James' intro to numeric methods

# 1 Euler's (an outline)

Euler's method is a process through which it is possible to 'project' the outcome of a curve via calculating its gradient at a point and extending from there. In general, for a curve described by

$$\frac{dy(x)}{dx} = f(x, y), \tag{1}$$

the Euler step can be found by rearranging this and the general equation for a derivative,

$$\frac{dy(x)}{dx} = lim\frac{y(x + dx) - y(x)}{dx}, \tag{2}$$

(calling the step size dx h) to give the equation for the next value of an equation, given the previous value:

$$y_{n+1} = y_n + h * f(x_n, y_n), \tag{3}$$

which is, of course, an approximation, the numerical. The exact solution's value for the 'next' value for y is $y(x_n + h)$, which can be found using a Taylor expansion of $y$ around point $x_n$:

$$y(x_n + h) = y_n + h\frac{dy(x)}{dx} + \frac{1}{2}h^2\frac{d^2y(x)}{dx^2} + O(h^3), \tag{4}$$

where $O(h^3)$ represents the values of this expansion with dependency on higher powers of $h$ than $h^2$. As such, we can find our local error by subtracting the numerical value from the actual; Subtracting (3) from (4) gives

$$y(x_n + h) - y_{n+1} = y_n + h\frac{dy(x)}{dx} + \frac{1}{2}h^2\frac{d^2y(x)}{dx^2} + O(h^3) - y_n + h\frac{dy(x)}{dx}. \tag{5}$$

Substituting (1) into (5) gives us

$$\text{Local Error} = y(x_n + h) - y_{n+1} = \frac{1}{2}h^2\frac{d^2y(x)}{dx^2} + O(h^3), \tag{6}$$

the right hand side of which can easily be seen to be $O(h^2)$. (This is a lot higher than the other methods we will see).

## 2 Runge Kutta

The second order Runge Kutta (also known as RK2) imporves Euler's method by 'averaging' the slope of the curve over the step, h. To do this we take the derivative of the curve at the beginning of the step, use Euler's method to find an rough value for y at the midpoint in the interval, and then take the derivative at this point and use Euler's method again to reach the end of the interval. Mathematically:

$$k1 = h * f(x_n, y_n) \tag{7}$$

$$k2 = h * f(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \tag{8}$$

$$y_{n+1} = y_n + k2 + O(h^3). \tag{9}$$

I've included a picture of this below.

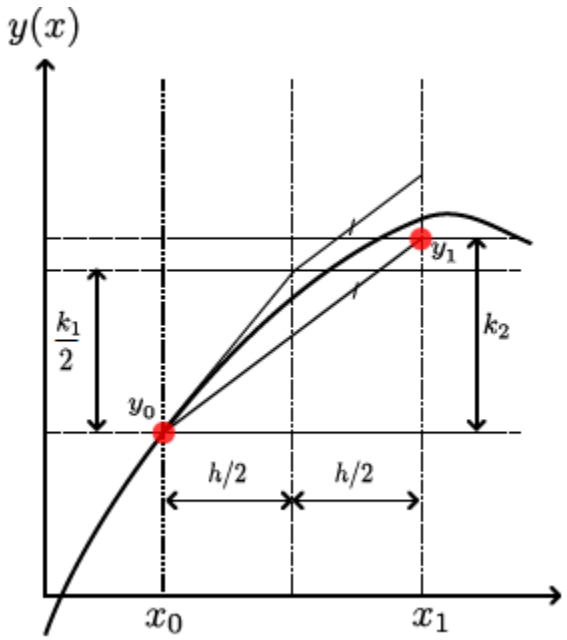


Figure 1: RK2 example over step h

Although each application of the Euler method generates errors of $O(h^2)$ you can show that these corrections at $O(h^2)$ cancel out, so the overall error is actually $O(h^3)$. . Higher order Runge Kutta methods are essentially the same idea as this but with more sub-intervals. There are many different ways of expanding $f(x, y)$ so that we can continue to use Euler's method for each sub-interval but successively higher order corrections in the different expansions from each sub-interval cancel each other out.

In order to look at the relative effectiveness of these methods, let's use them to solve a function with a known solution:

$$\frac{dy}{dt} = 1 + y^2 \tag{10}$$

by rearranging; dividing by $1 + y^2$ and multiplying $dt$, giving us the integral

$$\int \frac{1}{1+y^2} dy = \int dt. \tag{11}$$

We know that this gives

$$arctan(y) = t + c, \tag{12}$$

as such, $y = tan(t + c)$.Given that y(0) = 0, the constant is 0. Therefore, the solution of $y(\frac{\pi}{4})$ must be 1.

For 1000 iterations, the solution at $\pi/4$ is 0.9994542921848207 (pretty close to 1), with an error of 0.0005457078151791528. In addition, the error with only 10 steps is still less than 5 percent.

Belowis plotted a graph of the number of iterations, n, against the ratio of step size, h, to absolute error. This was in order to show that, not only were h and the error of the same order in 1 instance, they are both similarly proportional to n. This can be seen in Figure X, below. In this graph, we observe that as n increases, h becomes exponentially closer to the absolute error value; The error is O(h)

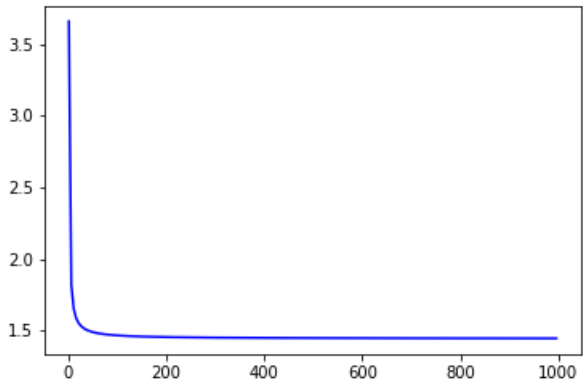


Figure 2: Plot of n against h/error

## 2.1 In regards to our project

In our simulation, we're going to have to somehow model an atom whose (acceleration) incident force is velocity dependent, a.k.a. a coupled set of equations. This makes finding a good equation to describe its velocity and position difficult, especially when we add in a position dependence to detuning like in a Zeeman slower. Because of this, an iterative method, like Euler's or Runge Kutta is helpful. Let's imagine that we have a SHO, but with all important constants equal to 1:

$$V(x) = \frac{1}{2}x^2 \tag{13}$$

The second order D.E. force equation gives us

$$-\frac{dU}{dx} = F(x), \tag{14}$$

so in this case,

$$F(x) = -x. \tag{15}$$

Knowing that F=ma, we can rewrite this as

$$F(x) = -x = \frac{d^2x}{dt^2} = \frac{dv}{dt}. \tag{16}$$

For the sake of our Runga Kutta method, our two first-order equations are thus:

$$1)\frac{dx(v,t)}{dt} = v \tag{17}$$

$$2)\frac{dv(x,t)}{dt} = -x \tag{18}$$

We know that our solutions should look like $x = cos(t + \phi)$ and $y = sin(t + \phi)$.

## 2.2 RK2

Coupling the two equations in the same loop, and assuming that phi = 0, that the initial condition is at rest but at maximum displacement, allows us to use the RK method to solve this second order ODE, and I've plotted below the functions x and v over the range 0 to 2pi, demonstrating the correctness of the approximation:

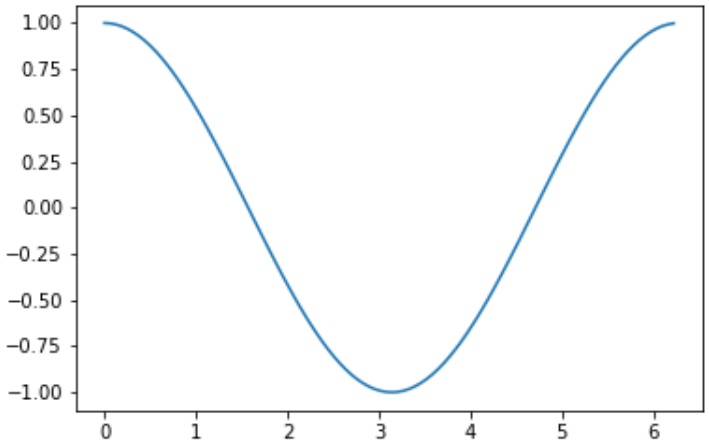


Figure 3: Plot of x against t

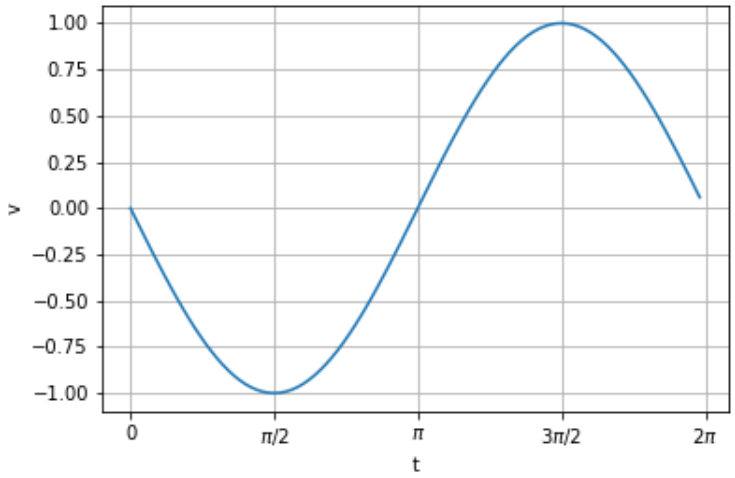


Figure 4: Plot of v against t

It is important to note that ( since we've got mass=1), velocity and momentum are equal. The Phase space Plot, momentum against displacement, for the RK2 method can be seen below.

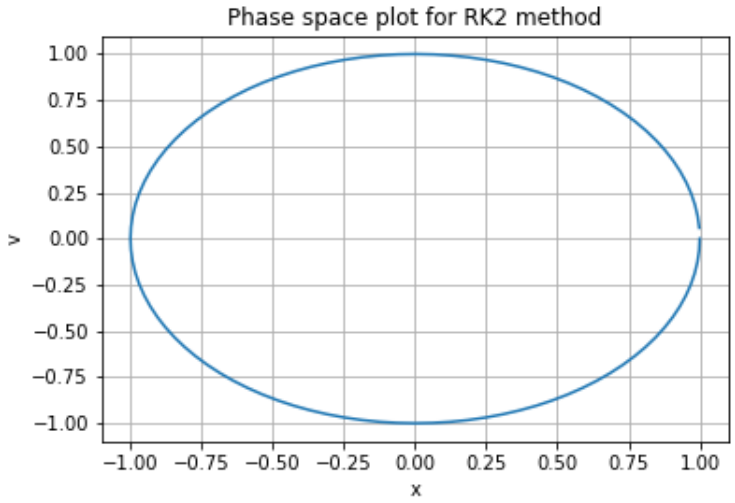


Figure 5: Phase space plot for RK2

Knowing that the energy of the SHO is constrained to

$$E = \frac{p^2}{2} + \frac{x^2}{2}, \tag{19}$$

and that for a value dependent on 2 other variables,

$$\delta E = \sqrt{(\frac{dE}{dp})^2 dp^2 + (\frac{dE}{dx})^2 dx^2}, \tag{20}$$

we can easily calculate the error on E throughout the process, which is show below to oscillate with time between 0 and 2, at twice the frequency of x and v.

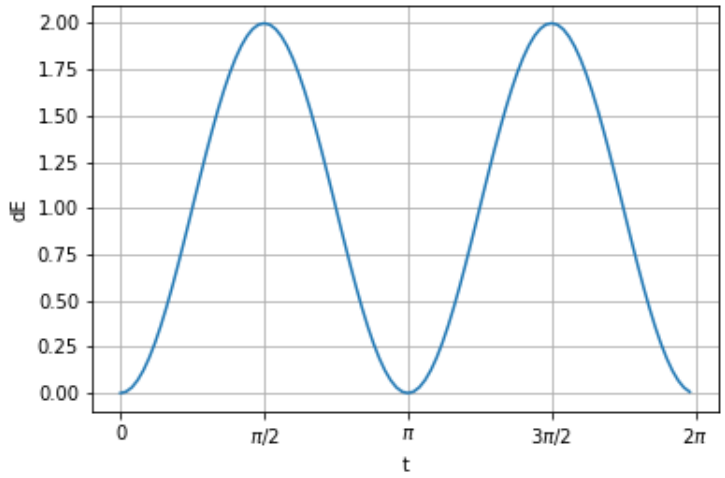


Figure 6: Error of the total energy with respect to time

A good way to see the error in the method is to plot a phase space plot over many cycles. Given that energy is conserved (and is the sum of the squares of the momentum and displacement), we would expect to see the plot form a circle, given that $E = \frac{x^2}{2} + \frac{p^2}{z}$. When we set the final time to $20\pi$ and do not change the number of steps, we observe a spiral towards the centre, clearly showing energy not being conserved in this RK2 method. We can see that the error in the model will contribute to this 'energy loss', which isn't appropriate for our sim.

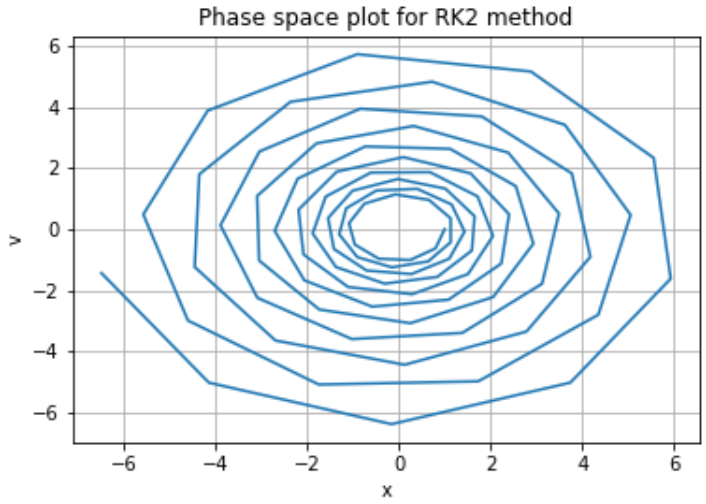


Figure 7: demonstration of error dE

## 2.3 RK4

The higher-order Runge Kutta 4 method is known to have error proportional to $O(h^5)$. And here we will demonstrate that it is sufficiently better that RK2 to be viable. The equations are:

$$
\begin{aligned}
k_1 &= h\, f(t_n, y_n), \\
k_2 &= h\, f\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right), \\
k_3 &= h\, f\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right), \\
k_4 &= h\, f\left(t_n + h, y_n + k_3\right).
\end{aligned}
$$

Figure 8:

The routine was shown to produce the same solution as our 'manual' method earlier, and still in agreement with the analytical solutions, as seen below.

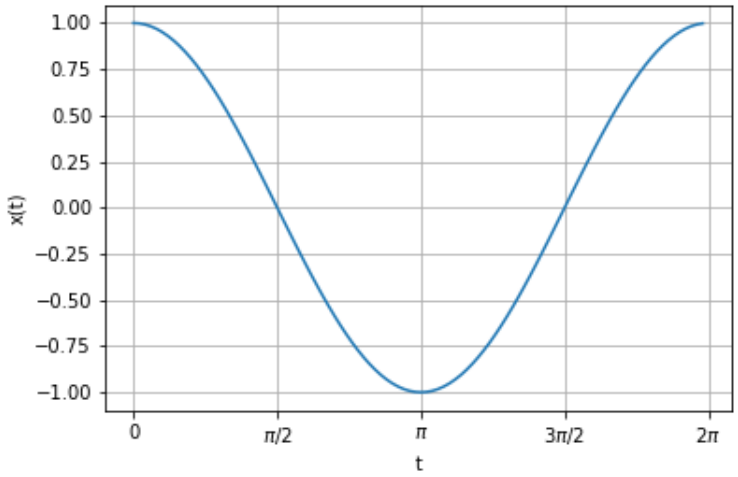


Figure 9: displacement with respect to time

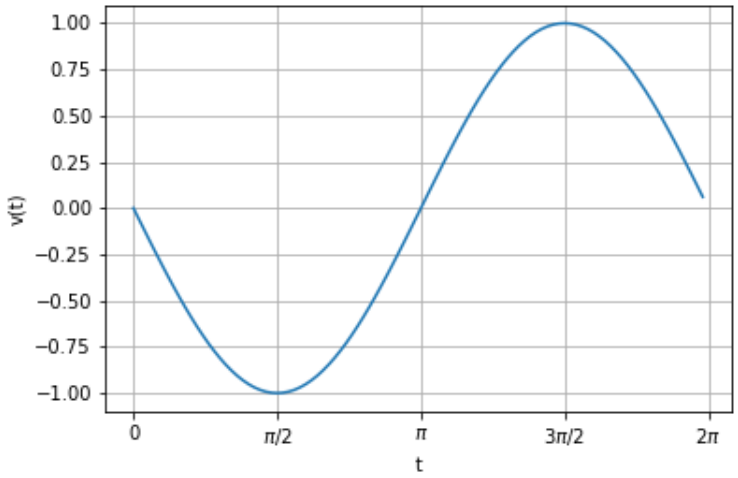


Figure 10: velocity with respect to time

The Phase space plot for 1 cycle is seen below, demonstrating that energy appears to be conserved over 1 period.

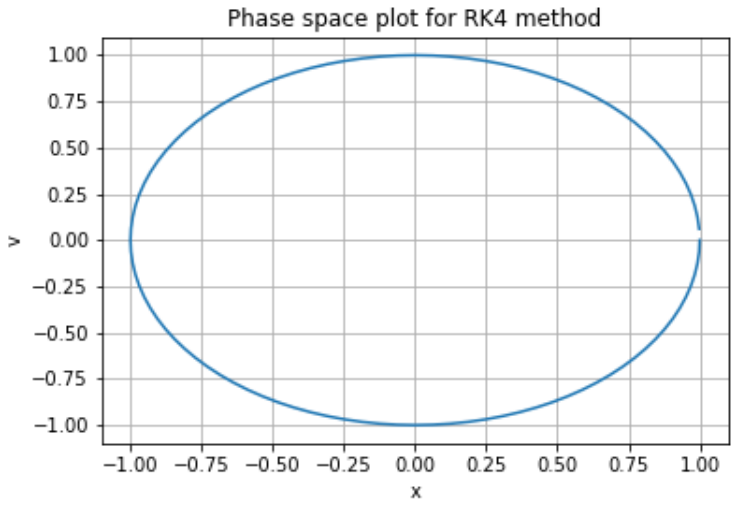


Figure 11: observing error dE

The most effective way to observe the behavior of the error is to, as before to plot a phase space plot over many cycles. Given that energy is conserved (and is the sum of the squares of the momentum and displacement), we would expect to see the plot form a circle. The figure below demonstrates this error dE, which is less than that of the manual RK2 method.

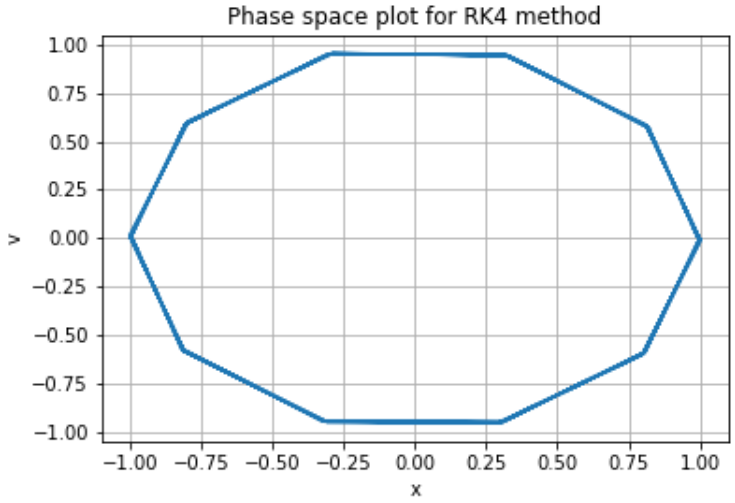


Figure 12: observing error dE

## 2.4 Adaptive step-size control

Another routine available with scipy is that of the .ode solver package. This offers several numerical solvers. Those considered for this part were those that are step size adaptive: the dopri 5 and 853 methods. The 853 is that that features in the figures, given that it is effectively a RK8 method and has a higher precision.

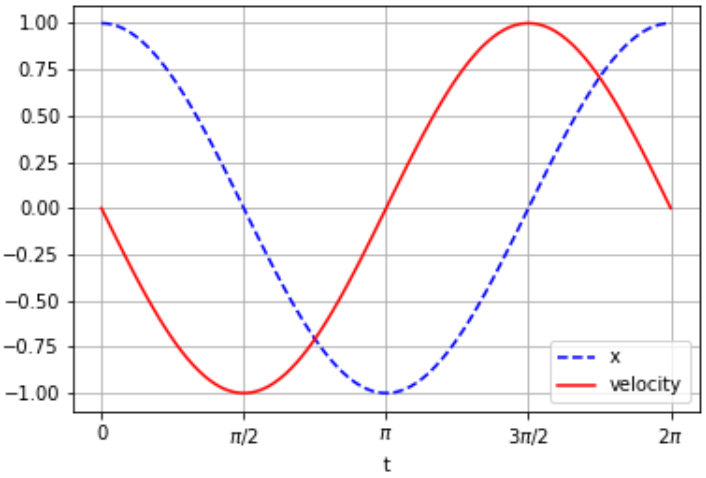


Figure 13: how x and v depend on time for the dopri method

The error can be seen below, as before with a phase space plot of many periods:

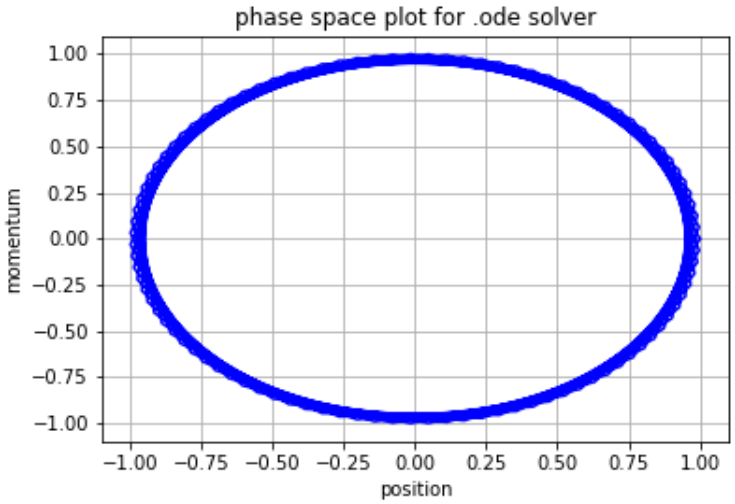


Figure 14: observing error dE

This clearly shows greatly reduced error over the other 2 method, as there is no exponential 'spiral' towards the centre, E is more conserved. And the phase space plot for 1 cycle:

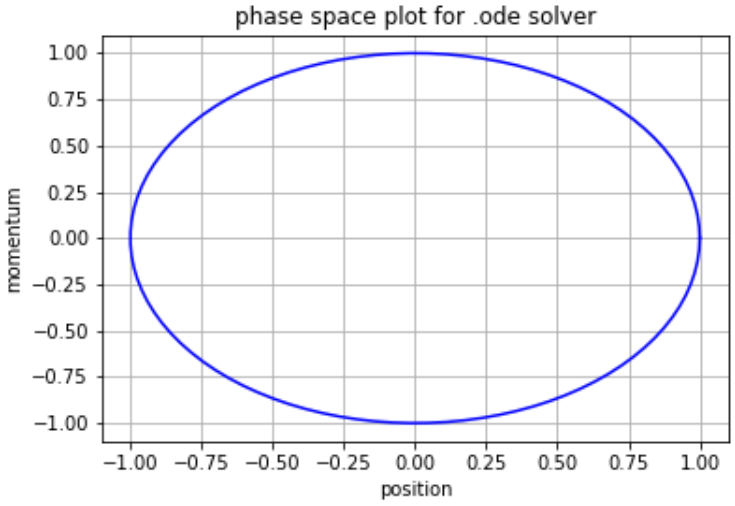


Figure 15: phase space plot for 1 cycle of dopri