Numerical Solution of Differential Equations I

Alberto Paganini

April 19, 2018

Contents

1 Initial value problems

A *first-order ordinary differential equation* (ODE) is an equation of the form[1]

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}) \,,$$

where the righthand side is a function $\mathbf{f} : I \times D \to \mathbb{R}^d$ ($d \in \mathbb{N}^+$) defined on the cartesian product of a *time interval* I (usually $I = [t_0, T]$) with an open subset $D \subset \mathbb{R}^d$ (that is often called *state space* or *phase space*).

An *initial value problem* is an ODE with an initial condition, that is,

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}) \,, \qquad \mathbf{y}(t_0) = \mathbf{y}_0 \,. \tag{1.1}$$

In the course **A1: Differential Equations 1**, we have learned that Picard's Theorem gives precise conditions to ensure that (1.1) admits a unique solution.

Theorem 1.1 (Picard's Theorem)**.** *Suppose that* $\mathbf{f}$ *is continuous in a neighborhood* $U \subset \mathbb{R}^{1+d}$ *of* $(t_0, \mathbf{y}_0)$ *that contains the cylinder*

$$R = \{(t, \mathbf{y}) : t_0 \le t \le T_M, \|\mathbf{y} - \mathbf{y}_0\| \le Y_M\} \,,$$

where $T_M > t_0$ *and* $Y_M > 0$ *are constants. Suppose also that there exists a positive constant* L *such that*

$$\|\mathbf{f}(t, \mathbf{y}) - \mathbf{f}(t, \mathbf{z})\| \le L\|\mathbf{y} - \mathbf{z}\| \tag{1.2}$$

holds whenever $(t, \mathbf{y})$ *and* $(t, \mathbf{z})$ *lie in* R*. Finally, letting*

$$M \coloneqq \max\{\|\mathbf{f}(t, \mathbf{y})\| : (t, \mathbf{y}) \in R\} \,,$$

suppose that $M(T_M - t_0) \le Y_M$*. Then, there exists a unique continuously differentiable function*

$$[t_0, T_M] \ni t \mapsto \mathbf{y}(t) \in \mathbb{R}^d$$

that is the solution to (1.1)*.*

Proof. See the lecture notes of the course **A1: Differential Equations 1**. □

Remark 1.1. *We briefly summarize some general facts about Theorem 1.1.*

- *Condition* (1.2) *is nothing but assuming that the function* $\mathbf{f}(t, \mathbf{y})$ *is* Lipschitz continuous *in the variable* $\mathbf{y}$ *and that its* Lipschitz constant L *is independent of* t*.*
- *Picard's Theorem guarantees the existence of a solution only up to a finite time* T_M*. For instance, the solution of the initial value problem* $y' = y^2$*,* $y(0) = 1$ *(which satisfies the hypotheses of Picard's theorem) is* $y(t) = (1-t)^{-1}$*, which exhibits finite time blow up at* $t = 1$ *(*$y(t) \to \infty$ *as* $t \to 1$*).*

[1] Notation: we use **bold symbols** to indicate vectors, matrices, and vector fields.

- *In general, Lipschitz continuity is important. For instance, the initial value problem $y' = y^{2/3}$, $y(0) = 0$, admits two solutions: $y(t) = 0$ and $y(t) = t^3/27$. However, this condition is sufficient but not necessary: there are IVPs whose righthand sides are not Lipschitz and that admit a unique solution. For instance, $y' = \sqrt{y}$, $y(0) = y_0$ (with $y_0 > 0$), has a solution $y(t) = (\sqrt{y_0} + t/2)^2$.*

Another important result is that if an IVP satisfies the hypotheses of Picard's Theorem, its solution is stable on the bounded interval $[t_0, T]$. This means that if $\mathbf{y} : [t_0, T] \to D$ solves the IVP

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}) \,, \qquad \mathbf{y}(t_0) = \mathbf{y}_0 \,,$$

and $\tilde{\mathbf{y}} : [t_0, T] \to D$ solves the same ODE with a perturbed initial condition $\tilde{\mathbf{y}}_0$, that is,

$$\tilde{\mathbf{y}}'(t) = \mathbf{f}(t, \tilde{\mathbf{y}}) \,, \qquad \tilde{\mathbf{y}}(t_0) = \tilde{\mathbf{y}}_0 \,,$$

then

$$\|\mathbf{y}(t) - \tilde{\mathbf{y}}(t)\| \leq e^{L(T-t_0)} \|\mathbf{y}_0 - \tilde{\mathbf{y}}_0\| \quad \text{for every } t \in [t_0, T] \,. \tag{1.3}$$

The estimate (1.3) implies that, for IVPs that satisfy the hypotheses of Picard's theorem, the solution operator that assigns to each initial condition the solution (defined on a bounded interval $[t_0, T]$) of the IVP is Lipschitz continuous. This is relevant for applications: it implies that a small error in the initial condition does not compromise dramatically the solution of the IVP. However, note that the Lipschitz constant may deteriorate exponentially as the finite time T increases.

In this part of the course we focus on numerical methods for first-order IVPs. Note that this is not necessarily a limitation since any higher-order IVP can be formulated as a first-order IVP in a higher-dimensional state space. For instance, the second-order IVP

$$\mathbf{y}''(t) = \mathbf{f}(t, \mathbf{y}, \mathbf{y}') \,, \quad \mathbf{y}(t_0) = \mathbf{y}_0 \,, \quad \mathbf{y}'(t_0) = \mathbf{y}_0' \,,$$

can be equivalently formulated as the first-order IVP

$$\mathbf{z}'(t) = \begin{pmatrix} \mathbf{z}_2 \\ \mathbf{f}(t, \mathbf{z}_1, \mathbf{z}_2) \end{pmatrix} , \quad \mathbf{z}(t_0) = \begin{pmatrix} \mathbf{y}_0 \\ \mathbf{y}_0' \end{pmatrix} ,$$

where the variable $\mathbf{z}$ corresponds to $\mathbf{z} = \binom{\mathbf{z}_1}{\mathbf{z}_2}$ with $\mathbf{z}_1 \coloneqq \mathbf{y}$ and $\mathbf{z}_2 \coloneqq \mathbf{y}'$.

An IVP is called *autonomous* when the righthand side $\mathbf{f}$ does not depend explicitly on the time variable t, that is, $\mathbf{f}(t, \mathbf{y}) = \mathbf{f}(\mathbf{y})$. We will often restrict our considerations to autonomous IVPs because this is notationally convenient (for instance, we could simply set $t_0 = 0$). Note that a nonautonomous IVP can always be transformed into an autonomous one introducing the variable $\mathbf{z} = \binom{\mathbf{z}_1}{z_2}$ with $\mathbf{z}_1 \coloneqq \mathbf{y}$ and $z_2 \coloneqq t$, which satisfies

$$\mathbf{z}'(t) = \begin{pmatrix} \mathbf{f}(z_2, \mathbf{z}_1) \\ 1 \end{pmatrix} , \quad \mathbf{z}(t_0) = \begin{pmatrix} \mathbf{y}_0 \\ t_0 \end{pmatrix} .$$

2 A first glimpse into one-step methods

Let assume that the IVP

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}) \,, \qquad \mathbf{y}(t_0) = \mathbf{y}_0 \,,$$

admits a stable solution $\mathbf{y} : [t_0, T] \to D$ that is defined on the bounded interval $[t_0, T]$. How can we compute a numerical approximation of $\mathbf{y}$ that can be made arbitrarily accurate? An idea could be to first divide the interval $[t_0, T]$ into $N \in \mathbb{N}^+$ subintervals defined by the equidistant points $t_n = t_0 + nh$, $n = 0, \ldots, N$, where the *step size* h is $h = (T - t_0)/N$. To each time step t_n, we want to associate an approximation $\mathbf{y}_n$ of $\mathbf{y}(t_n)$. To define how to compute these approximations, we can take inspiration from the following equality

$$\mathbf{y}(t_{n+1}) = \mathbf{y}(t_n) + \int_{t_n}^{t_{n+1}} \mathbf{f}(t, \mathbf{y}(t)) \, \mathrm{d}t \,, \tag{2.1}$$

which is obtained by integrating the IVP, and where the integration has to be understood componentwise.

Equality (2.1) suggests that, if we have already computed an approximation $\mathbf{y}_n$ of $\mathbf{y}(t_n)$, we could compute $\mathbf{y}_{n+1}$ by adding to $\mathbf{y}_n$ an approximation of the integral appearing on the righthand side. Starting with $n = 0$, we could iterate such a strategy to compute the entire sequence $\{\mathbf{y}_n\}_{n=0}^N$. In what follows, we construct three different schemes based on three different (and still very similar) approximations of the integral appearing in (2.1) and investigate the impact that this choice has on the properties of the resulting numerical method.

To construct an approximation of the integral from (2.1), we recall that by the mean value theorem there is a $\xi \in [t_n, t_{n+1}]$ such that

$$\int_{t_n}^{t_{n+1}} \mathbf{f}(t, \mathbf{y}(t)) \, \mathrm{d}t = h\mathbf{f}(\xi, \mathbf{y}(\xi)) \,. \tag{2.2}$$

Therefore, we can construct an approximation of (2.2) replacing ξ with a value we like. The resulting numerical approximation rule is called a *rectangle rule*.

For instance, we can choose $\xi = t_n$, so that

$$\int_{t_n}^{t_{n+1}} \mathbf{f}(t, \mathbf{y}(t)) \, \mathrm{d}t \approx h\mathbf{f}(t_n, \mathbf{y}(t_n)) \,. \tag{2.3}$$

Inserting (2.3) into (2.1) gives

$$\mathbf{y}(t_{n+1}) \approx \mathbf{y}(t_n) + h\mathbf{f}(t_n, \mathbf{y}(t_n)) \,,$$

which motivates the definition of the *explicit Euler method*

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\mathbf{f}(t_n, \mathbf{y}_n) \,. \tag{2.4}$$

Two other interesting choices are $\xi = t_{n+1}$ and $\xi = (t_n + t_{n+1})/2$, which give rise to the *implicit Euler method*

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\mathbf{f}(t_n + h, \mathbf{y}_{n+1}) \tag{2.5}$$

and the *implicit midpoint rule*

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\mathbf{f}(t_n + h/2, (\mathbf{y}_n + \mathbf{y}_{n+1})/2)\,, \tag{2.6}$$

respectively. Note the occurrence of $\mathbf{y}_{n+1}$ on the righthand side of (2.5) and (2.6).

The numerical methods (2.5) and (2.6) are labelled *implicit* because computing $\mathbf{y}_{n+1}$ requires solving a (generally nonlinear) system, which makes them computationally more expensive than (2.4) (we discuss how to implement implicit methods at the end of this lecture).

Next, we test this methods on two different examples. First, we consider the linear test case

$$y' = \lambda y\,, \quad y_0 = 1\,, \quad \text{on the interval } [0, 1]\,.$$

For $\lambda = 3$ and $N = 10$, we observe that all three methods compute a qualitatively correct solution, although the one computed with the implicit midpoint rule is way more accurate. Doubling the value of N, we see that the accuracy of the Euler methods improves, although they are never as precise as the implicit midpoint rule.

Next, we investigate what happens for negative values of λ. This case in interesting because the exact solution converges to 0 exponentially fast. We fix $N = 10$ and investigate different values of λ. For $\lambda \in [-1, -10]$, we see that all methods provide a qualitatively correct solution. For $\lambda < -10$, we see that the explicit Euler solution start oscillating, becoming equioscillatory for $\lambda = -20$, and diverging for $\lambda < -20$. For $\lambda < -20$, the solution computed with the implicit midpoint rule also starts to oscillate, although the level of these oscillations cannot be compared with the ones of the explicit Euler method, and the method does not diverge (not even for $\lambda = -9000$). On the other hand, it is surprising to see that the implicit Euler method provides excellent solutions for any negative number of λ. This example shows that the stability of a numerical method can vary drastically.

The second test case we consider is the following IVP:

$$\mathbf{y}' = \begin{pmatrix} y_2 \\ -y_1 \end{pmatrix}, \quad \mathbf{y}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \text{on the interval } [0, 2\pi]\,,$$

whose analytic solution is $\mathbf{y}(t) = \binom{\cos(t)}{\sin(t)}$. This case is interesting because the quantity $Q(\mathbf{y}) \coloneqq \|\mathbf{y}(t)\|$ is constant in time. We fix $N = 40$ and plot the orbit of the numerical solutions computed with the three methods above and the evolution of their quantity Q. We see that the implicit midpoint rule is the only method that preserves Q, and that it does it up to machine precision! Numerical methods that preserve certain quantities like length, energy, mass, momentum, and so, are very welcomed by engineers and physicists.

We conclude this lecture with a few words on implicit methods. In the previous example we considered linear IVPs, that is, IVPs whose righthand side was

a linear in the variable $\mathbf{y}$. In such cases, the increment formula of implicit methods is a linear system and can be solved with direct methods. However, it might be that the IVP is not linear, in which case the increment formula of implicit methods becomes a nonlinear equation. In this case, the standard approach consists in reformulating the increment formula as a root-finding problem and solving it with Newton's method. For instance, the increment equation of the implicit Euler method can be formulated as follows:

$$\text{Find } \mathbf{y}_{n+1} \text{ such that } \mathbf{F}(\mathbf{y}_{n+1}) \coloneqq \mathbf{y}_{n+1} - \mathbf{y}_n - h\mathbf{f}(t_n + h, \mathbf{y}_{n+1}) = \mathbf{0}\,. \tag{2.7}$$

Solving (2.7) with Newton's method means constructing a sequence of approximations $\{\mathbf{y}_{n+1}^{(m)}\}_{m=0}^{M}$ of $\mathbf{y}_{n+1}$ with the following algorithm (usually one starts with $\mathbf{y}_{n+1}^{(0)} = \mathbf{y}_n$)

$$\mathbf{y}_{n+1}^{(m+1)} = \mathbf{y}_{n+1}^{(m)} - \mathbf{DF}(\mathbf{y}_{n+1}^{(m)})^{-1}\mathbf{F}(\mathbf{y}_{n+1}^{(m)})\,, \tag{2.8}$$

where $\mathbf{DF}(\mathbf{y}_{n+1}^{(m)})$ is the Jacobian of $\mathbf{F}(\mathbf{x})$ evaluated at $\mathbf{x} = \mathbf{y}_{n+1}^{(m)}$. For the implicit Euler method, the Newton's step (2.8) becomes

$$\mathbf{y}_{n+1}^{(m+1)} = \mathbf{y}_{n+1}^{(m)} - \Big(\mathbf{I} - h\mathbf{D}_{\mathbf{y}}\mathbf{f}(t_n + h, \mathbf{y}_{n+1}^{(m)})\Big)^{-1} \Big(\mathbf{y}_{n+1}^{(m)} - \mathbf{y}_n - h\mathbf{f}(t_n + h, \mathbf{y}_{n+1}^{(m)})\Big)\,,$$

where $\mathbf{I} \in \mathbb{R}^{d,d}$ denotes the identity matrix and $\mathbf{D}_{\mathbf{y}}\mathbf{f}$ is the Jacobian of $\mathbf{f}(t, \mathbf{y})$ with respect to the variable $\mathbf{y}$.

3 Abstract one-step methods

We consider the IVP

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}) \,, \qquad \mathbf{y}(t_0) = \mathbf{y}_0 \,, \tag{3.1}$$

(for a generic but fixed initial value $\mathbf{y}_0 \in D$) and denote its solution by $\mathbf{y} : [t_0, T] \to D$ (note that T depends on $\mathbf{y}_0$). A *one-step method* is a discrete evolution operator $\boldsymbol{\Psi}$ that, given $(t, \mathbf{y}(t))$ and a (sufficiently small) time step h, computes an approximation of $\mathbf{y}(t+h)$. In particular, the evolution operator depends only on $(t, t+h, \mathbf{y}(t))$, that is,

$$\mathbf{y}(t+h) \approx \boldsymbol{\Psi}(t, t+h, \mathbf{y}(t)) \,.$$

It goes without saying that a one-step method is defined independently of the choice of t_0 and of the initial value $\mathbf{y}_0$.

A one-step method $\boldsymbol{\Psi}$ is said to be *consistent* with (3.1) if[2]

$$\boldsymbol{\Psi}(t, t, \mathbf{y}) = \mathbf{y} \quad \text{and} \quad \frac{\mathrm{d}}{\mathrm{d}s} \boldsymbol{\Psi}(t, t+s, \mathbf{y})\big|_{s=0} = \mathbf{f}(t, \mathbf{y}) \quad \text{for every } (t, \mathbf{y}) \in [t_0, T] \times D \,.$$

To assess how well a one-step method discretizes the original IVP, we introduce the *consistency error* $\boldsymbol{\tau}$, which is defined as

$$\begin{aligned} \boldsymbol{\tau}(t, h, \mathbf{y}_0) &\coloneqq \frac{\mathbf{y}(t+h) - \mathbf{y}(t)}{h} - \frac{\boldsymbol{\Psi}(t, t+h, \mathbf{y}(t)) - \mathbf{y}(t)}{h} \,, \\ &= \frac{\mathbf{y}(t+h) - \boldsymbol{\Psi}(t, t+h, \mathbf{y}(t))}{h} \,. \end{aligned}$$

Remark 3.1. *We explicitly write $\mathbf{y}_0$ among the inputs of $\boldsymbol{\tau}$ to specify that the consistency error is computed on the orbit $\{\mathbf{y}(t) : t \in [t_0, T)\}$ of the solution to* (3.1) *.*

Lemma 3.1. *Assume that $s \mapsto \boldsymbol{\Psi}(t, t+s, \mathbf{y})$ is continuously differentiable in a neighborhood of* 0 *(for $(t, \mathbf{y}) \in [t_0, T] \times D$). Then, $\boldsymbol{\Psi}$ is consistent with* (3.1) *if and only if*

$$\|\boldsymbol{\tau}(t, h, \mathbf{y}_0)\| \to 0 \quad \textit{as } h \to 0 \quad \textit{locally uniformly in } (t, \mathbf{y}_0) \in [t_0, T] \times D \,.$$

A one-step method has *consistency order* p if $\|\boldsymbol{\tau}(t, h, \mathbf{y}_0)\| = \mathcal{O}(h^p)$ locally uniformly in $[t_0, T] \times D$. The consistency order of a one-step method is usually determined via Taylor expansion. As an example, we consider the implicit Euler method. Its discrete evolution operator is (recall that the method is implicit)

$$\boldsymbol{\Psi}^{\mathrm{iE}}(t, t+h, \mathbf{y}) = \mathbf{y} + h\mathbf{f}(t+h, \boldsymbol{\Psi}^{\mathrm{iE}}(t, t+h, \mathbf{y})) \,.$$

[2] Note that we cannot pretend that, for $t < s < r$, $\boldsymbol{\Psi}(s, r, \boldsymbol{\Psi}(t, s, \mathbf{y})) = \boldsymbol{\Psi}(t, r, \mathbf{y})$, because this would imply that $\boldsymbol{\Psi}$ is the analytic solution to (3.1).

For simplicity, we introduce the notation $\tilde{\mathbf{y}}(h) \coloneqq \mathbf{\Psi}^{\mathrm{iE}}(t, t+h, \mathbf{y})$ and assume that the IVP is autonomous, that is, $\mathbf{f}(t, \mathbf{y}) = \mathbf{f}(\mathbf{y})$. Then, the implicit Euler step becomes

$$\tilde{\mathbf{y}}(h) = \mathbf{y} + h\mathbf{f}(\tilde{\mathbf{y}}(h))\,. \tag{3.2}$$

The first derivative of $\tilde{\mathbf{y}}(h)$ reads (iterating recursively)

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}h}\tilde{\mathbf{y}}(h) &= \mathbf{f}(\tilde{\mathbf{y}}(h)) + h\mathbf{D}_{\mathbf{y}}\mathbf{f}(\tilde{\mathbf{y}}(h))\left(\frac{\mathrm{d}}{\mathrm{d}h}\tilde{\mathbf{y}}(h)\right)\,,\\
&= \mathbf{f}(\tilde{\mathbf{y}}(h)) + h\mathbf{D}_{\mathbf{y}}\mathbf{f}(\tilde{\mathbf{y}}(h))\left(\mathbf{f}(\tilde{\mathbf{y}}(h)) + h\mathbf{D}_{\mathbf{y}}\mathbf{f}(\tilde{\mathbf{y}}(h))\left(\frac{\mathrm{d}}{\mathrm{d}h}\tilde{\mathbf{y}}(h)\right)\right)\,,\\
&= \mathbf{f}(\tilde{\mathbf{y}}(h)) + h\mathbf{D}_{\mathbf{y}}\mathbf{f}(\tilde{\mathbf{y}}(h))\big(\mathbf{f}(\tilde{\mathbf{y}}(h)) + \mathcal{O}(h)\big)\,.
\end{aligned}$$

The second derivative of $\tilde{\mathbf{y}}(h)$ reads

$$\begin{aligned}
\frac{\mathrm{d}^2}{\mathrm{d}h^2}\tilde{\mathbf{y}}(h) &= \mathbf{D}_{\mathbf{y}}\mathbf{f}(\tilde{\mathbf{y}}(h))\frac{\mathrm{d}}{\mathrm{d}h}\tilde{\mathbf{y}}(h) + \mathbf{D}_{\mathbf{y}}\mathbf{f}(\tilde{\mathbf{y}}(h))\big(\mathbf{f}(\tilde{\mathbf{y}}(h)) + \mathcal{O}(h)\big) + \mathcal{O}(h)\,,\\
&= 2\mathbf{D}_{\mathbf{y}}\mathbf{f}(\tilde{\mathbf{y}}(h))\mathbf{f}(\tilde{\mathbf{y}}(h)) + \mathcal{O}(h)\,.
\end{aligned}$$

Therefore, the Taylor expansion of $\tilde{\mathbf{y}}(h)$ with respect to h at $h = 0$ reads (note that $\tilde{\mathbf{y}}(0) = \mathbf{y}$ by (3.2))

$$\tilde{\mathbf{y}}(h) = \mathbf{y} + h\mathbf{f}(\mathbf{y}) + h^2\mathbf{D}_{\mathbf{y}}\mathbf{f}(\mathbf{y})\mathbf{f}(\mathbf{y}) + \mathcal{O}(h^3)\,. \tag{3.3}$$

On the other hand, the Taylor expansion of the exact solution $\mathbf{y}(h)$ reads

$$\mathbf{y}(h) = \mathbf{y} + h\mathbf{f}(\mathbf{y}) + \frac{h^2}{2}\mathbf{D}_{\mathbf{y}}\mathbf{f}(\mathbf{y})\mathbf{f}(\mathbf{y}) + \mathcal{O}(h^3)\,. \tag{3.4}$$

Therefore, the norm of the consistency error of the implicit Euler method satisfies

$$\|\boldsymbol{\tau}(h, \mathbf{y})\| \leq \frac{h}{2}\|\mathbf{D}_{\mathbf{y}}\mathbf{f}(\mathbf{y})\mathbf{f}(\mathbf{y})\| + \mathcal{O}(h^2) = \mathcal{O}(h)\,,$$

which implies that the implicit Euler method has consistency order 1.

We conclude this lecture by showing that consistency is important for convergence. The next lemma gives a representation of one-step methods.

Lemma 3.2. *Assume that $s \mapsto \mathbf{\Psi}(t, t+s, \mathbf{y})$ is continuously differentiable in a neighborhood of 0 (for $(t, \mathbf{y}) \in [t_0, T] \times D$). Then, $\mathbf{\Psi}$ is consistent with (3.1) if and only if there is a continuous increment function $h \mapsto \boldsymbol{\psi}(t, h, \mathbf{y})$ such that*

$$\mathbf{\Psi}(t, t+h, \mathbf{y}) = \mathbf{y} + h\boldsymbol{\psi}(t, h, \mathbf{y})\,, \quad \boldsymbol{\psi}(t, 0, \mathbf{y}) = \mathbf{f}(t, \mathbf{y})\,.$$

Let $t_n = t_0 + nh$, $n = 0, \ldots, N$, where the *step size* h is $h = (T - t_0)/N$, and consider the sequence $\{\mathbf{y}_n\}_{n=0}^N$ given by

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\boldsymbol{\psi}(t, h, \mathbf{y}_n)\,, \quad \text{for } n = 0, \ldots, N-1\,.$$

We define the *global error* e by

$$e \coloneqq \max_{n=1,\ldots,N} e_n\,, \quad \text{where} \quad e_n \coloneqq \|\mathbf{y}(t_n) - \mathbf{y}_n\|\,.$$

Theorem 3.1. *Let $\boldsymbol{\Psi}$ be a consistent one-step method and assume that its increment function $\boldsymbol{\psi}$ is Lipschitz continuous with respect to $\mathbf{y}$, that is, that there exists a positive constant $L_{\boldsymbol{\psi}}$ such that, for $0 \le h \le h_0$ and for the same region R of Picard's theorem,*

$$\|\boldsymbol{\psi}(t,h,\mathbf{y}) - \boldsymbol{\psi}(t,h,\mathbf{z})\| \le L_{\boldsymbol{\psi}}\|\mathbf{y}-\mathbf{z}\| \quad \textit{for } (t,\mathbf{y}),(t,\mathbf{z}) \textit{ in } R\,.$$

Then, assuming that $\|\mathbf{y}_n - \mathbf{y}_0\| \le Y_M$, it follows that

$$e \le \left(\frac{\exp\left(L_{\boldsymbol{\psi}}(t_N - t_0)\right) - 1}{L_{\boldsymbol{\psi}}}\right) \max_{n=0,\dots,N-1} \|\boldsymbol{\tau}(t_n,h,\mathbf{y}_0)\|\,.$$

Proof. For a generic $n \in \{1,\dots,N-1\}$,

$$\begin{aligned}
e_{n+1} &= \|\mathbf{y}(t_{n+1}) - \mathbf{y}_{n+1}\|\,,\\
&= \|\mathbf{y}(t_{n+1}) - \boldsymbol{\Psi}(t_n, t_n+h, \mathbf{y}_n)\|\,,\\
&= \|\mathbf{y}(t_{n+1}) - \boldsymbol{\Psi}(t_n, t_n+h, \mathbf{y}(t_n)) + \boldsymbol{\Psi}(t_n, t_n+h, \mathbf{y}(t_n)) - \boldsymbol{\Psi}(t_n, t_n+h, \mathbf{y}_n)\|\,,\\
&\le \|\mathbf{y}(t_{n+1}) - \boldsymbol{\Psi}(t_n, t_n+h, \mathbf{y}(t_n))\| + \|\boldsymbol{\Psi}(t_n, t_n+h, \mathbf{y}(t_n)) - \boldsymbol{\Psi}(t_n, t_n+h, \mathbf{y}_n)\|\,,\\
&\le h\|\boldsymbol{\tau}(t_n,h,\mathbf{y}_0)\| + \|\left(\mathbf{y}(t_n) + h\boldsymbol{\psi}(t,h,\mathbf{y}(t_n))\right) - \left(\mathbf{y}_n + h\boldsymbol{\psi}(t,h,\mathbf{y}_n)\right)\|\,,\\
&\le h\|\boldsymbol{\tau}(t_n,h,\mathbf{y}_0)\| + \|\mathbf{y}(t_n) - \mathbf{y}_n\| + h\|\boldsymbol{\psi}(t,h,\mathbf{y}(t_n)) - \boldsymbol{\psi}(t,h,\mathbf{y}_n)\|\,,\\
&= h\|\boldsymbol{\tau}(t_n,h,\mathbf{y}_0)\| + e_n + h\|\boldsymbol{\psi}(t,h,\mathbf{y}(t_n)) - \boldsymbol{\psi}(t,h,\mathbf{y}_n)\|\,,\\
&\le h\|\boldsymbol{\tau}(t_n,h,\mathbf{y}_0)\| + e_n + hL_{\boldsymbol{\psi}}\|\mathbf{y}(t_n) - \mathbf{y}_n\|\,,\\
&= h\|\boldsymbol{\tau}(t_n,h,\mathbf{y}_0)\| + (1+hL_{\boldsymbol{\psi}})e_n\,,
\end{aligned}$$

where the forelast step is allowed if h is sufficiently small (so that $(t,\mathbf{y}_n) \in R$). Iterating recursively, this implies that (note that $e_0 = 0$)

$$\begin{aligned}
e_{n+1} &\le (1+hL_{\boldsymbol{\psi}})^{n+1}e_0 + h\sum_{k=0}^{n}(1+hL_{\boldsymbol{\psi}})^k \max_{m=0,\dots,n}\|\boldsymbol{\tau}(t_m,h,\mathbf{y}_0)\|\\
&= \frac{(1+hL_{\boldsymbol{\psi}})^{n+1}-1}{L_{\boldsymbol{\psi}}} \max_{m=0,\dots,n}\|\boldsymbol{\tau}(t_m,h,\mathbf{y}_0)\|\,.
\end{aligned}$$

To conclude the proof, note that $1 + hL_{\boldsymbol{\psi}} \le \exp hL_{\boldsymbol{\psi}}$. □

The theorem above shows that, under suitable conditions, if $\boldsymbol{\Psi}$ has consistency order p, then the global error satisfies $e = \mathcal{O}(h^p)$. In such a case, we say that $\boldsymbol{\Psi}$ converges with order p.

Remark 3.2. *Note that the order of consistency and convergence of a numerical method is an asymptotic concept (i.e., it holds only in the limit $h \to 0$), and that depends on the regularity of the righthand side $\mathbf{f}$.*

4 Introduction to Runge-Kutta methods

The family of *s-stage Runge-Kutta methods* is defined by

$$\boldsymbol{\Psi}(t, t+h, \mathbf{y}) = \mathbf{y} + h \sum_{i=1}^{s} b_i \mathbf{k}_i \,, \tag{4.1}$$

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

$$\mathbf{k}_i \coloneqq \mathbf{f}(t + c_i h, \mathbf{y} + h \sum_{j=1}^{s} a_{ij} \mathbf{k}_j) \,, \quad i = 1, \ldots, s \,. \tag{4.2}$$

Lemma 4.1. *If* $\mathbf{f}$ *is locally Lipschitz continuous in* R *and* $h > 0$ *is sufficiently small,* (4.2) *admits a unique solution.*

A Runge-Kutta method is characterized by the coefficients $\{b_i\}_{i=1}^s$ and $\{a_{ij}\}_{i,j=1}^s$, whereas the coefficients $\{c_i\}_{i=1}^s$ are always given by

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

The coefficients of a Runge-Kutta method can be summarized in his Butcher table [3]

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

The explicit Euler method, the implicit Euler method, and the implicit midoint rule are Runge-Kutta methods. Their Butcher tables are

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

respectively.

In the first lecture, we saw that nonautonomous IVPs can be reformulated in an autonomous equivalent form. The next lemma states that, under suitable conditions, the numerical solution computed with Runge-Kutta methods is not affected by this change of variables. This implies that, to investigate the order of consistency of Runge-Kutta methods, we can restrict ourselves to autonomous IVPs.

In the following, $\{\mathbf{y}_k\}_{k=0}^N$ denotes an approximation of $\{\mathbf{y}(t_k)\}_{k=0}^N$ computed with a Runge-Kutta method on the time grid $\{t_k\}_{k=0}^N$.

Lemma 4.2. *The following diagram commutes*

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

$$\begin{array}{ccc} \mathbf{y}' = \mathbf{f}(t,\mathbf{y})\,, \mathbf{y}(t_0) = \mathbf{y}_0 & \xrightarrow{autonomization} & \mathbf{z}' = \mathbf{g}(\mathbf{z})\,, \mathbf{z}(t_0) = \mathbf{z}_0 \\ \downarrow RK & & \downarrow RK \\ \{(t_k, \mathbf{y}_k)\} & \xrightarrow{\hat{\mathbf{y}}_k := \binom{\mathbf{y}_k}{t_k}} & \{\hat{\mathbf{y}}_k\} \end{array}$$

if and only if

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

The previous lemma allows shortening the (omitted) proof of the following.

Lemma 4.3. *A Runge-Kutta method is consistent if and only if* $\sum_{i=1}^{s} b_i = 1$. *If the condition*

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

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

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

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

Remark 4.1. *The previous lemma is proven computing the Taylor expansion of a generic s-stage Runge-Kutta method and matching the terms with the Taylor expansion of the exact solution. The following table indicates the number of conditions 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 (consistency) order p of an s-stage Runge-Kutta method is bounded by the number of stages. More specifically, $p \leq 2s$. In the next lecture, we will see how to construct Runge-Kutta methods that achieve maximal order. However, high-order comes at a cost: computational cost! To elaborate on that, note that to evolve a numerical solution from t_n to t_{n+1} with a Runge-Kutta method, one needs to compute the stages $\mathbf{k}_i$s.

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 $j >= i$). However, the order of an explicit s-stage Runge-Kutta method can be at most s. In fact, Butcher barriers quantify the minimal amount of stages that an explicit Runge-Kutta method of order p requires. The following table shows some of these minimal amount of stages

p	1	2	3	4	5	6	7	8	≥ 9
minival 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. In this case, calculating the stages $\mathbf{k}_i$s can be computationally expensive, because the system (4.2) has dimension $(d \cdot s)$ and can be nonlinear.

If the system (4.2) is nonlinear, one can use Newton's method to compute an approximation of the stages. This approximation can affect the convergence order of the Runge-Kutta method. In an exercise, we will see that one Newton's step is generally sufficient for second order Runge-Kutta methods, but that higher-order methods require more Newton's steps.

If the system (4.2) is linear, one can employ Gaussian elimination. We recall that the computational cost to solve a general m-dimensional linear system with Gaussian elimination is $\mathcal{O}(m^3)$. This means that computing the stages for a general implicit Runge-Kutta method can cost $\mathcal{O}((ds)^3)$, which can be still very expensive.

The class of diagonally implicit Runge-Kutta methods[4] provide a tradeoff between computational efficiency and higher-order (compared to explicit methods). A Runge-Kutta method is diagonally implicit if its matrix $\mathbf{A}$ is a lower triangular matrix. In this case, the computational cost to compute the stages reduces to $\mathcal{O}(d^3 s)$. The implicit Euler method and the implicit midpoint rule are examples of diagonally implicit Runge-Kutta methods.

[4] NASA seems to be interested in these methods; see C. A. Kennedy and M. H. Carpenter, *Diagonally Implicit Runge-Kutta Methods for Ordinary Differential Equations. A Review.* (2016).

5 Construction of Runge-Kutta methods

To construct explicit Runge-Kutta methods, we start by recalling that the analytic solution of

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}) \,, \qquad \mathbf{y}(t_0) = \mathbf{y}_0 \,, \tag{5.1}$$

is given by the (implicit) formula

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

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

$$\mathbf{y}(t+h) \approx \mathbf{y}(t) + h \sum_{i=1}^{s} b_i \mathbf{f}(t + c_i h, \mathbf{y}(t + c_i h)) \,. \tag{5.2}$$

Note that the approximation (5.2) requires the values $\mathbf{y}(t + c_i h)$. To make the method explicit, we approximate the values $\mathbf{y}(t_0 + c_i h)$ with explicit Runge-Kutta methods we already know. This way, we can construct s-stage explicit Runge-Kutta methods by induction.

For instance, we can choose the 1-point Gauss quadrature rule in (5.2), that is,

$$\mathbf{y}(t+h) \approx \mathbf{y}(t) + h\mathbf{f}(t + h/2, \mathbf{y}(t + h/2)) \tag{5.3}$$

and approximate $\mathbf{y}(t + h/2)$ with the explicit Euler method. The resulting scheme reads

$$\boldsymbol{\Psi}(t, t+h, \mathbf{y}) = \mathbf{y} + h\mathbf{f}\left(t + h/2, \mathbf{y} + \frac{h}{2}\mathbf{f}(t, \mathbf{y})\right) \,. \tag{5.4}$$

Another simple scheme can be obtained using the trapezium rule in (5.2), that is,

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

and approximating $\mathbf{y}(t + h)$ with the explicit Euler method. The resulting scheme reads

$$\boldsymbol{\Psi}(t, t+h, \mathbf{y}) = \mathbf{y} + \frac{h}{2}\mathbf{f}(t, \mathbf{y}) + \frac{h}{2}\mathbf{f}\left(t + h, \mathbf{y} + h\mathbf{f}(t, \mathbf{y})\right) \,. \tag{5.5}$$

Both (5.4) and (5.5) are 2nd-order Runge-Kutta methods. Their Butcher tables read

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

respectively. A similar approach leads to the most famous explicit Runge-Kutta method *RK4*, a 4-stage 4th-order explicit Runge-Kutta method whose Butcher

table reads

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 2/6 & 2/6 & 1/6 \end{array} .$$

In the previous lecture, we saw 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. To do so, we consider the space $\mathcal{P}_s$ of univariate polynomials of degree s. Note that $\mathcal{P}_s$ has dimension $s+1$. Then, we choose s (pairwise distinct) collocation points $c_1, \dots, c_s \in [0,1]$ and compute the unique polynomial $\tilde{\mathbf{y}} \in \mathcal{P}_s$ that satisfies (5.1)

$$\tilde{\mathbf{y}}(0) = \mathbf{y}(t) \quad \text{and} \quad \tilde{\mathbf{y}}'(c_i h) = \mathbf{f}(t + c_i h, \tilde{\mathbf{y}}(c_i h)), \quad \text{for } i = 1, \dots, s. \tag{5.6}$$

Finally, we approximate $\mathbf{y}(t+h)$ by evaluating $\tilde{\mathbf{y}}(h)$. Such an approach is called *collocation method*.

Lemma 5.1. *Collocation methods are Runge-Kutta methods. Their coefficients are*

$$a_{ij} = \int_0^{c_i} L_j(\tau)\,\mathrm{d}\tau, \quad b_i = \int_0^1 L_i(\tau)\,\mathrm{d}\tau, \tag{5.7}$$

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

Proof. The Lagrange polynomials associated to $c_1, \dots, c_s$ are

$$L_i(\tau) \coloneqq \prod_{j \neq i} \frac{\tau - c_j}{c_i - c_j}, \quad i = 1, \dots, s,$$

satisfy $L_j(c_i) = \delta_{ij}$ for every pair $i, j = 1, \dots, s$, and are a basis of $\mathcal{P}_{s-1}$. Since $\tilde{\mathbf{y}} \in \mathcal{P}_s$, $\tilde{\mathbf{y}}' \in \mathcal{P}_{s-1}$, and there are some coefficients $\{\boldsymbol{\mu}_i\}_{i=1}^s$ (to be determined) such that

$$\tilde{\mathbf{y}}'(\tau h) = \sum_{i=1}^s \boldsymbol{\mu}_i L_i(\tau). \tag{5.8}$$

By the Fundamental Theorem of Calculus,

$$\begin{aligned} \tilde{\mathbf{y}}(\tau h) &= \tilde{\mathbf{y}}(0) + \int_0^{\tau h} \tilde{\mathbf{y}}'(x)\,\mathrm{d}x \\ &= \tilde{\mathbf{y}}(0) + h \int_0^{\tau} \tilde{\mathbf{y}}'(yh)\,\mathrm{d}y = \tilde{\mathbf{y}}(0) + h \sum_{i=1}^s \boldsymbol{\mu}_i \int_0^{\tau} L_i(y)\,\mathrm{d}y. \end{aligned} \tag{5.9}$$

Therefore, the interpolatory condition $L_j(c_i) = \delta_{ij}$, (5.6), (5.8), and (5.9) imply

$$\begin{aligned} \boldsymbol{\mu}_i = \sum_{j=1}^s \boldsymbol{\mu}_j L_j(c_i) = \tilde{\mathbf{y}}'(c_i h) &= \mathbf{f}(t + c_i h, \tilde{\mathbf{y}}(c_i h)) \\ &= \mathbf{f}\left(t + c_i h, \mathbf{y}(t) + h \sum_{i=1}^s \boldsymbol{\mu}_i \int_0^{c_i} L_i(y)\,\mathrm{d}y\right), \end{aligned}$$

that is, $\{\boldsymbol{\mu}_i\}_{i=1}^s$ are Runge-Kutta stages. A comparison with (4.2) motivates the formula of the coefficients a_{ij} given in (5.7). Finally, comparing formula (5.9) with $\tau = 1$ to (4.1) gives the formula for the coefficients b_is. □

Lemma 5.2. *Let Q be the highest-order quadrature rule on $[0,1]$ that can be constructed using the nodes $c_1, \ldots, c_s$, and let p_Q be its order (p_Q = 1 + the degree of polynomials it integrates exactly). If $\mathbf{f}$ is sufficiently smooth and $h > 0$ is sufficiently small, the collocation method associated to $c_1, \ldots, c_s$ has order p_Q.*

Corollary 5.1. *If $\mathbf{f}$ is sufficiently smooth and $h > 0$ is sufficiently small, the order of the collocation method associated to $c_1, \ldots, c_s$ is at least s and at most $2s$ (Gauss-Quadrature).*

6 Stability of Runge-Kutta methods

We want to investigate the behavior of Runge-Kutta methods when the initial value problem has a stable fixed point. A *fixed point* of the ODE[5] $\mathbf{y}' = \mathbf{f}(\mathbf{y})$ 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 $t \mapsto \mathbf{y}(t)$ of $\mathbf{y}' = \mathbf{f}(\mathbf{y})$, $\mathbf{y}(0) = \mathbf{y}_0$ satisfies $\lim_{t\to\infty} \mathbf{y}(t) = \mathbf{y}^*$.

Theorem 6.1. *A fixed point $\mathbf{y}^*$ is asymptotically stable if*

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

This theorem implies that, to study the asymptotic stability of $\mathbf{y}^*$, we can restrict our considerations to the linearized ODE $\mathbf{y}' = \mathbf{Df}(\mathbf{y}^*)(\mathbf{y} - \mathbf{y}^*)$, that is, we can restrict our attention to linear ODEs. For simplicity, we restrict our attentions to the *Dahlquist test equation*

$$y' = \lambda y\,, \quad y(0) = 1\,, \quad \text{and} \quad \operatorname{Re}\lambda < 0\,. \tag{6.1}$$

Clearly, the solution of (6.1) is $y(t) = \exp(\lambda t)$, which satisfies $\lim_{t\to\infty} y(t) = 0$. Therefore, $y^* = 0$ is an attractive fixed point.

The solution of (6.1) obtained with a Runge-Kutta method has a special structure. Let $\mathbf{K} \coloneqq (k_1, \dots, k_s)^\top \in \mathbb{C}^s$ be the vector that contains the (scalar) stages, and let $\mathbf{1} \coloneqq (1, \dots, 1)^\top \in \mathbb{R}^s$. Then, $\mathbf{K}$ satisfies

$$\mathbf{K} = \lambda\mathbf{1} + h\lambda\mathbf{A}\mathbf{K}\,, \quad \text{and thus,} \quad \mathbf{K} = (\mathbf{I} - h\lambda\mathbf{A})^{-1}\lambda\mathbf{1}\,. \tag{6.2}$$

Therefore,

$$y_1 = \mathbf{\Psi}(0, h, 1) = 1 + h\lambda\mathbf{b}^\top(\mathbf{I} - h\lambda\mathbf{A})^{-1}\mathbf{1} \coloneqq S(h\lambda). \tag{6.3}$$

The function S is called the *stability function* of the Runge-Kutta method and it is, in general, a rational function.

Iterating (6.3), we find $y_k = S(h\lambda)^k$. 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 (6.1). When this happens, we say that $\{y_k\}_{k\in\mathbb{N}}$ is *asymptotically stable*. The region in the complex plane

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

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

The stability function S appears also when solving the linear nonscalar initial value problem $\mathbf{y}' = \mathbf{M}\mathbf{y}$, $\mathbf{y}(0) = \mathbf{y}_0$. Indeed, $\mathbf{\Psi}(0, h, \mathbf{y}_0) = S(h\mathbf{M})\mathbf{y}_0$. In light of Theorem 6.1, we say that a Runge-Kutta method *inherits the asymptotic stability* of a fixed point $\mathbf{y}^*$ if $h\sigma\left(\mathbf{Df}(\mathbf{y}^*)\right) \subset S_{\mathbf{\Psi}}$.

[5] For simplicity, we consider only autonomous problems.

It is not so difficult to see that the stability function of an explicit Runge-Kutta method is a polynomial, which implies that $S_{\mathbf{\Psi}}$ 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 observed in the second lecture.

There are numerical methods that do not suffer from this limitation. A Runge-Kutta method is said to be *A-stable*[6] if $\mathbb{C}^- \subset S_{\mathbf{\Psi}}$. For instance, Gauss-collocation methods are A-stable, and their stability region is exactly $\mathbb{C}^-$.

A-stability may not be sufficient if the initial value problem has a strongly attractive fixed point. In such instances, it is advisable to employ an A-stable method that further satisfies $\lim_{\mathrm{Re}\, z \to -\infty} |S(z)| = 0$. Such a method is said to be *L-stable* (or stiffly accurate). An example of L-stable Runge-Kutta methods is the class of Gauss-Radau-collocation methods. These methods are constructed fixing $c_s = 1$ and choosing the remaining collocation points $c_1, \ldots, c_{s-1}$ to obtain an associated quadrature rule with maximal order (which is $2s-1$). The simplest Gauss-Radau-collocation method is the implicit Euler method.

To conclude, we mention that there are other relevant concepts of stability that we do not have the time to cover, like *algebraic stability* or *B-stability* (nonlinear stability).

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

7 Adaptivity and stiffness

We want to compute an approximate solution of an initial value problem that is accurate up to a certain (absolute/relative) precision. What shall we do? A simple strategy could be to choose a one-step method of order p and solve the problem once using a certain constant time step h (that is, $t_{k+1} = t_k + h$) and then using a smaller constant time step $\tilde{h} = \alpha h$ with $\alpha < 1$. This way, we obtain two approximations $\mathbf{y}_N$ and $\mathbf{y}_{\tilde{N}}$ of $\mathbf{y}(T)$ (where $T = Nh = \tilde{N}\tilde{h}$). The following (questionable) argumentation suggests that the (computable) difference $\|\tilde{\mathbf{y}}_{\tilde{N}} - \mathbf{y}_N\|$ can be used to estimate the (noncomputable) error $\|\mathbf{y}(T) - \mathbf{y}_N\|$.

For h sufficiently small,

$$\|\tilde{\mathbf{y}}_{\tilde{N}} - \mathbf{y}_N\| = \|\tilde{\mathbf{y}}_{\tilde{N}} - \mathbf{y}(T) + \mathbf{y}(T) - \mathbf{y}_N\| \leq C(\tilde{h}^p + h^p) = (1 + \alpha^p)Ch^p\,,$$

and thus,

$$\begin{aligned}\|\mathbf{y}(T) - \mathbf{y}_N\| &= \|\mathbf{y}(T) - \mathbf{y}_{\tilde{N}} + \mathbf{y}_{\tilde{N}} - \mathbf{y}_N\| \\ &\leq \|\mathbf{y}(T) - \mathbf{y}_{\tilde{N}}\| + \|\mathbf{y}_{\tilde{N}} - \mathbf{y}_N\| \\ &\leq C\tilde{h}^p + (1 + \alpha^p)Ch^p \\ &\leq \alpha^p\big(Ch^p\big) + (1 + \alpha^p)\big(Ch^p\big)\,,\end{aligned}$$

For $\alpha < 1$, $\alpha^p \ll 1 + \alpha^p$ (in relative terms). Therefore, the term $\|\mathbf{y}(T) - \mathbf{y}_{\tilde{N}}\|$ has a minor contribution, and $\|\mathbf{y}_{\tilde{N}} - \mathbf{y}_N\|$ may be used to estimate $\|\mathbf{y}(T) - \mathbf{y}_N\|$.

The strategy described above may deliver an accurate solution, but it is likely to be computationally inefficient: if the accuracy criteria are not met, we need to choose a smaller h and recompute everything from scratch.

A better (but still questionable) idea is try to control the local error at every time step. Indeed, Theorem 3.1 has been proved estimating the final error with the sum of the local errors (however, note the exponential term in the constant!). To control the local error introduced evolving a numerical solution $\mathbf{y}_k$ from t_k, it is common to proceed as follows:

1. select a local time step h_k,
2. evaluate $\mathbf{\Psi}^1(t_k, t_k + h_k, \mathbf{y}_k)$ and $\mathbf{\Psi}^2(t_k, t_k + h_k, \mathbf{y}_k)$ using two different one-step methods $\mathbf{\Psi}^1$ and $\mathbf{\Psi}^2$ (one of order p_1 and the other of order p_2, with $p_2 > p_1$),
3. estimate the local error computing
$$\mathrm{ERR}_k(h_k) = \|\mathbf{\Psi}^1(t_k, t_k + h_k, \mathbf{y}_k) - \mathbf{\Psi}^2(t_k, t_k + h_k, \mathbf{y}_k)\|\,, \tag{7.1}$$
4. if ERR_k is smaller than a chosen (absolute/relative) tolerance TOL, the step is accepted; otherwise, choose a smaller h_k and go back to step 2.

To make this algorithm more efficient, it is common to increase the step h_k every time this has been accepted (that is, $h_{k+1} = \beta h_k$ for a $\beta > 1$; the "optimal" β is $\beta_k = \sqrt[(p_1+1)]{\mathrm{TOL}/\mathrm{ERR}_k}$).

The use of (7.1) as an error indicator can be motivated similarly as above.

To make the computations more efficient, $\mathbf{\Psi}^1$ and $\mathbf{\Psi}^2$ are chosen to be two Runge-Kutta methods that use the same stages. This leads to the definition of *embedded Runge-Kutta methods*. Their Butcher table reads

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}_2^\top \\ & \\ & \mathbf{b}_1^\top \end{array}, \quad \text{where} \quad \begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}_2^\top \end{array} \quad \text{and} \quad \begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}_1^\top \end{array} \quad \text{are of order } p_2 \text{ and } p_1,$$

respectively. A simple example is the Heun-Euler method $\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \\ & 1 & 0 \end{array}$.

MATLAB integrators for ODEs (such as the functions `ode45`, `ode23`, etc.) are based on embedded Runge-Kutta methods[7].

Adaptive methods are also used to identify *stiffness*. Stiffness of an ODE is a concept that lacks a rigorous definition[8]. An historic and pragmatic definition by Curtis and Hirschfelder[9] (adapted to our lecture) reads: stiff equations are equations where the implicit Euler method works tremendously better than the explicit Euler method. The idea behind this definition is that stability of the numerical methods requires the choice of a very small time step, much smaller than the one required by accuracy. In such case, it is not a surprise that the implicit Euler method, being L-stable, performs way better than his explicit counterpart. In general, it has been observed that if the solution of a system of ODEs has components that decay rapidly and at notably different time scales, than that system of ODEs is stiff.

[7] See L. F. Shampine and M. W. Reichelt, *The MATLAB ODE suite* (1997).

[8] See G. Söderlind, L. Jay, and M. Calvo, *Stiffness 1952-2012: Sixty years in search of a definition*(2015).

[9] *Integration to stiff equations* (1952).

8 Structure preserving integrators

In this section[10], we restrict ourselves to the autonomous ODE

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}), \quad \text{where} \quad \mathbf{f} : D \to \mathbb{R}^d. \tag{8.1}$$

Its *flow* is a family of functions $\mathbf{\Phi}^t : D \to \mathbb{R}^d$ that map an initial condition $\mathbf{y}_0$ to $\mathbf{y}(t)$, the solution of $\mathbf{y}' = \mathbf{f}(\mathbf{y})$, $\mathbf{y}(0) = \mathbf{y}_0$ at time t (tacitly assuming that $\mathbf{y}(t)$ exists).

A *first integral* of (8.1) is a function $I : D \to \mathbb{R}$ that satisfies $I(\mathbf{\Phi}^t\mathbf{y}_0) = I(\mathbf{y}_0)$ for every (admissible) $t \geq 0$ and every $\mathbf{y}_0 \in D$. Note that I is a first integral of (8.1) if and only if $\frac{\mathrm{d}}{\mathrm{d}t} I(\mathbf{\Phi}^t\mathbf{y}_0) = 0$ for every for every (admissible) $t \geq 0$ and every $\mathbf{y}_0 \in D$, which is equivalent to $\mathbf{grad}\, I(\mathbf{y}) \cdot \mathbf{f}(\mathbf{y}) = 0$ for every $\mathbf{y} \in D$.

A *first integral* is polynomial of degree $n \in \mathbb{N}$ if it is a multivariate polynomial of degree n, that is,

$$I(\mathbf{y}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}_0^d, |\boldsymbol{\alpha}| \leq n} \beta_{\boldsymbol{\alpha}} \mathbf{y}^{\boldsymbol{\alpha}}, \tag{8.2}$$

where $\beta_{\boldsymbol{\alpha}} \in \mathbb{R}$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d)$, $|\boldsymbol{\alpha}| = \sum_i \alpha_i$, and $\mathbf{y}^{\boldsymbol{\alpha}} = y_1^{\alpha_1} \cdots y_d^{\alpha_d}$. For instance, a linear first integral is of the form $I(\mathbf{y}) = \mathbf{b}^\top \mathbf{y} + c$, whereas a quadratic first integral is of the form $I(\mathbf{y}) = \mathbf{y}^\top \mathbf{M} \mathbf{y} + \mathbf{b}^\top \mathbf{y} + c$ (with $\mathbf{M} \in \mathbb{R}^{d,d}$, $\mathbf{b} \in \mathbb{R}^d$, and $c \in \mathbb{R}$).

Theorem 8.1. *Every Runge-Kutta method preserves linear first integrals.*

Theorem 8.2. *Gauss-collocation methods preserve quadratic first integrals.*

Theorem 8.3. *If $n \geq 3$, there is no consistent Runge-Kutta method that preserves every polynomial first integrals of degree n for every autonomous ODE.*

A differentiable map $\mathbf{\Phi} : D \to \mathbb{R}^d$ is called *volume preserving* if, for every measurable subset $V \subset D$, $\mathrm{Vol}(\mathbf{\Phi}(V)) = \mathrm{Vol}(V)$.

Lemma 8.1. *A continuously differentiable map $\mathbf{\Phi} : D \to \mathbb{R}^d$ preserves volumes if and only if $|\det \mathbf{D}\mathbf{\Phi}(\mathbf{y})| = 1$ for every $\mathbf{y} \in D$.*

The flow $\mathbf{\Phi}^t$ is said to be volume preserving if for every compact subset $V \subset D$ there is a $\delta > 0$ such that $\mathrm{Vol}(\mathbf{\Phi}^t(V)) = \mathrm{Vol}(V)$ for every $0 \leq t \leq \delta$. The theorem of Liouville states that, for a continuously differentiable $\mathbf{f}$, $\mathbf{\Phi}^t$ is volume preserving if and only if $\operatorname{div} \mathbf{f}(\mathbf{y}) = 0$ for every $\mathbf{y} \in D$.

A one-step method is volume preserving if it preserves the volume in each time step. In light of Lemma 8.1, we can verify that a one-step method is volume preserving by checking that $|\det \mathbf{D}\mathbf{\Psi}(0, h, \mathbf{y})| = 1$ for every $\mathbf{y} \in D$.

Theorem 8.4. *For $d \leq 2$, every Runge-Kutta method that preserves quadratic first integrals is volume preserving.*

[10] We could have named this section *Geometric numerical integration*, a term introduced by J. M. Sans-Serna; see his article *Geometric integration* in the proceedings *The state of the art in numerical analysis* (1997).

A one-step method is said to be *reversible* (or *symmetric*) if $\boldsymbol{\Psi}(h, 0, \boldsymbol{\Psi}(0, h, \mathbf{y})) = \mathbf{y}$ for every $\mathbf{y} \in \mathbf{D}$ whenever h is sufficiently small.

Theorem 8.5. *The maximal order of a reversible one-step method is always even.*

Theorem 8.6. *Gauss-collocation methods are reversible.*

Theorem 8.7. *The stability domain $S_{\boldsymbol{\Psi}}$ of a reversible and A-stable Runge-Kutta method satisfies $S_{\boldsymbol{\Psi}} = \mathbb{C}^-$.*

Next, we focus on Hamiltonian differential equations, that is, on ODEs of the form

$$\mathbf{p}' = -\mathbf{D}_{\mathbf{q}} H(\mathbf{p}, \mathbf{q}) \,, \quad \mathbf{q}' = \mathbf{D}_{\mathbf{p}} H(\mathbf{p}, \mathbf{q}) \,, \tag{8.3}$$

where the function $H : \mathbb{R}^d \times M \to \mathbb{R}$ ($M \subset \mathbb{R}^d$) is called the Hamiltonian. It is easy to see that H is a first integral of (8.3).

With the change of variables $\mathbf{y} \coloneqq \binom{\mathbf{p}}{\mathbf{q}} \in \mathbb{R}^{2d}$, it can be shown that (8.3) is equivalent to

$$\mathbf{y}' = \mathbf{J}^{-1} \operatorname{\mathbf{grad}} H(\mathbf{y}) \,, \quad \text{where} \quad \mathbf{J} = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{2d,2d} \,. \tag{8.4}$$

For $\mathbf{v}, \mathbf{w} \in \mathbb{R}^{2d}$, we define their *symplectic product* by $\omega(\mathbf{v}, \mathbf{w}) \coloneqq \mathbf{v}^\top \mathbf{J} \mathbf{w}$. A continuously differentiable map $\boldsymbol{\Phi} : D \subset \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ is called *symplectic* if $\omega(\mathbf{D}\boldsymbol{\Phi}(\mathbf{y})\mathbf{v}, \mathbf{D}\boldsymbol{\Phi}(\mathbf{y})\mathbf{w}) = \omega(\mathbf{v}, \mathbf{w})$ for every $\mathbf{v}, \mathbf{w} \in \mathbb{R}^{2d}$.

Theorem 8.8. *If H is C^2, the flow $\boldsymbol{\Phi}^t$ of (8.4) satisfies: for every $\mathbf{y} \in D$ there is a $\delta > 0$ such that*

$$\omega(\mathbf{D}\boldsymbol{\Phi}^t(\mathbf{y})\mathbf{v}, \mathbf{D}\boldsymbol{\Phi}^t(\mathbf{y})\mathbf{w}) = \omega(\mathbf{v}, \mathbf{w}) \quad \textit{for every } \mathbf{v}, \mathbf{w} \in \mathbb{R}^{2d} \,, \, 0 \le t < \delta \,.$$

A one-step method is *symplectic* if, when applied to (8.4), $\boldsymbol{\Psi}(0, h, \cdot)$ defines a symplectic map on every compact subset $K \subset D$ provided that $h > 0$ is sufficiently small.

Theorem 8.9. *Every Runge-Kutta method that preserves quadratic first integrals is symplectic.*

We conclude by mentioning that there exist also explicit numerical methods that are symplectic and/or preserve volumes. The most important examples for ODEs with "separate" righthand side are the Störmer-Verlet method[11]. and the class of Splitting methods. For more details, we refer to Hairer, Lubich, and Wanner, *Geometric Numerical Integration* (2006).

[11] In 1687, I. Newton proved Keplers second law using the Störmer-Verlet method, as proudly advertised by Richard Feynman in his *Messenger Lectures* (1964).

9 Gap lecture, questions

10 Introduction to linear multi-step methods

Runge-Kutta methods tacitly assume that it is possible to evaluate anywhere the right-hand side of an ODE $\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y})$ and use a lot of such function evaluations. Instead, linear multi-step methods require values of $\mathbf{f}$ at grid points only[12].

Given a sequence of equally spaced[13] grid points $t_n = t_0 + hn$, a *linear k-step method* is an iterative method that determines $\mathbf{y}_{n+k}$ by

$$\sum_{j=0}^{k} \alpha_j \mathbf{y}_{n+j} = h \sum_{j=0}^{k} \beta_j \mathbf{f}(t_{n+j}, \mathbf{y}_{n+j}) \,,$$

where $\{\alpha_j\}_{j=0}^k$ and $\{\beta_j\}_{j=0}^k$ are real coefficients. To avoid degenerate cases, we assume that $\alpha_k \neq 0$ and that $\alpha_0^2 + \beta_0^2 \neq 0$. Note that if $\beta_k = 0$, the method is explicit.

A simple linear 3-step method can be constructed using Simpson's quadrature rule. Indeed,

$$\begin{aligned} \mathbf{y}(t_{n+1}) &= \mathbf{y}(t_{n-1}) + \int_{t_{n-1}}^{t_{n+1}} \mathbf{f}(t, \mathbf{y}(t)) \, \mathrm{d}t \\ &\approx \mathbf{y}(t_{n-1}) + \frac{2h}{6} \left(\mathbf{f}(t_{n-1}, \mathbf{y}(t_{n-1})) + 4\mathbf{f}(t_n, \mathbf{y}(t_n)) + \mathbf{f}(t_{n+1}, \mathbf{y}(t_{n+1})) \right) . \end{aligned}$$

There is a formal calculus that can be used to construct families of multi-step methods. For a fixed small $h > 0$, consider the shift operator $E : \mathbf{y}(t) \mapsto \mathbf{y}(t+h)$, its inverse $E^{-1} : \mathbf{y}(t) \mapsto \mathbf{y}(t-h)$, and the difference operator $\Delta : \mathbf{y}(t) \mapsto \mathbf{y}(t) - \mathbf{y}(t-h)$. By formal calculation, it is easy to see that $E^{-1} = \mathbf{I} - \Delta$ (where $\mathbf{I}$ denotes the identity operator) and, therefore, $E = (\mathbf{I} - \Delta)^{-1}$. On the other hand, let D denote the differential operator $D : \mathbf{y}(t) \mapsto \mathbf{y}'(t)$. Taylor expansion implies that $E\mathbf{y}(t) = \exp(hD)\mathbf{y}(t)$, and thus, $E = \exp(hD)$. Therefore, by Taylor expansion of the logarithm,

$$hD = \log(E) = -\log(\mathbf{I} - \Delta) = \left(\Delta + \frac{1}{2}\Delta^2 + \frac{1}{3}\Delta^3 + \dots \right) ,$$

and thus

$$h\mathbf{f}(t, \mathbf{y}(t)) = \left(\Delta + \frac{1}{2}\Delta^2 + \frac{1}{3}\Delta^3 + \dots \right) \mathbf{y}(t) \,. \tag{10.1}$$

We can construct a family of multi-step methods by truncating the infinite series (10.1) at different orders and replacing $\mathbf{y}(t_n)$ with $\mathbf{y}_n$. This methods are called

[12] Runge-Kutta methods have been developed around 1900, whereas linear multi-step methods origin to at least 1855

[13] It is a bit more challenging to construct multi-step methods on nonregular grids.

backward differentiation formulas, and their simplest instances are

$$\begin{aligned} \mathbf{y}_n - \mathbf{y}_{n-1} &= h\mathbf{f}(t_n, \mathbf{y}_n)\,, \quad \text{(implicit Euler)} \\ \frac{3}{2}\mathbf{y}_n - 2\mathbf{y}_{n-1} + \frac{1}{2}\mathbf{y}_{n-2} &= h\mathbf{f}(t_n, \mathbf{y}_n)\,, \\ \frac{11}{6}\mathbf{y}_n - 3\mathbf{y}_{n-1} + \frac{3}{2}\mathbf{y}_{n-2} - \frac{1}{3}\mathbf{y}_{n-3} &= h\mathbf{f}(t_n, \mathbf{y}_n)\,. \end{aligned}$$

Explicit Euler's method arises from truncating the series

$$hD = \left(\Delta - \frac{1}{2}\Delta^2 - \frac{1}{6}\Delta^3 + \dots\right) E\,,$$

which can be derived similarly. Another two important families are the *Adams-Moulton methods* and the *Adams-Bashforth methods*, which originate from the formal equalities

$$\begin{aligned} E\Delta &= h\left(\mathbf{I} - \frac{1}{2}\Delta - \frac{1}{12}\Delta^2 - \frac{1}{24}\Delta^3 - \frac{19}{720}\Delta^4 + \dots\right) D\,, \\ E\Delta &= h\left(\mathbf{I} + \frac{1}{2}\Delta + \frac{5}{12}\Delta^2 + \frac{3}{8}\Delta^3 + \frac{251}{720}\Delta^4 + \dots\right) D\,, \end{aligned}$$

respectively.

To compute $\mathbf{y}_k$ with a linear k-step methods, we need the values $\mathbf{y}_0, \dots, \mathbf{y}_{k-1}$. These (except $\mathbf{y}_0$) must be approximated with either a one step method or another multi-step method that uses less steps. At any rate, they will contain numerical error. Clearly, a meaningful multistep method should be robust with respect to small perturbations of these initial values. A linear k-step method is said to be *zero-stable* if the iterates $\mathbf{y}_n$ and $\tilde{\mathbf{y}}_n$ that result from two different sets of initial data $\mathbf{y}_0, \dots, \mathbf{y}_{k-1}$ and $\tilde{\mathbf{y}}_0, \dots, \tilde{\mathbf{y}}_{k-1}$ satisfy (for a constant K)

$$\|\mathbf{y}_n - \tilde{\mathbf{y}}_n\| \le K \max_{j \le k-1} \|\mathbf{y}_j - \tilde{\mathbf{y}}_j\| \tag{10.2}$$

for every $n \le (T - t_0)/h$ if h is sufficiently small.

Zero-stability of a k-step method can be verified algebraically with the *root condition*. We denote by

$$\rho(z) = \sum_{j=0}^{k} \alpha_j z^j \quad \text{and} \quad \sigma(z) = \sum_{j=0}^{k} \beta_j z^j \tag{10.3}$$

the *first* and *second characteristic polynomials* of the k-step method.

Theorem 10.1. *A linear multi-step method is zero-stable for any ODE $\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y})$ with Lipschitz right-hand side, if and only if all zeros of its first characteristic polynomial lie inside the closed unit disc, and every zero that lies on the unit circle is simple.*

This theorem implies that zero-stability of a multi-step method can be determined by merely considering its behavior when applied to the trivial differential equation $y' = 0$; it is for this reason that it is called *zero*-stability.

11 Consistency and convergence of linear multi-step methods

The definition of the *consistency error* $\boldsymbol{\tau}$ for a linear multi-step method is less straightforward than the one for one-step methods because to compute $\mathbf{y}_{n+k}$ we need to rely on $\mathbf{y}_n, \dots, \mathbf{y}_{n+k-1}$, which may be perturbed by numerical errors. To overcome this difficulty, we start by defining

$$\boldsymbol{\tau}(h, \mathbf{y}_0) \coloneqq \frac{\mathbf{y}(t_k) - \tilde{\mathbf{y}}_k}{h}\,, \tag{11.1}$$

where $\tilde{\mathbf{y}}_k$ is the solution of

$$\left(\sum_{j=0}^{k-1} \alpha_j \mathbf{y}(t_j)\right) + \alpha_k \tilde{\mathbf{y}}_k = h\left(\sum_{j=0}^{k-1} \beta_j \mathbf{f}(t_j, \mathbf{y}(t_j))\right) + h\beta_k \mathbf{f}(t_k, \tilde{\mathbf{y}}_k)\,. \tag{11.2}$$

Then, adding and subtracting $\alpha_k \mathbf{y}(t_k) - h\beta_k \mathbf{f}(t_k, \mathbf{y}(t_k))$ to (11.2), we can show that

$$\alpha_k(\mathbf{y}(t_k) - \tilde{\mathbf{y}}_k) - h\beta_k\left(\mathbf{f}(t_k, \mathbf{y}(t_k) - \mathbf{f}(t_k, \tilde{\mathbf{y}}_k)\right) = \left(\sum_{j=0}^{k} \alpha_j \mathbf{y}(t_j) - h\sum_{j=0}^{k} \beta_j \mathbf{f}(t_j, \mathbf{y}(t_j))\right)\,.$$

If $\mathbf{f}(t, \mathbf{y})$ if continuously differentiable, the mean value theorem implies that there is an $\boldsymbol{\eta} \in \{t\mathbf{y}(t_k) + (1-t)\tilde{\mathbf{y}}_k : t \in [0,1]\}$ such that

$$\mathbf{f}(t_k, \mathbf{y}(t_k)) - \mathbf{f}(t_k, \tilde{\mathbf{y}}_k) = \mathbf{D}\mathbf{f}(t_k, \boldsymbol{\eta})(\mathbf{y}(t_k) - \tilde{\mathbf{y}}_k). \tag{11.3}$$

Therefore,

$$\mathbf{y}(t_k) - \tilde{\mathbf{y}}_k = \Big(\alpha_k \mathbf{I} - h\beta_k \mathbf{D}\mathbf{f}(t_j, \boldsymbol{\eta})\Big)^{-1}\left(\sum_{j=0}^{k} \alpha_j \mathbf{y}(t_j) - h\sum_{j=0}^{k} \beta_j \mathbf{f}(t_j, \mathbf{y}(t_j))\right)\,.$$

For $h > 0$ sufficiently small, $\big(\alpha_k \mathbf{I} - h\beta_k \mathbf{D}\mathbf{f}(t_j, \boldsymbol{\eta})\big)^{-1} = \mathbf{I}/\alpha_k + \mathcal{O}(h)$ (by Neumann series expansion). Therefore, the consistency error $\boldsymbol{\tau}$ satisfies

$$\boldsymbol{\tau}(h, \mathbf{y}_0) = \frac{\sum_{j=0}^{k} \alpha_j \mathbf{y}(t_j) - h\sum_{j=0}^{k} \beta_j \mathbf{f}(t_j, \mathbf{y}(t_j))}{\alpha_k h}(1 + \mathcal{O}(h))\,. \tag{11.4}$$

Equation (11.4) is intuitive, but ends up giving uncorrect error constants[14]. For this reason, we replace the definition of consistency error (11.1) with

$$\boldsymbol{\tau}(h, \mathbf{y}_0) = \frac{\sum_{j=0}^{k} \alpha_j \mathbf{y}(t_j) - h\sum_{j=0}^{k} \beta_j \mathbf{f}(t_j, \mathbf{y}(t_j))}{h\sum_{j=0}^{k} \beta_j}$$

[14] For details, we refer to the section *The error constant of a multistep method* in chapter III.2 of the book *Solving ordinary differential equations 1* (1987) by Hairer, Nørsett, and Wanner.

and require that $\sum_{j=0}^{k} \beta_j = \sigma(1) \neq 0$.

A linear multi-step method has *(consistency) order* p if $\boldsymbol{\tau}(h, \mathbf{y}_0) = \mathcal{O}(h^p)$ for sufficiently smooth data. By adequate Taylor expansion, we can obtain the following theorem.

Theorem 11.1. *A linear multi-step method has consistency order p if and only if $\sigma(1) \neq 0$ and*

$$\sum_{j=0}^{k} \alpha_j = 0 \quad \textit{and} \quad \sum_{j=0}^{k} \alpha_j j^q = q \sum_{j=0}^{k} \beta_j j^{q-1} \quad \textit{for} \quad q = 1, \ldots, p\,. \tag{11.5}$$

A multi-step method is said to be *consistent* if conditions (11.5) are satisfied at least for $p = 1$. This condition can be equivalently formulated as

$$\rho(1) = 0 \quad \text{and} \quad \rho'(1) = \sigma(1) \neq 0.$$

In general, conditions (11.5) can be equivalently and elegantly reformulated as

$$\rho(e^h) - h\sigma(e^h) = \mathcal{O}(h^{p+1})\,.$$

A k-step method is said to be *convergent* if, for every initial value problem $\mathbf{y} = \mathbf{f}(t, \mathbf{y})$, $\mathbf{y}(t_0) = \mathbf{y}_0$, that satisfies the assumptions of Picard's theorem, we have that

$$\lim_{h \to 0} \mathbf{y}_N = \mathbf{y}(T) \quad (\text{with } N = (T - t_0)/h)$$

holds for every choice of *consistent starting conditions*

$$\mathbf{y}_0 = \boldsymbol{\eta}_0(h), \ldots, \mathbf{y}_{k-1} = \boldsymbol{\eta}_{k-1}(h)\,,$$

where $\boldsymbol{\eta}_s(h) \to \mathbf{y}_0$ as $h \to 0$ for every $s = 0, \ldots, k-1$.

Theorem 11.2 (Dahlquist's Equivalence Theorem)**.** *For a linear k-step method that is consistent with the ordinary differential equation $\mathbf{y} = \mathbf{f}(t, \mathbf{y})$ with Lipschitz $\mathbf{f}$, and with consistent starting values, zero-stability is necessary and sufficient for convergence. Moreover, if the solution $\mathbf{y}$ has continuous derivatives of order $p + 1$, the consistency order is p, and $\|\mathbf{y}(t_s) - \boldsymbol{\eta}_s(h)\| = \mathcal{O}(h^p)$ for $s = 0, \ldots, k-1$, then the global error of the method $\mathbf{y}(t_n) - \mathbf{y}_n$ is also $\mathcal{O}(h^p)$.*

Theorem 11.3 (The first Dahlquist-barrier)**.** *The order p of a zero-stable linear k-step method satisfies*

- *$p \leq k + 2$ if k is even,*
- *$p \leq k + 1$ if k is odd,*
- *$p \leq k$ if $\beta_k / \alpha_k \leq 0$ (in particular if the method is explicit).*

Theorem 11.4. *Zero-stable multistep methods of order $k + 2$ are reversible.*

12 Stability of linear multi-step methods

Similarly to one-step methods, stability is investigated applying a linear multi-step methods to the Dahlquist's test equation $y' = \lambda y$, $\lambda \in \mathbb{C}$, $y(0) = 1$. Recall that the solution to this ODE is $y(t) = \exp(\lambda t)$, that $|y(t)| \to 0$ as $t \to \infty$ whenever $\mathrm{Re}(\lambda) < 0$, and that we call its numerical approximation $\{y_n\}_{n\in\mathbb{N}}$ (absolutely) stable if it $y_n \to 0$ as $n \to \infty$ when $\mathrm{Re}(\lambda) < 0$.

The iterate y_n satisfies

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j \lambda y_{n+j}\,, \quad \text{or equivalently,} \quad \sum_{j=0}^{k} (\alpha_j - h\lambda\beta_j) y_{n+j} = 0\,. \tag{12.1}$$

We define the *stability polynomial*

$$\pi(\zeta) = \pi(\zeta; z) \coloneqq \sum_{j=0}^{k} (\alpha_j - z\beta_j)\zeta^j = \rho(\zeta) - z\sigma(\zeta)\,. \tag{12.2}$$

Lemma 12.1. *Let $\zeta_1, \dots, \zeta_\ell$ be the roots (of respective multiplicity $m_1, \dots, m_\ell$) of* (12.2) *(with $z = h\lambda$). Then, the general solution of* (12.1) *is given by*

$$y_n = p_1(n)\zeta_1^n + \cdots + p_\ell(n)\zeta_\ell^n\,,$$

where the $p_j(n)$s are polynomials of degree $m_j - 1$.

This lemma[15] implies that if the polynomial $\pi(\zeta)$ has a zero ζ_j out of the unit disc, than y_n grows as $|\zeta_j|^n$, and that if there is a zero ζ_j on the unit circle with multiplicity $m_j > 1$, then y_n grows as $n^{m_j - 1}$. For this reason, we define the *stability domain* of a linear multistep method as follows

$$S \coloneqq \{z \in \mathbb{C} : \text{if } \pi(\zeta; z) = 0, \text{ then } |\zeta| \le 1; \text{ multiple zeros satisfy } |\zeta| < 1\}\,.$$

Note that $0 \in S$ if the method is zero-stable. Unfortunately, determining the stability domain of a linear multistep method is a challenging task. We can gather some information on S by asking ourselves for which z the polynomial $\pi(\cdot\,; z)$ has a zero on the unit circle, that this, by looking at the curve

$$\{z = \rho(e^{i\theta})/\sigma(e^{i\theta}) : \theta \in [0, 2\pi]\}$$

This curve is called *zero lotus curve* and represents the points of z which could constitute the boundary of S. Note that a zero lotus curve might intersect itself. Therefore, the boundary of S consists only of parts of the zero lotus curve, in general.

Theorem 12.1 (Dahlquist's second barrier)**.** *An A-stable linear multi-step method must be implicit and of order $p \le 2$. The trapezium rule is the second-order A-stable linear multi-step method with the smallest error constant.*

[15] A similar lemma and the same reasoning lies at the core of Theorem 10.1.

To break this barrier there are two options: weakening the stability requirement or improving the method. Here, we consider only the first option.

A linear multi-step method is said to be $A(\theta)$-stable, $\theta \in (0, \pi/2)$, if stability region S contains the infinite wedge

$$\{z : \pi - \theta < \arg(z) < \pi + \theta\}\,.$$

For instance, the k-step BDF methods are $A(\theta)$-stable with

k	1	2	3	4	5	6
θ	$90°$	$90°$	$86.03°$	$73.35°$	$51.84°$	$17.84°$.

A linear multi-step method is $A(0)$-*stable* if is it $A(\theta)$-stable for some $\theta \in (0, \pi/2)$

Theorem 12.2. *No explicit linear multi-step method is $A(0)$-stable. The only $A(0)$-stable linear k-step method whose order exceed k is the trapezium rule.*

It has been shown that[16] for any given $\theta < \pi/2$ and every $k \in \mathbb{N}$ there is an $A(\theta)$-stable linear k-step method of order $p = k$. Unfortunately, it has also been proven that such methods cannot be of practical use because their error constants are enormous[17].

We conclude by mentioning the concept of *stiff-stability* introduced by Gear: a linear multi-step method is stiffly stable if there exist positive real numbers a and c such that S contains the two sets

$$\{z \in \mathbb{C} : \mathrm{Re}(z) < -a\} \quad \text{and} \quad \{z \in \mathbb{C} : -a \le \mathrm{Re}(z) < 0\,, -c \le \mathrm{Im}(z) \le c\}\,.$$

For instance, BDF methods are stiffly stable with $c = a\tan(\theta)$ where

k	1	2	3	4	5	6
a	0	0	0.083	0.667	2.327	6.075.

In general, we have that (with $\theta = \arctan(c/a)$)

$$A\text{-stability} \Rightarrow \text{stiff-stability} \Rightarrow A(\theta)\text{-stability} \Rightarrow A(0)\text{-stability} \Rightarrow \text{zero-stability}\,.$$

[16] See Grigorieff and Schroll, *Über $A(\alpha)$-stabile Verfahren hoher Konsistenzordnung* (1978).

[17] See Jeltsch and Nevanlinna *Stability and accuracy of time discretizations for initial value problems* (1982).

13 Initial boundary value problems: Introduction

The flow of heat in a homogeneous unchanging bounded medium with no heat source can be modeled with the boundary value problem (subscripts denote partial derivatives)

$$u_t(t,x) = u_{xx}(t,x)\,, \quad \text{for } (t,x) \in (0,T] \times (0,1)\,, \tag{13.1}$$

($T > 0$ denotes a final time), together with the homogeneous Dirichlet boundary condition

$$u(t,0) = u(t,1) = 0\,, \quad \text{for } t \in [0,T]\,, \tag{13.2}$$

and an initial condition

$$u(0,x) = u^0(x)\,, \quad \text{for } x \in [0,1]\,, \tag{13.3}$$

(we tacitly assume that $u^0(0) = u^0(1) = 0$). Different boundary conditions, such as nonhomogeneous, Neumann, Robin, or mixed, as well as different (bounded or unbounded) space intervals could be considered, but for convenience we restrict ourselves to this simple test case.

The solution of such a simple problem can be computed analytically using the ansatz $u(t,x) = f(x)g(t)$. Substituting into (13.1) and (temporarily) assuming that $f, g \neq 0$, we obtain

$$g(t)'/g(t) = f''(x)/f(x)\,. \tag{13.4}$$

Since the variable t is independent of x and viceversa, equation (13.4) can hold only if the ratio g'/g is constant. An educated guess[18] is to write this constant as $-k^2$ for a real value k. Then, (13.4) can be rewritten as the following two separate differential equations

$$g'(t) = -k^2 g(t) \quad \text{and} \quad f''(x) = -k^2 f(x)\,,$$

whose solutions are (for $A_1, A_2, A_3 \in \mathbb{R}$)

$$g(t) = A_1 e^{-k^2 t} \quad \text{and} \quad f(x) = A_2 \sin(kx) + A_3 \cos(kx)\,.$$

The boundary condition (13.2) imposes the restrictions $A_3 = 0$ and $k = m\pi$, for an $m \in \mathbb{N}$. By the superposition principle (note that (13.1) is linear), we conclude that

$$u(t,x) = \sum_{m=1}^{\infty} a_m e^{-(m\pi)^2 t} \sin(m\pi x)\,, \tag{13.5}$$

for certain real coefficients $\{a_m\}_{m=1}^{\infty}$. Equation (13.3) determines the remaining coefficients. Indeed,

$$u(0,x) = \sum_{m=1}^{\infty} a_m \sin(m\pi x) = u^0(x)\,,$$

[18] This constant must be real, because heat is a real variable, and negative, because otherwise heat (and thus energy) would increase exponentially in time.

which implies that (assuming that $u^0 \in C^1(0,1)$[19])

$$a_m = 2\int_0^1 u^0(x)\sin(m\pi x)\,\mathrm{d}x\,.$$

This suggests a simple numerical method to approximate u: compute sufficiently many coefficients of the sine series of u^0 (that is, of the Fourier series of the odd extension of u^0) and truncate (13.5). This approach is very efficient if the Fourier coefficient decay rapidly, but it cannot be used if the PDE is even slightly more complicated.

To devise a more general numerical scheme, we begin by discretizing (13.1) in space with a central difference scheme: we introduce a uniform spatial grid $x_j = j\Delta x$, $j = 0,\dots,N$ (with $\Delta x = 1/N$) and approximate

$$u_{xx}(t,x_j) \approx u_{xx}^{\Delta x}(t,x_j) \coloneqq \frac{u(t,x_{j+1}) - 2u(t,x_j) + u(t,x_{j-1})}{\Delta x^2} \quad \text{for } j = 1,\dots,N-1\,.$$

Then, denoting $u_j(t) \approx u(t,x_j)$, (13.1) becomes a system of $N-1$ ODEs

$$\mathbf{u}'(t) = \frac{1}{\Delta x^2}\mathbf{K}\mathbf{u}(t) \tag{13.6}$$

where

$$\mathbf{K} = \begin{pmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{N-1,N-1} \quad \text{and} \quad \mathbf{u} = \begin{pmatrix} u_1(t) \\ \vdots \\ u_{N-1}(t) \end{pmatrix} \in \mathbb{R}^{N-1}\,.$$

This resulting initial value problem can be solved with any numerical scheme for ODEs we studied in the previous lectures. Just be aware that stability may be an issue due to the factor $1/\Delta x^2$ in the right-hand side of (13.6), and that the dimension of the system depends on the space discretization.

Discretizing first in space and then in time is know as "method of lines". The advantage of this approach is that one can combine and experiment with different kind of discretizations. Additionally, decoupling time and space allows for a simpler error analysis.

[19] For weaker assumptions, have a look at `https://en.wikipedia.org/wiki/Convergence_of_Fourier_series` .

14 Initial boundary value problems: Stability

Discretizing in space, the heat equation (13.1) becomes the following $(N-1)$-dimensional initial value problem

$$\mathbf{u}'(t) = \frac{1}{\Delta x^2}\mathbf{K}\mathbf{u}(t)\,, \quad \mathbf{u}(0) = \mathbf{u}^0\,, \tag{14.1}$$

where $\Delta x = 1/N$,

$$\mathbf{K} = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{N-1,N-1} \quad \text{and} \quad \mathbf{u}^0 = \begin{pmatrix} u^0(\Delta x) \\ u^0(2\Delta x) \\ \vdots \\ u^0(1-\Delta x) \end{pmatrix} \in \mathbb{R}^{N-1}\,.$$

Solving (14.1) with a time-stepping schemes poses stability issues due to the term $1/\Delta x^2$ in front of the matrix $\mathbf{K}$. To see why, we begin by studying the eigenvalues of $\mathbf{K}$, which arises from the discretization

$$u_{xx}(t,x_j) \approx \frac{u(t,x_{j+1}) - 2u(t,x_j) + u(t,x_{j-1})}{\Delta x^2} \quad \text{for } j = 1,\dots,N-1\,.$$

Lemma 14.1. *For any $k, \Delta x \in \mathbb{R}$ and $j \in \mathbb{N}$,*

$$\sin\big(k(j+1)\Delta x\big) - 2\sin\big(kj\Delta x\big) + \sin\big(k(j-1)\Delta x\big) = 2\big(\cos(k\Delta x) - 1\big)\sin\big(kj\Delta x\big)\big)\,.$$

Proof. It is convenient to replace $\sin(x)$ with e^{ix}. By direct calculation,

$$\begin{aligned} e^{ik(j+1)\Delta x} - 2e^{ikj\Delta x} + e^{ik(j-1)\Delta x} &= (e^{ik\Delta x} - 2 + e^{-ik\Delta x})e^{ikj\Delta x}\,, \\ &= 2\big(\cos(k\Delta x) - 1\big)e^{ikj\Delta x}\,. \end{aligned}$$

To conclude, note that $\sin(x) = (e^{ix} - e^{-ix})/2i$ and $\cos(k\Delta x) = \cos(-k\Delta x)$. □

Lemma 14.1, which is inspired by Fourier analysis, implies that the vectors

$$\mathbf{z}_p^\top = (\sin(p\pi\Delta x), \sin(2p\pi\Delta x), \dots, \sin((N-1)p\pi\Delta x))\,, \quad p = 1,\dots,N-1\,, \tag{14.2}$$

are a basis of eigenvectors of $\mathbf{K}$, and that their eigenvalues are

$$\lambda_p = 2\big(\cos(p\pi\Delta x) - 1\big) \quad p = 1,\dots,N-1\,, \tag{14.3}$$

respectively (we choose $k = p\pi$ because we need $\sin(N\Delta x k) = \sin(k) = 0$; note also that $\lambda_p \neq \lambda_q$ if $p \neq q$ and $1 \leq p, q \leq N-1$). This implies that the matrix $\mathbf{K}$ is diagonalizable. In particular, let $\mathbf{Z}$ be the matrix whose columns are the eigenvectors $\mathbf{z}_p$. Then, $\mathbf{D} \coloneqq \mathbf{Z}^{-1}\mathbf{K}\mathbf{Z}$ is a diagonal matrix whose nonzero entries are the eigevalues λ_p's.

This information is relevant to study the stability of a numerical solution obtained applying a Runge-Kutta method to (14.1), because Runge-Kutta methods are affine covariant, that is, their behavior is invariant with respect to

a linear change of variables. In practice, this means that the approximation $\mathbf{u}^1 \approx \mathbf{u}(\Delta t)$ satisfies $\mathbf{u}^1 = \mathbf{Z}\mathbf{y}^1$, where $\mathbf{y}^1$ is the Runge-Kutta approximation of the solution $\mathbf{y}(\Delta t)$ to

$$\mathbf{y}' = \frac{1}{\Delta x^2}\mathbf{D}\mathbf{y}\,, \quad \mathbf{y}^0 = \mathbf{Z}^{-1}\mathbf{u}^0\,. \tag{14.4}$$

This initial value problem is obtained performing the linear change of variables $\mathbf{y} = \mathbf{Z}^{-1}\mathbf{u}$ to (14.1), because

$$\mathbf{y}' = \mathbf{Z}^{-1}\mathbf{u}' = \frac{1}{\Delta x^2}\mathbf{Z}^{-1}\mathbf{K}\mathbf{u} = \frac{1}{\Delta x^2}\mathbf{Z}^{-1}\mathbf{Z}\mathbf{D}\mathbf{Z}^{-1}\mathbf{u} = \frac{1}{\Delta x^2}\mathbf{D}\mathbf{y}\,.$$

The initial value problem (14.4) is a system of decoupled scalar differential equations. Similarly to section 6, it is not too difficult to see that

$$\mathbf{y}^1 = S(\frac{\Delta t}{\Delta x^2}\mathbf{D})\mathbf{y}^0\,, \tag{14.5}$$

where the stability function S acts componentwise on the diagonal elements of $\mathbf{D}$ (that is, the jth component y_j^1 of $\mathbf{y}^1$ satisfies $y_j^1 = S(\frac{\Delta t}{\Delta x^2}\lambda_j)y_j^0$).

Note that $\lambda_p \in [-4, 0)$. Therefore, numerical instabilities arise if $\frac{\Delta t}{\Delta x^2}\lambda_p$ is not in the stability region of the Runge-Kutta method[20]. If we employ an explicit method, then the time-step size Δt must be proportional to Δx^2 to ensure stability (because the stability region of explicit methods is always bounded) and this implies that one needs to perform roughly $T/\Delta x^2$ time-steps. For a more concrete example, we consider the explicit Euler. The value $\frac{\Delta t}{\Delta x^2}\lambda_p$ is in its stability region if $|1 + \frac{\Delta t}{\Delta x^2}\lambda_p| < 1$. Since $\lambda_p \in [-4, 0)$, the worst case scenario happens if $\lambda_p = -4$. In this case, we need to make sure that $1 - 4\Delta t/\Delta x^2 > -1$, that is, $\Delta t/\Delta x^2 < 1/2$.

To get rid of the time-step size restriction, we can employ an A-stable (or even better, an L-stable) Runge-Kutta method. The price to pay is that an A-stable method is necessarily implicit, and the numerical effort can become very expensive because (14.1) is inherently high dimensional (its dimension is inversely proportional to the number of space-grid points). However, for (14.1) this computational cost is affordable because $\mathbf{K}$ is tridiagonal. For instance, for the theta-scheme, $\mathbf{u}_1$ is the solution of

$$\left(\mathbf{I} - \theta\frac{\Delta t}{\Delta x^2}\mathbf{K}\right)\mathbf{u}^1 = \left(\mathbf{I} + (1-\theta)\frac{\Delta t}{\Delta x^2}\mathbf{K}\right)\mathbf{u}^0. \tag{14.6}$$

This system is implicit for $\theta > 0$, but can be solved in linear cost with respect to N using Thomas algorithm[21], because the matrix $\left(\mathbf{I} - \theta\frac{\Delta t}{\Delta x^2}\mathbf{K}\right)$ is tridiagonal (since $\mathbf{K}$ is).

To conclude, we mention that Von Neumann analysis is an alternative technique to investigate stability. For details, we refer to chapter 9.6 of LeVeque's book *Finite difference methods for ordinary and partial differential equations.*

[20] In fact, the biggest eigenvalue of $\mathbf{K}/\Delta x^2$ is $\lambda_1 \approx -\pi^2$ whereas the smallest is proportional to $-4/\Delta x^2$. Therefore, the initial value problem (14.1) is stiff because its presents very different time scales. For more details, see chapter 9.4 of LeVeque's book.

[21] See `https://en.wikipedia.org/wiki/Tridiagonal_matrix_algorithm` .

15 Initial boundary value problems: Consistency

In lecture 3, we considered the IVP $\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \mathbf{y}(t_0) = \mathbf{y}_0$, and studied the consistency error

$$\boldsymbol{\tau}(t, h, \mathbf{y}_0) = \frac{\mathbf{y}(t+h) - \boldsymbol{\Psi}(t, t+h, \mathbf{y}(t))}{h}.$$

of a generic one-step method $\boldsymbol{\Psi}$. Here, we study a similar concept for the heat equation. For simplicity, we restrict ourselves to the 2-stage familiy Runge-Kutta methods[22]

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1-\theta & \theta \\ \hline & 1-\theta & \theta \end{array}, \tag{15.1}$$

where $\theta \in [0, 1]$. A glimpse at Lemma 4.3 tells us that this Runge-Kutta method has at least order 1, and has order 2 for $\theta = 1/2$. It is also easy to see that this method is A-stable for $\theta \in [1/2, 1]$ and that is is L-stable for $\theta = 1$.

We fix a generic space point $\tilde{x} \in (0, 1)$ and consider the initial value problem

$$u_t(t, \tilde{x}) = u_{xx}(t, \tilde{x}), \quad u(0, \tilde{x}) = u^0(\tilde{x}). \tag{15.2}$$

For $\Delta x > 0$, we introduce the perturbed initial value problem

$$u_t(t, \tilde{x}) = u_{xx}^{\Delta x}(t, \tilde{x}), \quad u(0, \tilde{x}) = u^0(\tilde{x}), \tag{15.3}$$

where

$$u_{xx}^{\Delta x}(t, \tilde{x}) \coloneqq \frac{u(t, \tilde{x} + \Delta x) - 2u(t, \tilde{x}) + u(t, \tilde{x} - \Delta x)}{\Delta x^2}.$$

Following the method of lines policy, we compute an approximation of the solution $u(t+\Delta t, \tilde{x})$ of (15.2) by applying one step of (15.1) to (15.3). The resulting numerical approximation reads

$$u(t + \Delta t, \tilde{x}) \approx u(t, \tilde{x}) + \Delta t \left((1-\theta) u_{xx}^{\Delta x}(t, \tilde{x}) + \theta u_{xx}^{\Delta x}(t + \Delta t, \tilde{x}) \right). \tag{15.4}$$

The *truncation error* of the scheme is defined as

$$\tau(t, \tilde{x}) \coloneqq \frac{u(t + \Delta t, \tilde{x}) - \left(u(t, \tilde{x}) + \Delta t \left((1-\theta) u_{xx}^{\Delta x}(t, \tilde{x}) + \theta u_{xx}^{\Delta x}(t + \Delta t, \tilde{x}) \right) \right)}{\Delta t} \tag{15.5}$$

To estimate this truncation error, note that (15.5) can be rewritten as

$$\frac{u(t + \Delta t, \tilde{x}) - \left(u(t, \tilde{x}) + \Delta t \left((1-\theta) u_t(t, \tilde{x}) + \theta u_t(t + \Delta t, \tilde{x}) \right) \right)}{\Delta t} \tag{15.6}$$

$$+ \left((1-\theta)(u_{xx}(t, \tilde{x}) - u_{xx}^{\Delta x}(t, \tilde{x})) + \theta (u_{xx}(t + \Delta t, \tilde{x}) - u_{xx}^{\Delta x}(t + \Delta t, \tilde{x})) \right). \tag{15.7}$$

[22] The simplicity comes from the fact that $c_0 = 0$ and $c_1 = 1$. Note that the 1-stage Runge-Kutta family $\begin{array}{c|c} \theta & \theta \\ \hline & 1 \end{array}$ would have led exactly to the same numerical solution, because the right-hand side of (15.2) is linear in u.

If $\theta \neq 1/2$, the term (15.6) is bounded by $C\|u_{tt}(\cdot,\tilde{x})\|_{C^0(t,t+\Delta t)}\Delta t$, whereas if $\theta = 1/2$, this bound improves to $C\|u_{ttt}(\cdot,\tilde{x})\|_{C^0(t,t+\Delta t)}\Delta t^2$. On the other hand, the following lemma readily implies that (15.7) is bounded by $C\|u_{xxxx}(\cdot,\tilde{x})\|_{C^0(0,1)}\Delta x^2$. Therefore, the truncation error (15.5) behaves as

$$\mathcal{O}(\Delta t + \Delta x^2) \quad \text{for } \theta \neq 1/2\,, \quad \text{and as} \quad \mathcal{O}(\Delta t^2 + \Delta x^2) \quad \text{for } \theta = 1/2\,.$$

Lemma 15.1. $|u_{xx}(t,\tilde{x}) - u_{xx}^{\Delta x}(t,\tilde{x})| = \mathcal{O}(\Delta x^2)$.

Proof. The result follows by replacing $u(t,\tilde{x}+\Delta x)$ and $u(t,\tilde{x}-\Delta x)$ with the Taylor expansions

$$u(t,\tilde{x}+\Delta x) = u(t,\tilde{x}) + \Delta x\, u_x(t,\tilde{x}) + \frac{\Delta x^2}{2}u_{xx}(t,\tilde{x}) + \frac{\Delta x^3}{3!}u_{xxx}(t,\tilde{x}) + \mathcal{O}(\Delta x^4)\,,$$
$$u(t,\tilde{x}-\Delta x) = u(t,\tilde{x}) - \Delta x\, u_x(t,\tilde{x}) + \frac{\Delta x^2}{2}u_{xx}(t,\tilde{x}) - \frac{\Delta x^3}{3!}u_{xxx}(t,\tilde{x}) + \mathcal{O}(\Delta x^4)\,.$$

□

To conclude, we study when a consistent scheme converges. Note that the numerical solution obtained by applying a Runge-Kutta scheme to the spatially discretized equation is given by

$$\mathbf{u}^k = \mathbf{S}^k\mathbf{u}^0\,, \quad \text{where} \quad \mathbf{S} \coloneqq S\left(\frac{\Delta t}{\Delta x^2}\mathbf{K}\right)\,. \tag{15.8}$$

The "error vector"

$$\mathbf{e}^k \coloneqq (u(k\Delta t,\Delta x), u(k\Delta t,2\Delta x),\dots,u(k\Delta t,1-\Delta x))^\top - \mathbf{u}^k$$

satisfies

$$\mathbf{e}^{k+1} = \mathbf{S}\mathbf{e}^k + \Delta t\boldsymbol{\tau}^k\,,$$

where $\boldsymbol{\tau}^k \coloneqq (\tau(k\Delta t,\Delta x),\tau(k\Delta t,2\Delta x),\dots,\tau(k\Delta t,1-\Delta x))^\top$. Therefore (with $M = T/\Delta t$),

$$\|\mathbf{e}^M\| = \|\mathbf{S}^M\mathbf{e}^0 + \Delta t\sum_{n=1}^{M}\mathbf{S}^{M-n}\boldsymbol{\tau}^{n-1}\| \leq \|\mathbf{S}^M\|\|\mathbf{e}^0\| + \Delta t\sum_{n=1}^{M}\|\mathbf{S}^{M-n}\|\|\boldsymbol{\tau}^{n-1}\|\,.$$

It is easy to see that $\|\mathbf{e}^M\|$ inherits the asymptotic behavior of the truncation error τ, provided that the following condition is satisfied (*Lax-Richtmyer* stability[23]): for every time T there is a constant C_T such that $\|\mathbf{S}^k\| \leq C_T$ for every ratio $(\Delta t/\Delta x^2) > 0$ and every integer $k \leq T/\Delta t$.

Theorem 15.1 (Lax Equivalence Theorem)**.** *A consistent linear method of the form* (15.8) *is convergent if and only if it is Lax-Richtmyer stable.*

[23]This notion of stability is less strict than the notion of stability of one-step methods. For instance, it is satisfied also if $\|\mathbf{S}^k\| \leq 1 + \alpha\Delta t/\Delta x^2$ for an $\alpha > 0$.

16 Initial boundary value problems: Two space dimensions

In two space dimensions, the heat equation takes the form

$$u_t(t,x,y) = u_{xx}(t,x,y) + u_{yy}(t,x,y)\,, \quad \text{for } (t,(x,y)) \in (0,T]\times\Omega\,. \tag{16.1}$$

For simplicity, we assume that the computation domain Ω is the square $(0,1)\times(0,1)$. As for the 1D case, (16.1) is accompanied by an initial condition

$$u(0,x,y) = u^0(x,y)\,,$$

and by some boundary conditions. Here, we consider homogeneous Dirichlet boundary conditions, that is,

$$u(t,x,y) = 0 \quad \text{when} \quad (x,y)\in\partial\Omega\,.$$

Following the method of lines policy, we discretize the right-hand side of (16.1) with $u_{xx}^{\Delta x}(t,x,y)+u_{yy}^{\Delta y}(t,x,y)$, that is (with $h=\Delta x=\Delta y$, and $N=1/h$), with

$$\frac{u(t,x+h,y)+u(t,x-h,y)+u(t,x,y+h)+u(t,x,y-h)-4u(t,x,y)}{h^2}\,.$$

Let $\{(ih,jh) : i=0,\dots,N,\ j=0,\dots,N\}$ be the set of vertices of a Cartesian grid of Ω. The vector

$$\Big(u(t,h,h),u(t,h,2h),\dots,u(t,h,1-h),\\ u(t,2h,h),u(t,2h,2h),\dots,u(t,2h,1-h),\\ \dots,u(t,1-h,h),u(t,1-h,2h),\dots,u(t,1-h,1-h)\Big)^\top$$

can be approximated solving the system of initial value problems

$$\mathbf{U}' = \frac{1}{h^2}\mathbf{K}\mathbf{U}\,, \quad \mathbf{U}(0)=\mathbf{U}^0\,, \tag{16.2}$$

where $\mathbf{K}$ is the sparse tridiagonal block matrix

$$\mathbf{K} = \begin{pmatrix} \mathbf{B} & \mathbf{I} & & & \\ \mathbf{I} & \mathbf{B} & \mathbf{I} & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{I} & \mathbf{B} & \mathbf{I} \\ & & & \mathbf{I} & \mathbf{B} \end{pmatrix} \in \mathbb{R}^{(N-1)^2,(N-1)^2}$$

with sparse blocks

$$\mathbf{B} = \begin{pmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ & & & 1 & -4 \end{pmatrix} \in \mathbb{R}^{N-1,N-1} \quad \text{and} \quad \mathbf{I} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \in \mathbb{R}^{N-1,N-1}\,.$$

Similarly to the 1D case, one can show that the eigenvalues of $\mathbf{K}$ are

$$\lambda_{p,q} = 2\left(\cos(p\pi h) + \cos(q\pi h) - 2\right), \quad \text{where} \quad p, q = 1, \ldots, N-1\,. \tag{16.3}$$

Since $\lambda_{p,q} \in [-8, 0)$, the system of ODEs (16.2) is more stiff than the one for the 1D case. Therefore, stability imposes even stricter requirements on the ratio $\Delta t/h^2$, if one employs an explicit one-step method. For instance, for the explicit Euler method one need $\Delta t/h^2 < 1/4$. On the other hand, note that $\mathbf{K}$ is not tridiagonal in 2D, so that solving a linear system that arises from the use of an implicit one-step method becomes computationally more demanding. We just mention in passing that often one can employ iterative methods to solve these systems[24], although current implementations of direct solvers are hard to beat for 2D problems.

Consistency of the fully discretized methods (that stems from choosing a one-step method to solve (16.2)) can be inferred/estimated repeating the procedure for the theta-scheme explained in Section 15.

[24] See the discussion in chapter 9.7 of LeVeque's book *Finite difference methods for ordinary and partial differential equations*.