Numerical Analysis Hilary Term 2024
Lecture 14–15: Runge–Kutta methods

Runge–Kutta methods: Runge–Kutta (RK) methods form a broad class of algorithms for the numerical solution of IVPs. The idea is to achieve a higher order of accuracy than 1 (as in Euler's method) by approximately finding the derivative f at 'intermediate values' (called *stages*) between x_n and x_{n+1} (and similarly $\mathbf{y}_n$ and $\mathbf{y}_{n+1}$).

The class includes both explicit and implicit schemes. When applications call for an integrator with some kind of stability or conservation property, there usually exists a suitable RK method. In particular, RK methods can be made arbitrarily high-order without the loss of stability.

Definition 1. *The family of* s-stage Runge–Kutta methods *is defined by*

$$\mathbf{\Psi}(x, \mathbf{y}, h, \mathbf{f}) = \mathbf{y} + h \sum_{i=1}^{s} b_i \mathbf{k}_i \,, \tag{1}$$

where the stages $\mathbf{k}_i$ *(recall that* $\mathbf{y} \in \mathbb{R}^d$*, and also* $\mathbf{k}_i \in \mathbb{R}^d$*) are the solutions of the coupled system of (generally nonlinear) equations*

$$\mathbf{k}_i := \mathbf{f}(x + c_i h, \mathbf{y} + h \sum_{j=1}^{s} a_{ij} \mathbf{k}_j) \,, \quad i = 1, \ldots, s \,. \tag{2}$$

The coefficients $\{c_i\}_{i=1}^s$ *are always given by*

$$c_i := \sum_{j=1}^{s} a_{ij} \quad i = 1, \ldots, s \,.$$

Definition 2. *The coefficients of a Runge–Kutta method are commonly summarized in a* Butcher tableau[1]

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^\top \end{array} .$$

Example 3. *The explicit Euler method, the implicit Euler method, and the implicit midpoint rule are Runge–Kutta methods. Their Butcher tables are*

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}, \quad \begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}, \quad \text{and} \quad \begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array}, \quad \text{respectively.}$$

Example 4. *Let us derive two famous RK methods using Taylor expansions. Consider the following family of methods (where for simplicity we assume* $d = 1$*, i.e.,* $\mathbf{y}_n \in \mathbb{R}$ *is a scalar and write* y_n, k_i *etc):*

$$y_{n+1} = y_n + h(b_1 k_1 + b_2 k_2), \tag{3}$$

[1]The use of this tableau was introduced by J. C. Butcher in 1963 with the article *Coefficients for the study of Runge–Kutta integration processes.*

where

$$k_1 = f(x_n, y_n), \tag{4}$$
$$k_2 = f(x_n + c_2 h, y_n + a_{21} h k_1), \tag{5}$$

and where the parameters b_1, b_2, c_2 *and* a_{21} *are to be determined.*[2] *The method is consistent iff* $b_1 + b_2 = 1$. *Further conditions on the parameters are obtained by attempting to maximise the order of accuracy of the method. Indeed, expanding the consistency error* τ_n *of (3)–(5) in powers of* h, *after some algebra we obtain*

$$\tau_n = \frac{1}{2}hy''(x_n) + \frac{1}{6}h^2 y'''(x_n) - b_2 h[c_2 f_x + a_{21} f_y f] - b_2 h^2 \left[\frac{1}{2}c_2^2 f_{xx} + c_2 a_{21} f_{xy} f + \frac{1}{2}a_{21}^2 f_{yy} f^2\right] + \mathcal{O}(h^3).$$

Here we have used the abbreviations $f = f(x_n, y(x_n))$, $f_x = \frac{\partial f}{\partial x}(x_n, y(x_n))$, *etc. On noting that* $y'' = f_x + f_y f$, *it follows that* $\tau_n = \mathcal{O}(h^2)$ *for any* f *provided that*

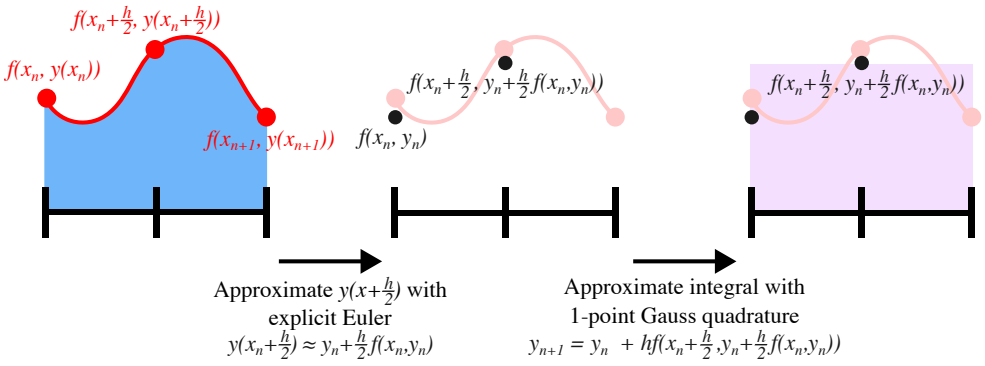
$$c_2 b_2 = a_{21} b_2 = \frac{1}{2},$$

which implies that if $a_{21} = c_2$, $b_2 = 1/(2a_2)$ *and* $b_1 = 1 - 1/(2c_2)$ *then the method is second-order accurate; while this still leaves one free parameter,* c_2, *it is easy to see that no choice of the parameters will make the method generally third-order accurate. There are two well-known examples of second-order explicit Runge–Kutta methods of the form (3), (5):*

a) ***The modified Euler method:*** *In this case we take* $c_2 = \frac{1}{2}$ *to obtain*

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\, f\left(x_n + \frac{1}{2}h, \mathbf{y}_n + \frac{1}{2}hf(x_n, \mathbf{y}_n)\right);$$
$$\boldsymbol{\Psi}(x, \mathbf{y}, h, f) = x + hf\left(x + \frac{h}{2}, \mathbf{y} + \frac{h}{2}f(x, \mathbf{y})\right). \tag{6}$$

The figure below illustrates this method.



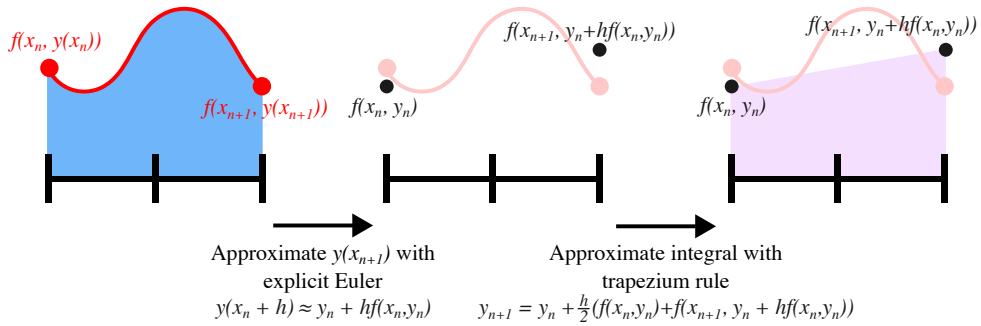
[2] We note in passing that Euler's explicit method is a member of this family of methods, corresponding to $b_1 = 1$ and $b_2 = 0$. However we are now seeking methods that are at least second-order accurate.

b) ***The improved Euler method:*** *This is obtained by choosing* $c_2 = 1$*, which gives*

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{1}{2}h\left[f(x_n, \mathbf{y}_n) + f(x_n + h, \mathbf{y}_n + hf(x_n, \mathbf{y}_n))\right].$$

$$\boldsymbol{\Psi}(x, \mathbf{y}, h, \mathbf{f}) = \mathbf{y} + \frac{h}{2}\mathbf{f}(x, \mathbf{y}) + \frac{h}{2}\mathbf{f}\left(x + h, \mathbf{y} + h\mathbf{f}(x, \mathbf{y})\right). \tag{7}$$

Here's the corresponding figure.



For these two methods it is easily verified by Taylor series expansion that the consistency error is of the form, respectively,

$$\begin{aligned}\tau_n &= \frac{1}{6}h^2\left[f_y F_1 + \frac{1}{4}F_2\right] + \mathcal{O}(h^3),\\ \tau_n &= \frac{1}{6}h^2\left[f_y F_1 - \frac{1}{2}F_2\right] + \mathcal{O}(h^3),\end{aligned}$$

where

$$F_1 = f_x + f f_y \quad \textit{and} \quad F_2 = f_{xx} + 2f f_{xy} + f^2 f_{yy}.$$

The family (3)–(5) is referred to as the class of explicit two-stage explicit Runge–Kutta methods.

Their Butcher tables read

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array} \quad \textit{and} \quad \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array},$$

respectively.

Example 5. *Last but not least, RK4, a 4-stage 4th-order explicit Runge–Kutta method is a famous and very popular choice.*

$$\boldsymbol{\Psi}(x, \mathbf{y}, h, \mathbf{f}) = \mathbf{y} + \frac{h}{6}\left(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4\right),$$

where

$$\begin{aligned}\mathbf{k}_1 &= f(x,\mathbf{y}), \qquad \mathbf{k}_2 = f\left(x+\frac{1}{2}h, \mathbf{y}+\frac{1}{2}h\mathbf{k}_1\right),\\ \mathbf{k}_3 &= f\left(x+\frac{1}{2}h, \mathbf{y}+\frac{1}{2}h\mathbf{k}_2\right), \quad \mathbf{k}_4 = f(x+h, \mathbf{y}+h\mathbf{k}_3).\end{aligned}$$

Its Butcher table reads

0	*0*	*0*	*0*	*0*
1/2	*1/2*	*0*	*0*	*0*
1/2	*0*	*1/2*	*0*	*0*
1	*0*	*0*	*1*	*0*
	1/6	*2/6*	*2/6*	*1/6*

.

We now turn to the convergence properties of RK methods. As mentioned at the end of the last lecture, convergence in the sense that $\mathbf{y}$ converges to the exact solution as $h \to 0$ holds under mild assumptions[3]. We shall thus look at two questions:

- What is the order of accuracy? (Which, as last lecture's theorem showed, is essentially the same as the consistency error)
- Stability: How small does h need to be?

We treat these questions separately.

Order of accuracy/consistency order of RK methods.

It is convenient at this point to restrict our attention to autonomous IVPs. (Recall that a nonautonomous system can always be made autonomous by increasing its dimension.) The process of making an IVP autonomous commutes with Runge–Kutta discretisation if and only if

$$\sum_{i=1}^{s} b_i = 1, \quad c_k = \sum_{j=1}^{s} a_{kj} \quad k = 1, \ldots, s\,,$$

which we assume henceforth. (In other words, if these conditions hold, the RK discretisation of the autonomised system is the autonomisation of the RK discretisation of the original problem.)

By computing appropriate Taylor expansions (as in Example 4), it is possible to derive algebraic conditions the Runge–Kutta coefficients must satisfy for the method to have a targeted consistency order. For example:

Lemma 6. *A Runge–Kutta method is consistent if and only if* $\sum_{i=1}^{s} b_i = 1$. *If the condition*

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

[3]Strictly speaking, it is not trivial to apply that theorem to general RK methods, which cannot be written easily, as writing a step as $\boldsymbol{\Psi}(s, \mathbf{y}, h, \mathbf{f})$ is not trivial, especially for implicit methods. Nonetheless an analogous result still holds; we refer to Section II.3 of the book by Hairer, Norsett, and Wanner for details.

is also satisfied, the Runge–Kutta method has consistency order 2, and if the conditions

$$\sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3} \quad \textit{and} \quad \sum_{i=1}^{s} b_i \sum_{j=1}^{s} a_{ij} c_j = \frac{1}{6}$$

are also satisfied, the Runge–Kutta method has consistency order 3.

Nonexaminable: The following table indicates the number of conditions as described above that a Runge–Kutta method must satisfy to have order p:

p	1	2	3	4	5	6	7	8	9	10	20
#conditions	1	2	4	8	17	37	85	200	486	1205	20247374

.

The number of stages of a Runge–Kutta method provides an interesting upper bound on its consistency order.

Lemma 7. *The (consistency) order p of an s-stage Runge–Kutta method is bounded by $p \leq 2s$. If the Runge–Kutta method is explicit, then $p \leq s$.*

To evolve a numerical solution from x_n to x_{n+1} with a Runge–Kutta method, one needs to compute the stages $\mathbf{k}_i$. If the Runge–Kutta method is explicit, these stages can be computed sequentially (and at a low-cost) starting from $\mathbf{k}_1$ (a Runge–Kutta method is explicit if $a_{ij} = 0$ whenever $j \geq i$, i.e. the matrix $\mathbf{A}$ is strictly lower-triangular). An example of this is the explicit Euler method. If $\mathbf{A}$ is lower-triangular (i.e. possibly $a_{ii} \neq 0$), then the scheme is said to be *diagonally-implicit*; one can compute the stages $\mathbf{k}_i$ sequentially, solving a sequence of nonlinear problems. The implicit Euler and implicit midpoint rules are examples of diagonally-implicit RK methods. Finally, if $\mathbf{A}$ enjoys neither of these structures, the RK method is said to be fully implicit; one must solve a large coupled nonlinear system for all stages simultaneously.

It is possible to construct Runge–Kutta methods that achieve maximal order. So-called *Butcher barriers* quantify the minimal amount of stages that an explicit Runge–Kutta method of order p requires (nonexaminable):

p	1	2	3	4	5	6	7	8	≥ 9
minimal value of s	1	2	3	4	6	7	9	11	$\geq p+3$

.

This implies that a Runge–Kutta method that has maximal order must be implicit.

Stability of Runge–Kutta methods

As discussed above, if the step size h is sufficiently small the computed solution with a RK method tends to the exact solution. This does not tell us how small h needs to be. Indeed, we have seen that numerical methods for IVPs may encounter stability issues in the last lecture. We now study the appropriate values of h; taking h too small means we'll do many ($O(h^{-1})$) steps to find $y(X)$, which is undesirable. For simplicity, we only consider autonomous ODEs.

Definition 8. *A* fixed point *of $\mathbf{y}' = \mathbf{f}(\mathbf{y})$ is a point $\mathbf{y}^*$ such that $\mathbf{f}(\mathbf{y}^*) = \mathbf{0}$. A fixed point $\mathbf{y}^*$ is* asymptotically stable *(or attractive) if there exists a ball $B_\delta(\mathbf{y}^*)$ (of radius $\delta > 0$ and centered at $\mathbf{y}^*$) such that, whenever $\mathbf{y}_0 \in B_\delta(\mathbf{y}^*)$, the solution to $\mathbf{y}' = \mathbf{f}(\mathbf{y})$, $\mathbf{y}(0) = \mathbf{y}_0$ satisfies $\lim_{x\to\infty} \mathbf{y}(x) = \mathbf{y}^*$.*

Theorem 9. *A fixed point* $\mathbf{y}^*$ *of an autonomous ODE is asymptotically stable if*

$$\sigma\left(\mathbf{Df}(\mathbf{y}^*)\right) \subset \mathbb{C}^- := \{z \in \mathbb{C} : \mathrm{Re} z < 0\}\,,$$

where $\sigma\left(\mathbf{Df}(\mathbf{y}^*)\right)$ *denotes the set of eigenvalues of the matrix* $\mathbf{Df}(\mathbf{y}^*)$.

This theorem implies that, to study the asymptotic stability of $\mathbf{y}^*$, we can restrict our considerations to the linearised ODE $\mathbf{y}' = \mathbf{Df}(\mathbf{y}^*)(\mathbf{y} - \mathbf{y}^*)$, that is, we can restrict our attention to linear ODEs. To further simplify the analysis, we restrict our attention to a single eigenvalue, yielding the *Dahlquist test equation*

$$y' = zy\,, \quad y(0) = 1\,, \quad \text{and} \quad \mathrm{Re} z < 0\,. \tag{8}$$

Clearly, the solution of the Dahlquist test equation is $y(x) = \exp(zx)$, which satisfies $\lim_{x\to\infty} y(x) = 0$. Therefore, $y^* = 0$ is an attractive fixed point.

In what follows we ask the question: Is the fact that $y \to 0$ respected by the computed solution obtained by a RK method? This can be seen as a 'minimum requirement' for a method to be reasonable; if the exact solution is tending to 0 but the computed one isn't, that surely cannot be a good method. Conversely, if a method (with a given step size h) does satisfy $y_n \to 0$, then it tends to work well also for more general problems with similar h, in addition to respecting the asymptocic behavior of fixed points as discussed above; this is why we look specifically at the Dahlquist test equation (8).

The solution of the Dahlquist test equation obtained with a Runge–Kutta method has a special structure:

Definition 10. *Let* $\mathbf{\Psi}$ *be a Runge–Kutta method. The function*

$$S : \mathbb{C} \to \mathbb{C}\,, \quad z \mapsto S(z) := \mathbf{\Psi}(0, 1, 1, f : y \mapsto zy)\,,$$

is called the stability function *of* $\mathbf{\Psi}$ *(defined in (1)). To shorten the notation, we henceforth write* $\mathbf{\Psi}(0, 1, 1, z)$ *instead of* $\mathbf{\Psi}(0, 1, 1, f : y \mapsto zy)$.

Lemma 11. *If* $\mathbf{\Psi}$ *is a Runge–Kutta method, then* $\mathbf{\Psi}(0, \ell, h, z) = \mathbf{\Psi}(0, 1, 1, zh)\ell$.

Theorem 12. *Let* $\{y_k\}_{k\in\mathbb{N}}$ *be the Runge–Kutta solution to the Dahlquist test equation obtained with a time step* $h > 0$. *Then,* $y_k = S(zh)^k$.

Proof. By direct computation, we can see that

$$y_1 = \mathbf{\Psi}(0, 1, h, z) = \mathbf{\Psi}(0, 1, 1, zh) = S(zh)$$

and that

$$y_2 = \mathbf{\Psi}(0, y_1, h, z) = \mathbf{\Psi}(0, 1, 1, zh)y_1 = S(zh)y_1 = S(zh)^2\,.$$

Therefore, we conclude that $y_k = S(h\lambda)^k$. □

As discussed, it is desirable that the discrete solution $\{y_k\}_{k\in\mathbb{N}}$ satisfies $\lim_{k\to\infty} y_k = 0$, mimicking the behavior of the exact solution to the Dahlquist test equation. When this happens, we say that $\{y_k\}_{k\in\mathbb{N}}$ is *asymptotically stable*.

Definition 13. *The region in the complex plane*

$$S_{\boldsymbol{\Psi}} := \{z \in \mathbb{C} : |S(z)| < 1\}$$

is called the stability region *of the Runge–Kutta method. Clearly,* $\{y_k\}_{k\in\mathbb{N}}$ *is asymptotically stable if* $zh \in S_\Psi$.

It is not so difficult to see that the stability function of an explicit Runge–Kutta method is a polynomial, which implies that $S_{\boldsymbol{\Psi}}$ is bounded. Therefore, the numerical approximation computed with an explicit Runge–Kutta method cannot be asymptotically stable if the time step h is too large. This is what we saw in our numerical experiments in the last lecture. However, the stability function of an implicit Runge–Kutta method is a rational function, and hence may not suffer from this limitation.

Definition 14. *A Runge–Kutta method is said to be* A-stable[4] *if* $\mathbb{C}^- \subset S_{\boldsymbol{\Psi}}$.

A-stability guarantees that $\{y_k\}_{k\in\mathbb{N}}$ will eventually converge to zero. However, the decay can be very slow compared to that of the exact solution.

Example 15. *Let* $\{y_k\}$ *be the approximate solution to the Dalhquist test equation obtained with the implicit midpoint rule and a fixed step size* h. *By direct computation, we can see that stability function of the implicit midpoint rule is*

$$S(z) = \frac{1 + z/2}{1 - z/2}.$$

The exact solution converges exponentially to zero with rate $\mathrm{Re}z$*; the smaller (more negative) the* $\mathrm{Re}z$*, the quicker the convergence. On the other hand,* $\{y_k\}$ *is a geometric sequence with ratio* $S(zh)$. *This also converges to zero, but the more negative the* $\mathrm{Re}z$*, the closer* $|S(zh)|$ *to* 1*, and the slower the decay of* $\{y_k\}$. *This implies that, if* $\mathrm{Re}z \ll 0$*, the qualitative behavior of* $\{y_k\}$ *can be very different from the one of the exact solution.*

Therefore, if the initial value problem has a strongly attractive fixed point, it is advisable to further ensure that $\lim_{\mathrm{Re}z\to-\infty} |S(z)| = 0$.

Definition 16. *An A-stable method that further satisfies* $\lim_{\mathrm{Re}z\to-\infty} |S(z)| = 0$ *is said to be* L-stable.

One can verify that the implicit Euler method is L-stable, but it is not the only one.

The remainder is nonexaminable.

Construction of explicit/implicit RK methods To construct explicit Runge–Kutta methods, we start by recalling that the analytic solution of

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}), \qquad \mathbf{y}(x_0) = \mathbf{y}_0, \tag{9}$$

[4]This concept was introduced by G. Dahlquist in 1963 with the article *A special stability problem for linear multistep methods.*

is given by the (implicit) formula

$$\mathbf{y}(x+h) = \mathbf{y}(x) + \int_x^{x+h} \mathbf{f}(\tau, \mathbf{y}(\tau))\,\mathrm{d}\tau = \mathbf{y}(x) + h\int_0^1 \mathbf{f}(x+h\tau, \mathbf{y}(x+h\tau))\,\mathrm{d}\tau\,.$$

Approximating the latter integral with a quadrature rule on $[0,1]$ with s nodes $c_1,\dots,c_s$ and weights $b_1,\dots,b_s$ returns

$$\mathbf{y}(x+h) \approx \mathbf{y}(x) + h\sum_{i=1}^{s} b_i \mathbf{f}(x+c_i h, \mathbf{y}(x+c_i h))\,. \tag{10}$$

Note that this approximation requires the values $\mathbf{y}(x+c_i h)$. To make the method explicit, we approximate the values $\mathbf{y}(x_0+c_i h)$ with explicit Runge–Kutta methods we already know. This way, we can construct s-stage explicit Runge–Kutta methods by induction.

Example 17. *If we choose the 1-point Gauss quadrature rule, that is,*

$$\mathbf{y}(x+h) \approx \mathbf{y}(x) + h\mathbf{f}(x+h/2, \mathbf{y}(x+h/2)) \tag{11}$$

and approximate $\mathbf{y}(x+h/2)$ *with the explicit Euler method, the resulting scheme is (6).*

Example 18. *If we use the trapezium rule, that is,*

$$\mathbf{y}(x+h) \approx \mathbf{y}(x) + \frac{h}{2}\mathbf{f}(x, \mathbf{y}(x)) + \frac{h}{2}\mathbf{f}(x+h, \mathbf{y}(x+h))\,,$$

and approximate $\mathbf{y}(x+h)$ *with the explicit Euler method, the resulting scheme is (7).*

A similar approach leads to the most famous explicit Runge–Kutta method *RK4*.

We have seen that s-stage explicit Runge–Kutta methods have at most order s. Next, we construct s-stage implicit Runge–Kutta methods whose order is at least s.

Definition 19. *Let* $c_1,\dots,c_s \in [0,1]$ *be (pairwise distinct) collocation points. The corresponding collocation method is the one-step method defined by*

$$\boldsymbol{\Psi}(x, \mathbf{y}, h, \mathbf{f}) = \tilde{\mathbf{y}}(h)\,,$$

where $\tilde{\mathbf{y}}$ *is the unique polynomial of degree* s *that satisfies*

$$\tilde{\mathbf{y}}(0) = \mathbf{y} \quad \textit{and} \quad \tilde{\mathbf{y}}'(c_i h) = \mathbf{f}(x+c_i h, \tilde{\mathbf{y}}(c_i h))\,, \quad \textit{for } i = 1,\dots,s\,. \tag{12}$$

Lemma 20. *Let* Q *be the highest-order quadrature rule on* $[0,1]$ *that can be constructed using the nodes* $c_1,\dots,c_s$*, and let* p_Q *be its order (*p_Q *= 1 + the maximal degree of polynomials it integrates exactly). If* $\mathbf{f}$ *is sufficiently smooth and* $h > 0$ *is sufficiently small, the collocation method associated to* $c_1,\dots,c_s$ *has order* p_Q*.*

Corollary 21. *If* $\mathbf{f}$ *is sufficiently smooth and* $h > 0$ *is sufficiently small, the order of the collocation method associated to* $c_1,\dots,c_s$ *is at least* s *and at most* $2s$ *(Gauss quadrature).*

It is not obvious, but collocation methods are indeed Runge–Kutta methods.

Lemma 22. *Collocation methods are Runge–Kutta methods. Their coefficients are*

$$a_{ij} = \int_0^{c_i} L_j(\tau)\,\mathrm{d}\tau\,, \quad b_i = \int_0^1 L_i(\tau)\,\mathrm{d}\tau\,, \tag{13}$$

where $\{L_i\}_{i=1}^s$ *are the Lagrange polynomials associated to* $c_1, \ldots, c_s$*.*

The Gauss collocation methods form a family of arbitrarily high-order A-stable methods whose stability region is exactly $\mathbb{C}^-$.

An example of a family of L-stable RK methods is the Gauss–Radau family. This is a family of collocation methods where the final quadrature point is fixed to $c_s = 1$ and the remaining points $c_1, \ldots, c_{s-1}$ are chosen to obtain an associated quadrature rule of maximal order $2s - 1$.

For more detailed discussions on RK methods, we refer to the books

- Süli and Mayer, "Introduction to Numerical Analysis"
- Hairer, Norsett, and Wanner, "Solving Ordinary Differential Equations"
- Butcher, "Numerical Methods for Ordinary Differential Equations"