}$$

and

$$\frac{dy_2}{dt} = (1 - y_1^2)y_2 - y_1 \tag{32}$$

Solving the above system of ODE's using the fourth order Range-Kutta Methods we get solution as shown in FIG.11.

IV. CONCLUSION

Most of the used Algorithms in the commercial packages such as ADAMS, ANASYS and ABAQUS rely on

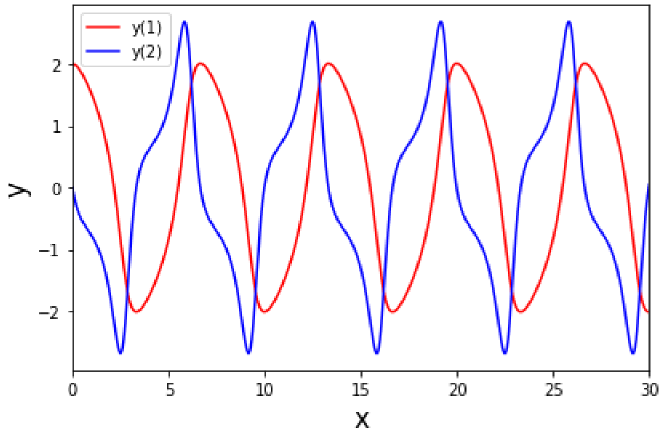

FIG. 11. Step length = 0.01

these algorithms to compute the numerical definite integral. Among the methods that we discussed the best and efficient way to compute the definite integral is the 4th order Range-Kutta Method. As we saw in that a big step length can also solve a definite integral with a maximum error coming in the order of 1e-5. The Euler's forward method is the implicit way but less efficient as it works in higher time complexity. The Euler's backward (Explicit) method is more efficient in those terms when we talk about the error and time complexity.

So, the implicit way to calculate a definite integral in some interval is generally more accurate but the only drawback we face there is of the time complexity of the codes when we try to implement them on the modern computers. In this report most of the integration we computed were on small intervals. But when we try the same thing on large intervals (in the order of 1e9 to 1e12), this won't run on any of the modern computers if we try the implicit way. That's why we say that the explicit ways are better methods to approximate the definite integrals for large intervals.

Maybe as the more and more supercomputers come, we'd be able to compute all the integrals in the large intervals in some near future with more accuracy.