Computer Simulation Assignment 4

Numerical Solution of an Ordinary Differential Equation

Prof. Stephan Hutzler

Alexandra Mulholland 17336557 04/11/2019

Aim and Abstract

The aim of this investigation was to analyse the result of three numerical methods, used to solve an ordinary differential equation (1) in terms of x and t, which is otherwise impossible to solve analytically. This was done so about a near point to the critical of the ODE, x(0)=0.0655. Comparing the Simple Euler, Improved Euler and Rung-Kutta methods about this point provides a good representation of which method is most reliable. As expected, the Runge-Kutta resulted in the most accurate result, regardless of step size. The Simple Euler method at step size 0.04 resulted in an entirely incorrect trajectory tending towards +∞. The improved Euler method at this step size produced a slightly advanced result but certainly not one which is representative of the system. Upon halving the step size, both methods showed significant improvement, tending towards -∞, with the Improved Euler result almost perfectly matching that of the Runge-Kutta. The Runge-Kutta was concluded to be the most representative approximation, involving higher order step size powers leading to smaller error and less sensitivity to being evaluated near the critical point of the ODE.

Mathematical Background

As is often the case, first order differential equations may not have any explicit solutions- since the differential equations are not linear, separable nor exact. This is the case with the first order, ordinary differential equation (1), which cannot be solved analytically.

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

In order to analyse the equation, however, a direction field may be plotted, which in this case is a plot of x(t) versus t. The direction field demonstrates a sketch of the solutions, whereby the arrows in the field are tangents to the actual solutions to the differential equation. Direction fields can depict the long-term behaviour of the solutions as the independent variable, t, increases. However, if one were to execute a detailed analysis of one of these differential equations, a direction field may not suffice. Numerical methods are used to approximate solutions to the equations, and thus provide a greater insight into the behaviour of said solutions. The methods utilised in this investigation were the Euler's method, the Improved Euler's method and the Runge-Kutta method.

Euler's method

Using the initial conditions given, one can insert the values for t and x into the differential equation to obtain a gradient for a line at this point. Using this, the initial value of x, and the constant step size (h=$\Delta t = t_{n+1} - t_n$) one can form an equation of a straight line, tangent to this initial solution. While this point is unlikely to lie on the function curve, an appropriate step size can ensure it is a close approximation. This next point is used to carry out the same method- constructing another line tangent to this point to the next approximation and so on. The general formula for this method is equation (2), where f_n is the equation for the ODE at the point (x_n, t_n).

$$x_{n+1} = x_n + hf_n \tag{2}$$

Improved Euler method

The original Euler method suffers from a truncation error upon further decrease of the step size. There is a substantial increase in the rounding error and so accuracy ultimately deteriorates. The improved Euler method utilises not only the tangent line to the original coordinate (x_0, t_0), but the tangent line to the next approximation (x_1, t_1). The average of the gradient of each line is determined and then run through the initial condition. This results in a much closer approximation to the actual value. This is iterated again as in the Euler method. The general formula for this numerical method is shown in equation (3), where f_n has the same meaning and equation (2) is shown in blue.

$$x_{n+1} = x_n + h\left[\frac{f_n + f(t_n + h, x_n + hf_n))}{2}\right] \tag{3}$$

Runge-Kutta

This is a fourth-order numerical approximation which, again, utilises the initial conditions as well as the recurrence formula. This is the superior of the methods as it uses more than just the function at the previous point, and the fourth-order aspect means that if we were to decrease the step size by just a half, then the error is reduced by a factor of 2^5. The general formula is shown in equations (4a- The recurrence formula) and (4b) below.

$$x_{n+1} = x_n + hT_4(t_n, x_n, h) \tag{4a}$$

$$T_4 = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{4b}$$

Where:

- $k_1 = f(t, x)$
- $k_2 = f\left(t + \frac{h}{2}, x + \frac{h}{2}k_1\right)$
- $k_3 = f\left(t + \frac{h}{2}, x + \frac{h}{2}k_2\right)$
- $k_4 = f(t + h, x + hk_3)$

Method

Firstly, the ODE (equation (1)) was defined within the code. A direction field was plotted within x ranging from -3 to 3 and t ranging from 0 to 5. A sequence of evenly spaced figures for x and t were plotted, in 25 steps using np.linspace. In order to plot the direction field, the function np.meshgrid() and subsequently plt.quiver() was used, creating a 25x25 grid point field.

Next, the simple Euler method was defined, beginning with the initial conditions $t_0 = 0$ and $x_0 = 0.0655$, with step size equal to 0.04. This initial value of x is very close to the critical value of approximately 0.065923 (explained later). Creating arrays within the ranges allows for the iterations to continue to a certain point. The result was then plotted atop the direction field.

```
n = int((end-start)/step)
t1 = np.arange(start,end,step)

seul = np.zeros(n)
ieul = np.zeros(n)
ruku = np.zeros(n)

seul[0] = X_zero
ieul[0] = X_zero
ruku[0] = X_zero

for i in range(1,n):
        seul[i] = seuler(t1[i-1], seul[i-1], step)
        ieul[i] = ieuler(t1[i-1], ieul[i-1], step)
        ruku[i] = rk(t1[i-1], ruku[i-1], step)
```

In part 3, the improved Euler method and Runge-Kutta method was defined, again plotted with the same parameters and use of arrays. The number of steps was defined for each, where this was the range of t divided by the step size, shown in the code above. Each of these methods was then plotted atop the direction field. The above process was then repeated using a step size value of 0.02, and the array parameters were redefined, utilising the new step size, which was called “step1”.

Results

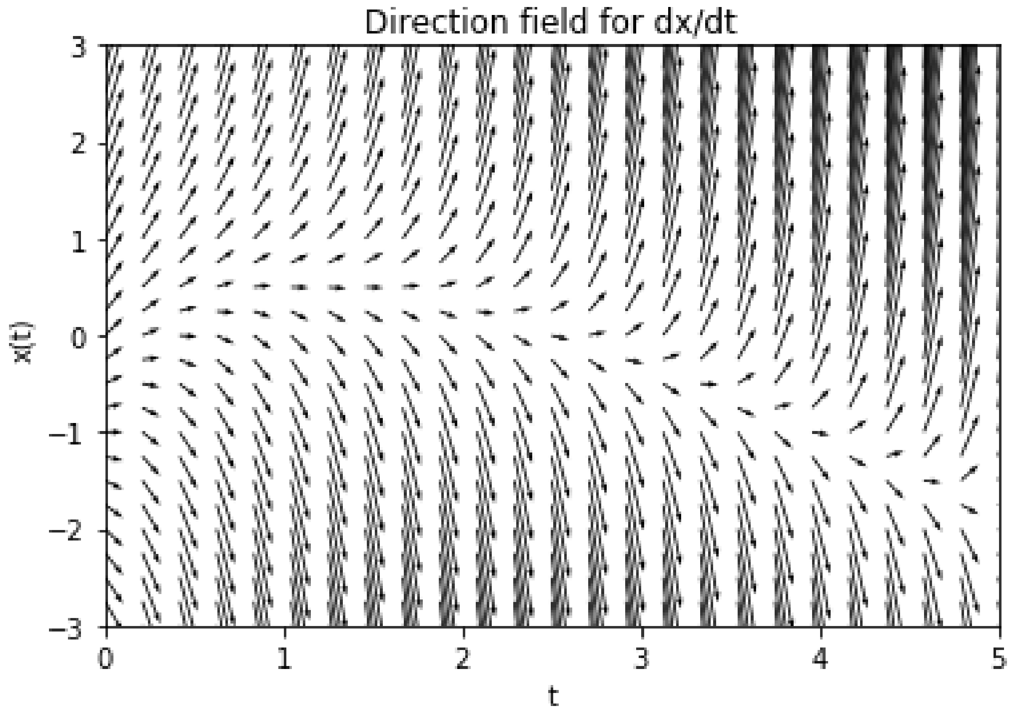

Figure 1
Direction field for the ODE described by equation (1), about the initial condition x(0)=0.0655

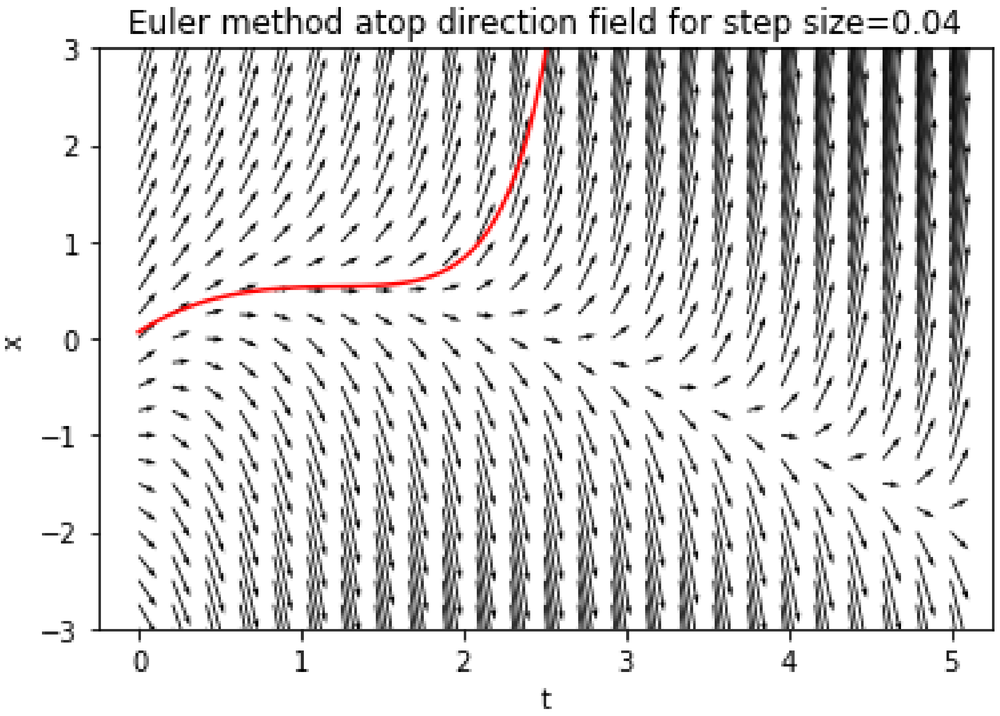

Figure 2

Simple Euler method with step=0.04 plotted against the direction field for the ODE described by equation (1)

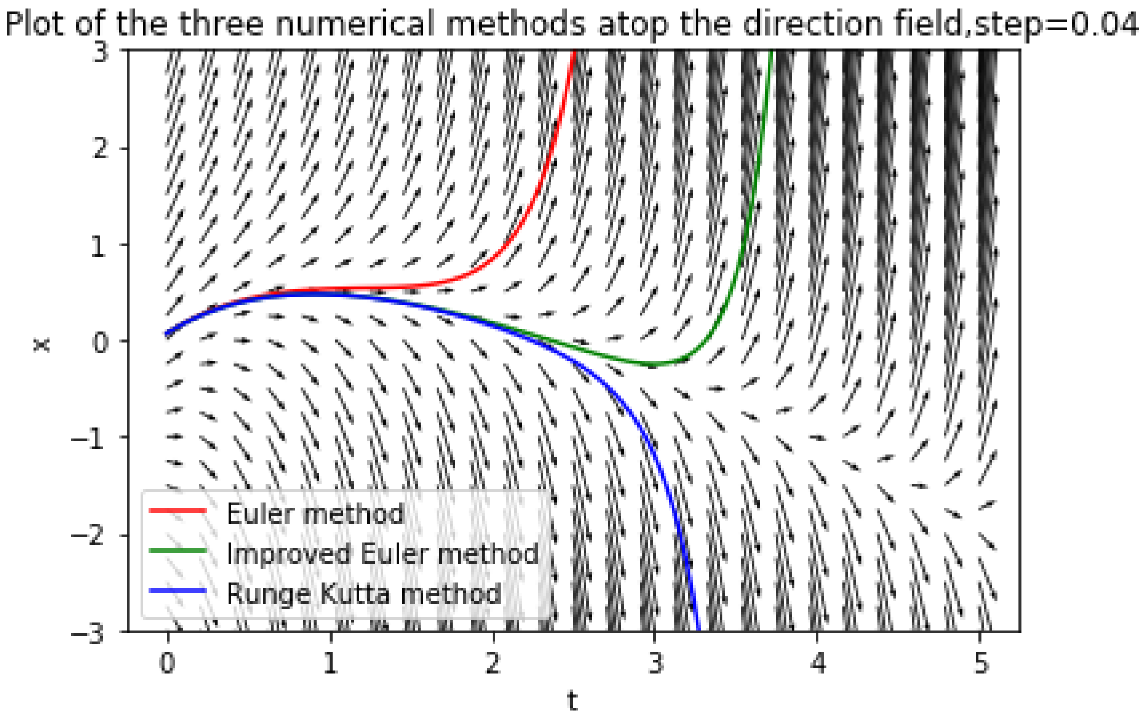

Figure 3.1

Simple Euler, improved Euler and Runge-Kutta methods with step=0.04, plotted against the direction field of the ODE

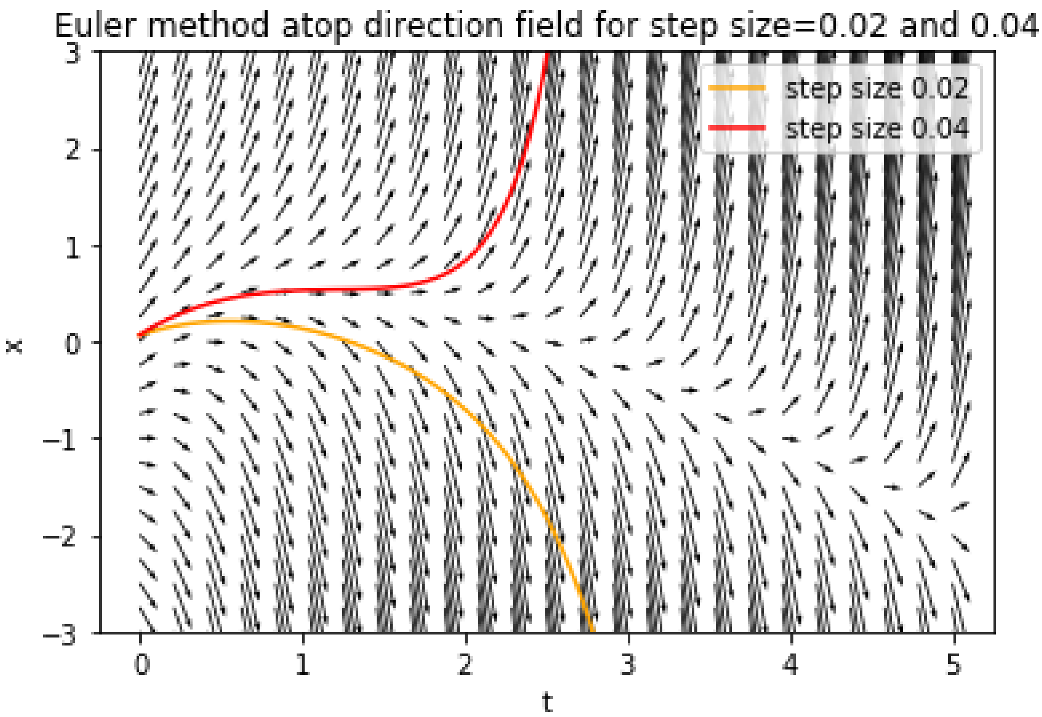

Figure 3.2a

Comparison of Simple Euler method when using step sizes h=0.04 and h=0.02.

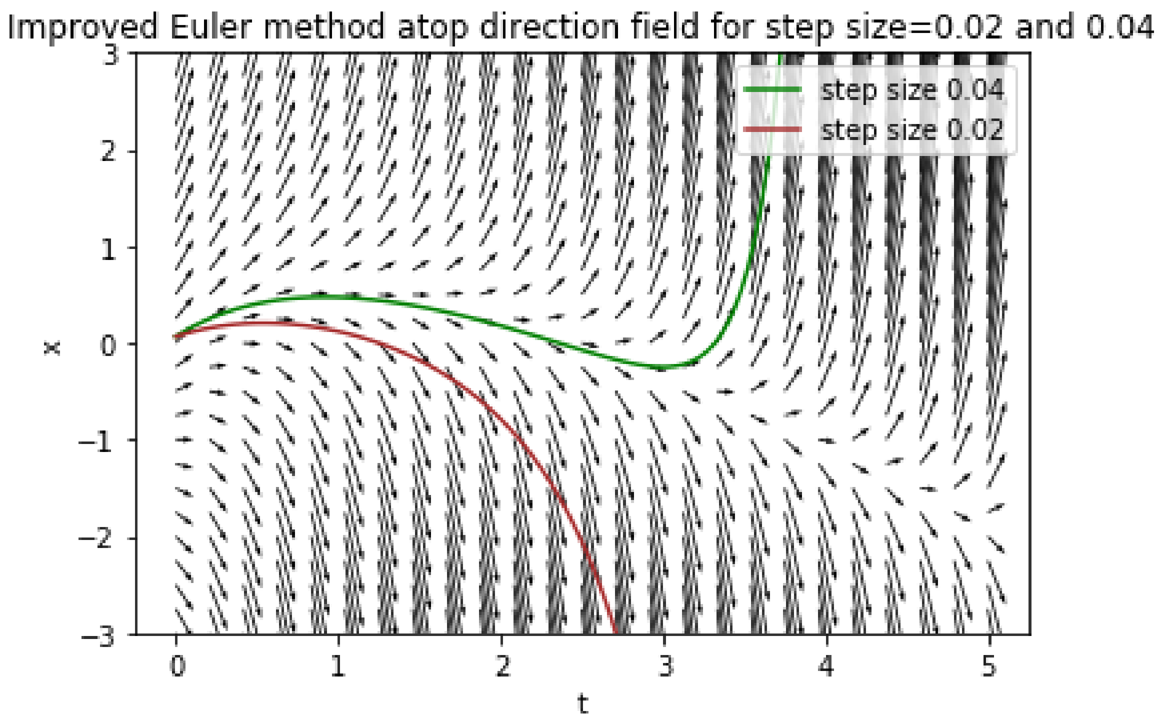

Figure 3.2b

Comparison of Improved Euler method when using step sizes h=0.04 and h=0.02.

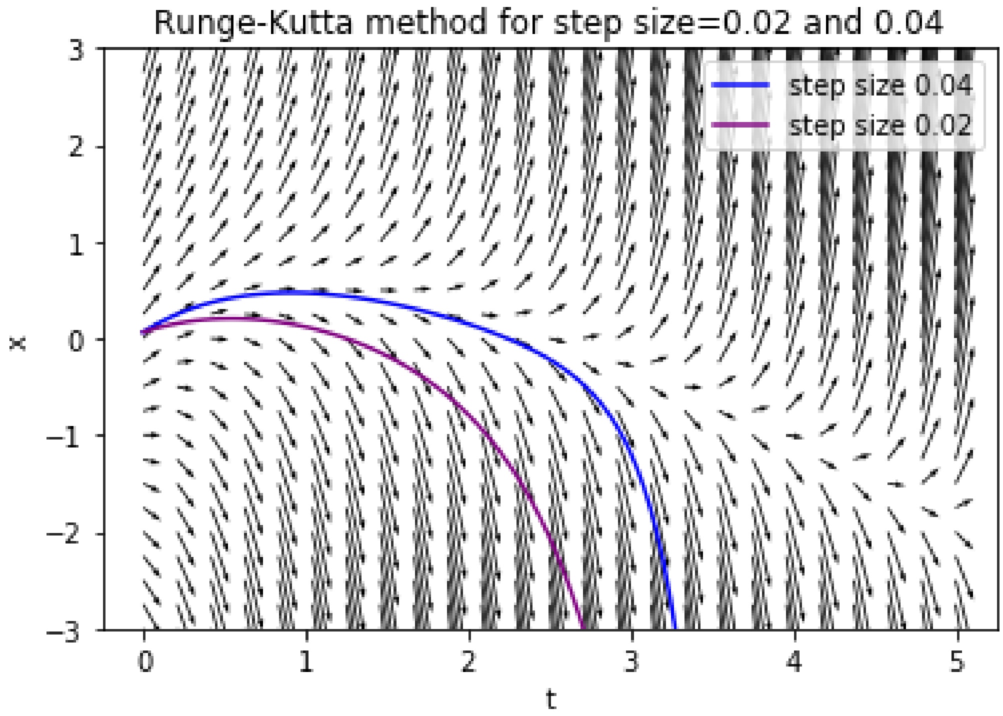

Figure 3.2b
Comparison of Runge-Kutta method when using step sizes h=0.04 and h=0.02.

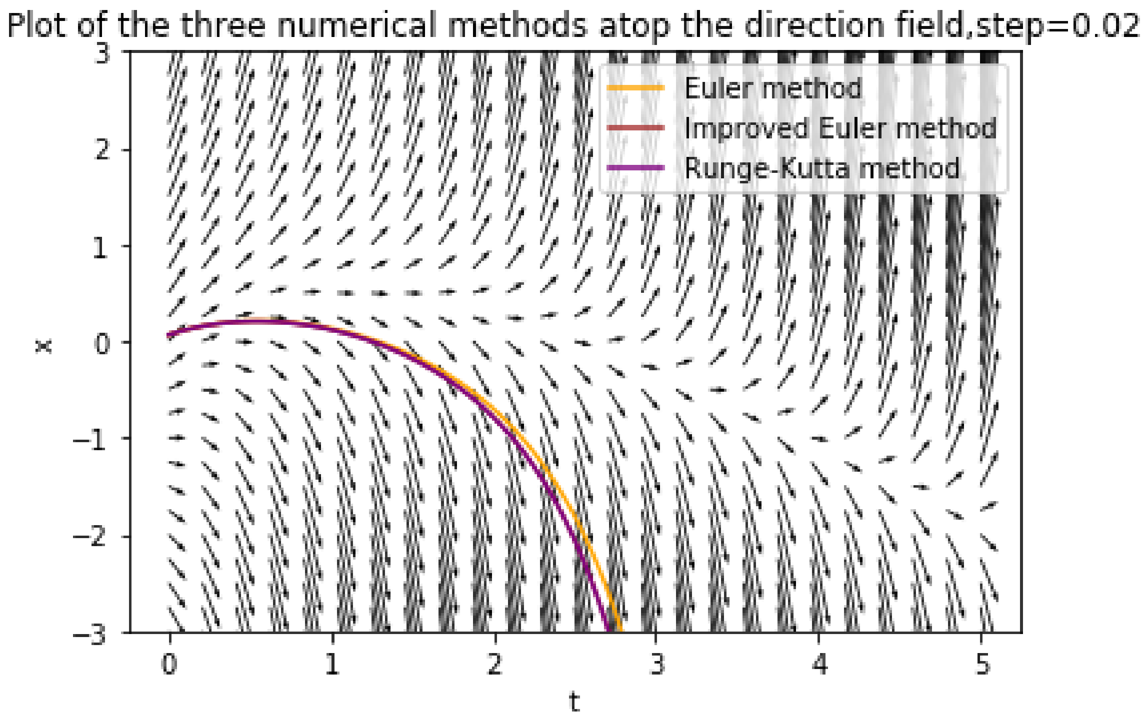

Figure 3.3
Plot of each numerical method using step size h=0.02

Discussion

Figure 1, a plot of the direction field for the ODE, demonstrates the ability to describe a function without explicitly knowing the solutions. The arrows shown in the field demonstrate the direction and gradient of the tangents to the actual solutions.

Figure 2 shows a plot of the trajectory of the Simple Euler method from the initial conditions of t=0 and x(t)=0.0655. Values higher than the critical value (0.065923…) result in trajectories of solutions to tend to +∞ and values lower result in trajectories tending to -∞ (in the x direction). Minor deviations about this critical point can therefore lead to extremely different results. This is demonstrated by the Euler method, whereby this numerical approximation drops higher power orders of the step size, and so the path tends to +∞ as opposed to -∞.

Figure 3.1 shows a comparison of all three numerical methods using a step size of 0.04. While the Improved Euler method is indeed an improvement upon the Simple method, it is an inferior approximation to the Runge-Kutta in this context, as it remains to tend to +∞. As predicted, the Runge-Kutta method results in the most accurate approximation for the trajectory of the three methods. The higher order power utilised by the Runge Kutta method provides this more accurate result.

Figure 3.2a, b and c provide comparisons of each numerical method at full and half step sizes. All three methods demonstrate improvements- this is as expected for each method with decreasing step size. However, it is common the Simple Euler method will demonstrate a truncation and increased rounding error should we continue to decrease the step size. On the other hand, the Runge-Kutta method will result in a decrease in error with decreasing step size- as shown.

This would imply that this result, as it is highly comparable with the result of the Improved Euler method (shown in figure 3.3), is extremely close to the true solution.

Testament to the accuracy of the Runge-Kutta regardless of the step size, this method seems to be the one of least improvement upon halving the step size (again suggesting the second result for the Runge-Kutta is the most accurate). Both Euler methods show significant advancement to the true solution upon halving the step size, providing evidence that this step size is not too small for the Simple Euler method.

Figure 3.3 shows a comparison of all three methods at step size=0.02. As described, all three show convergence to the true value, with a significant change in the Improved Euler method being made apparent by sitting on top the Runge-Kutta result. Furthermore, a significant change for the Simple Euler method, which now tends to -∞.

Conclusion

By comparing these three numerical methods at step sizes 0.02 and 0.04, it can be concluded that the Runge-Kutta approximation is the most accurate and reliable. It is not subject to extreme changes upon decreasing the step size and provides a more representative trajectory due to the continuous improvements at each iteration. The Euler method proved the least reliable. Although showing no evidence of rounding error upon decreasing the step size, was subject to an extreme alternation between the two step sizes. At h=0.04, the Euler method resulted in an approximation which tended to +∞, as did the Improved Euler method. Upon decreasing the step size, both results tended towards -∞, the correct result. However, the unreliable nature of these methods demonstrates how one should not conclude the initial results to be the definitive nature of the system, particularly when evaluating about the critical point of the ODE.