Brief note on numerical solution of ordinary differential equation

A first order differential equation (ODE), with initial value(s) given, can generically be written as

$$\frac{dy}{dx} = f(y(x), x), \quad \text{with } \; y(x_0) = y_0. \tag{1}$$

The dy/dx is basically a tangent to the solution curve $y = y(x)$ at the point x and the initial condition states that at $x = x_0$, $y = y_0$. Examples of such ODE (1) are rate of radioactive decay $dN/dt = -\lambda N$ with $N(t_0) = N_0$, discharging of CR circuit $dQ/dt = -Q/CR$ with $Q(t_0) = Q_0$ etc. In many cases, such as harmonic oscillator (say, damped), it is possible to rewrite second-order differential equation in terms of two coupled first order ODE,

$$\frac{d^2x}{dt^2} = -\omega^2 x \quad \Rightarrow \quad v = \frac{dx}{dt} \;\text{ and }\; \frac{dv}{dt} = -\omega^2 x - \mu v \tag{2}$$

with the initial conditions being $x(t_0) = x_0$ and $v(t_0) = v_0$.

To solve the above first order ODEs given an initial condition, we will discuss the following methods,

1. Forward (explicit) and Backward (implicit) Euler's method
2. Predictor-Corrector method
3. Runge-Kutta 4th order

Forward Euler's method

Consider Taylor expansion of the function $y(x_0 + h)$ about x_0, where h is small,

$$y(x_0 + h) = y(x_0) + h \left.\frac{dy}{dx}\right|_{x_0} + \frac{h^2}{2!} \left.\frac{d^2y}{dx^2}\right| x_0 + \cdots \approx y(x_0) + h\, f(y(x_0), x_0) + \mathcal{O}(h^2) \tag{3}$$

If we consider $x_1 = x_0 + h$ to be a small h step away from x_0, then the Forward Euler's method gives the solution of our ODE (1) at x_1 as

$$y(x_1) = y(x_0) + h\, f(y(x_0), x_0) + \mathcal{O}(h^2) \tag{4}$$

In the next step, starting at $x = x_1$ we can use (4) to go to the next step $x_2 = x_1 + h$ and then to $x_3 = x_2 + h$ and so on. Therefore, at the n-th step the solution of the ODE (1) is

$$y(x_n + h) = y(x_n) + h\, f(y(x_n), x_n) \quad \text{or, equivalently } \; y_{n+1} = y_n + \kappa_1 \tag{5}$$

where $\kappa_1 = f(y(x_n), x_n)$. This forward Euler method depends on the tangent dy/dx calculated at earlier point x_n (*i.e.* beginning of the interval) to obtain the solution at the end of the interval y_{n+1}. Hence, this method is also regarded as *explicit* Euler method. Why it is called *forward*? It becomes obvious if we look at (1) as discrete derivative,

$$\begin{aligned} & \frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} = \lim_{\Delta x \to 0} \frac{y(x + \Delta x) - y(x)}{\Delta x} = f(y(x), x) \\ \Rightarrow \quad & y(x + \Delta x) \approx y(x) + \Delta x \, f(y(x), x). \end{aligned} \tag{6}$$

Although Euler method often return fairly good approximation to the actual solution $y = y(x)$, it is extremely slow in the sense that h has to be quite small to achieve some desired accuracy. This also has stability problem and can easily veer away from the solution. So the next obvious step one can consider is the *backward* Euler where we re-define $\Delta y/\Delta x$,

$$\begin{aligned} \frac{\Delta y}{\Delta x} \quad \approx \quad & \frac{y(x) - y(x - \Delta x)}{\Delta x} = f(y(x), x) \quad \Rightarrow \quad y(x) = y(x - \Delta x) + \Delta x \, f(y(x), x) \\ \Rightarrow \quad & y(x + \Delta x) = y(x) + f(y(x + \Delta x), x + \Delta x) \\ \equiv \quad & y(x + h) = y(x) + h \, f(y(x + h), x + h) \end{aligned} \tag{7}$$

The backward Euler thus $y(x+h)$ is determined from the tangent at $x+h$ which, rather strangely, implies that in order to calculate $y(x + h)$ one needs to know $y(x + h)$!! In reality $y(x + h)$ is solved by using Newton-Raphson by solving

$$y^{\mathrm{NR}}(x + h) = y(x) + h \, f(y^{\mathrm{NR}}(x + h), x + h) \tag{8}$$

and then using the solution to determine $y(x + h)$,

$$y(x + h) = y(x) + h \, f(y^{\mathrm{NR}}(x + h), x + h). \tag{9}$$

The backward or implicit Euler method, in spite of having an extra step of Newton-Raphson, is advantageous because of better stability over backward method.

The algorithm for forward method is absolutely straight forward (no pun intended).

1. Choose an h and take the first step from x_0 to $x_1 = x_0 + h$ using (4) to solve $y(x_1)$.
2. Use x_1 to solve $y(x_2)$, x_2 to solve $y(x_3)$ and so on till you reach your desired end point x_N.
3. A plot of $\{x_i, y(x_i)\}$ will give you an approximate solution $y = y(x)$ curve, smaller the h better the solution.

Predictor-Corrector method

The Predictor-Corrector predicts the $y(x_n + h)$ using forward Euler (5), say $y^p(x_n + h)$, and then use it to estimate the slope at $x_n + h$ which is $f(y^p(x_n + h), x_n + h)$. Taking the average of the two slopes $f(y(x_n), x_n)$ and $f(y^p(x_n + h), x_n + h)$, we obtain the corrected value $y^c(x_n + h)$,

$$\begin{aligned}
y^p(x_n + h) &= y(x_n) + h\, f(y(x_n), x_n) + \mathcal{O}(h^2) \\
y^c(x_n + h) &= y(x_n) + \frac{h}{2}\Big[f(y(x_n), x_n) + f(y^p(x_n + h), x_n + h)\Big] \\
&= y(x_n) + \frac{1}{2}\Big(\kappa_1 + \kappa_2\Big) \qquad (10) \\
\text{where, } \kappa_1 &= h\, f(y(x_n), x_n) \;\text{ and }\; \kappa_2 = h\, f(y(x_n + h), x_n + h)
\end{aligned}$$

The algorithm for Predictor-Corrector method is almost like forward Euler,

1. Compute the slope at x_n and define $\kappa_1 = h\, f(y(x_n), x_n)$.
2. Calculate the predicted $y^p(x_n + h) = y(x_n) + \kappa_1$.
3. Use y^p compute $\kappa_2 = h\, f(y^p(x_n + h), x_n + h) = h\, f(y(x_n) + \kappa_1, x_n + h)$.
4. Obtain the corrected $y^c(x_n + h) = y(x_n) + (\kappa_1 + \kappa_2)/2$, which the equation (10).
5. Go to step 1 and continue iterating through the steps till you reach your x_N.

Runge-Kutta method

Runge-Kutta (RK) methods are based as usual on Taylor expansion but gives in general better algorithm for solutions of an ODE for the same step size and stability. It involves use of numerical integration methods for computing $y(x_n + h)$. Consider the following

$$\begin{aligned}
\frac{dy}{dt} &= f(y(x), x) \\
\int_{y_n}^{y_{n+h}} dy &= \int_{x_n}^{x_n+h} f(y(x), x)\, dx \\
y(x_n + h) &= y(x_n) + \int_{x_n}^{x_n+h} f(y(x), x)\, dx \qquad (11)
\end{aligned}$$

To numerically estimate the integral in (11), we can use any of Midpoint, Trapezoidal or Simpson rule for numerical integration. Let us begin with using the Midpoint rule, where

$$
\begin{aligned}
\overline{x}_n &= \frac{(x_n + h) + x_n}{2} = x_n + \frac{h}{2} \\
y(x_n + h) &= y(x_n) + h\, f(y(x_n + h/2), x_n + h/2) + \mathcal{O}(h^3) \qquad (12)
\end{aligned}
$$

But since $y(x_n + h/2)$ is not known, so the next approximation involves forward Euler's method to compute it,

$$
y(x_n + h/2) = y(x_n) + \frac{h}{2} f(y(x_n), x_n) \qquad (13)
$$

Therefore, *second order Runge-Kutta method (RK2)* goes as,

$$
\begin{aligned}
\kappa_1 &= h\, f(y(x_n), x_n) \qquad (14) \\
\kappa_2 &= h\, f(y(x_n) + \kappa_1/2, x_n + h/2) \qquad (15) \\
y(x_n + h) &\approx y(x_n) + \kappa_2 + \mathcal{O}(h^3) \qquad (16)
\end{aligned}
$$

So the difference between the previous one-step methods is the addition of an intermediate half-step method (13). The order of error follows from the maximum error bound of the Midpoint method.

Similarly, we can use the Trapezoidal rule but it simply reproduce the Predictor-Corrector formula with the same $\mathcal{O}(h^3)$ error.

The next obvious step is using the Simpson rule to develop the *fourth order Runge-Kutta (RK4)* which is thus far the most popular method for solving ODE. Unless stated or asked differently, it will always be assumed that you are using RK4 to solve an ODE. The integral in the expression (11) using Simpson rule can be written as,

$$
\begin{aligned}
y(x_n + h) &= y(x_n) + \frac{h}{6}\Big[f(y(x_n), x_n) + 4f(y(x_n + h/2), x_n + h/2) + f(y(x_n + h), x_n + h)\Big] \\
&= y(x_n) + \frac{h}{6}\Big[f(y(x_n), x_n) + 2f(y(x_n + h/2), x_n + h/2) + \\
&\qquad 2f(y(x_n + h/2), x_n + h/2) + f(y(x_n + h), x_n + h)\Big] \qquad (17)
\end{aligned}
$$

Essentially, we split up the expression for slopes at interval midpoint $f(y(x_n + h/2), x_n + h/2)$ into two – one *predicts* the tangent at the interval and the later *corrects* it. Now, we define the following

$$
\begin{aligned}
\kappa_1 &= h\, f(y(x_n), x_n) \qquad (18) \\
\kappa_2 &= h\, f(y(x_n) + \kappa_1/2, x_n + h/2) \qquad (19) \\
\kappa_3 &= h\, f(y(x_n) + \kappa_2/2, x_n + h/2) \qquad (20) \\
\kappa_4 &= h\, f(y(x_n) + \kappa_3, x_n + h) \qquad (21)
\end{aligned}
$$

which when combined, we finally arrive at the RK4 solution

$$y(x_n + h) = y(x_n) + \frac{1}{6}\Big(\kappa_1 + 2\kappa_2 + 2\kappa_3 + \kappa_4\Big) + \mathcal{O}(h^5) \tag{22}$$

Once again the $\mathcal{O}(h^5)$ error in RK4 estimate of $y(x_n + h)$ follows from the maximum error bound of Simpson integration rule.

In the following we discuss two problems – *(i)* second order ODE to solve damped SHO and *(ii)* coupled ODE to solve Lorentz equations.

First consider damped SHO,

$$\frac{d^2x}{dt^2} = -\mu\,\frac{dx}{dt} - \omega^2 x \quad \text{where,}\;\; x(t=0) = 1, \;\; v(t=0) = dx/dt|_{t=0} = 0 \tag{23}$$

which we split in two first order ODEs for numerical solution,

$$v \;=\; \frac{dx}{dt} \quad \text{where} \;\; v(t=0) = 0 \tag{24}$$

$$\frac{dv}{dt} \;=\; -\mu\, v - \omega^2 x \tag{25}$$

The RK4 for the above equations takes the following appearance, where we the name of the functions are self explanatory.

```
k1x   =   dt*dxdt(v,t);
k1v   =   dt*dvdt(x,v,t);

k2x   =   dt*dxdt(v+k1v/2,t+dt/2);
k2v   =   dt*dvdt(x+k1x/2,v,t+dt/2);

k3x   =   dt*dxdt(v+k2v/2,t+dt/2);
k3v   =   dt*dvdt(x+k2x/2,v,t+dt/2);

k4x   =   dt*dxdt(v+k3v,t+dt);
k4v   =   dt*dvdt(x+k3x,v,t+dt);

x    +=   (k1x + 2*k2x + 2*k3x + k4x)/6;
v    +=   (k1v + 2*k2v + 2*k3v + k4v)/6;
t    +=   dt;
```

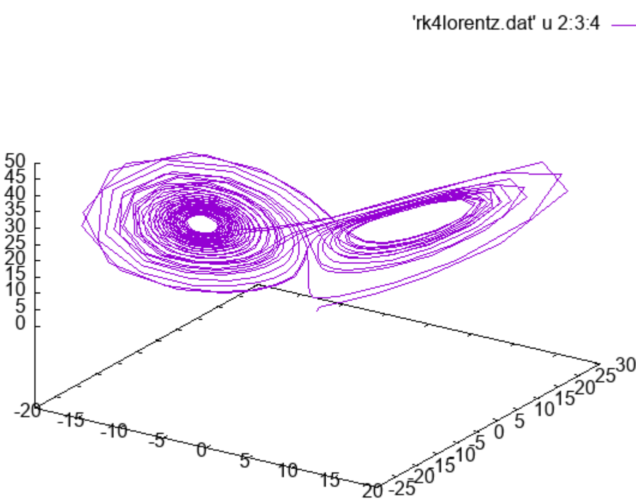
The second example is to solve Lorentz equations which are coupled first order ODE and the solution are sensitive to the initial condition. The equations are,

$$
\begin{aligned}
\frac{dx}{dt} &= \sigma\,(y-x) \\
\frac{dy}{dt} &= x(\rho - z) - y \\
\frac{dz}{dt} &= xy - \beta z
\end{aligned}
\tag{26}
$$

The equations relate the properties of a two-dimensional fluid layer uniformly warmed from below and cooled from above, and σ, ρ, β are three parameters whose certain values give rise to chaotic behavior. The implementation is similar to the above damped SHO, except that here we have three $\kappa_i(x, y, z)$ and each are like, for example,

$$
\mathtt{k3x} = \mathtt{dt} * \mathtt{dxdt}(\mathtt{x} + \mathtt{k2x}/2, \mathtt{y} + \mathtt{k2y}/2, \mathtt{z} + \mathtt{k2z}/2, \mathtt{t} + \mathtt{dt}/2) \tag{27}
$$

For $\sigma = 10$, $\rho = 28$, $\beta = 8/3$, the 3-dimensional plot of the solution shows the famous Lorentz attractor.



We can generalized the above RK4 solution of 3 coupled differential equation to n-

coupled first order ODE.

$$\left.\begin{array}{rcl} \frac{dy_1}{dx} & = & f_1(y_1, y_2, \ldots, y_n, x) \\ \frac{dy_2}{dx} & = & f_2(y_1, y_2, \ldots, y_n, x) \\ & \vdots & \\ \frac{dy_n}{dx} & = & f_n(y_1, y_2, \ldots, y_n, x) \end{array}\right\} \Rightarrow \frac{d\vec{y}}{dx} = \vec{f}(\vec{y}, x) \tag{28}$$

where $\vec{y} = (y_1, y_2, \ldots, y_n)$ and $\vec{f} = (f_1, f_2, \ldots, f_n)$. The vector sign simply implies collection of variables. In such case the RK4 equations take the forms,

$$\begin{aligned} \vec{\kappa}_1 &= h\,\vec{f}(\vec{y}_i, x_i) \\ \vec{\kappa}_2 &= h\,\vec{f}(\vec{y}_i + \vec{\kappa}_1/2, x_i + h/2) \\ \vec{\kappa}_3 &= h\,\vec{f}(\vec{y}_i + \vec{\kappa}_2/2, x_i + h/2) \\ \vec{\kappa}_4 &= h\,\vec{f}(\vec{y}_i + \vec{\kappa}_3, x_i + h) \\ \vec{y}_{i+h} &= \vec{y}_i + \frac{1}{6}\Big[\vec{\kappa}_1 + 2\vec{\kappa}_2 + 2\vec{\kappa}_3 + \vec{\kappa}_4\Big] \end{aligned} \tag{29}$$

where $\vec{y}_i$ are the values at the i-th interval boundary. The above set of equations (29) have to be read only in terms of components.

Shooting Method : Boundary value problem

Many problems in physics are, in fact, boundary value problems. For instance the Laplace equation in electrostatics or, more famously, Schrödinger equations. In boundary value problems, we have conditions specified at two different space (and/or time) points. We can have either

Dirichlet condition : $y(x_0) = Y_0$ and $y(x_N) = Y_n$

Neumann condition : $y'(x_0) = Y'_0$ and $y'(x_N) = Y'_N$

The strategy to numerically solve boundary value problem is to reduce the second order ODE to a system of first order initial value ODE.

$$\frac{d^2y}{dx^2} = f(x, y, y') \quad \text{where } \; a \le x \le b \quad \text{and} \quad y(a) = \alpha,\; y(b) = \beta \tag{30}$$

$$\Rightarrow \qquad \frac{dy}{dx} = z \quad \text{with} \;\; y(a) = \alpha \tag{31}$$

$$\frac{dz}{dx} = \frac{d^2y}{dx^2} = f(x, y, z) \quad \text{with} \;\; z(a) = \zeta_h \tag{32}$$

where $z(a) = \zeta_h$ *i.e.* slope at $x = a$ is a guess. The next step involves solving (31) and (32) by Euler/RK2/RK4 using the initial values $y(a) = \alpha$ and $z(a) = \zeta_h$. The solution obtained at the end point x_N is compared with the boundary condition $y(b) = \beta$. If $y_{\zeta_h}(b) = \beta$ within tolerance then the ODE is solved. But suppose it is not and $y_{\zeta_h}(b) > \beta$.

Change the guess initial value to ζ_l and the system is solved again as above. The choice should be such (assuming it does not land bang on the solution) that $y_{\zeta_l}(b) < \beta$. This implies that the actual slope at initial point is $\zeta_l < z(a) < \zeta_h$.

Use Lagrange's interpolation formula to choose the next $z(a) = \zeta$,

$$\zeta = \zeta_l + \frac{\zeta_h - \zeta_l}{y_{\zeta_h}(b) - y_{\zeta_l}(b)} \Big(y(b) - y_{\zeta_l}(b) \Big) \tag{33}$$

The $z(a) = \zeta$ is our new guess and chances are this choice will lead us to the solution of the ODE *i.e.* $y_\zeta(b) \approx \beta$. If not, go through the above procedure until $y_\zeta(b)$ converges to β reasonably well.

Let us study the following example,

$$\frac{d^2y}{dx^2} = 2y \quad \text{with} \;\; y(x = 0.0) = \alpha = 1.2, \;\; y(x = 1.0) = \beta = 0.9, \;\; h = 0.02. \tag{34}$$

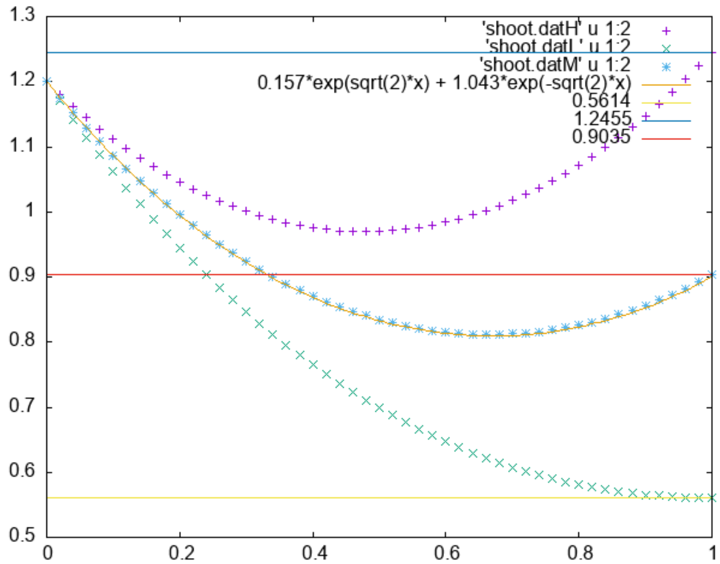
The solution of the above ODE (34) is

$$y(x) = c_1\, e^{\sqrt{2}\,x} + C_2\, e^{-\sqrt{2}\,x}, \quad \text{where} \;\; C_1 = 0.157, \; C_2 = 1.043 \tag{35}$$

Suppose we start with $z(x = 0.0) = -1.5$, using RK4 we obtain $y(x = 1.0) = 0.5614$ which is less than $\beta = 0.9$. So $\zeta_l = -1.5$ and $y_{\zeta_l}(1.0) = 0.5614$. Next we try (presently the choice is deliberate) $z(x = 0.0) = -1.0 = \zeta_h$ which yields $y_{\zeta_h}(x = 1.0) = 1.2455 > \beta = 0.9$. Using Lagrange's linear interpolating formula (33) we get,

$$\zeta = -1.5 + \frac{-1.0 - (-1.5)}{1.2455 - 0.5614} \times \Big(0.9 - 0.5614 \Big) = -1.2525 \tag{36}$$

Using $z(x = 0.0) = -1.25$, we obtain $y_\zeta(x = 1.0) = 0.9035 \approx \beta = 0.9$. Thus we have solved our ODE in (34). A graphical view of the process is shown below.



In practice, you may not be so lucky to guess ζ_h and ζ_l appropriately, so you may end up going through multiple iterations of the above method.