An Introduction to Runge-Kutta Integrators

Robin Leroy (eggrobin)

2014-02-15

In this post I shall assume understanding of the concepts described in chapter 8 (Motion) as well as sections 11–4 and 11–5 (Vectors and Vector algebra) of chapter 11 of Richard Feynmann's *Lectures on Physics*.

1 Motivation

We want to be able to predict the position $\boldsymbol{s}(t)$ as a function of time of a spacecraft (without engines) around a fixed planet of mass M. In order to do this, we recall that the velocity is given by

$$\boldsymbol{v} = \frac{\mathrm{d}\,\boldsymbol{s}}{\mathrm{d}\,t}$$

and the acceleration by

$$\boldsymbol{a} = \frac{\mathrm{d}\,\boldsymbol{v}}{\mathrm{d}\,t} = \frac{\mathrm{d}^2\,\boldsymbol{s}}{\mathrm{d}\,t^2}.$$

We assume the mass of the spacecraft is constant and that the planet sits at the origin of our reference frame. Newton's law of universal gravitation tells us that the magnitude (the length) of the acceleration vector will be

$$a = \frac{GM}{s^2},$$

where s is the length of $\boldsymbol{s}$, and that the acceleration will be directed towards the planet, so that

$$\boldsymbol{a} = -\frac{GM}{s^2}\frac{\boldsymbol{s}}{s}.$$

We don't really care about the specifics, but we see that this is a function of $\boldsymbol{s}$. We'll write it $\boldsymbol{a}(\boldsymbol{s})$. Putting it all together we could rewrite this as

$$\frac{\mathrm{d}^2\,\boldsymbol{s}}{\mathrm{d}\,t^2} = \boldsymbol{a}(\boldsymbol{s})$$

and go ahead and solve this kind of problem, but we don't like having a second derivative. Instead we just write down both equations,

$$\begin{cases} \frac{\mathrm{d}\,\boldsymbol{s}}{\mathrm{d}\,t} = \boldsymbol{v} \\ \frac{\mathrm{d}\,\boldsymbol{v}}{\mathrm{d}\,t} = \boldsymbol{a}(\boldsymbol{s}) \end{cases}.$$

Let us define a vector $\boldsymbol{y}$ with 6 entries instead of 3,

$$\boldsymbol{y} = (\boldsymbol{s}, \boldsymbol{v}) = (s_x, s_y, s_z, v_x, v_y, v_z).$$

Similarly, define a function $\boldsymbol{f}$ as follows:

$$\boldsymbol{f}(\boldsymbol{y}) = (\boldsymbol{v}, \boldsymbol{a}(\boldsymbol{s})).$$

Our problem becomes

$$\frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t} = \left(\frac{\mathrm{d}\,\boldsymbol{s}}{\mathrm{d}\,t}, \frac{\mathrm{d}\,\boldsymbol{v}}{\mathrm{d}\,t}\right) = (\boldsymbol{v}, \boldsymbol{a}(\boldsymbol{s})) = \boldsymbol{f}(\boldsymbol{y}).$$

So we have gotten rid of that second derivative.

2 Ordinary differential equations

We are interested computing solutions to equations of the form

$$\frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t} = \boldsymbol{f}(t, \boldsymbol{y}).$$

Such an equation is called an *ordinary differential equation* (ODE). The function $\boldsymbol{f}$ is called the *right-hand side* (RHS).

Recall that if the right-hand side didn't depend on $\boldsymbol{y}$, the answer would be the integral,

$$\frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t} = \boldsymbol{f}(t) \implies \boldsymbol{y} = \int \boldsymbol{f}(t)\,\mathrm{d}\,t.$$

In the case where the right-hand side doesn't depend on t (but depends on $\boldsymbol{y}$), as was the case in the previous section, the equation becomes

$$\frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t} = \boldsymbol{f}(\boldsymbol{y}).$$

We call such a right-hand side *autonomous*.

In order to compute a particular solution (a particular trajectory of our spacecraft), we need to define some initial conditions (the initial position and velocity of our spacecraft) at $t = t_0$. We write them as

$$\boldsymbol{y}(t_0) = \boldsymbol{y}_0.$$

The ODE together with the initial conditions form the *initial value problem* (IVP)

$$\begin{cases} \frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t} = \boldsymbol{f}(t, \boldsymbol{y}) \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \end{cases}.$$

3 Euler's method

As we want to actually store the solution in a computer, we can't compute $\boldsymbol{y}(t)$ for all values of t. Instead we approximate $\boldsymbol{y}(t)$ at discrete time steps.

How do we compute the first point $\boldsymbol{y}_1$, the approximation for $\boldsymbol{y}(t_0 + \Delta t)$? By definition of the derivative, we get

$$\lim_{\Delta t \to 0} \frac{\boldsymbol{y}(t_0 + \Delta t) - \boldsymbol{y}(t_0)}{\Delta t} = \boldsymbol{f}(t_0, \boldsymbol{y}_0).$$

This means that if we take a sufficiently small Δt, we have

$$\frac{\boldsymbol{y}(t_0 + \Delta t) - \boldsymbol{y}(t_0)}{\Delta t} \approx \boldsymbol{f}(t_0, \boldsymbol{y}_0),$$

where the approximation gets better as Δt gets smaller. Our first method for approxmimating the solution is therefore to compute

$$\boldsymbol{y}_1 = \boldsymbol{y}(t_0) + \Delta t \boldsymbol{f}(t_0, \boldsymbol{y}_0).$$

For the rest of the solution, we just repeat the same method, yielding

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}(t_n) + \Delta t \boldsymbol{f}(t_n, \boldsymbol{y}_n).$$

This is called *Euler's method*, after Swiss mathematician Leonhard Euler (1707–1783). A good visualisation of this method, as well as a geometric interpretation, can be found on *Wikipedia*.

We want to know two things: how good our approximation is, and how much we need to reduce Δt in order to make it better. In order to do that, we use Taylor's theorem.

Taylor's theorem

Recall that if $\frac{\mathrm{d}\,y}{\mathrm{d}\,t}$ is constant,

$$\boldsymbol{y}(t_0 + \Delta t) = \boldsymbol{y}(t_0) + \Delta t \frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t}(t_0).$$

If $\frac{\mathrm{d}^2\,y}{\mathrm{d}\,t^2}$ is constant,

$$\boldsymbol{y}(t_0 + \Delta t) = \boldsymbol{y}(t_0) + \Delta t \frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t}(t_0) + \frac{\Delta t^2}{2}\frac{\mathrm{d}^2\,\boldsymbol{y}}{\mathrm{d}\,t^2}(t_0).$$

In general, if you assume the nth derivative is constant,

$$\boldsymbol{y}(t_0 + \Delta t) = \boldsymbol{y}(t_0) + \sum_{j=1}^{n} \frac{\Delta t^j}{j!}\frac{\mathrm{d}^j\,\boldsymbol{y}}{\mathrm{d}\,t^j}(t_0),$$

where $j! = 1 \times 2 \times 3 \times \cdots \times j$ is the factorial of j.

Taylor's theorem roughly states that this is a good approximation, which gets better as n gets higher. Formally, if $\boldsymbol{y}$ is differentiable n times, for sufficiently small Δt,

$$\boldsymbol{y}(t_0 + \Delta t) = \boldsymbol{y}(t_0) + \sum_{j=1}^{n} \frac{\Delta t^j}{j!}\frac{\mathrm{d}^j\,\boldsymbol{y}}{\mathrm{d}\,t^j}(t_0) + \Delta t^n \boldsymbol{r}(\Delta t), \tag{3.1}$$

where

$$\lim_{\Delta t \to 0} \boldsymbol{r}(\Delta t) = 0.$$

A proof can be found on *Wikipedia*.

It is often convenient to absorb the highest order terms using Landau notation. We can rewrite the above as

$$\boldsymbol{y}(t_0 + \Delta t) = \boldsymbol{y}(t_0) + \sum_{j=1}^{n-1} \frac{\Delta t^j}{j!}\frac{\mathrm{d}^j\,\boldsymbol{y}}{\mathrm{d}\,t^j}(t_0) + \mathcal{O}(\Delta t^n), \tag{3.2}$$

where $\mathcal{O}(\Delta t^n)$, read "big $\mathcal{O}$ of Δt^n", is not a specific function, but stands for "some function whose magnitude is bounded by $K\Delta t^n$ for some constant K as Δt goes to 0".

We shall always assume that the solution to our IVP is sufficiently differentiable.

Error analysis

Armed with this theorem, we can look back at Euler's method. We computed the approximation for $\boldsymbol{y}(t_0 + \Delta t)$ as

$$\boldsymbol{y}_1 = \boldsymbol{y}(t_0) + \Delta t \boldsymbol{f}(t_0, \boldsymbol{y}_0) = \boldsymbol{y}(t_0) + \Delta t \frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t}(t_0).$$

In order to compute the magnitude of the error, we'll use Taylor's theorem for $n = 2$. We have, for sufficiently small Δt,

$$\begin{aligned} |\boldsymbol{y}(t_0 + \Delta t) - \boldsymbol{y}_1| &= \left|\boldsymbol{y}(t_0) + \Delta t \frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t}(t_0) + \mathcal{O}(\Delta t^2) - \boldsymbol{y}(t_0) + \Delta t \frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t}(t_0)\right| \\ &= |\mathcal{O}(\Delta t^2)| \leq K\Delta t^2 \text{ by definition of the big } \mathcal{O} \text{ notation.} \end{aligned}$$

for some constant K which does not depend on Δt. This means that the error on *one step* behaves as the square of the time step: it is divided by four when the time step is halved.

We should remember however that when we reduce the time step, we need more steps to compute the solution over the same duration. What is the error when we reach some t_{end}? There are obviously $\frac{t_{\text{end}} - t_0}{\Delta t}$ steps, so intuitively, the error should behave as $\frac{t_{\text{end}} - t_0}{\Delta t}\Delta t^2 = (t_{\text{end}} - t_0)\Delta t$, and indeed this is the case. In order to properly show

that, some additional assumptions must be made, the description of which is beyond the scope of this introduction.[1]

The conclusion about Euler is then that when computing the solution over a fixed duration $t_{\text{end}} - t_0$, the error behaves like Δt—linearly: halving the time step will halve the error. We say Euler's method is a first order method.

Can we do better? In order to answer that question, we note that the reason why the error of Euler's method was linear for a fixed duration is that it was quadratic for a fixed timestep. The reason why it was quadratic for a fixed timestep is that our approximation matched the first derivative term in the Taylor expansion. If we could match higher-order terms in the expansion, we would get a higher-order method. Specifically, if our approximation matches the Taylor expansion up to and including the kth derivative, we'll get a kth order method.

4 The midpoint method

How do we match the higher derivatives? We don't know what they are: the first derivative is given to us by the problem (it's $\boldsymbol{f}(t, \boldsymbol{y}(t))$ at time t), the other ones are not. However, if we look at $\boldsymbol{g}(t) = \boldsymbol{f}(t, \boldsymbol{y}(t))$ as a function of t, we have

$$\begin{aligned} \boldsymbol{g} &= \frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t} \\ \frac{\mathrm{d}\,\boldsymbol{g}}{\mathrm{d}\,t} &= \frac{\mathrm{d}^2\,\boldsymbol{y}}{\mathrm{d}\,t^2}. \end{aligned}$$

Of course, we can't directly compute the derivative of $\boldsymbol{g}$, because we don't even know what $\boldsymbol{g}$ itself looks like: that would entail knowing $\boldsymbol{y}$, which is what we are trying to compute.

However, let us assume for a moment that we could compute $\boldsymbol{g}\left(t_0 + \frac{\Delta t}{2}\right)$. Using Taylor's theorem on $\boldsymbol{g}$,

$$\boldsymbol{g}\left(t_0 + \frac{\Delta t}{2}\right) = \boldsymbol{g}(t_0) + \frac{\Delta t}{2}\frac{\mathrm{d}\,\boldsymbol{g}}{\mathrm{d}\,t}(t_0) + \mathcal{O}(\Delta t^2).$$

Substituting $\boldsymbol{g}$ yields.

$$\boldsymbol{g}\left(t_0 + \frac{\Delta t}{2}\right) = \frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t}(t_0) + \frac{\Delta t}{2}\frac{\mathrm{d}^2\,\boldsymbol{y}}{\mathrm{d}\,t^2}(t_0) + \mathcal{O}(\Delta t^2).$$

This looks like the first and second derivative terms in the Taylor expansion of $\boldsymbol{y}$. This means if we could compute it,

$$\boldsymbol{y}_1 = \boldsymbol{y}_0 + \Delta t \boldsymbol{g}\left(t_0 + \frac{\Delta t}{2}\right) = \boldsymbol{y}_0 + \Delta t \boldsymbol{f}\left(t_0 + \frac{\Delta t}{2}, \boldsymbol{y}\left(t_0 + \frac{\Delta t}{2}\right)\right)$$

would yield a second-order method:

$$\boldsymbol{y}_0 + \Delta t \boldsymbol{g}\left(t_0 + \frac{\Delta t}{2}\right) = \boldsymbol{y}_0 + \Delta t \frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t}(t_0) + \frac{\Delta t^2}{2}\frac{\mathrm{d}^2\,\boldsymbol{y}}{\mathrm{d}\,t^2}(t_0) + \mathcal{O}(\Delta t^3).$$

The problem is we can't compute $\boldsymbol{y}\left(t_0 + \frac{\Delta t}{2}\right)$ exactly. Instead, we try using a second-order approximation for it, computed using one step of Euler's method, namely

$$\boldsymbol{y}_0 + \frac{\Delta t}{2}\boldsymbol{f}(t_0, \boldsymbol{y}_0) = \boldsymbol{y}\left(t_0 + \frac{\Delta t}{2}\right) + \mathcal{O}(\Delta t^2).$$

This yields the following value for $\boldsymbol{y}_1$.

$$\boldsymbol{y}_1 = \boldsymbol{y}_0 + \Delta t \boldsymbol{f}\left(t_0 + \frac{\Delta t}{2}, \boldsymbol{y}_0 + \frac{\Delta t}{2}\boldsymbol{f}(t_0, \boldsymbol{y}_0)\right)$$

[1] For the advanced reader: the solution has to be Lipschitz continuous and its second derivative has to be bounded.

In order to show that the half Euler step was good enough, use our error bound on that step and compute the Taylor expansion of $\boldsymbol{f}$ for $n = 1$ in its second argument,[2]

$$\begin{aligned}\boldsymbol{f}\left(t_0 + \frac{\Delta t}{2}, \boldsymbol{y}_0 + \frac{\Delta t}{2}\boldsymbol{f}(t_0, \boldsymbol{y}_0)\right) &= \boldsymbol{f}\left(t_0 + \frac{\Delta t}{2}, \boldsymbol{y}\left(t_0 + \frac{\Delta t}{2}\right) + \mathcal{O}(\Delta t^2)\right) \\ &= \boldsymbol{f}\left(t_0 + \frac{\Delta t}{2}, \boldsymbol{y}\left(t_0 + \frac{\Delta t}{2}\right)\right) + \mathcal{O}(\Delta t^2).\end{aligned}$$

Substituting yields

$$\begin{aligned}\boldsymbol{y}_1 &= \boldsymbol{y}_0 + \Delta t\boldsymbol{f}\left(t_0 + \frac{\Delta t}{2}, \boldsymbol{y}_0 + \frac{\Delta t}{2}\boldsymbol{f}(t_0, \boldsymbol{y}_0)\right) \\ &= \boldsymbol{y}_0 + \Delta t\boldsymbol{f}\left(t_0 + \frac{\Delta t}{2}, \boldsymbol{y}\left(t_0 + \frac{\Delta t}{2}\right)\right) + \mathcal{O}(\Delta t^3) \\ &= \boldsymbol{y}_0 + \Delta t\frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t}(t_0) + \frac{\Delta t^2}{2}\frac{\mathrm{d}^2\,\boldsymbol{y}}{\mathrm{d}\,t^2}(t_0) + \mathcal{O}(\Delta t^3) \\ &= \boldsymbol{y}(t + \Delta t) + \mathcal{O}(\Delta t^3).\end{aligned}$$

The idea here was to say that the derivative at t_0 is not a good enough approximation for the behaviour between t_0 and $t_0 + \Delta t$, and to compute the derivative halfway through $\boldsymbol{g}\left(t_0 + \frac{\Delta t}{2}\right)$ instead. In order to do that, we had to use a lower-order method (our Euler half-step).

A good visualisation of this method, as well as a geometric interpretation, can be found on *Wikipedia*.

5 Heun's method

Before we move on to the description of general Runge-Kutta methods, let us look at another take on second-order methods.

Instead of approximating the behaviour between t_0 and $t_0 + \Delta t$ by the derivative halfway through, what if we averaged the derivatives at the end and at the beginning?

$$\boldsymbol{y}_1 = \boldsymbol{y}_0 + \Delta t\frac{\boldsymbol{g}(t_0) + \boldsymbol{g}(t_0 + \Delta t)}{2}.$$

Let us compute the error:

$$\begin{aligned}\boldsymbol{y}_1 &= \boldsymbol{y}_0 + \Delta t\frac{\frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t}(t_0) + \frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t}(t_0) + \Delta t\frac{\mathrm{d}^2\,\boldsymbol{y}}{\mathrm{d}\,t^2}(t_0) + \mathcal{O}(\Delta t^2)}{2} \\ &= \boldsymbol{y}_0 + \Delta t\frac{\mathrm{d}\,\boldsymbol{y}}{\mathrm{d}\,t}(t_0) + \frac{\Delta t^2}{2}\frac{\mathrm{d}^2\,\boldsymbol{y}}{\mathrm{d}\,t^2}(t_0) + \mathcal{O}(\Delta t^3) = \boldsymbol{y}(t_0 + \Delta t).\end{aligned}$$

So this is indeed a second-order method. As in the midpoint method, we can't actually compute $\boldsymbol{g}(t_0 + \Delta t)$. Instead we approximate it using Euler's method, so that our step becomes:

$$\boldsymbol{y}_1 = \boldsymbol{y}_0 + \Delta t\frac{\boldsymbol{f}(t_0, \boldsymbol{y}_0) + \boldsymbol{f}(t_0 + \Delta t, \boldsymbol{y}_0 + \Delta t\boldsymbol{f}(t_0, \boldsymbol{y}_0))}{2}.$$

It looks like this is slower than the midpoint method, as there are three evaluations of $\boldsymbol{f}$. However, two of those are with the same arguments, so we can rewrite things as follows:

$\boldsymbol{k}_1 = \Delta t\boldsymbol{f}(t_0, \boldsymbol{y}_0)$ is our approximation for $\boldsymbol{g}(t_0)$;
$\boldsymbol{k}_2 = \Delta t\boldsymbol{f}(t_0 + \Delta t, \boldsymbol{y}_0 + \Delta t\boldsymbol{k}_1)$ is our approximation for $\boldsymbol{g}(t_0 + \Delta t)$;
$\boldsymbol{y}_1 = \boldsymbol{y}_0 + \Delta t\frac{\boldsymbol{k}_1 + \boldsymbol{k}_2}{2}$ is our approximation for $\boldsymbol{y}(t + \Delta t)$.

[2] We are actually using Taylor's theorem in the multivariate case here, which is a little more complicated than the one stated above. This doesn't really matter since we are only using the first order.

This is the idea behind Runge-Kutta methods. We compute *increments* $\boldsymbol{k}_i$ which approximate the derivative $\boldsymbol{g}$ at various points between t_0 and $t_0 + \Delta t$, and we take a weighted average of these as our overall linear approximation. In order to compute each increment, we can use the previous ones to construct an approximation that has high enough order.

6 Runge-Kutta methods

A Runge-Kutta method is defined by its *weights* $\boldsymbol{b} = (b_1, \dots, b_s)$, its *nodes* $\boldsymbol{c} = (c_1, \dots, c_s)$ and its Runge-Kutta matrix

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & \cdots & a_{1s} \\ \vdots & \ddots & \vdots \\ a_{s1} & \cdots & a_{ss} \end{pmatrix}.$$