Numerical Methods Assignment 3

Name: Benjamin Stott, ID:18336161,JS Physics

November 2020

1 Methods:

The aim of this task is to find the numerical solution of an ordinary differential equation. The equation in question is

$$f(X,t) = \frac{dx}{dt} = (1+t)X + 1 - 3t + t^2$$

A solution to this ODE can be found using many methods, three of which we shall focus on; the simple euler method, the improved euler method and the 4th order runge kutta method.
Simple euler:

$$X_{n+1} = X_n + hf(t_n, X_n)$$

where h= step size. This is very easy to understand, we have an initial point which moves depending on the slope at each point multiplied by step size in order to get a jump in the y value corresponding to this. This method, while intuitive is also crude and only gives a rough approximation.
Improved euler:

$$X_{n+1} = X_n + \frac{h}{2}(f(t_n, X_n) + f(t_{n+1}, Y_n + hf(t_n, X_n))$$

The improved euler method increases the accuracy of this method by taking the average of the slopes at point t_n and t_{n+1} before multiplying this by step size.The approximate slope for this step interval is closer to the true value of the curve than the simple euler method.
4th order Runge kutta method:

$$X_{n+1} = X_n + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = f(t_n, X_n)$$

$$k_2 = f(t_n + \frac{h}{2}, X_n + h\frac{k_1}{2})$$

$$k_3 = f(t_n + \frac{h}{2}, X_n + h\frac{k_2}{2})$$

$$k_4 = f(t_n + h, X_n + hk_3)$$

Essentially, what we are doing here is averaging four slopes to obtain an even more accurate approximate slope for an interval. Two endpoint slopes(k1 and k4) and two midpoint slopes(k2 and k3). Doing this once again refines the precision of the shape of the curve and brings it closer to the real ODE.

2 Results+Discussion

Figures 1 and 2 are a quiver and streamplot of the specified ODE. A quiverplot is a plot of the direction of the ODE for a given starting value X(on y axis here) as a function of t(on x-axis here). It is essentially many traces of the ODE for a large array of starting points overlaid onto each other, retaining only the directional information as small arrows. A streamplot is also included here to better visualise the solutions to this ODE. There is a very clear separation at around X=0.0655 from solutions that go to infinity above this and solutions that tend to negative infinity below this value.

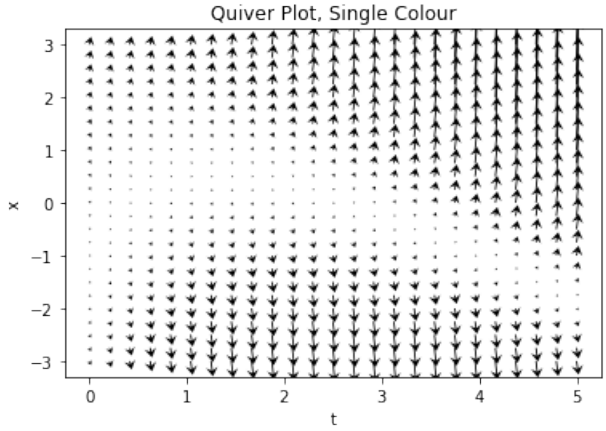

Figure 1:(quiver plot of specified ODE)

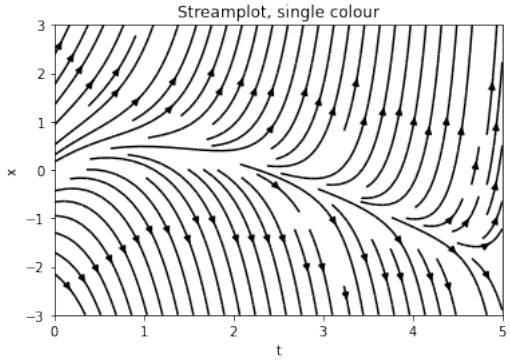

Figure 2:(Streamplot of ODE for better visualisation)

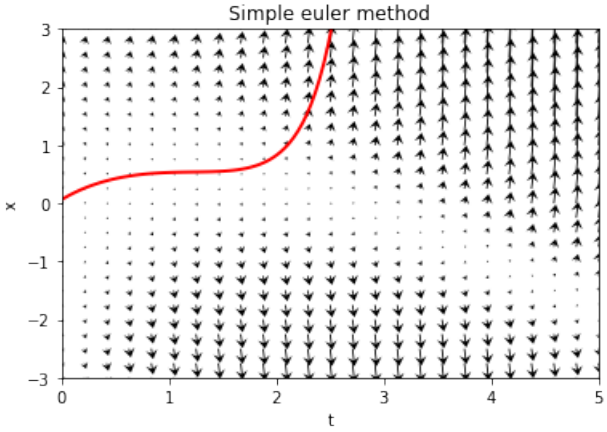

Figure 3:simple Euler method approximation for X initial = 0.0655, h=0.04

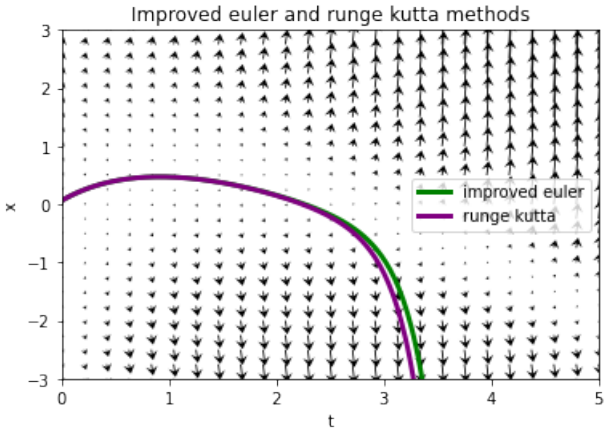

Figure 4:Improved Euler and Runge Kutta approximations for X initial=0.0655, h=0.02

We are attempting to plot the solution for the ODE at the separation point between upwards tending and downwards tending solutions. This means that any even slight error in the approximation may cause a solution to diverge greatly from the true value. The simple euler method tends to infinity while the other two methods tend to negative infinity.

The simple euler method, as said before, is crude and only returns an over-approximation of the real curve. As such it went past the critical point in this graph and went to positive infinity, diverging greatly from the real solution.

The improved euler and runge kutta methods both tend to negative infinity as they should. It can also be observed that the improved euler method is a very slight approximation from the difference between the green and purple curves above which makes sense since the Runge kutta method is the more accurate method.

After testing my simple euler for various step sizes, I found it only approximates

the true shape of the curve at step size h=0.001. In other words, the simple euler method requires 20 more iterations than the other two methods to reach a similar level of accuracy. This is inefficient and time consuming especially for larger scale programs.
In conclusion, more accurate integration methods of ODEs are essential, not just for numerical precision, but for ODEs which have very sensitive direction fields(i.e small error throws them off) to remain accurate and to do this in a manner that minimizes the amount of iterations, thereby remaining efficient.