Chapter 8

Runge-Kutta Methods

Main concepts: Generalized collocation method, consistency, order conditions

In this chapter we introduce the most important class of one-step methods that are generically applicable to ODES (1.2). The formulas describing Runge-Kutta methods look the same as those of the collocation methods of the previous chapter, but are abstracted away from the ideas of quadrature and collocation. In particular, the "quadrature nodes" need no longer be distinct and collocation conditions need not hold at each stage.

8.1 The Family of Runge-Kutta Methods

Recall from the previous chapter the definion of a collocation method (7.9)

$$F_i = f\left(y_n + h\sum_{j=1}^{s} a_{ij}F_j\right), \quad i = 1, \dots, s,$$

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i F_i.$$

For collocation methods, the coefficients b_i and a_{ij} are defined by certain integrals of the Lagrange interpolating polynomials associated with a chosen set of quadrature nodes or *abscissae* c_i, $i = 1, \dots, s$.

A natural generalization of collocation methods is obtained by allowing the coeffcients c_i, b_i, and a_{ij}, $i = 1, \dots, s$ to take on arbitrary values, not necessarily related to quadrature rules. In fact, we no longer assume the c_i to be distinct[1] The result is the class of **Runge-Kutta methods**. The formulas above still hold; however, an alternative form introduces the **stage values** Y_i which can be viewed as intermediate values of the solution y at time $t_n + c_i h$ and are defined by $Y_i = y_n + h\sum_j a_{ij}F_j$, so that $F_i = f(Y_i)$ holds for each i. We have,

$$Y_i = y_n + h\sum_{j=1}^{s} a_{ij} f(Y_j), \quad i = 1, \dots, s, \tag{8.1a}$$

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i f(Y_i). \tag{8.1b}$$

Here, s is termed the *number of stages* of the Runge-Kutta method, the b_i, $i = 1, \dots, s$ are the *weights*, and a_{ij} are the *internal coefficients*.

[1] In the formulation for autonomous ODEs, the c_i do not appear explicitly. However, we will assume the internal consistency condition $c_i = \sum_{j=1}^{s} a_{ij}$ to hold throughout.

It is easy to see that with this definition, Euler's method and trapezoidal rule are Runge-Kutta methods. For example Euler's method can be put into the form (8.1b)-(8.1a) with $s = 1$, $b_1 = 1$, $a_{11} = 0$. Trapezoidal rule has $s = 1$, $b_1 = b_2 = 1/2$, $a_{11} = a_{12} = 0$, $a_{21} = a_{22} = 1/2$.

Each Runge-Kutta method generates an approximation of the flow map. To see this, note that from the formula (8.1a) it follows that each Y_i has a functional dependence on y_n and h, $Y_i = Y_i(y_n, h)$. Thus the Runge-Kutta method can be viewed as

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i f(Y_i(y_n, h)),$$

and the method generates a discrete flow-map approximation

$$\Psi_h(y) = y + h \sum_{i=1}^{s} b_i f(Y_i(y, h)).$$

8.1.1 Some Runge-Kutta methods

Let $s = 2$ and choose $c_1 = 0$ and $c_2 = 2/3$, $b_1 = 1/4$, $b_2 = 3/4$. We need $Y_1 \approx y(t_n)$ and $Y_2 \approx Y(t_n + \frac{2}{3}h)$. The latter can be obtained using an Euler step:

$$\begin{aligned} Y_1 &= y_n, \\ Y_2 &= y_n + \frac{2}{3} h f(Y_1), \\ y_{n+1} &= y_n + h\left(\frac{1}{4} f(Y_1) + \frac{3}{4} f(Y_2)\right). \end{aligned} \tag{8.2}$$

The vectors Y_1 and Y_2 are called *stage vectors*. They are intermediate values computed within each step and discarded at the end of the step. Another example with $s = 2$ is the popular method due to Heun:

$$\begin{aligned} Y_1 &= y_n, \\ Y_2 &= y_n + h f(Y_1), \\ y_{n+1} &= y_n + \frac{h}{2}(f(Y_1) + f(Y_2)). \end{aligned} \tag{8.3}$$

The most famous Runge-Kutta method has four stages (this method is sometimes referred to as *the* Runge-Kutta method):

$$\begin{aligned} Y_1 &= y_n, \\ Y_2 &= y_n + \frac{h}{2} f(Y_1), \\ Y_3 &= y_n + \frac{h}{2} f(Y_2), \\ Y_4 &= y_n + h f(Y_3), \\ y_{n+1} &= y_n + h\left(\frac{1}{6} f(Y_1) + \frac{1}{3} f(Y_2) + \frac{1}{3} f(Y_3) + \frac{1}{6} f(Y_4)\right). \end{aligned} \tag{8.4}$$

The formulas above are often represented schematically in a *Butcher table*:

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} \quad = \quad \begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array}$$

where $A = (a_{ij})$, $b = (b_i)$ and $c = (c_i)$. For instance, the methods (8.2), (8.3) and (8.4) listed in this section are represented by

$$\begin{array}{c|cc} 0 & & \\ 2/3 & 2/3 & \\ \hline & 1/4 & 3/4 \end{array}, \qquad \begin{array}{c|cc} 0 & & \\ 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array}, \qquad \begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

It is standard practice to leave the trailing zero elements in the matrix A blank in the Butcher table.

The previous examples of this subsection are all explicit methods. A method for which the matrix $A = (a_{ij})$ is strictly lower triangular is explicit. A simple example of an implicit RK method is the *implicit midpoint rule*

$$\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array}, \quad Y_1 = y_n + \frac{h}{2} f(Y_1), \quad y_{n+1} = y_n + hf(Y_1)$$

which can also be written in simplified form as (just substitute $Y_1 = (y_{n+1} + y_n)/2$ in the above wherever it occurs):

$$y_{n+1} = y_n + hf(\frac{y_{n+1} + y_n}{2}) \tag{8.5}$$

The class of collocation methods from the previous section are a subset of the class of Runge-Kutta methods.

8.1.2 Local accuracy of Runge-Kutta methods

We will now investigate the accuracy of the methods introduced above in terms of their approximation of the numerical solution over one timestep.

To do so, we need to work with higher order derivatives of the function $f(y) : \mathbb{R}^d \to \mathbb{R}^d$. We will denote by f' the Jacobian matrix

$$f' = (\frac{\partial f_i}{\partial y_j}),$$

which can be seen as an linear operator: $\mathbb{R}^d \to \mathbb{R}^d$.

The second derivative

$$f'' = (\frac{\partial^2 f_i}{\partial y_j \partial y_k})$$

is a bilinear operator. We denote its contraction by

$$f''(a, b) = \sum_{j,k} \frac{\partial^2 f_i}{\partial y_j \partial y_k} a_j b_k, \qquad a, b \in \mathbb{R}^d.$$

This contraction is symmetric by the equivalence of mixed partial derivatives, so the order of the arguments a and b is irrelevant. The third derivative is similarly trilinear and symmetric in all three arguments, denoted $f'''(\cdot, \cdot, \cdot)$.

In general, we cannot have an exact expression for $le(y; h)$, but we can approximate this by computing its Taylor series in powers of h. For the continuous dynamics (i.e. the exact solution of (1.13)) we have the Taylor series expansion

$$y(t + h) = y(t) + hy'(t) + \frac{1}{2}h^2 y''(t) + \frac{1}{6}h^3 y'''(t) + \mathcal{O}(h^4).$$

The derivatives can be related directly to the solution itself by using the differential equation

$$\begin{aligned} y'(t) &= f(y(t)) \\ y''(t) &= f'(y(t))y'(t) \\ y'''(t) &= f''(y(t))(y'(t), y'(t)) + f'(y(t))y''(t) \\ &\vdots \end{aligned}$$

Then we can recursively use the differential equation again to obtain

$$\begin{aligned} y' &= f, \\ y'' &= f'f, \\ y''' &= f''(f,f) + f'f'f, \\ &\vdots \end{aligned}$$

where we have dropped the arguments of the various expressions. In all cases, y and its derivatives are assumed to be evaluated at t and f and its derivatives at y.

Alternatively, we can write

$$\Phi_h(y) = y + hf + \frac{h^2}{2} f'f + \frac{h^3}{6} \left[f''(f,f) + f'ff \right] + \mathcal{O}(h^4) \tag{8.6}$$

The same procedure can be carried out for the Runge-Kutta method itself. For example, for Euler's method, we have

$$\Psi_h(y) = y + hf(y)$$

(that is it!). This means that the discrete and continuous series match to $\mathcal{O}(h^2)$ and the local error expansion can be written

$$le(y,h) = \frac{h^2}{2} f'f + \mathcal{O}(h^3).$$

again, we have dropped the argument y of f for notational compactness.

For trapezoidal rule, and other implicit schemes, the derivatives need to be computed by implicit differentiation. For simplicity, with y fixed, write $z = z(h) = \Psi_h(y)$. Then $z(h)$ must satisfy the implicit relation

$$z = y + \frac{h}{2} \left(f(y) + f(z) \right)$$

Observe that $z(0) = y$. Next we differentiate the expression for z,

$$z' = dz/dh = \frac{1}{2} \left(f(y) + f(z) \right) + \frac{h}{2} f'(z) z', \tag{8.7}$$

(Note that z does not satisfy the differential equation, so we *cannot* replace z' here by $f(z)$!) For $h = 0$, the last term of (8.7) vanishes and we have

$$z'(0) = f(y).$$

Differentiate (8.7) once more with respect to h to obtain

$$z'' = \frac{1}{2} f'(z) z' + \frac{1}{2} f'(z) z' + \frac{h}{2} (f''(z)(z',z') + f'(z) z''), \tag{8.8}$$

for $h = 0$, the last term of (8.8) drops out and we have

$$z''(0) = f'(z(0)) z'(0) = f'(y) f(y).$$

Using these expressions, we may write the first few terms of the Taylor series in h of $\Psi_h(y)$:

$$\Psi_h(y) = z(h) = z(0) + hz'(0) + \frac{h^2}{2} z''(0) + \ldots = y + hf + \frac{h^2}{2} f'f + \ldots \tag{8.9}$$

Comparing this with the expansion for the exact solution (8.6) we see that the first few terms are identical. Thus the local error vanishes to *at least* $\mathcal{O}(h^3)$. To get the next term in the local error expansion, we differentiate (8.8):

$$\begin{aligned} z''' = {} & f''(z)(z',z') + f'(z) z'' + (1/2) \left(f''(z)(z',z') + f'(z) z'' \right) + \\ & \frac{h}{2} \left(f'''(z)(z',z',z') + 2f''(z)(z',z'') + f''(z)(z',z'') + f'(z) z''' \right), \end{aligned}$$

which evaluates to

$$z'''(0) = (3/2)\left(f''(f,f) + f'f'f\right),$$

which means that the expansions (8.9) and (8.6) *do not* match in the term of 3rd order: for trapezoidal rule, we have

$$le(y,h) = -\frac{1}{12}\left(f''(f,f) + f'f'f\right)h^3 + \mathcal{O}(h^4).$$

8.1.3 Order conditions for Runge-Kutta methods

The methods discussed above were relatively simple in that they could be written without using the Runge-Kutta formalism. Nonetheless the same procedure can be used to determine order conditions in the general case; we just need to compute the derivatives of $\Psi_h(y) = z(h)$. To do this we write the general RK method in the following form:

$$z(h) = y + h\sum_{i=1}^{s} b_i F_i,$$

where

$$\begin{aligned} F_i &= f(Y_i), && (8.10)\\ Y_i &= y + h\sum_{j=1}^{s} a_{ij}F_j. && (8.11) \end{aligned}$$

Now the calculations required are straightforward, viewing F_i, Y_i, and z as functions of h with y assumed fixed (i.e. the prime denotes differentiation with respect to h):

$$z' = \sum_{i=1}^{s} b_i F_i + h\sum_{i=1}^{s} b_i F_i'.$$

which immediately tells us that

$$z'(0) = \sum_{i=1}^{s} b_i F_i(0) = \sum_{i=1}^{s} b_i f(Y_i(0)) = (\sum_{i=1}^{s} b_i)f(y).$$

For the method to have order $p = 1$, we must have

$$\sum_{i=1}^{s} b_i = 1. \qquad (8.12)$$

This is the first of what are termed the “order conditions”. Any RK method satisfying (8.12) is consistent, and by Theorem 6.2.1, convergent independent of the a_{ij}. To get the next one, we compute the second derivative of z:

$$\begin{aligned} z'' &= \sum_{i=1}^{s} b_i F_i' + \sum_{i=1}^{s} b_i F_i' + h\sum_{i=1}^{s} b_i F_i'' && (8.13)\\ &= 2\sum_{i=1}^{s} b_i F_i' + h\sum_{i=1}^{s} b_i F_i''. \end{aligned}$$

$$(8.14)$$

From (8.10)-(8.11) we obtain

$$\begin{aligned} F_i' &= f'(Y_i)Y_i',\\ Y_i' &= \sum_{j=1}^{s} a_{ij}F_j + h\sum_{j=1}^{s} a_{ij}F_j'. \end{aligned}$$

At $h = 0$, we have

$$Y_i'(0) = \sum_{j=1}^{s} a_{ij} F_j(0) = \sum_{j=1}^{s} a_{ij} f(y)$$

Let $c_i = \sum_{j=1}^{s} a_{ij}$, so $Y_i'(0) = c_i f(y)$. Then

$$F_i'(0) = c_i f'(y) f(y)$$

and

$$z''(0) = 2(\sum_{i=1}^{s} b_i c_i) f'(y) f(y).$$

For this term to cancel with y'', we must have

$$\sum_{i=1}^{s} b_i c_i = \frac{1}{2}. \tag{8.15}$$

For $p = 2$, we must therefore have *both* conditions (8.12),(8.15).

To continue this process, we would need to differentiate (8.13). This is a bit tedious. To make it a little simpler, we will not bother to compute terms which vanish when we evaluate at $h = 0$:

$$z''' = 3\sum_{i=1}^{s} b_i F_i'' + h(\ldots).$$

Then

$$F_i'' = f''(Y_i)(Y_i', Y_i') + f'(Y_i) Y_i''$$
$$Y_i'' = 2\sum_{j=1}^{s} a_{ij} F_j' + h(\ldots).$$

Now we evaluate everything at $h = 0$ and find

$$z'''(0) = 3\sum_{i=1}^{s} b_i (f''(Y_i(0))(Y_i'(0), Y_i'(0)) + f'(Y_i(0)) Y_i''(0))$$

Recall from the earlier work that $Y_i(0) = y$, $Y_i'(0) = c_i f(y)$, and we also have

$$Y_i'' = 2\sum_{j=1}^{s} a_{ij} F_j'(0) = 2\sum_{j=1}^{s} a_{ij} c_j f'(y) f(y)$$

so finally,

$$z'''(0) = 3\sum_{i=1}^{s} b_i \left(c_i^2 f''(y)(f(y), f(y)) + 2(\sum_{j=1}^{s} a_{ij} c_j) f'(y) f'(y) f(y) \right)$$

which means that, for the method to have order $p = 3$, we must have *two* more conditions satisfied (in addition to (8.12) and (8.15)):

$$\sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}, \qquad \sum_{i=1}^{s} \sum_{j=1}^{s} b_i a_{ij} c_j = \frac{1}{6}.$$

The number of order conditions explodes with increasing order. For order 4, there are four additional conditions or 8 in all, there are 17 at order 5, 37 at order 6, 85 at order 7, and 200 at order 8. The task of enumerating the conditions can be greatly simplified by using techniques from graph theory ("Butcher trees," the ingenious invention of one J. C. Butcher from New Zealand who has pioneered much of the study of Runge-Kutta methods). In order to actually find methods, it is not enough to work out the order conditions: we need to identify sets of coefficients, a_{ij} and b_i which allow us to exactly satisfy a large system of multivariate polynomials. Butcher also developed a set of simplifying conditions, which greatly ease the development of higher order methods.

There exist explicit methods of order $p = s$ for up to four stages ($s \leq 4$). It can be shown that the conditions for order 5 cannot be solved with only 5 stages under the restriction that the method be explicit. ($a_{ij} = 0,\ j \geq i$). Obviously, from the previous chapter, it is clear that an implicit s-stage method can reach the maximum order $p = 2s$.

8.2 Exercises

1. Compute the leading terms of the local error expansions for the methods (8.2) and (8.3).

2. Show that the "traditional Runge-Kutta" method (8.4) satisfies all the conditions mentioned for order 3, and, additionally, the four equations

$$\sum_{i=1}^{s} b_i c_i^3 = \frac{1}{4}, \tag{8.16}$$

$$\sum_{i=1}^{s}\sum_{j=1}^{s} b_i c_i a_{ij} c_j = \frac{1}{8}, \tag{8.17}$$

$$\sum_{i=1}^{s}\sum_{j=1}^{s} b_i a_{ij} c_j^2 = \frac{1}{12}, \tag{8.18}$$

$$\sum_{i=1}^{s}\sum_{j=1}^{s}\sum_{k=1}^{s} b_i a_{ij} a_{jk} c_k = \frac{1}{24}. \tag{8.19}$$

   which are the order 4 conditions.

3. Determine the order of the two-stage Gauss collocation method from Chapter 7.

4. Choose a Runge-Kutta method of order at least two and demonstrate the order by integrating the (nonlinear, nonscalar, smooth) initial value problem of your choice over a fixed interval with successively smaller step sizes until the predicted convergence rate is observed. (In other words, choose an interval $[t_0, t_0 + T]$ and a number of time steps $N_k = 2^k N_0$, $k = 0, 1, \ldots$ for some $N_0 > 0$. For each k, choose a time step $h_k = T/N_k$ and compute the solution over N_k time steps. Denote the solution at the end of the interval by $\bar{y}^{N_k}$. Approximate the error in the kth solution by comparing it with the $(k+1)$th, i.e. $\mathrm{err}_k = \|\bar{y}^{N_{k+1}} - \bar{y}^{N_k}\|$. Continue in this way until you observe the predicted convergence rate.)