

Worksheet 5

Integration of Ordinary Differential Equations

Josh Wainwright
UID:1079596

Contents

| | | |
|----------|---|----------|
| 5 | Integration of Ordinary Differential Equations | 1 |
| 5.1 | Euler's Method | 1 |
| 5.2 | Runge-Kutta Method | 3 |
| 5.2.1 | Runge-Kutta 2nd Order | 3 |
| 5.2.2 | Runge-Kutta 4th Order - GSL | 5 |
| 5.2.3 | Adaptive Step Control | 6 |

5 Integration of Ordinary Differential Equations

Differential equations can be used to model a very wide range of physical systems. It takes just a small set of coupled differential equations to generate complex behaviour. However, most physical systems are not able to be integrated directly and so, when given a set of initial conditions, the equations are tackled numerically. Using these initial conditions, and observing changes in the solutions when the dependant variables are edited by small amounts, the behaviour can be inferred and approximate solutions estimated.

Only the special case of the ordinary differential equation shall be considered here, though with the expansion to partial differential equations, a greater number of physical situations can be modelled.

5.1 Euler's Method

A system of N th order coupled differential equations can be written as

$$\frac{dy_i}{dx} = f_i(x, y_1, y_2, y_3, \dots, y_N)$$

where $i = 1, 2, 3, \dots, N$. The aim from this equation is to find the solutions, y_i .

A differential equation can be written in terms of the small step change h as

$$\frac{dy}{dx}(x) = f(x, y) = \lim_{h \rightarrow 0} \frac{y(x+h) - y(x)}{h}$$

This can be written in terms of y_n to find an iterative formula to move from

$$y_0 = y(x_0)$$

to

$$y_1 = y(x_0 + h)$$

through an interval of length h as

$$y_1 = y_0 + hf(x_0, y_0) + O(h^2)$$

The error can be shown to be $O(h^2)$ by considering the expansion of $y(x_1)$ as shown below.

The Taylor expansion of $y(x)$ is given by

$$y(x_1) = y(a) + y'(a)(x - a) + \frac{y''(a)}{2!}(x - a)^2 + \frac{y'''(a)}{3!}(x - a)^3 + \dots$$

When $a = x_0$,

$$y(x_1) = y(x_0) + y'(x_0)(x - x_0) + \frac{y''(x_0)}{2}(x - x_0)^2 + \dots$$

The higher order terms can be truncated to give

$$y(x_1) = y(x_0) + y'(x_0)(x - x_0) + O(h^2)$$

$y(x_0) = y_0$ and the value of x_1 is given by the previous value, x_0 plus the step size h , i.e. $x_1 = x_0 + h$ and so

$$\begin{aligned} y(x_1) &= y_0 + f(x_0, y_0) \times h + O(h^2) \\ y(x_1) &= y_0 + hf(x_0, y_0) + O(h^2) \end{aligned}$$

This means that the error for a single iteration can be represented as $O(h^2)$. When this is applied to Euler's method for n iterations, the error can be written as

$$\begin{aligned} y(x_n) &= y(x_{n-1}) + \underbrace{hf(x_{n-1}, y_{n-1})}_{O(h)} + O(h^2) \\ y(x_n) &= y(x_{n-1}) + O(h) \end{aligned}$$

To demonstrate Euler's method, we shall consider the ordinary differential equation

$$\frac{dy}{dt} = 1 + y^2$$

with the boundary condition $y(0) = 0$. This equation can be solved analytically thus,

$$\begin{aligned} \frac{dy}{dt} &= 1 + y^2 \\ \frac{dy}{1 + y^2} &= dt \\ \int \frac{1}{1 + y^2} dy &= \int dt \\ \tan^{-1}(y) + C &= t \\ \therefore y(t) &= \tan(t - C) \end{aligned}$$

The boundary conditions mean that at $y(0) = 0$, $0 = \tan(C)$ and so

$$y(t) = \tan(t)$$

So for the given time at $t = \frac{\pi}{4}$, the solution is $y = \tan(\frac{\pi}{4}) = 1$. This can be solved relatively easily analytically by hand, but would be extremely difficult for a computer to solve analytically. An estimate can, however, be found using numerical methods, in this case Euler's Method. Figure 1 shows how the value of the estimate gets closer to the actual value, using a starting value of $y(0) = 0$.

The approximate error for this method can be calculated by considering the global error. The global error at a position, y , is defined to be $\frac{y - y_0}{h}$ which is proportional to $\frac{1}{h}$. We have already seen that the error for one iteration is proportional to h^2 , therefore, the approximate error for this method is $O(h^2) \times O(h^{-1}) = O(h)$.

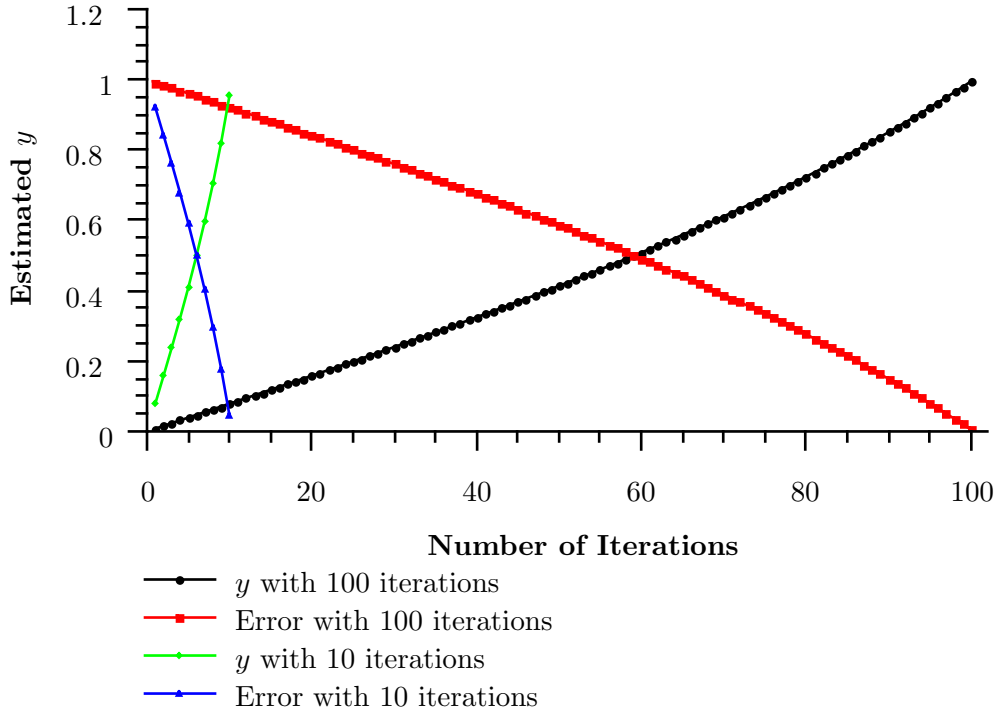


Figure 1: The estimated value of y up to $y(\frac{\pi}{4})$ is shown along with the error for both 10 and 100 iterations. As can be seen, the final error for the value of $y(\frac{\pi}{4})$ gets much closer to zero when the number of iterations is increased.

5.2 Runge-Kutta Method

An extension to Euler's Method for numerically solving differential equations is the Runge-Kutta Method. This is based on Euler's Method, but adds a step to average the gradient over the interval so that the error is much closer to the actual value. This average is taken from the midpoint of the interval where the gradient is taken again. This secondary gradient is then used, with Euler's Method as before, to reach the end of the interval. The effect of this intermediate gradient is to reduce the overshoot that Euler's method generates when the step size, h is finite. This can be expressed as

$$\begin{aligned}
 k_1 &= hf(x_n, y_n) \\
 k_2 &= hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \\
 y_{n+1} &= y_n + k_2 + O(h^3)
 \end{aligned}$$

and is shown diagrammatically in figure 2

5.2.1 Runge-Kutta 2nd Order

To demonstrate the Runge-Kutta Method, we shall consider the simple harmonic oscillator potential equation

$$V(x) = \frac{1}{2}x^2$$

For a physical system, the force equation is given by

$$\begin{aligned}
 F &= ma = -\frac{dV}{dx} \\
 m \frac{d^2x}{dt^2} &= -\frac{dV}{dx}
 \end{aligned}$$

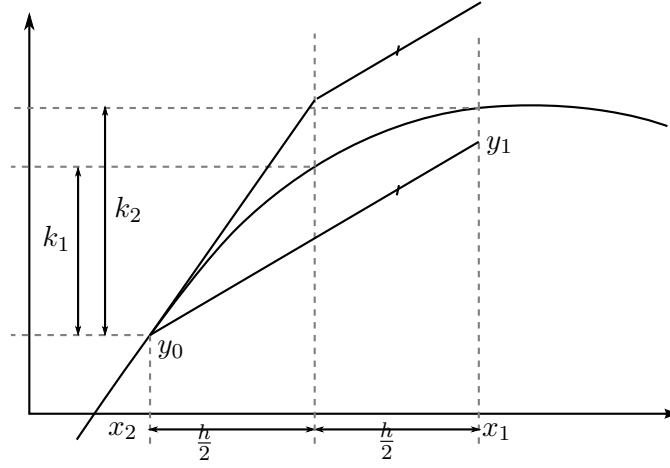


Figure 2: The Runge-Kutta Method can be used in place of Euler's Method to solve differential equations.

We know that $\frac{dV}{dx} = x$, so

$$m \frac{d^2x}{dt^2} = -x$$

This second order differential equation can be decomposed into a system of coupled first order equations by introducing another variable. We shall use $v = \frac{dx}{dt}$ so that now we have two equations linked by v .

$$\frac{dv}{dx} = -mx \qquad v = \frac{dx}{dt}$$

We shall assume that the mass, m is 1,

$$\frac{dv}{dx} = -x \qquad v = \frac{dx}{dt}$$

Again, this equation can be solved analytically, shown below, and will be used to calculate absolute errors.

The only solution that would remain the same after two differentials is,

$$\begin{aligned} x(t) &= C_1 i e^{ix} + C_1 i e^{-ix} + C_2 e^{ix} + C_2 e^{-ix} \\ x(t) &= C_1 \sin(t) + C_2 \cos(t) \end{aligned}$$

In order to compare, we shall set some initial values in order to determine the values of the constants. Let $x(0) = 1$,

$$\begin{aligned} 1 &= C_1 \sin(0) + C_2 \cos(0) \\ 1 &= C_2 \cos(0), \quad C_2 = 1 \end{aligned}$$

Also, we shall assume that the velocity at $t = 0$ is zero, so

$$\begin{aligned} v = \frac{dx}{dt} &= 0 = C_1 \cos(t) - \sin(t) \\ \therefore C_1 &= 1 \end{aligned}$$

Also, we shall assume that the velocity at $t = 0$ is zero, so

$$\begin{aligned} v = \frac{dx}{dt} &= 0 = C_1 \cos(t) - \sin(t) \\ \therefore C_1 &= 1 \\ \Rightarrow x &= \sin(t) + \cos(t) \end{aligned}$$

Solving this function analytically, again, is relatively easy, though not possible on a computer. As such we shall investigate how this can be solved numerically. We have already looked at Euler's method for solving differential equations, we shall now apply the Runge-Kutta method to this function.

A second order Runge-Kutta is used to solve this equation, using the same initial conditions as were used above. When this is performed, the data in figure 3 is calculated.

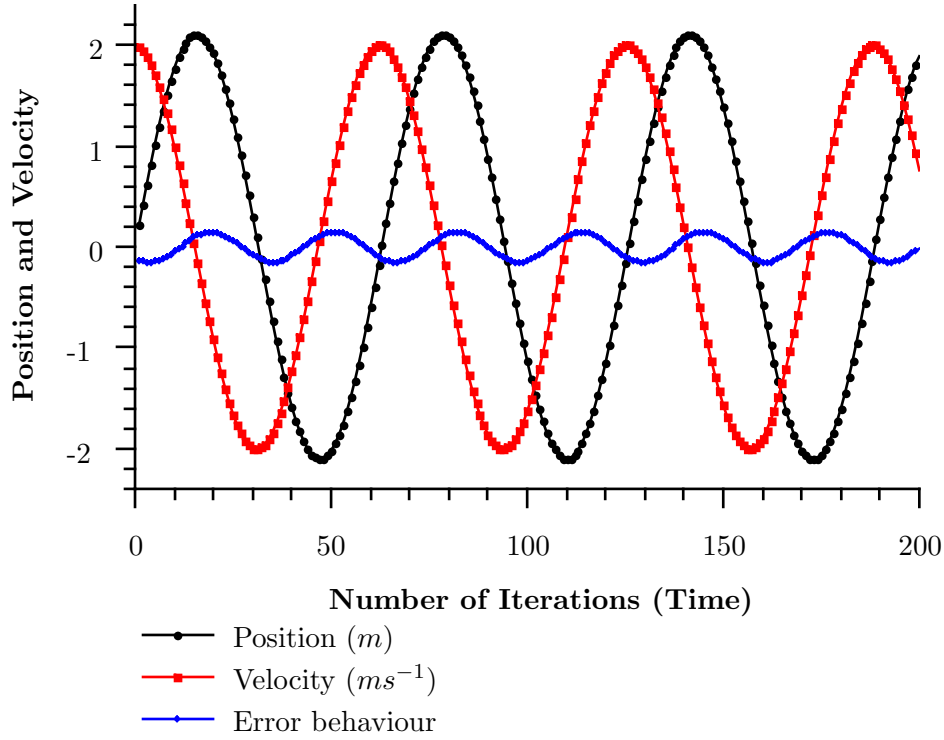


Figure 3: The plots of position and velocity against time with the error behaviour. A step size of 0.1 seconds was chosen, therefore this plot represents 20 seconds. As would be expected, the position lags the velocity by $\frac{\pi}{4}$.

The plot in 3 contains a representation of the error behaviour. This is calculated from the fact that the energy is conserved for the system. The energy is given by

$$E = \frac{p^2}{2} + \frac{x^2}{2}$$

Where p is the momentum. But we have defined the mass to be 1, so this can be written as

$$E = \frac{v^2}{2} + \frac{x^2}{2}$$

This means that if the Energy is plotted, it should be constant. For the data, the energy is plotted offset by the magnitude to show the error throughout the duration of the time calculated. For the second order Runge-Kutta method, figure 3, the error oscillates with the data. When the position is a maximum, the error is also maximum. This is because, at this turning point, the overshoot that is created by the finite step size is at the largest and so creates a larger error. The error is associated with the position and not the velocity simply because of the order that they are calculated in, meaning the position depends on the velocity calculation and so is less accurate.

5.2.2 Runge-Kutta 4th Order - GSL

The Runge-Kutta method is adaptable in that it can be used to generate data with smaller errors if desired. To do this, the order of the iterations is increased. Previously, a Runge-Kutta of order

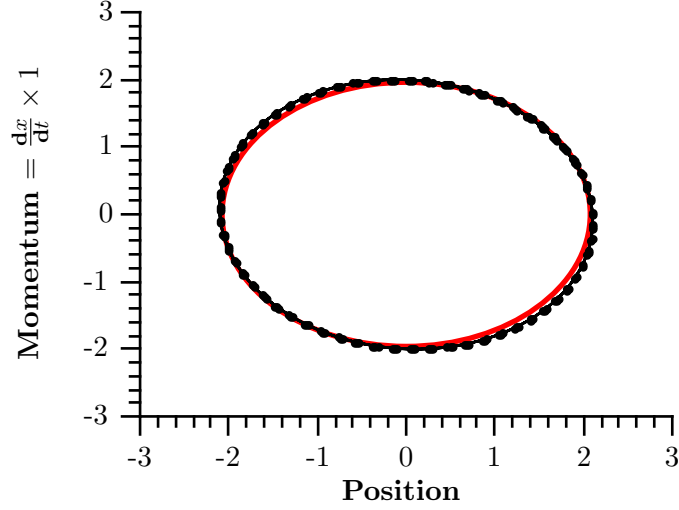


Figure 4: A phase plot, $p = \frac{dx}{dt}$ vs x , of the Runge-Kutta Method 2nd order. The errors can be seen as slight deformations in the shape.

2 was performed meaning that there we two steps for each iteration. If this is increased, then the calculated point for the next iteration will be closer to the actual value and so errors will propagate more slowly.

The SHO potential problem is re-calculated to take advantage of a higher order algorithm. The data in figure 5 is using a fourth order Runge-Kutta method. This means that for every iteration of the algorithm, 4 separate corrections are made to the estimate, each bringing that estimate closer to the true value.

By examining this new set of data, we can see that this is indeed more accurate. For example, due to the over shooting introduced by the errors in each step of the Runge-Kutta method, though the system starts at a maximum displacement of 1, the first system increases the displacement to a maximum of around 2.1. This is clearly incorrect since the system being modelled has no driving force and no resistive force, thus, it should oscillate from the starting point, to the same magnitude in the negative direction and back to the starting point but no further. The new data in figure 5 does not do this.

We can also see that this method increases the accuracy from the energy plot. Again, the energy is plotted as $E = \frac{v^2}{2} + \frac{x^2}{2}$ when the mass is 1. This time however, the energy is constant to a factor of around 10^{-7} .

5.2.3 Adaptive Step Control

Finally, using a dedicated algorithm, the accuracy of the Runge-Kutta method can be increased still further by replacing the slowly increasing iterative steps with step of varying sizes, the width of each being calculated by the complexity of the curve at that point, so as to reduce the error to as low as possible. This is shown in figure 7. The curve can be clearly seen to start at the initial starting point of 1 with very low error. This error is again confirmed to be small by using the energy of the system which is defined above. The variation of the energy from a constant value is representative of the error in the data and so is shown normalised to 0 in figure 7.

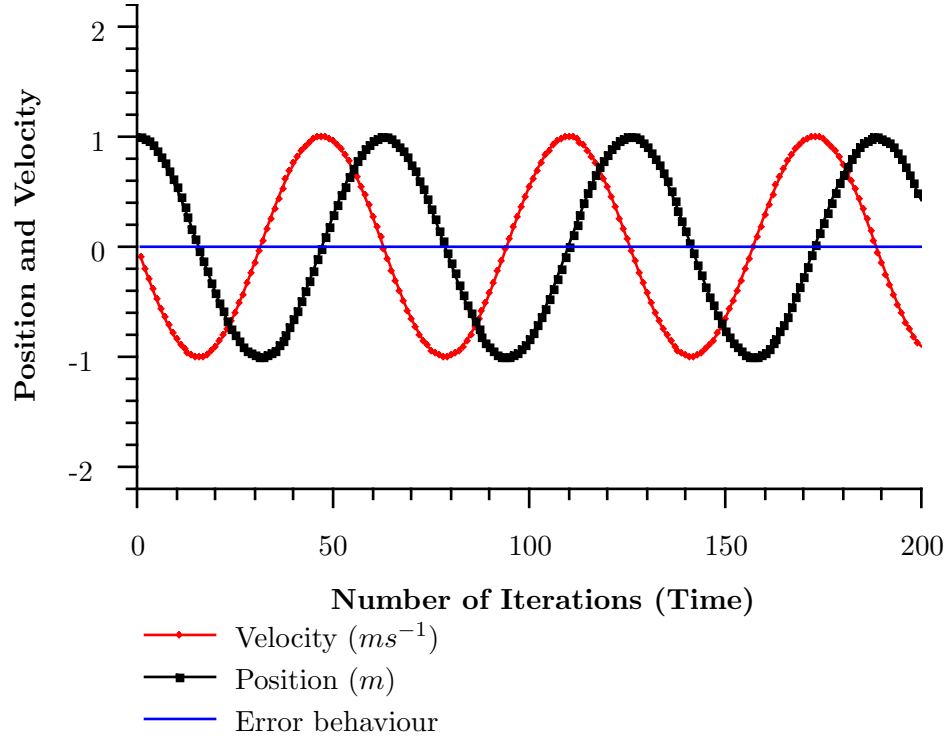


Figure 5: The plots of position and velocity against time with the error behaviour. A step size of 0.1 seconds was chosen, therefore this plot represents 20 seconds. As would be expected, the position lags the velocity by $\frac{\pi}{4}$.

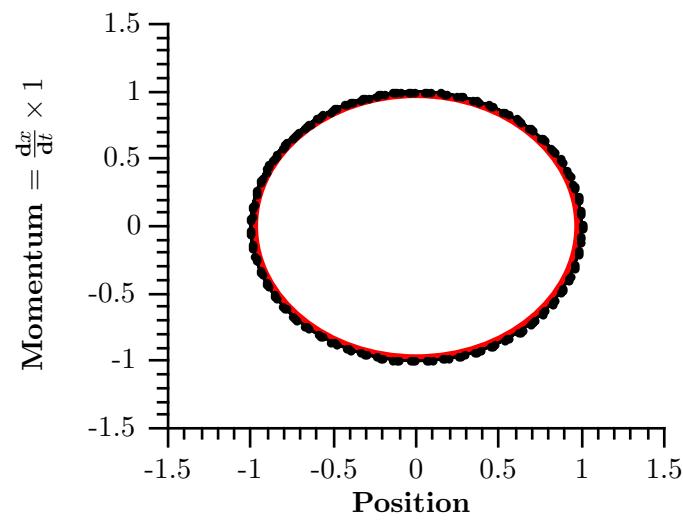


Figure 6: A phase plot, $p = \frac{dx}{dt}$ vs x , of the Runge-Kutta Method 4th order. This time the data exactly matches the ellipse showing that the error has reduced.

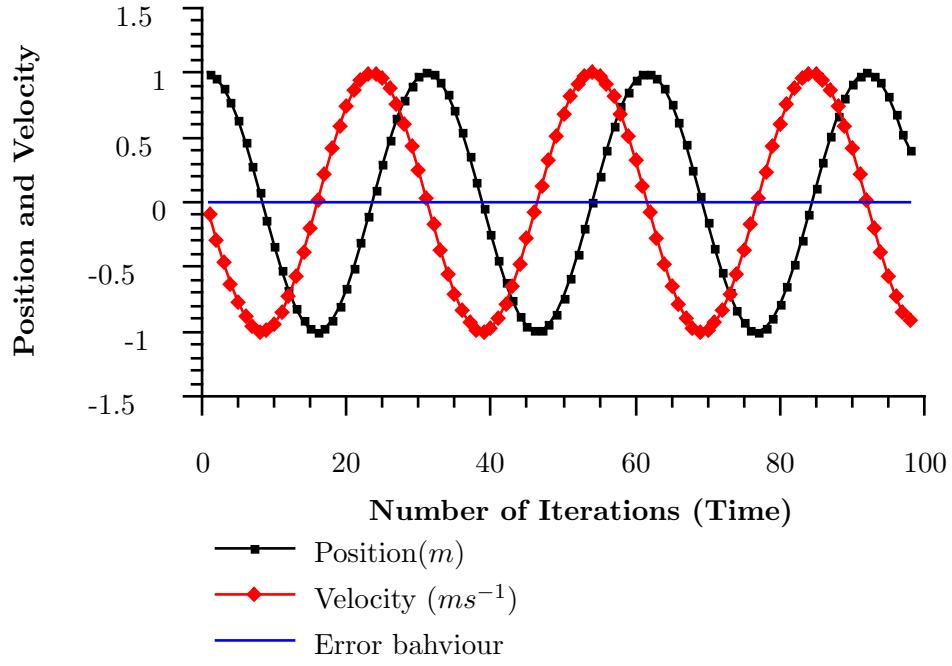


Figure 7: A plot of the data generated by the Runge-kUtta method when using adaptive step size control. Also shown is the error behaviour, again calculated from the energy. The energy is a constant value to an error calculated to be smaller than the error specified for the data.

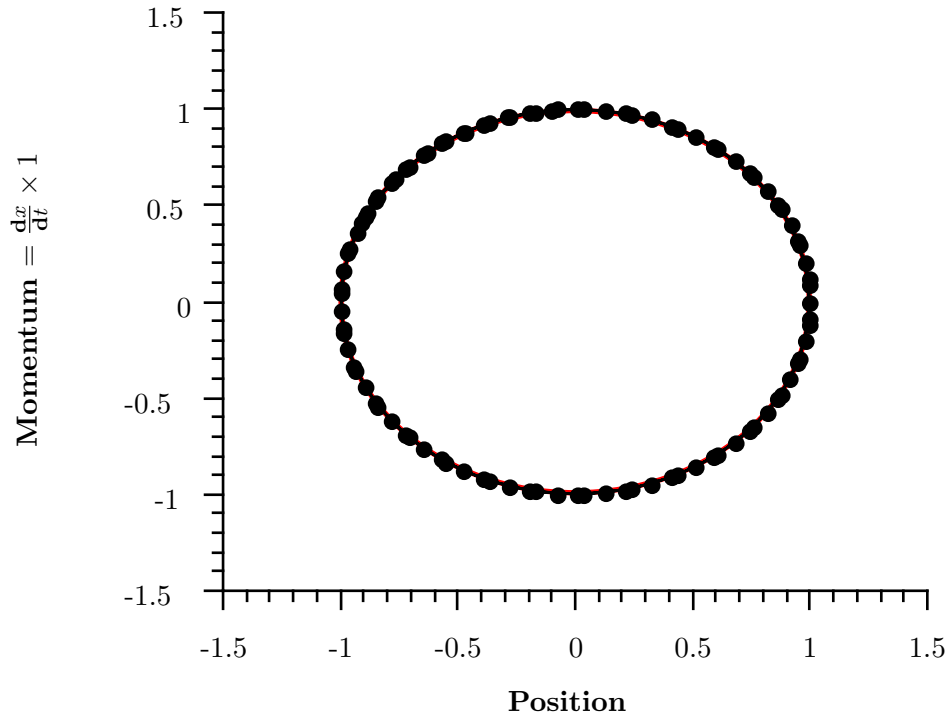


Figure 8: A phase plot, $p = \frac{dx}{dt}$ vs x , of the Runge-Kutta Method 4th order with adaptive step size. Again the data exactly matches the ellipse showing that the error has reduced.