An Introduction to Runge-Kutta Integrators

Robin Leroy (eggrobin)

2014-07-05

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 Feynman'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 that 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 go back to our first order equations and we write both of them down,

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

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, at the cost of making the problem 6-dimensional instead of 3-dimensional.

2 Ordinary differential equations

We are intereſted in 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,

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

General ODEs are a generalisation of this, so the methods we use to compute their solutions are called *integrators*.

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

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

For future reference, 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)

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

3 Euler's method

As we want to actually solve the equation using 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 have

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

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

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

Note that this is an approximation:

$$\boldsymbol{y}_1 \approx \boldsymbol{y}(t_0 + \Delta t).$$

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

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

Again these are approximations:

$$\boldsymbol{y}_n \approx \boldsymbol{y}(t_0 + n\Delta t).$$

This is called *Euler's method*, after the Swiss mathematician and physiciſt Leonhard Euler (1707–1783). A good visualisation of this method, as well as a geometric interpretation, can be found in the *Wikipedia* article `http://en.wikipedia.org/wiki/Euler_method`.

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 $\frac{d\boldsymbol{y}}{dt}$ is conſtant,

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

If $\frac{d^2\boldsymbol{y}}{dt^2}$ is conſtant,

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

In general, if we assume the nth derivative to be conſtant,

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

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,

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

where $\mathcal{O}(\Delta t^n)$ is read "big $\mathcal{O}$ of Δt^n". It 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". This big $\mathcal{O}$ notation indicates the quality of the approximation: it represents error terms that vanish at leaſt as faſt as Δt^n.

There is a version of Taylor's theorem for multivariate functions[1]; the idea is the same, but ſtating it in its general form is complicated. Inſtead let us look at the cases we will need here.

For a function $\boldsymbol{f}(t, \boldsymbol{y})$, We have the analogue to the $n = 1$ version of Taylor's theorem:

$$\boldsymbol{f}(t_0, \boldsymbol{y}_0 + \Delta\boldsymbol{y}) = \boldsymbol{f}(t_0, \boldsymbol{y}_0) + \mathcal{O}(|\Delta\boldsymbol{y}|) \tag{3.2}$$

and the analogue to the $n = 2$ version:

$$\boldsymbol{f}(t_0, \boldsymbol{y}_0 + \Delta\boldsymbol{y}) = \boldsymbol{f}(t_0, \boldsymbol{y}_0) + \frac{d\boldsymbol{f}}{d\boldsymbol{y}}(t_0, \boldsymbol{y}_0)\Delta\boldsymbol{y} + \mathcal{O}(|\Delta\boldsymbol{y}|^2). \tag{3.3}$$

Knowing what $\frac{d\boldsymbol{f}}{d\boldsymbol{y}}(t_0, \boldsymbol{y}_0)$ actually means is not important here, it is juſt something that you can multiply a vector with to get another vector.[2]

[1]Functions of a vector.

[2]It is a linear map, so if you know what a matrix is, you can see it as one.

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

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

By definition of the derivative, we have seen that as Δt approaches 0, $\boldsymbol{y}_1$ will become a better approximation for $\boldsymbol{y}(t + \Delta t)$. However, 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 we should multiply the error on a single step by $\frac{t_{\text{end}} - t_0}{\Delta t}$. This means that the error on a single step needs to vanish[3] *faster than* Δt.

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

$$\begin{aligned} |\boldsymbol{y}(t_0 + \Delta t) - \boldsymbol{y}_1| &= \left|\boldsymbol{y}(t_0) + \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t_0)\Delta t + \mathcal{O}(\Delta t^2) - \boldsymbol{y}(t_0) - \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t_0)\Delta t\right| \\ &= |\mathcal{O}(\Delta t^2)| \leq K\Delta t^2 \end{aligned}$$

for some constant K which does not depend on Δt (recall that this is the definition of the big $\mathcal{O}$ notation). 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.

It follows that the error in the approximation at t_{end} should intuitively 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.[4]

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

We remark that we can rewrite Euler's method as follows.

$$\begin{aligned} \boldsymbol{k}_1 &= \boldsymbol{f}(t_0, \boldsymbol{y}_0); \\ \boldsymbol{y}_1 &= \boldsymbol{y}_0 + \boldsymbol{k}_1\Delta t. \end{aligned}$$

This will be useful in the wider scope of Runge-Kutta integrators.

Can we do better than first-order? In order to answer this question, we note that the reason why the error in Euler's method was linear for a fixed duration is that it was quadratic for a single time step. The reason why it was quadratic for a single time step 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 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

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

[3]For the advanced reader: uniformly.

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

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 $\boldsymbol{g}$,

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

Substituting $\boldsymbol{g}$ yields.

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

This looks like the first and second derivative terms in the Taylor expansion of $\boldsymbol{y}$. Therefore, the following expression would yield a third-order approximation for the step $\boldsymbol{y}(t + \Delta t)$ (and thus a second-order method), if only we could compute it:

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

Indeed,

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

Unfortunately, we can't compute $\boldsymbol{g}\left(t_0 + \frac{\Delta t}{2}\right)$ exactly, because for that we would need to know $\boldsymbol{y}\left(t_0 + \frac{\Delta t}{2}\right)$. Instead, we try using a second-order approximation for it, obtained using one step of Euler's method, namely

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

We use it to get an approximation $\boldsymbol{y}_1$ of $\hat{\boldsymbol{y}}_1$.

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

In order to show that $\boldsymbol{y}_1$ is a third-order approximation for $\boldsymbol{y}(t + \Delta t)$, we show that it is a third-order approximation for $\hat{\boldsymbol{y}}_1$. In order to do that, we use our error bound on the step of Euler's method and compute the multivariate first-order Taylor expansion of $\boldsymbol{f}$ in its second argument (3.2),

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

Substituting yields

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

The method is third-order on a single step, so it is a second order method. 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 the *Wikipedia* article `http://en.wikipedia.org/wiki/Midpoint_method`.

Again we remark that we can rewrite the midpoint method as follows.

$$\begin{aligned}\boldsymbol{k}_1 &= \boldsymbol{f}(t_0, \boldsymbol{y}_0);\\ \boldsymbol{k}_2 &= \boldsymbol{f}\left(t_0, \boldsymbol{y}_0 + \boldsymbol{k}_1\frac{\Delta t}{2}\right);\\ \boldsymbol{y}_1 &= \boldsymbol{y}_0 + \boldsymbol{k}_2\Delta t.\end{aligned}$$

This will be useful in the wider scope of Runge-Kutta integrators.

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?

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

Let us compute the error:

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

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

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

We leave checking that the approximation $\hat{\boldsymbol{y}}_1 \approx \boldsymbol{y}_1$ is indeed third-order as an exercise to the reader. This method is called *Heun's method*, after German mathematician Karl Heun (1859–1929). A good visualisation of this method, as well as a geometric interpretation, can be found on the *Wikipedia* article `http://en.wikipedia.org/wiki/Heun's_method#Description`.

It looks like Heun's method 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:

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

This process can be generalised, yielding so-called Runge-Kutta methods.

6 Runge-Kutta methods

In a *Runge-Kutta method*,[5] we compute the ſtep $\boldsymbol{y}_1 \approx \boldsymbol{y}(t_0 + \Delta t)$ as a linear approximation

$$\boldsymbol{y}_1 = \boldsymbol{y}_0 + \boldsymbol{\lambda}\Delta t.$$

[5] Named after German mathematicians Carl David Tolmé Runge (1856–1927) and Martin Wilhelm Kutta (1867–1944).

The idea is that we want to use a weighted average (with weights b_i) of the derivative $\boldsymbol{g}$ of $\boldsymbol{y}$ at s points between t_0 and $t_0 + \Delta t$ as our approximation $\boldsymbol{\lambda}$,

$$\hat{\boldsymbol{y}}_1 = \boldsymbol{y}_0 + (b_1\boldsymbol{g}(t_0) + b_2\boldsymbol{g}(t_0 + c_2\Delta t) + \cdots + b_s\boldsymbol{g}(t_0 + c_s\Delta t))\Delta t,$$

but we cannot do that because we do not know how to compute $\boldsymbol{g}$; we only know how to compute $\boldsymbol{f}$. Inﬅead we compute *increments* $\boldsymbol{k}_i$ which approximate the derivative, $\boldsymbol{k}_i \approx \boldsymbol{g}(t_0 + c_i\Delta t)$, and we take a weighted average of these as our overall linear approximation:

$$\begin{aligned}\boldsymbol{\lambda} &= b_1\boldsymbol{k}_1 + b_2\boldsymbol{k}_2 + \cdots + b_s\boldsymbol{k}_s,\\ \boldsymbol{y}_1 &= \boldsymbol{y}_0 + (b_1\boldsymbol{k}_1 + b_2\boldsymbol{k}_2 + \cdots + b_s\boldsymbol{k}_s)\Delta t.\end{aligned}$$

In order to compute each increment, we can use the previous ones to conﬅruct an approximation that has high enough order.

Surprisingly, we will see that the approximation

$$b_1\boldsymbol{k}_1 + b_2\boldsymbol{k}_2 + \cdots + b_s\boldsymbol{k}_s \approx b_1\boldsymbol{g}(t_0) + b_2\boldsymbol{g}(t_0 + c_2\Delta t) + \cdots + b_s\boldsymbol{g}(t_0 + c_s\Delta t)$$

is generally better than the individual approximations $\boldsymbol{k}_i \approx \boldsymbol{g}(t_0 + c_i\Delta t)$.

Definition

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

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

It is typically written as a *Butcher tableau*:

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array}$$

We will only consider *explicit* Runge-Kutta methods, *i.e.*, those where $\boldsymbol{A}$ is ﬅrictly lower triangular, so that the Butcher tableau is as follows (blank spaces in $\boldsymbol{A}$ are zeros).

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array}$$

The ﬅep is computed using the weighted sum of the increments as a linear approximation,

$$\boldsymbol{y}_1 = \boldsymbol{y}_0 + (b_1\boldsymbol{k}_1 + b_2\boldsymbol{k}_2 + \cdots + b_s\boldsymbol{k}_s)\Delta t,$$

where the increments are computed in s *ﬅages* as follows:[6]

$$\begin{aligned}\boldsymbol{k}_1 &= \boldsymbol{f}(t_0, y_0)\\ \boldsymbol{k}_2 &= \boldsymbol{f}(t_0 + c_2\Delta t, y_0 + a_{21}\boldsymbol{k}_1\Delta t)\\ \boldsymbol{k}_3 &= \boldsymbol{f}(t_0 + c_3\Delta t, y_0 + (a_{31}\boldsymbol{k}_1 + a_{32}\boldsymbol{k}_2)\Delta t)\\ &\vdots\\ \boldsymbol{k}_i &= \boldsymbol{f}(t_0 + c_i\Delta t, y_0 + (a_{i1}\boldsymbol{k}_1 + a_{i2}\boldsymbol{k}_2 + \cdots + a_{i,i-1}\boldsymbol{k}_{i-1})\Delta t)\\ &\vdots\\ \boldsymbol{k}_s &= \boldsymbol{f}(t_0 + c_s\Delta t, y_0 + (a_{s1}\boldsymbol{k}_1 + a_{s2}\boldsymbol{k}_2 + \cdots + a_{s,s-1}\boldsymbol{k}_{s-1})\Delta t).\end{aligned}$$

[6] *Caveat lector*: $\boldsymbol{k}_i$ is often defined as $\Delta t\boldsymbol{f}(t_0 + c_i\Delta t, y_0 + \Delta t(a_{i1}\boldsymbol{k}_1 + a_{i2}\boldsymbol{k}_2 + \cdots + a_{i,i-1}\boldsymbol{k}_{i-1}))$. In this case it is an approximation of the increment using the derivative at $t_0 + c_i\Delta t$ rather than an approximation of the derivative itself.

Recall that $\boldsymbol{k}_i$ is an approximation for $\boldsymbol{g}(t_0 + c_i\Delta t)$, the derivative of $\boldsymbol{y}$ at $t_0 + c_i\Delta t$, so that the

$$y_0 + (a_{i1}\boldsymbol{k}_1 + a_{i2}\boldsymbol{k}_2 + \cdots + a_{i,i-1}\boldsymbol{k}_{i-1})\Delta t$$

are themselves linear approximations obtained by weighted averages of approximated derivatives.

Note that each k_i only depends on the k_j for $j < i$, so that they can be computed in order.[7]

Note that all of the methods described above were Runge-Kutta methods. We invite the reader to check that the $\boldsymbol{k}_i$ described in the relevant sections correspond to the following tableaux: Euler's method has Butcher tableau

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array},$$

The midpoint method is described by

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array}$$

and Heun's method by

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}.$$

An example: Kutta's third-order method

We will now consider the Runge-Kutta method given by the following Butcher tableau.

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & -1 & 2 & \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

We have

$$\boldsymbol{y}_1 = \boldsymbol{y}_0 + \left(\frac{\boldsymbol{k}_1}{6} + \frac{2\boldsymbol{k}_2}{3} + \frac{\boldsymbol{k}_3}{6}\right)\Delta t.$$

This is an approximation for

$$\hat{\boldsymbol{y}}_1 = \boldsymbol{y}_0 + \left(\frac{\boldsymbol{g}(t_0)}{6} + \frac{2\boldsymbol{g}\left(t_0 + \frac{\Delta t}{2}\right)}{3} + \frac{\boldsymbol{g}(t_0 + \Delta t)}{6}\right)\Delta t$$

Let us look at the order of $\hat{\boldsymbol{y}}_1$ as an approximation of $\boldsymbol{y}(t_0 + \Delta t)$.

$$\begin{aligned}
\hat{\boldsymbol{y}}_1 &= \boldsymbol{y}_0 + \left(\frac{\boldsymbol{g}(t_0)}{6} + \frac{2\boldsymbol{g}\left(t_0 + \frac{\Delta t}{2}\right)}{3} + \frac{\boldsymbol{g}(t_0 + \Delta t)}{6}\right)\Delta t \\
&= \boldsymbol{y}_0 + \frac{1}{6}\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t_0)\Delta t \\
&\quad + \frac{2}{3}\left(\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t_0) + \frac{\mathrm{d}^2\boldsymbol{y}}{\mathrm{d}t^2}(t_0)\frac{\Delta t}{2} + \frac{\mathrm{d}^3\boldsymbol{y}}{\mathrm{d}t^3}(t_0)\frac{\Delta t^2}{8}\right)\Delta t \\
&\quad + \frac{1}{6}\left(\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t_0) + \frac{\mathrm{d}^2\boldsymbol{y}}{\mathrm{d}t^2}(t_0)\Delta t + \frac{\mathrm{d}^3\boldsymbol{y}}{\mathrm{d}t^3}(t_0)\frac{\Delta t^2}{2}\right)\Delta t \\
&\quad + \mathcal{O}(\Delta t^3)\Delta t \\
&= \boldsymbol{y}_0 + \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t_0)\Delta t + \frac{\mathrm{d}^2\boldsymbol{y}}{\mathrm{d}t^2}(t_0)\frac{\Delta t^2}{2} + \frac{\mathrm{d}^3\boldsymbol{y}}{\mathrm{d}t^3}(t_0)\frac{\Delta t^3}{6} + \mathcal{O}(\Delta t^4) \\
&= \boldsymbol{y}(t_0 + \Delta t) + \mathcal{O}(\Delta t^4),
\end{aligned}$$

[7]This is why we call the method *explicity.* In an implicit method, each k_i can depend on all the k_j, so that you need to solve a system of algebraic equations in order to compute them.

so it looks like this could be a third-order method ($\hat{\boldsymbol{y}}_1 \approx \boldsymbol{y}(t_0 + \Delta t)$ is a fourth-order approximation).

In order for that to work however, we need $\boldsymbol{y}_1 \approx \hat{\boldsymbol{y}}_1$ to be fourth-order, in other words, we need the difference between

$$\frac{\boldsymbol{g}(t_0)}{6} + \frac{2\boldsymbol{g}\left(t_0 + \frac{\Delta t}{2}\right)}{3} + \frac{\boldsymbol{g}(t_0 + \Delta t)}{6}$$

and

$$\frac{\boldsymbol{k}_1}{6} + \frac{2\boldsymbol{k}_2}{3} + \frac{\boldsymbol{k}_3}{6}$$

to be $\mathcal{O}(\Delta t^3)$.

We have $\boldsymbol{k}_1 = \boldsymbol{g}(t_0)$. If we compute $\boldsymbol{k}_2$, we see that it is *only* a second-order approximation for $\boldsymbol{g}\left(t_0 + \frac{\Delta t}{2}\right)$,

The fourth line follows from (3.3), the fifth by taking the Taylor expansion of $\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}}(t, \boldsymbol{y}(t))$ at a function of t.

$$\begin{aligned}
\boldsymbol{k}_2 &= \boldsymbol{f}\left(t_0 + \frac{\Delta t}{2}, \boldsymbol{y}_0 + \boldsymbol{k}_1 \frac{\Delta t}{2}\right) \\
&= \boldsymbol{f}\left(t_0 + \frac{\Delta t}{2}, \boldsymbol{y}_0 + \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t_0)\frac{\Delta t}{2}\right) \\
&= \boldsymbol{f}\left(t_0 + \frac{\Delta t}{2}, \boldsymbol{y}\left(t_0 + \frac{\Delta t}{2}\right) - \frac{\mathrm{d}^2\boldsymbol{y}}{\mathrm{d}t^2}(t_0)\frac{\Delta t^2}{8} + \mathcal{O}(\Delta t^3)\right) \\
&= \boldsymbol{f}\left(t_0 + \frac{\Delta t}{2}, \boldsymbol{y}\left(t_0 + \frac{\Delta t}{2}\right)\right) - \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}}\left(t_0 + \frac{\Delta t}{2}, \boldsymbol{y}\left(t_0 + \frac{\Delta t}{2}\right)\right)\frac{\mathrm{d}^2\boldsymbol{y}}{\mathrm{d}t^2}(t_0)\frac{\Delta t^2}{8} + \mathcal{O}(\Delta t^3) \\
&= \boldsymbol{f}\left(t_0 + \frac{\Delta t}{2}, \boldsymbol{y}\left(t_0 + \frac{\mathrm{d}^2\boldsymbol{y}}{\mathrm{d}t^2}(t_0)\frac{\Delta t}{2}\right)\right) - \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}}(t_0, \boldsymbol{y}(t_0))\frac{\mathrm{d}^2\boldsymbol{y}}{\mathrm{d}t^2}(t_0)\frac{\Delta t^2}{8} + \mathcal{O}(\Delta t^3) \\
&= \boldsymbol{g}(t_0 + \Delta t) - \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}}(t_0, \boldsymbol{y}(t_0))\frac{\mathrm{d}^2\boldsymbol{y}}{\mathrm{d}t^2}(t_0)\frac{\Delta t^2}{8} + \mathcal{O}(\Delta t^3),
\end{aligned}$$

so in order for the method to be third-order, we need this second-order term to be *cancelled* by the error in $\boldsymbol{k}_3$. We can compute this error,

$$\begin{aligned}
\boldsymbol{k}_3 &= \boldsymbol{f}(t_0 + \Delta t, \boldsymbol{y}_0 - \boldsymbol{k}_1\Delta t + 2\boldsymbol{k}_2\Delta t) \\
&= \boldsymbol{f}\left(t_0 + \Delta t, \boldsymbol{y}_0 - \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t_0)\Delta t + 2\boldsymbol{f}\left(t_0 + \frac{\Delta t}{2}, \boldsymbol{y}\left(t_0 + \frac{\Delta t}{2}\right)\right)\Delta t + \mathcal{O}(\Delta t^3)\right) \\
&= \boldsymbol{f}\left(t_0 + \Delta t, \boldsymbol{y}_0 - \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t_0)\Delta t + 2\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}\left(t_0 + \frac{\Delta t}{2}\right)\Delta t + \mathcal{O}(\Delta t^3)\right) \\
&= \boldsymbol{f}\left(t_0 + \Delta t, \boldsymbol{y}_0 - \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t_0)\Delta t + 2\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t_0)\Delta t + \frac{\mathrm{d}^2\boldsymbol{y}}{\mathrm{d}t^2}(t_0)\Delta t^2 + \mathcal{O}(\Delta t^3)\right) \\
&= \boldsymbol{f}\left(t_0 + \Delta t, \boldsymbol{y}(t_0 + \Delta t) + \frac{\mathrm{d}^2\boldsymbol{y}}{\mathrm{d}t^2}(t_0)\frac{\Delta t^2}{2} + \mathcal{O}(\Delta t^3)\right) \\
&= \boldsymbol{f}(t_0 + \Delta t, \boldsymbol{y}(t_0 + \Delta t)) + \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}}(t_0, \boldsymbol{y}(t_0))\frac{\mathrm{d}^2\boldsymbol{y}}{\mathrm{d}t^2}(t_0)\frac{\Delta t^2}{2} + \mathcal{O}(\Delta t^3) \\
&= \boldsymbol{g}(t_0 + \Delta t) + \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}}(t_0, \boldsymbol{y}(t_0))\frac{\mathrm{d}^2\boldsymbol{y}}{\mathrm{d}t^2}(t_0)\frac{\Delta t^2}{2} + \mathcal{O}(\Delta t^3),
\end{aligned}$$

and indeed the second-order error term from $\boldsymbol{k}_3$ cancels with the one from $\boldsymbol{k}_2$ in the weighted average, so that for the whole ſtep we get:

$$\begin{aligned}
\boldsymbol{y}_1 &= \boldsymbol{y}_0 + \left(\frac{\boldsymbol{k}_1}{6} + \frac{2\boldsymbol{k}_2}{3} + \frac{\boldsymbol{k}_3}{6}\right)\Delta t \\
&= \boldsymbol{y}_0 + \Bigg(\frac{\boldsymbol{g}(t_0)}{6} \\
&\quad + \frac{2\boldsymbol{g}\left(t_0 + \frac{\Delta t}{2}\right)}{3} - \frac{2}{3}\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}}(t_0, \boldsymbol{y}(t_0))\frac{\mathrm{d}^2\boldsymbol{y}}{\mathrm{d}t^2}(t_0)\frac{\Delta t^2}{8} \\
&\quad + \frac{\boldsymbol{g}(t_0 + \Delta t)}{6} + \frac{1}{6}\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}}(t_0, \boldsymbol{y}(t_0))\frac{\mathrm{d}^2\boldsymbol{y}}{\mathrm{d}t^2}(t_0)\frac{\Delta t^2}{2} \\
&\quad + \mathcal{O}(\Delta t^3)\Bigg)\Delta t \\
&= \boldsymbol{y}_0 + \Delta t\left(\frac{\boldsymbol{g}(t_0)}{6} + \frac{2\boldsymbol{g}\left(t_0 + \frac{\Delta t}{2}\right)}{3} + \frac{\boldsymbol{g}(t_0 + \Delta t)}{6}\right) + \mathcal{O}(\Delta t^4) \\
&= \hat{\boldsymbol{y}}_1 + \mathcal{O}(\Delta t^4) \\
&= \boldsymbol{y}(t_0 + \Delta t) + \mathcal{O}(\Delta t^4).
\end{aligned}$$

The error on the ſtep is fourth-order and thus the is accurate to the third order.

Closing remarks

Fiddling with Taylor's theorem in order to find a high-order method by trying to make low-order terms cancel out is hard and involves a lot of guesswork. This is where the Runge-Kutta formulation shines: one can check the order of the method by seeing whether the coefficients $\boldsymbol{A}$, $\boldsymbol{b}$, $\boldsymbol{c}$ satisfy the corresponding *order conditions*.

A method has order 1 if and only if it satisfies

$$\sum_{i=1}^{s} b_i = 1.$$

It has order 2 if and only if, in addition to the above equation, it satisfies

$$\sum_{i=1}^{s} b_i c_i = \frac{1}{2}.$$

It has order 3 if and only if, in addition to satisfying the above two equations, it satisfies

$$\begin{cases} \displaystyle\sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3} \\ \displaystyle\sum_{i=1}^{s}\sum_{j=1}^{s} b_i a_{ij} c_j = \frac{1}{6} \end{cases}.$$

It has order 4 if and only if, in addition to satisfying the above four equations, it satisfies

$$\begin{cases} \displaystyle\sum_{i=1}^{s} b_i c_i^3 = \frac{1}{4} \\ \displaystyle\sum_{i=1}^{s}\sum_{j=1}^{s} b_i c_i a_{ij} c_j = \frac{1}{8} \\ \displaystyle\sum_{i=1}^{s}\sum_{j=1}^{s} b_i a_{ij} c_j^2 = \frac{1}{12} \\ \displaystyle\sum_{i=1}^{s}\sum_{j=1}^{s}\sum_{k=1}^{s} b_i a_{ij} a_{jk} c_k = \frac{1}{24} \end{cases}.$$

The number of order conditions explodes with increasing order, and they are not easy to solve. There are cases where only numerical values are known for the coefficients.

We leave the following as an exercise to the reader: characterise all explicit second-order methods with two stages ($s = 2$). Check your result by computing Taylor expansions.