Initial value problems for ODEs

In this lecture we briefly review the mathematical theory of initial value problems for systems of first-order ordinary differential equations (ODEs). While systems of ODEs are of great importance on their own (many real-world systems are modeled in terms of ODEs), they also play a fundamental role in the numerical approximation of PDEs.

The initial value problem for one ODE. Let us begin with the following initial value problem for just one ODE

$$\begin{cases} \dfrac{dy}{dt} = f(y,t) \\ y(0) = y_0 \end{cases} \tag{1}$$

where $f : D \times [0,T] \mapsto \mathbb{R}$ and $D \subseteq \mathbb{R}$ is a subset of $\mathbb{R}$. In order for the initial value problem (1) to be well-posed, i.e., for the problem to have a unique solution in a certain space of functions, we need to impose some mild restrictions on $f(y,t)$. As we will see, it is sufficient for f to be continuous in time and Lipschitz continuous in the domain D.

Definition 1. Let $D \subseteq \mathbb{R}$ be a subset of $\mathbb{R}$. We say that $f : D \times [0,T] \to \mathbb{R}$ is Lipschitz continuous in D if there exists a positive constant $0 \leq L < \infty$ (Lipschitz constant) such that

$$|f(y_1,t) - f(y_2,t)| \leq L\,|y_1 - y_2| \qquad \text{for all } t \in [0,T]. \tag{2}$$

The smallest number L^* such that the inequality above is satisfied is called "best" Lipschitz constant.

Lipschitz continuity is stronger than just continuity, which requires only that[1]

$$\lim_{y_1 \to y_2^{\pm}} |f(y_1,t) - f(y_2,t)| = 0 \quad \text{for all } t \in [0,T] \text{ and for all } y_2 \text{ in the interior of } D. \tag{3}$$

Indeed, Lipschitz continuity implies that the rate at which $f(y_1,t)$ approaches $f(y_2,t)$ cannot be larger than L for all y_1 and y_2 in D. In other words, a Lipschitz continuous function $f(y,t)$ has a growth rate that is bounded by L for all y_1 and y_2 in D.

Example: Let $D = [-1,1]$ be a closed interval, i.e., an interval including the endpoints -1 and 1. The function $f(y,t) = e^{-t^2} y^{1/3}$ is continuous in D for all $t \in \mathbb{R}$ (see Figure 1). However, $f(y,t)$ is not Lipschitz continuous in D. The problem here is that $f(y,t)$ has infinite "slope" at the point $y = 0$ for all $t \in \mathbb{R}$. In other words, there is no constant $0 \leq L < \infty$ such that

$$|f(y,t) - f(0,t)| \leq L\,|y - 0| \quad \text{for all} \quad y \in D. \tag{4}$$

This can be seen by substituting $f(y,t) = e^{-t^2} y^{1/3}$ in (4)

$$|f(y,t)| \leq L\,|y| \quad \Rightarrow \quad e^{-t^2} \left| \frac{y^{1/3}}{y} \right| = e^{-t^2} \left| \frac{1}{y^{2/3}} \right| \leq L \quad \text{for all} \quad y \in D. \tag{5}$$

Clearly, if we send y to zero we have that L goes to infinity, and therefore $f(y,t)$ is not Lipschitz continuous in D. Note that $f(y,t)$ is Lipschitz continuous (actually infinitely differentiable with continuous derivatives), e.g., in

$$D = [-1,1] \setminus \{0\} = [-1,0[\cup]0,1] \quad \text{or in} \quad D = [1,10]. \tag{6}$$

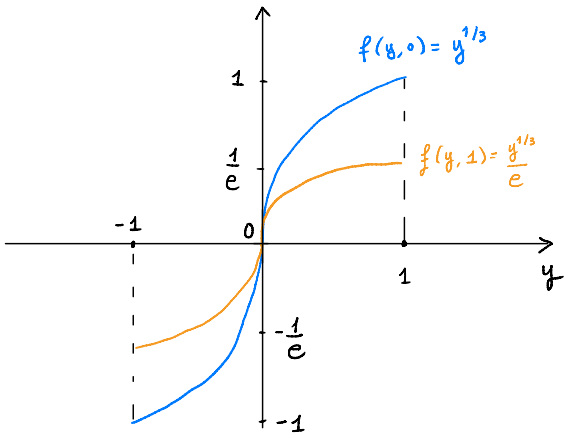


Figure 1: Sketch of $f(y,t) = e^{-t^2} y^{1/3}$ in $[-1,1]$ at $t = 0$ and $t = 1$. The function has infinite slope at $y = 0$.

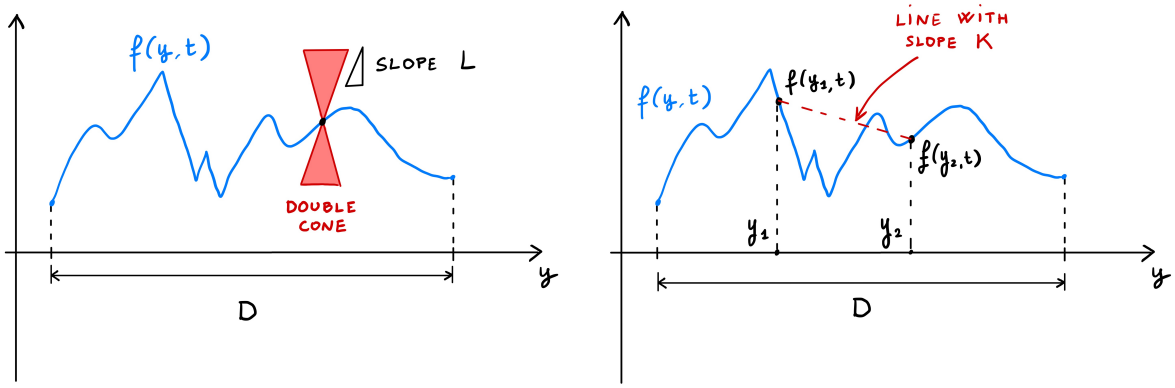


Figure 2: Geometric meaning of Lipschitz continuity.

The Lipschitz continuity condition (2) has a nice geometric interpretation. In practice it says that the function $f(y,t)$ can never enter a double cone with slope L and vertex on any point $(y, f(y,t)))$ where $y \in D$. In other words, if we can slide the vertex of the double cone over the (continuous) function $f(y,t)$ for $y \in D$ and the function never enters the cone then $f(y,t)$ is Lipschitz continuous in D. To explain this, let us divide the inequality (2) by $|y_1 - y_2|$ (for $y_1 \neq y_2$). This yields

$$\underbrace{\left|\frac{f(y_1,t) - f(y_2,t)}{y_1 - y_2}\right|}_{|K|} \leq L \quad \text{for all} \quad y_1, y_2 \in D. \tag{7}$$

For each fixed y_1 and y_2 in D we see that K represents the slope of the line connecting the points $(y_1, f(y_1,t))$ and $(y_2, f(y_2,t))$ (see Figure 2). Clearly, the best Lipshitz constant is obtained as

$$L^* = \max_{y_1, y_2 \in D} \left|\frac{f(y_1,t) - f(y_2,t)}{y_1 - y_2}\right|. \tag{8}$$

[1]The notation $y_1 \to y_2^{\pm}$ means that y_1 is approaching y_2 either from the left ("$-$") or from the right ("$+$"). Note that we can equivalently write (3) as

$$\lim_{y_1 \to y_2^+} f(y_1,t) = \lim_{y_1 \to y_2^-} f(y_1,t) = f(y_2,t).$$

Any $L \geq L^*$ is still a Lipschitz constant. If the function $f(y,t)$ is continuously differentiable in $y \in D$ and D is compact then

$$L^* = \max_{y \in D} \left| \frac{\partial f(y,t)}{\partial y} \right| < \infty. \tag{9}$$

Lemma 1. If $f(y,t)$ is of class C^1 in a compact subset $D \subseteq \mathbb{R}$ for all $t \in [0,T]$ then $f(y,t)$ is Lipschitz continuous in D.

Proof. By assumption the derivative of $\partial f(y,t)/\partial y$ is continuous on the compact domain $D \subseteq \mathbb{R}$. This implies that the minimum and the maximum of $\partial f(y,t)/\partial y$ is attained at some points in D. By using the mean value theorem we immediately see that

$$|f(y_1,t) - f(y_2,t)| = \left| \frac{\partial f(y^*,t)}{\partial y} \right| |y_1 - y_2| \,. \tag{10}$$

where y^* is some point within the interval $[y_1, y_2]$. The point y^* depends on f, y_1 and y_2. The right hand side of (10) can be bounded as

$$|f(y_1,t) - f(y_2,t)| \leq \underbrace{\max_{y \in D} \left| \frac{\partial f(y,t)}{\partial y} \right|}_{L^*} |y_1 - y_2| \qquad \text{for all} \quad y_1, y_2 \in D. \tag{11}$$

□

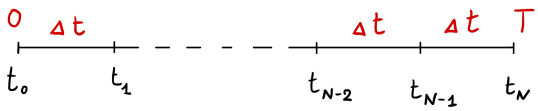
Example: The function $f(y) = y^2$ is of class C^∞ (infinitely differentiable with continuous derivative) in any bounded subset of $\mathbb{R}$. The function is not Lipshitz continuous at $y = \pm\infty$, since the slope of the first-order derivative $f'(y) = 2y$ grows unboundedly as $y \to \pm\infty$.

Remark: The initial value problem (1) can be equivalently written as

$$y(t) = y_0 + \int_0^T \frac{dy(s)}{ds} ds = y_0 + \int_0^t f(y(s), s) ds \tag{12}$$

i.e., as an integral equation for $y(s)$. This formulation is quite convenient for developing numerical methods for ODEs based on *numerical quadrature formulas*, i.e., numerical approximations of the temporal integral appearing at the right hand side of (12). For example, consider a discretization of the time interval $[0,T]$ in terms of $N+1$ evenly-spaced time instants

$$t_i = i\Delta t \qquad i = 0, 1, \ldots, N \qquad \text{where} \quad \Delta t = \frac{T}{N}. \tag{13}$$



By applying (12) within each time interval $[t_i, t_{i+1}]$ we obtain

$$y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} f(y(s), s) ds. \tag{14}$$

At this point we can approximate the integral at the right hand side if (14), e.g., by using the simple rectangle rule (see Figure 3)

$$\int_{t_i}^{t_{i+1}} f(y(s), s) ds \simeq \Delta t f(y(t_i), t_i) \tag{15}$$

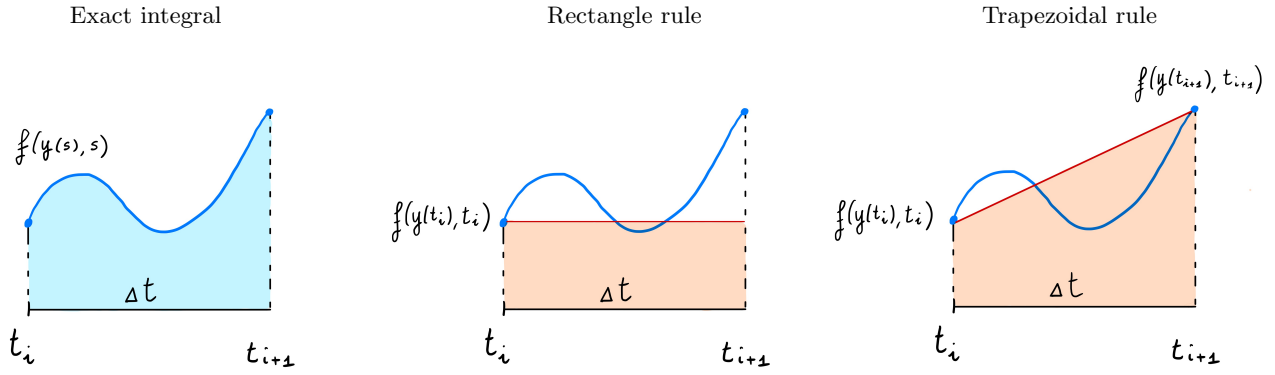


Figure 3: Approximations of the integral $\int_{t_i}^{t_{i+1}} f(y(s),s)ds$ in (14) leading to well-known numerical schemes: Euler forward (rectangle rule), Crank-Nicolson (trapezoidal rule)

This yields the *Euler forward scheme*

$$u_{i+1} = u_i + \Delta t f(u_i, t_i), \tag{16}$$

where u_i is an approximation of $y(t_i)$. The Euler forward scheme is an explicit one-step scheme. The adjective "explicit" emphasizes the fact that u_{i+1} can be computed explicitly based on the knowledge of f and u_i using (16). On the other hand, if we approximate the integral at the right hand side of (12) with the trapezoidal rule

$$\int_{t_i}^{t_{i+1}} f(y(s),s)ds \simeq \frac{\Delta t}{2}\left[f(y(t_{i+1}),t_{i+1}) + f(y(t_i),t_i)\right] \tag{17}$$

we obtain the *Crank-Nicolson scheme*

$$u_{i+1} = u_i + \frac{\Delta t}{2}\left[f(u_i,t_i) + f(u_{i+1},t_i)\right]. \tag{18}$$

The Crank-Nicolson scheme is "implicit" because the approximate solution at time t_{i+1}, i.e., u_{i+1}, cannot be computed explicitly based on u_i, but requires the solution of a nonlinear equation. Such a solution can be computed numerically by using any method to solve nonlinear equations. These methods are usually iterative, e.g., the bisection method, or the Newton method if f is continuously differentiable. Iterative methods for nonlinear equations can be formulated as fixed point iteration problems. In the specific case of (18) we have

$$u_{i+1} = G(u_{i+1}) \qquad \text{where} \qquad G(u_{i+1}) = u_i + \frac{\Delta t}{2}\left[f(u_i,t_i) + f(u_{i+1},t_i)\right]. \tag{19}$$

If Δt is small then u_i is close to u_{i+1}. Moreover, if Δt is sufficiently small we have that the Lipschitz constant of G is smaller than 1, which implies that the fixed point iterations will convergence globally to a unique solution u_{i+1} (see, e.g., [3, Ch. 6]).

Next, we formulate a well-known result for existence and uniqueness of the solution to the Cauchy problem for one ODE.

Theorem 1 (Well-posedness of the initial value problem for one ODE)**.** Let $D \subset \mathbb{R}$ be an open set, $y_0 \in D$. If $f : D \times [0,T] \to \mathbb{R}$ is Lipschitz continuous in D and continuous in $[0,T]$ then there exists a unique solution to the initial value problem (1) within the time interval $[0,\tau[$, where τ is the instant at which $y(t)$ exists the domain D. The solution $y(t)$ is continuously differentiable in $[0,\tau[$.

Clearly, if $f(y,t)$ is Lipschitz continuous in $y \in \mathbb{R}$ and continuous in t then the solution to the initial value problem (1) is *global* in the sense that it exists and is unique for all $t \geq 0$. This can be seen by noting that $y(t)$ never exits the domain in which $f(y,t)$ is Lipschitz continuous.

Hereafter we provide a simple example of an initial value problem that blows-up in a finite time, and an initial value problem that is not well posed.

- *Finite-time blow-up*: Consider the initial value problem

$$\frac{dy}{dt} = y^2 \qquad y(0) = 1. \tag{20}$$

  We know that $f(y) = y^2$ is non-Lipschitz at infinity. By using separation of variables it is straightforward to show that the solution to (20) is

$$y(t) = \frac{1}{1-t}. \tag{21}$$

  The function $y(t)$ clearly blows up to infinity as t approaches one (from the left).

- *Non-uniqueness of solutions:* Consider the initial value problem

$$\frac{dy}{dt} = y^{1/3} \qquad y(0) = 0. \tag{22}$$

  We have seen that $f(y) = y^{1/3}$ is not Lipshitz in any compact domain D including the point $y = 0$. In this case we are setting the initial condition exactly at the point in which the slope of $f(y)$ is infinity. By using separation of variables it can be shown that a solution to (22) is

$$y(t) = \left(\frac{2}{3}t\right)^{3/2}. \tag{23}$$

  However, as easily seen, the functions

$$y(t) = \begin{cases} 0 & \text{for } 0 \leq t < c \\ \pm\left(\frac{2}{3}(t-c)\right)^{3/2} & \text{for } t \geq c \end{cases} \tag{24}$$

  are also solutions to (22) for every $c \geq 0$.

Theorem 2 (Dependency of the ODE solution on the initial condition y_0)**.** Let $D \subset \mathbb{R}$ be an open set, $y_0 \in D$. If $f : D \times [0,T] \to \mathbb{R}$ is Lipschitz continuous in D and continuous in $[0,T]$ then the solution to (1) $y(t; y_0)$ (i.e., the flow generated by the ODE) is continuous in y_0. Moreover, if $f(y,t)$ is of class C^k in D (continuously differentiable k-times in D) then $y(t; y_0)$ is of class C^k in D.

Remark: By applying Theorem 1 iteratively (in the sense that we restart the the system from a new initial condition) we conclude that f can also be piece-wise continuous in time. This case is studied quite extensively in control of ODEs where a piecewise constant function in time is used as a control to minimize or maximize some performance metric. In this case the solution to (1) is continuous in time and piecewise diffentiable in time. The non-differentiability is at the times where the right hand side is not continuous (in t). And example of an ODE with piecewise constant control $v(t)$ is

$$\frac{dy}{dt} = g(y,t) + \underbrace{v(t)}_{\text{control}}, \qquad y(0) = y_0. \tag{25}$$

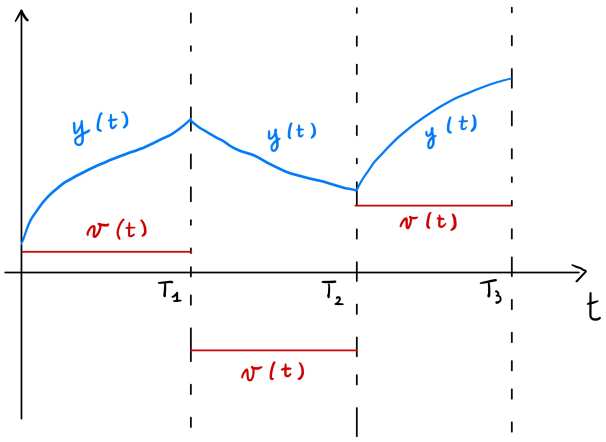


Figure 4: Piecewise differentiability of the solution in case the control $v(t)$ in equation (25) is piecewise continuous in time.

The control $v(t)$ can be computed, e.g., by solving the optimization problem

$$\min_{v(t)\in S} |y(T) - y^*|^2 \qquad \text{subject to (25)}, \tag{26}$$

where S is some function space, e.g., the space of piecewise continuous functions in $[0, T]$. Clearly, $y(T)$ depends on the whole time history of the function $v(t)$. Such functional dependence is often denoted as $y(t, [v(t)])$.

The initial value problem for systems of ODEs. Consider the following systems of nonlinear ODEs

$$\begin{cases} \dfrac{d\boldsymbol{y}}{dt} = \boldsymbol{f}(\boldsymbol{y}, t) \\ \boldsymbol{y}(0) = \boldsymbol{y}_0 \end{cases} \tag{27}$$

where $\boldsymbol{y}(t) = [y_1(t) \cdots y_n(t)]^T$ is a vector of phase variables, $\boldsymbol{f} : D \times [0, T] \to \mathbb{R}^n$, and D is a subset of $\mathbb{R}^n$. In an extended notation the system of ODEs (27) is written as

$$\begin{cases} \dfrac{dy_1}{dt} = f_1(y_1, \ldots, y_n, t) \\ \dfrac{dy_2}{dt} = f_2(y_1, \ldots, y_n, t) \\ \vdots \\ \dfrac{dy_n}{dt} = f_n(y_1, \ldots, y_n, t) \\ y_1(0) = y_{10} \\ y_2(0) = y_{20} \\ \vdots \\ y_n(0) = y_{n0} \end{cases} \tag{28}$$

Systems of ODE such as (1) or (28) arise, e.g., when modeling physical systems (e.g., pendulum equations, UAV models, etc.) or when performing a discretization of a partial differential equation to remove

dependence on spatial variables. Let us provide a simple example of a particular type of such a discretization.

Example: Consider the following initial-boundary value problem for the heat equation

$$\begin{cases} \dfrac{\partial y(t,x)}{\partial t} = \alpha \dfrac{\partial^2 y(x,t)}{\partial x^2} & \text{diffusion equation} \\ y(0,x) = y_0(x) & \text{initial condition} \\ y(t,0) = y(t,2\pi) & \text{periodic boundary conditions} \end{cases} \tag{29}$$

Since this problem is defined on a periodic domain, i.e., on the circle $\mathbb{T}$, we can use a *Fourier spectral method* to discretize it in space. To this end, consider the truncated Fourier series expansion[2]

$$y_N(t,x) = \sum_{k=-N}^{N} c_k(t) e^{ikx}, \tag{32}$$

where $c_k(t)$ are time dependent functions with values in $\mathbb{C}$. The series (32) automatically satisfies the periodic boundary conditions of the problem. A substitution of (32) into (29) yields,

$$\frac{\partial y_N(t,x)}{\partial t} = \alpha \frac{\partial^2 y_N(x,t)}{\partial x^2} + \underbrace{R_N(x,t)}_{\text{residual}} \tag{33}$$

i.e.,

$$\sum_{k=-N}^{N} \frac{dc_k(t)}{dt} e^{ikx} = -\alpha \sum_{k=-N}^{N} k^2 c_k(t) e^{ikx} + R_N(x,t) \tag{34}$$

At this point we impose that the residual PDE $R_N(x,t)$ is orthogonal o the span of the basis $B_N = \{e^{ikx}\}_{k=-N}^{N}$ in the sense of the standard inner product

$$(u,v)_{L^2([0,2\pi])} = \int_0^{2\pi} u(x)v(x)dx, \tag{35}$$

i.e.,

$$\left(R_N(x,t), e^{-ijx}\right)_{L^2([0,2\pi])} = 0 \qquad j = -N, ..., N. \tag{36}$$

This is called Fourier-Galerkin method [2, p.43], and yields a linear systems of $2N+1$ ODEs for the Fourier coefficients c_k

$$\frac{dc_k(t)}{dt} = -\alpha k^2 c_k(t), \qquad c_k(0) = \frac{1}{2\pi} \int_0^{2\pi} y_0(x) e^{-ikx} dx \qquad k = -N, ..., N. \tag{37}$$

Note that this system can be solved analytically. The solution is as

$$c_k(t) = \frac{e^{-\alpha k^2 t}}{2\pi} \int_0^{2\pi} y_0(x) e^{-ikx} dx \tag{38}$$

[2]The convergence rate of the Fourier series (32) to $y(x,t)$ depends on the smoothness of $y(x,t)$ in $x \in [0, 2\pi]$. Specifically, it can be shown that if $y(x,t) \in H^q([0,2\pi])$ (Sobolev space with degree q) for all t then [2, p. 35]

$$\|y(x,t) - y_N(x,t)\|^2_{L^2([0,2\pi])} \leq C N^{-q} \left\| \frac{d^q y}{dx^q} \right\|^2_{L^2([0,2\pi])}. \tag{30}$$

This type of convergence is called *spectral converge*. Moreover, if $y(x,t)$ is analytic in x for all t then it can be shown that

$$\|y(x,t) - y_N(x,t)\|^2_{L^2([0,2\pi])} \leq Q e^{-cN} \|y\|^2_{L^2([0,2\pi])}, \tag{31}$$

i.e., convergence is *exponential* [2, p. 36].

which allows yields the approximate solution

$$y_N(x,t) = \frac{1}{2\pi}\sum_{k=-N}^{N} e^{ikx-\alpha k^2 t}\int_0^{2\pi} y_0(x)e^{-ikx}dx. \tag{39}$$

As we will see, the IBVP (29) can be discretized in space by using many other techniques including finite-difference methods, pseudo-spectral collocation methods, finite-elements methods, etc. The proper way to formulate these methods often goes through the so-called weak (or variational) form of the PDE. AM 213B focuses mostly on finite-difference approximation methods of PDEs. For example, a central finite-difference approximation of the PDE (29) yields the ODE system

$$\frac{du(x_k,t)}{dt} = \frac{\alpha}{\Delta x^2}\left(u(x_{k+1},t) - 2u(x_k,t) + u(x_{k-1},t)\right) \qquad u(x_{N+j},t) = u(x_j,t) \tag{40}$$

where

$$x_k = k\Delta x \qquad k = 0,\ldots,N, \qquad \Delta x = \frac{2\pi}{(N+1)} \quad \text{(uniform grid spacing)}. \tag{41}$$

Clearly, this system can be written in the form (28) provided we define

$$y_k(t) = u(x_k,t) \qquad f_k(y_1,\ldots,y_n) = \frac{\alpha}{\Delta x^2}\left(y_{k+1} - 2y_k + y_{k-1}\right) \tag{42}$$

As before, we can re-write the Cauchy problem as an integral equation

$$\boldsymbol{y}(t) = \boldsymbol{y}(0) + \int_0^t \boldsymbol{f}(\boldsymbol{y}(s),s)ds, \tag{43}$$

which is very handy to derive numerical methods based on numerical quadrature of the one-dimensional integral at the right hand side. For instance, consider a partition of the $[0,T]$ into an evenly spaced grid points such that $t_{i+1} = t_i + \Delta t$, and write (43) within each time interval

$$\boldsymbol{y}(t_{i+1}) = \boldsymbol{y}(t_i) + \int_{t_i}^{t_{i+1}} \boldsymbol{f}(\boldsymbol{y}(s),s)ds. \tag{44}$$

By approximating the integral at the right hand side of (44), e.g., using the midpoint rule yields

$$\int_{t_i}^{t_{i+1}} \boldsymbol{f}(\boldsymbol{y}(s),s)ds \simeq \Delta t \boldsymbol{f}\left(\boldsymbol{y}\left(t_i + \frac{\Delta t}{2}\right), t_i + \frac{\Delta t}{2}\right) \tag{45}$$

At this point, we can approximate $\boldsymbol{y}(t_i + \Delta t/2)$ using the Euler forward method

$$\boldsymbol{y}\left(t_i + \frac{\Delta t}{2}\right) \simeq \boldsymbol{y}(t_i) + \frac{\Delta t}{2}\boldsymbol{f}(\boldsymbol{y}(t_i),t_i) \tag{46}$$

to obtain the *explicit midpoint method*

$$\boldsymbol{u}_{i+1} = \boldsymbol{u}_i + \Delta t \boldsymbol{f}\left(\boldsymbol{u}_i + \frac{\Delta t}{2}\boldsymbol{f}(\boldsymbol{u}_i,t_i), t_i + \frac{\Delta t}{2}\right) \tag{47}$$

where $\boldsymbol{u}_i$ is an approximation of $\boldsymbol{f}(t_i)$. The explicit midpoint method is a one-step method that belongs to the class of Runge-Kutta methods[3]. The integral formulation (43) is also at the basis of the Picard iteration method which is used to prove the following theorem.

[3] As we will see, the explicit midpoint method (47) is a two-stage explicit Runge-Kutta method.

Theorem 3 (Well-posedness of initial value problems for systems of ODEs)**.** Let $D \subset \mathbb{R}^n$ be an open set, $\boldsymbol{y}_0 \in D$. If $\boldsymbol{f} : D \times [0,T] \to \mathbb{R}$ is Lipschitz continuous in D and continuous in $[0,T]$ then there exists a unique solution to the initial value problem (27) within the time interval $[0,\tau[$, where τ is defined to be the instant at which $\boldsymbol{y}(t)$ exits the domain D in which $\boldsymbol{f}$ is Lipschitz continuous. The solution $\boldsymbol{y}(t)$ is continuously differentiable in $[0,\tau[$.

How do we define Lipschitz continuity for a vector-valued function $\boldsymbol{f}(\boldsymbol{y},t)$ defined in subset of $\mathbb{R}^d$? By a simple generalization of the definition we gave for one-dimensional functions.

Definition 2. Let D be a subset of $\mathbb{R}^n$, $\boldsymbol{f} : D \times [0,T] \to \mathbb{R}^n$. We say that $\boldsymbol{f}$ is Lipschitz continuous in D if there exists a constant $0 \leq L < \infty$ such that

$$\|\boldsymbol{f}(\boldsymbol{y}_1,t) - \boldsymbol{f}(\boldsymbol{y}_2,t)\| \leq L \|\boldsymbol{y}_1 - \boldsymbol{y}_2\| \quad \text{for all} \quad \boldsymbol{y}_1, \boldsymbol{y}_2 \in D, \tag{48}$$

where $\|\cdot\|$ is any norm defined in $\mathbb{R}^n$.

Remark: As is well known, all norms defined in a finite-dimensional vector space (such as $\mathbb{R}^n$) are *equivalent*. This means that if we pick two arbitrary norms in $\mathbb{R}^n$, say $\|\cdot\|_a$ and $\|\cdot\|_b$, then there exist two numbers c_1 and c_2 such that

$$c_1 \|\boldsymbol{y}\|_a \leq \|\boldsymbol{y}\|_b \leq c_2 \|\boldsymbol{y}\|_a \quad \text{for all} \quad \boldsymbol{y} \in \mathbb{R}^n. \tag{49}$$

The most common norms in $\mathbb{R}^n$ are

$$\|\boldsymbol{y}\|_\infty = \max_{k=1,..,n} |y_k| , \tag{50}$$

$$\|\boldsymbol{y}\|_1 = \sum_{k=1}^{n} |y_k| , \tag{51}$$

$$\|\boldsymbol{y}\|_2 = \left(\sum_{k=1}^{n} |y_k|^2 \right)^{1/2} , \tag{52}$$

$$\vdots \tag{53}$$

$$\|\boldsymbol{y}\|_p = \left(\sum_{k=1}^{n} |y_k|^p \right)^{1/p} \qquad p \in \mathbb{N} \setminus \{\infty\}. \tag{54}$$

Based on these definitions it is easy to show, e.g., that

$$\|\boldsymbol{y}\|_\infty \leq \|\boldsymbol{y}\|_1 \leq n \|\boldsymbol{y}\|_\infty , \tag{55}$$

$$\|\boldsymbol{y}\|_2 \leq \|\boldsymbol{y}\|_1 \leq \sqrt{n} \|\boldsymbol{y}\|_2 , \tag{56}$$

$$\|\boldsymbol{y}\|_\infty \leq \|\boldsymbol{y}\|_2 \leq \sqrt{n} \|\boldsymbol{y}\|_\infty . \tag{57}$$

Therefore if the $\boldsymbol{f}(\boldsymbol{y},t)$ is Lipschitz continuous in D with respect to the 1-norm, i.e.,

$$\|\boldsymbol{f}(\boldsymbol{y}_1,t) - \boldsymbol{f}(\boldsymbol{y}_2,t)\|_1 \leq L_1 \|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_1 \quad \text{for all} \quad \boldsymbol{y}_1, \boldsymbol{y}_2 \in D, \quad \text{for all} \quad t \geq 0 \tag{58}$$

then it is also Lipschitz continuous with respect to the uniform norm. In fact, by using (55) we have

$$\|\boldsymbol{f}(\boldsymbol{y}_1,t) - \boldsymbol{f}(\boldsymbol{y}_2,t)\|_\infty \leq \underbrace{L_1 n}_{L_\infty} \|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_\infty . \tag{59}$$

Of course, $\boldsymbol{f}(\boldsymbol{y},t)$ is also Lipschitz continuous with respect to the 2-norm.

Theorem 4. If $\boldsymbol{f}(\boldsymbol{y},t)$ is of class C^1 in a compact convex domain $D \subset \mathbb{R}^n$, then $\boldsymbol{f}(\boldsymbol{y},t)$ is Lipschitz continuous in D.

Proof. Let $D \subseteq \mathbb{R}^n$ be a compact convex domain and let

$$M = \max_{\boldsymbol{y}\in D} \left| \frac{\partial f_j(\boldsymbol{y},t)}{\partial y_i} \right|. \tag{60}$$

Clearly M exists and is finite because we assumed that D is compact and that $\boldsymbol{f}$ is of class C^1 in D[4]. Consider two points $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ in D, and the line that connects $\boldsymbol{y}_1$ to $\boldsymbol{y}_2$, i.e.,

$$\boldsymbol{z}(s) = (1-s)\boldsymbol{y}_1 + s\boldsymbol{y}_2 \qquad s \in [0,1]. \tag{61}$$

Since D is convex, we have that the line $\boldsymbol{z}(s)$ lies entirely within D. Therefore we can use the mean value theorem applied to the one-dimensional function $f_i(\boldsymbol{z}(s),t)$ ($s \in [0,1]$) to obtain

$$f_i(\boldsymbol{y}_2,t) - f_i(\boldsymbol{y}_1,t) = \nabla f_i(\boldsymbol{z}(s^*),t)\cdot(\boldsymbol{y}_2-\boldsymbol{y}_1) \qquad \text{for some } s^* \in [0,1]. \tag{62}$$

By taking the absolute value and using the Cauchy-Schwartz inequality we obtain

$$\begin{aligned} |f_i(\boldsymbol{y}_2,t) - f_i(\boldsymbol{y}_1,t)|^2 =& \left| \sum_{j=1}^n \frac{\partial f_i(\boldsymbol{z}(s^*))}{\partial y_j}(y_{2j}-y_{1j}) \right|^2 \\ \leq& \left| \sum_{j=1}^n \frac{\partial f_i(\boldsymbol{z}(s^*),t)}{\partial y_j} \right|^2 \left| \sum_{j=1}^n (y_{2j}-y_{1j}) \right|^2 \\ \leq& nM^2 \left\| \boldsymbol{y}_2 - \boldsymbol{y}_1 \right\|_2^2. \end{aligned} \tag{63}$$

This implies that

$$\left\| \boldsymbol{f}(\boldsymbol{y}_2,t) - \boldsymbol{f}(\boldsymbol{y}_1,t) \right\|_2 \leq \underbrace{nM}_{L_2} \left\| \boldsymbol{y}_2 - \boldsymbol{y}_1 \right\|_2. \tag{64}$$

i.e., $\boldsymbol{f}(\boldsymbol{y},t)$ is Lipschitz continuous in the 2-norm, or any other norm that is equivalent to the 2-norm. In particular, by using the inequalities (55)-(57) we have that $\boldsymbol{f}(\boldsymbol{y},t)$ is Lipschitz continuous relative to the 1-norm and the uniform norm (∞-norm).

□

Lemma 2. If $\boldsymbol{f}(\boldsymbol{y},t)$ is of class C^1 in $D \subseteq \mathbb{R}^n$ and has bounded derivatives $\partial f_i/\partial y_j$ then $\boldsymbol{f}(\boldsymbol{y},t)$ is Lipschitz continuous in D.

Linear systems of ODEs. Consider the following autonomous system of linear differential equations

$$\frac{d\boldsymbol{y}(t)}{dt} = \boldsymbol{A}\boldsymbol{y}(t) \qquad \boldsymbol{y}(0) = \boldsymbol{y}_0. \tag{65}$$

We have seen in AM214 that this system admits a global solution, i.e., the solution exists and is unique for all $t \geq 0$. Such an analytic solution can be expressed in terms of generalized eigenvectors of $\boldsymbol{A}$ as

$$\begin{aligned} \boldsymbol{y}(t) =& e^{t\boldsymbol{A}}\boldsymbol{y}_0 \\ =& \boldsymbol{P}e^{t\boldsymbol{J}}\boldsymbol{P}^{-1}\boldsymbol{y}_0, \end{aligned} \tag{66}$$

[4] A compact domain is by definition bounded and closed. The minimum and maximum of a continuous function in defined on a compact domain is attained at some points within the domain or on its boundary. Note that this is not true if the domain is not compact. For example, the function $f(y) = 1/y$ is continuously differentiable on $]0,1]$ (bounded domain by not compact), but the function is unbounded on $]0,1]$.

where $\boldsymbol{P}$ is a matrix that has the generalized eigenvectors of $\boldsymbol{A}$ as columns, and $\boldsymbol{J}$ is the Jordan form of $\boldsymbol{A}$. While the formula (66) is nice and compact, its computation requires the knowledge of the eigenvalues and and generalized eigenvectors of $\boldsymbol{A}$ which is something that is not easy to compute, especially in high-dimensions[5]. Moreover, the matrix $\boldsymbol{A}$ can be time-dependent (i.e., $\boldsymbol{A}(t)$), in which case the matrix exponential $e^{t\boldsymbol{A}}$ has to be replaced by a Magnus series (see, e.g., [1]).

Matrix norms compatible with vector norms Let us define the following matrix norm

$$\|\boldsymbol{A}\| = \sup_{\boldsymbol{y}\neq\boldsymbol{0}_{\mathbb{R}^n}} \frac{\|\boldsymbol{A}\boldsymbol{y}\|}{\|\boldsymbol{y}\|} = \sup_{\|\boldsymbol{y}\|=1} \|\boldsymbol{A}\boldsymbol{y}\|\,. \tag{67}$$

Clearly, $\|\boldsymbol{A}\|$ is matrix norm (prove it as exercise), which satisfies, by definition, the following inequality

$$\|\boldsymbol{A}\| \geq \frac{\|\boldsymbol{A}\boldsymbol{y}\|}{\|\boldsymbol{y}\|} \qquad \text{i.e.} \qquad \|\boldsymbol{A}\boldsymbol{y}\| \leq \|\boldsymbol{A}\|\,\|\boldsymbol{y}\|\,. \tag{68}$$

It is straightforward to show that

$$\|\boldsymbol{A}\|_\infty = \max_{i=1,..,n} \left(\sum_{j=1}^n |A_{ij}| \right), \tag{69}$$

$$\|\boldsymbol{A}\|_1 = \max_{j=1,..,n} \left(\sum_{i=1}^n |A_{ij}| \right), \tag{70}$$

$$\|\boldsymbol{A}\|_2 = \sqrt{\lambda_{\max}\left(\boldsymbol{A}^T\boldsymbol{A}\right)} = \sigma_{\max}(\boldsymbol{A}), \tag{71}$$

where $\sigma_{\max}(\boldsymbol{A})$ is the largest singular value of the matrix $\boldsymbol{A}$. For example,

$$\|\boldsymbol{A}\boldsymbol{y}\|_\infty = \max_{i=1,\ldots,n} \left| \sum_{j=1}^n A_{ij} y_j \right| \leq \max_{i=1,\ldots,n} \left(\sum_{j=1}^n |A_{ij}|\,|y_j| \right) \leq \|\boldsymbol{y}\|_\infty \max_{i=1,\ldots,n} \left(\sum_{j=1}^n |A_{ij}| \right) \tag{72}$$

which implies that

$$\frac{\|\boldsymbol{A}\boldsymbol{y}\|_\infty}{\|\boldsymbol{y}\|_\infty} \leq \max_{i=1,\ldots,n} \left(\sum_{j=1}^n |A_{ij}| \right) \qquad \text{for all } \boldsymbol{y} \neq \boldsymbol{0}_{\mathbb{R}^n}, \tag{73}$$

i.e.,

$$\sup_{\boldsymbol{y}\neq\boldsymbol{0}_{\mathbb{R}^n}} \frac{\|\boldsymbol{A}\boldsymbol{y}\|_\infty}{\|\boldsymbol{y}\|_\infty} = \max_{i=1,\ldots,n} \left(\sum_{j=1}^n |A_{ij}| \right) = \|\boldsymbol{A}\|_\infty\,. \tag{74}$$

With any compatible matrix norm available we immediately see that the function $\boldsymbol{f}(\boldsymbol{y}) = \boldsymbol{A}\boldsymbol{y}$ is Lipschitz continuous in $\mathbb{R}^n$. In fact, we have

$$\|\boldsymbol{A}\boldsymbol{y}_1 - \boldsymbol{A}\boldsymbol{y}_2\| \leq \|\boldsymbol{A}\|\,\|\boldsymbol{y}_1 - \boldsymbol{y}_2\| \qquad \text{for all} \qquad \boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{R}^n, \tag{75}$$

where $L = \|\boldsymbol{A}\|$ is the Lipschitz constant. Equation (75) implies that the solution to (65) is global in time. This can be also shown by noticing that $\boldsymbol{A}$ is the Jacobian matrix of $\boldsymbol{f}(\boldsymbol{y}, t)$ and that all entries of such a matrix are of course bounded in $\mathbb{R}^n$ (see Lemma 2).

[5] If the matrix $\boldsymbol{A}$ has a particular structure, e.g., if $\boldsymbol{A}$ is a tridiagonal differentiation matrix (Toeplitz matrix), then there are formulas available for the eigenvalues and the eigenvectors of $\boldsymbol{A}$.

The following result on the regularity of the flow generated by the initial value problem (27) holds true.

Theorem 5 (Dependency of the ODE solution on the initial condition $\boldsymbol{y}_0$)**.** Let $D \subset \mathbb{R}^n$ be an open set, $\boldsymbol{y} \in D$. If $\boldsymbol{f} : D \times [0,T] \to \mathbb{R}^n$ is Lipschitz continuous in D and continuous in $[0,T]$ then the solution $\boldsymbol{y}(t;\boldsymbol{y}_0)$ to the initial value problem (27), i.e., flow generated by the ODE system, is continuous in $\boldsymbol{y}_0$. Moreover, if $\boldsymbol{f}(\boldsymbol{y},t)$ is of class C^k (continuously differentiable k-times in D) in D then $\boldsymbol{y}(t;\boldsymbol{y}_0)$ is of class C^k in D relative to $\boldsymbol{y}_0$.

References

[1] S. Blanes, F. Casas, J. A. Oteo, and J. Ros. The Magnus expansion and some of its applications. *Physics Reports*, 470:151–238, 2009.

[2] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb. *Spectral methods for time-dependent problems*, volume 21 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2007.

[3] A. Quarteroni, R. Sacco, and F. Salieri. *Numerical mathematics*. Springer, 2007.

Overview of numerical methods for ODEs

In this lecture we provide a brief overview of the most common numerical methods to approximate the solution of an initial value problem for systems of ODEs of the form

$$\begin{cases} \dfrac{d\boldsymbol{y}}{dt} = \boldsymbol{f}(\boldsymbol{y}, t) \\ \boldsymbol{y}(0) = \boldsymbol{y}_0 \end{cases} \tag{1}$$

where $\boldsymbol{y}(t) = [y_1(t) \cdots y_n(t)]^T$ is a (column) vector of phase variables, $\boldsymbol{f} : D \times [0, T] \to \mathbb{R}^n$, D is a subset of $\mathbb{R}^n$, and T is the integration period. Most of the material presented in this lecture can be found, e.g., in [2, 3, 4]. We assume that the initial value problem (1) well-posed, i.e., that it has a unique solution[1] (at least for some time $t\tau > 0$). We have seen that this is equivalent to assume that $\boldsymbol{f}(\boldsymbol{y}, t)$ is at least Lipschitz continuous in D and that $\boldsymbol{y}_0 \in D$.

The initial value problem (1) can be equivalently written as

$$\boldsymbol{y}(t) = \boldsymbol{y}(0) + \int_0^t \boldsymbol{f}(\boldsymbol{y}(s), s) ds. \tag{2}$$

i.e., as an integral equation for $\boldsymbol{y}(t)$.

Picard iteration method. The Picard iteration method is rarely used in practice to compute numerical solutions to ODEs, but rather to prove existence and uniqueness of solutions to ODEs. Picard's method is essentially a fixed point iteration in which the solution $\boldsymbol{y}(t)$ is approximated within a fixed time interval $[0, t]$ by a *sequence of functions*

$$\boldsymbol{y}^{[0]}(t) \to \boldsymbol{y}^{[1]}(t) \to \cdots \to \boldsymbol{y}^{[k]}(t) \to \cdots \tag{3}$$

generated by the fixed iteration rule

$$\boldsymbol{y}^{[k+1]}(t) = \boldsymbol{y}_0 + \int_0^t \boldsymbol{f}\left(\boldsymbol{y}^{[k]}(s), s\right) ds. \tag{4}$$

In (4) we can set $\boldsymbol{y}^{[0]}(t) = \boldsymbol{y}_0$ which, in principle, allows us to compute $\boldsymbol{y}^{[1]}(t)$. With $\boldsymbol{y}^{[1]}(t)$ available we can compute $\boldsymbol{y}^{[2]}(t)$, and so on and so forth. For the sequence of functions $\boldsymbol{y}^{[k]}(t)$ to converge from an arbitrary $\boldsymbol{y}^{[0]}(t)$, we need to make sure that the Frechét derivative of the nonlinear operator at the right hand side of (4) has a norm smaller than one, i.e., that the operator is a contraction. This translates to an upper bound for t, which depends on the Lipschitz constant of $\boldsymbol{f}$. The larger the Lipschitz constant, the smaller t. In other words, Picard iterations converge only if t is smaller than some t_{max} that depends on the Lipschitz constant of $\boldsymbol{f}$. Clearly, once $\boldsymbol{y}(t)$ has been computed to the desired accuracy we can use $y(t)$ as initial condition for the next sequence of Picard iterations, e.g., the sequence that yields the solution within the time interval $[t, 2t]$. The Picard iteration method is clearly not practical, as it involves the evaluation of a time integral at each iteration. Moreover, the function $y^{[k]}(t)$ that is being integrated is defined by an integral with t as one of the endpoints of the integral (see equation (4)).

Next, consider a partition of time interval $[0, T]$ into an evenly-spaced set of grid points:

$$t_0 = 0, \qquad t_N = T, \qquad t_{i+1} = t_i + \Delta t \qquad i = 0, \ldots, N-1, \tag{5}$$

[1] Of course, if the initial value problem (1) does not admit a unique solution then its numerical approximation might just pick one of the possible solutions.

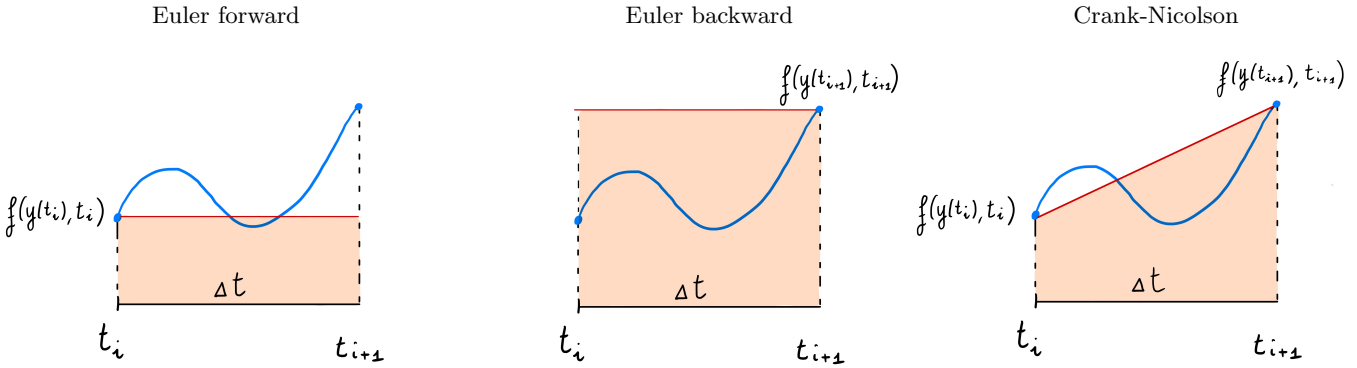


Figure 1: Approximations of the integral $\int_{t_i}^{t_{i+1}} \boldsymbol{f}(\boldsymbol{y}(s), s)ds$ in (6) leading to well-known numerical schemes.

Figure 2: Show how you integrate rectangle and trapezoidal rule to obtain the three methods (Euler forward/backward and CN - 3 Figures).

where $\Delta t = T/N$.

By using the semi-group property of (1) we can write (2) within each time interval $[t_k, t_{k+1}]$ as

$$\boldsymbol{y}(t_{k+1}) = \boldsymbol{y}(t_k) + \int_{t_k}^{t_{k+1}} \boldsymbol{f}\left(\boldsymbol{y}(s), s\right) ds. \tag{6}$$

This formulation is quite convenient for developing numerical methods based on *numerical quadrature formulas*, i.e., numerical approximations of the one-dimensional temporal integral appearing at the right hand side of (6).

Euler and Crank-Nicolson methods These are elementary methods obtained by approximating the integral at the right hand side of (6) by using the rectangle rule or the trapezoidal rule (see Figure 1). Specifically, consider the approximations

$$\int_{t_k}^{t_{k+1}} \boldsymbol{f}\left(\boldsymbol{y}(s), s\right) ds \simeq \Delta t \boldsymbol{f}\left(\boldsymbol{y}(t_k), t_k\right), \tag{7}$$

$$\int_{t_k}^{t_{k+1}} \boldsymbol{f}\left(\boldsymbol{y}(s), s\right) ds \simeq \Delta t \boldsymbol{f}\left(\boldsymbol{y}(t_{k+1}), t_{k+1}\right), \tag{8}$$

$$\int_{t_k}^{t_{k+1}} \boldsymbol{f}\left(\boldsymbol{y}(s), s\right) ds \simeq \frac{\Delta t}{2}\left(\boldsymbol{f}\left[\boldsymbol{y}(t_k), t_k\right) + \boldsymbol{f}\left(\boldsymbol{y}(t_{k+1}), t_{k+1}\right)\right]. \tag{9}$$

These quadrature rules yield, respectively, the following numerical schemes

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \Delta t \boldsymbol{f}\left(\boldsymbol{u}_k, t_k\right) \qquad \text{Euler forward method (explicit)}, \tag{10}$$

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \Delta t \boldsymbol{f}\left(\boldsymbol{u}_{k+1}, t_{k+1}\right) \qquad \text{Euler backward method (implicit)}, \tag{11}$$

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \frac{\Delta t}{2}\left[\boldsymbol{f}\left(\boldsymbol{u}_k, t_k\right) + \boldsymbol{f}\left(\boldsymbol{u}_{k+1}, t_{k+1}\right)\right] \qquad \text{Crank-Nicolson method (implicit)}. \tag{12}$$

In these methods $\boldsymbol{u}_k$ represents an approximation of the exact solution $\boldsymbol{y}(t_k)$, and we set $\boldsymbol{u}_0 = \boldsymbol{y}_0$ (initial condition). Both Euler and Crank-Nicolson methods are *one-step methods*. This means that the approximate solution at time t_{k+1}, i.e., $\boldsymbol{u}_{k+1}$, can be computed by knowing only the solution (or its approximation)

at time t_k. The Euler forward method allows us to compute $\boldsymbol{u}_{k+1}$ explicitly, given $\boldsymbol{u}_k$ and $\boldsymbol{f}(\boldsymbol{u}_k, t_k)$. On the other hand, the Euler backward and Crank-Nicolson methods are "implicit". This is because the approximate solution at time t_{k+1}, i.e., $\boldsymbol{u}_{k+1}$, cannot be computed explicitly based on $\boldsymbol{u}_k$, but it requires solving a nonlinear equation. Specifically, in the case of the Euler backward method we need to solve the nonlinear equation

$$\boldsymbol{u}_{k+1} = \boldsymbol{G}(\boldsymbol{u}_{k+1}) \qquad \text{where} \qquad \boldsymbol{G}(\boldsymbol{u}_{k+1}) = \boldsymbol{u}_k + \Delta t \boldsymbol{f}(\boldsymbol{u}_{k+1}, t_{k+1}). \tag{13}$$

Nonlinear equations of this form are essentially fixed point problems which can be solved numerically using iterative methods such as the Newton's method (provided $\boldsymbol{f}$ is of class C^1). Upon definition of

$$\boldsymbol{F}(\boldsymbol{u}_{k+1}) = \boldsymbol{u}_{k+1} - \boldsymbol{G}(\boldsymbol{u}_{k+1}) \tag{14}$$

we can equivalently write (13) as

$$\boldsymbol{F}(\boldsymbol{u}_{k+1}) = \boldsymbol{0}. \tag{15}$$

The Newton's method for nonlinear systems: The solution to the system of nonlinear equations (15) can be approximated by using the Newton's method or any other method for root finding. As is well known, the Newton's method generates a sequence of vectors $\boldsymbol{u}_{k+1}^{[j]}$ $(j = 0, 1, \ldots)$ converging to $\boldsymbol{u}_{k+1}$ under rather mild assumptions (see [4, Ch. 7]). The Newton's method can be formulated as[2]

$$\underbrace{\left[\boldsymbol{I} - \boldsymbol{J}_{\boldsymbol{G}}\left(\boldsymbol{u}_{k+1}^{[j]}\right)\right]}_{\text{matrix}} \underbrace{\left(\boldsymbol{u}_{k+1}^{[j+1]} - \boldsymbol{u}_{k+1}^{[j]}\right)}_{\text{vector}} = \underbrace{\boldsymbol{G}\left(\boldsymbol{u}_{k+1}^{[j]}\right) - \boldsymbol{u}_{k+1}^{[j]}}_{\text{vector}} \qquad j = 0, 1, \ldots, \tag{18}$$

where $\boldsymbol{J}_{\boldsymbol{G}}\left(\boldsymbol{u}_{k+1}^{[j]}\right)$ is the Jacobian matrix of $\boldsymbol{G}$ defined in equation (13), evaluated at $\boldsymbol{u}_{k+1}^{[j]}$. It is convenient to set the initial guess $\boldsymbol{u}_{k+1}^{[0]}$ for Newton's iteration as $\boldsymbol{u}_{k+1}^{[0]} = \boldsymbol{u}_k$, i.e., the numerical solution at previous time step. For Δt sufficiently small this guarantees that $\boldsymbol{u}_{k+1}^{[0]}$ is within the basin of attraction of $\boldsymbol{u}_{k+1}$. Note also that for Δt sufficiently small the matrix at the left hand side of (18) is a perturbation of the identity (the norm of $\boldsymbol{J}_{\boldsymbol{G}}$ goes to zero linearly in Δt), and therefore $\boldsymbol{I} - \boldsymbol{J}_{\boldsymbol{G}}$ is always invertible for sufficiently small Δt. Indeed $\boldsymbol{I} - \boldsymbol{J}_{\boldsymbol{G}}$ is a diagonally dominant matrix for small Δt. More rigorously, we have the following convergence result (see [4, Theorem 7.1]).

Theorem 1 (Convergence of Newton's method)**.** Let $\boldsymbol{F}(\boldsymbol{x})$ in equation (14) be of class C^1 in a convex open set $D \subseteq \mathbb{R}^n$ that contains a zero of $\boldsymbol{F}$, i.e., a point $\boldsymbol{x}^* \in D$ such that $\boldsymbol{F}(\boldsymbol{x}^*) = \boldsymbol{0}$. If $\boldsymbol{J}_{\boldsymbol{F}}(\boldsymbol{x})$ is invertible at $\boldsymbol{x}^*$ (it always is for sufficiently small Δt), and $\boldsymbol{J}_{\boldsymbol{F}}(\boldsymbol{x})$ is Lipschitz continuous in a neighborhood of $\boldsymbol{x}^*$, i.e.[3],

$$\|\boldsymbol{J}_{\boldsymbol{F}}(\boldsymbol{x}) - \boldsymbol{J}_{\boldsymbol{F}}(\boldsymbol{y})\| \leq L \|\boldsymbol{x} - \boldsymbol{y}\| \tag{19}$$

then there exists a neighborhood of $\boldsymbol{x}^*$ such that for any initial guess $\boldsymbol{x}^{[0]}$ in such a neighborhood we have that the sequence $\boldsymbol{x}^{[k]}$ generated by

$$\boldsymbol{J}_{\boldsymbol{F}}\left(\boldsymbol{x}^{[k]}\right)\left(\boldsymbol{x}^{[k+1]} - \boldsymbol{x}^{[k]}\right) = -\boldsymbol{F}\left(\boldsymbol{x}^{[k]}\right) \tag{20}$$

[2]Consider the Taylor series

$$\boldsymbol{F}\left(\boldsymbol{u}_{k+1}^{[j+1]}\right) = \boldsymbol{F}\left(\boldsymbol{u}_{k+1}^{[j]}\right) + \boldsymbol{J}_{\boldsymbol{F}}\left(\boldsymbol{u}_{k+1}^{[j]}\right)\left(\boldsymbol{u}_{k+1}^{[j+1]} - \boldsymbol{u}_{k+1}^{[j]}\right) + \cdots \qquad j = 0, 1, \ldots. \tag{16}$$

Setting $\boldsymbol{F}\left(\boldsymbol{u}_{k+1}^{[j+1]}\right) = \boldsymbol{0}$ yields

$$\boldsymbol{J}_{\boldsymbol{F}}\left(\boldsymbol{u}_{k+1}^{[j]}\right)\left(\boldsymbol{u}_{k+1}^{[j+1]} - \boldsymbol{u}_{k+1}^{[j]}\right) = -\boldsymbol{F}\left(\boldsymbol{u}_{k+1}^{[j]}\right) \qquad j = 0, 1, \ldots \tag{17}$$

which, upon substitution of (14), coincides with the Newton method (18).

[3]In equation (19) the matrix norm $\|\cdot\|$ at the left hand side is induced by the vector norm at the right hand side.

converges to $\boldsymbol{x}^*$ with order 2. In other words, there exists $m \in \mathbb{N}$ such that

$$\left\|\boldsymbol{x}^{[k+1]} - \boldsymbol{x}^*\right\| \leq C \left\|\boldsymbol{x}^{[k]} - \boldsymbol{x}^*\right\|^2 \quad \text{for all } k \geq m. \tag{21}$$

If we replace $\boldsymbol{u}_{k+1}$ at the right hand side of (12) with one step of the Euler forward scheme (10) we obtain the *Heun method*

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \frac{\Delta t}{2}\left[\boldsymbol{f}\left(\boldsymbol{u}_k, t_k\right) + \boldsymbol{f}\left(\boldsymbol{u}_k + \Delta t \boldsymbol{f}(\boldsymbol{u}_k, t_k), t_{k+1}\right)\right] \qquad \text{Heun method (explicit)} \tag{22}$$

The Heun method is a one-step explicit method that belongs to the class of two-stage explicit Runge-Kutta methods.

Remark: The Euler methods (10)-(11) can be derived also by replacing $d\boldsymbol{y}/dt$ in (1) with the first-order forward and backward finite-differentiation formulas

$$\frac{d\boldsymbol{y}(t_k)}{dt} \simeq \frac{\boldsymbol{y}(t_{k+1}) - \boldsymbol{y}(t_k)}{\Delta t} = \boldsymbol{f}(\boldsymbol{y}(t_k), t_k), \tag{23}$$

$$\frac{d\boldsymbol{y}(t_{k+1})}{dt} \simeq \frac{\boldsymbol{y}(t_{k+1}) - \boldsymbol{y}(t_k)}{\Delta t} = \boldsymbol{f}(\boldsymbol{y}(t_{k+1}), t_{k+1}). \tag{24}$$

The midpoint method. By approximating the integral at the right hand side of (6) using the midpoint rule yields

$$\int_{t_i}^{t_{i+1}} \boldsymbol{f}(\boldsymbol{y}(s), s) ds \simeq \Delta t \boldsymbol{f}\left(\boldsymbol{y}\left(t_i + \frac{\Delta t}{2}\right), t_i + \frac{\Delta t}{2}\right). \tag{25}$$

At this point, we approximate $\boldsymbol{y}(t_i + \Delta t/2)$ using the Euler forward method to obtain

$$\boldsymbol{y}\left(t_i + \frac{\Delta t}{2}\right) \simeq \boldsymbol{y}(t_i) + \frac{\Delta t}{2}\boldsymbol{f}(\boldsymbol{y}(t_i), t_i) \tag{26}$$

to obtain

$$\boldsymbol{u}_{i+1} = \boldsymbol{u}_i + \Delta t \boldsymbol{f}\left(\boldsymbol{u}_i + \frac{\Delta t}{2}\boldsymbol{f}(\boldsymbol{u}_i, t_i), t_i + \frac{\Delta t}{2}\right) \qquad \text{explicit mipoint method} \tag{27}$$

where $\boldsymbol{u}_i$ is an approximation of $\boldsymbol{y}(t_i)$. The explicit midpoint method is a one-step method that belongs to the class of two-stage explicit Runge-Kutta methods.

Approximating $\boldsymbol{y}(t_i + \Delta t/2)$ by the average

$$\boldsymbol{y}\left(t_i + \frac{\Delta t}{2}\right) \simeq \frac{\boldsymbol{y}(t_i) + \boldsymbol{y}(t_{i+1})}{2} \tag{28}$$

yields

$$\boldsymbol{u}_{i+1} = \boldsymbol{u}_i + \Delta t \boldsymbol{f}\left(\frac{\boldsymbol{u}_i + \boldsymbol{u}_{i+1}}{2}, t_i + \frac{\Delta t}{2}\right) \qquad \text{implicit mipoint method} \tag{29}$$

The implicit midpoint method is a *one-step symplectic integrator*, i.e., the scheme preserves the Hamiltonian when applied to Hamiltonian dynamical systems, e.g., pendulum or double-pendulum equations. There is a vast literature on *structure-preserving* integration methods for ordinary differential equations (see, e.g., [1] and the references therein).

Exercise: Solve the pendulum equations with the implicit midpoint method and the Euler forward method. Verify that the Hamiltonian is preserved in the case of the implicit midpoint method is used.

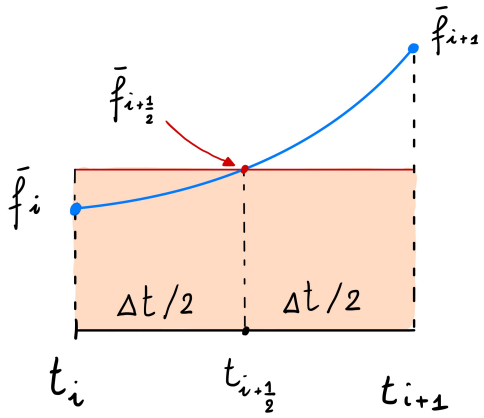


Figure 3: Approximation of the integral $\int_{t_i}^{t_{i+1}} \boldsymbol{f}(\boldsymbol{y}(s),s)ds$ in (6) leading to midpoint method. In this figure we set $\boldsymbol{f}_k = \boldsymbol{f}(\boldsymbol{y}(t_k),t_k)$.

Adams-Bashforth methods. These are *explicit multistep methods* constructed by replacing the integral at the right hand side of (6) with the integral of a polynomial interpolant of $\boldsymbol{f}(\boldsymbol{y}(s),s)$ at $\{t_k, t_{k-1}, \cdots t_{k-q}\}$ which is then extrapolated into $[t_k, t_{k+1}]$ to compute the integral. In other words, we introduce the following approximation

$$\int_{t_k}^{t_{k+1}} \boldsymbol{f}(\boldsymbol{y}(s),s)ds \simeq \int_{t_k}^{t_{k+1}} \Pi_q \boldsymbol{f}(\boldsymbol{y}(s),s)ds \tag{30}$$

where $\Pi_q \boldsymbol{f}(\boldsymbol{y}(s),s)$ is a polynomial of degree q interpolating

$$\{\boldsymbol{f}_k, \boldsymbol{f}_{k-1}, \ldots, \boldsymbol{f}_{k-q}\} \qquad \text{at} \qquad \{t_k, t_{k-1}, \ldots, t_{k-q}\} \tag{31}$$

where $\boldsymbol{f}_k = \boldsymbol{f}(\boldsymbol{y}(t_k),t_k)$. It is very convenient to use Lagrangian interpolation to derive the polynomial $\Pi_q \boldsymbol{f}$. Hereafter we derive the Adams-Bashforth (AB) methods for $q = 0, 1, 2$.

- *One-step Adams-Bashforth method (AB1)*:

$$q = 0 \quad \Rightarrow \quad \Pi_0 \boldsymbol{f}(\boldsymbol{y}(s),s) = \boldsymbol{f}_k \tag{32}$$

  where $\boldsymbol{f}_k = \boldsymbol{f}(\boldsymbol{y}(t_k),t_k)$. Hence,

$$\int_{t_k}^{t_{k+1}} \boldsymbol{f}(\boldsymbol{y}(s),s)ds \simeq \int_{t_k}^{t_{k+1}} \Pi_0 \boldsymbol{f}(\boldsymbol{y}(s),s)ds = \Delta t \boldsymbol{f}(\boldsymbol{y}(t_k),t_k). \tag{33}$$

  This yields

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \Delta t \boldsymbol{f}(\boldsymbol{u}_k,t_k) \qquad \text{AB1 method.} \tag{34}$$

  Note that the AB1 method coincides with the Euler forward method.

- *Two-step Adams-Bashforth method (AB2)*:

$$q = 1 \quad \Rightarrow \quad \Pi_1 \boldsymbol{f}(\boldsymbol{y}(s),s) = \boldsymbol{f}_k l_k(s) + \boldsymbol{f}_{k-1} l_{k-1}(s) \tag{35}$$

  where $l_k(s)$ and $l_{k-1}(s)$ are Lagrange characteristic polynomials

$$l_k(s) = \frac{s - t_{k-1}}{t_k - t_{k-1}}, \qquad l_{k-1}(s) = \frac{s - t_k}{t_{k-1} - t_k}. \tag{36}$$

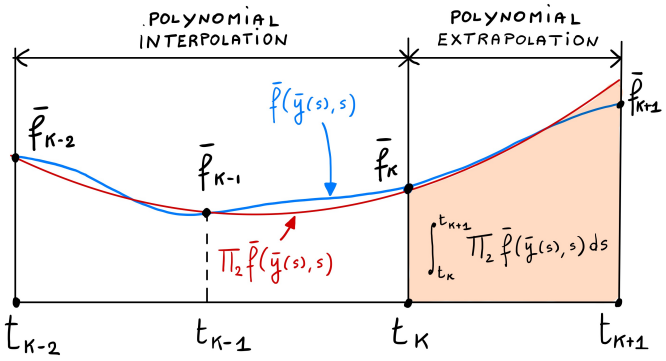


Figure 4: Derivation of the three-step Adams-Bashforth scheme (AB3). We first construct the polynomial $\Pi_2\boldsymbol{f}$ that interpolates $\boldsymbol{f}(\boldsymbol{y}(s),s)$ at t_{k-2}, t_{k-1} and t_k. Subsequently, we extrapolate $\Pi_2\boldsymbol{f}$ to $[t_k,t_{k+1}]$ so that we can compute the integral in equation (30).

This yields

$$\Pi_1\boldsymbol{f}(\boldsymbol{y}(s),s)=\boldsymbol{f}_k\frac{s-t_{k-1}}{t_k-t_{k-1}}+\boldsymbol{f}_{k-1}\frac{s-t_k}{t_{k-1}-t_k}. \tag{37}$$

The (linear) polynomial $\Pi_1\boldsymbol{f}(\boldsymbol{y}(s),s)$ is constructed in $[t_{k-1},t_k]$ and it can be integrated exactly in $[t_k,t_{k+1}]$ using the trapezoidal rule. To this end, we notice that

$$\Pi_1\boldsymbol{f}(\boldsymbol{y}(t_k),t_k)=\boldsymbol{f}_k, \tag{38}$$

$$\begin{aligned}\Pi_1\boldsymbol{f}(\boldsymbol{y}(t_{k+1}),t_{k+1})=&\boldsymbol{f}_k\frac{t_{k+1}-t_{k-1}}{t_k-t_{k-1}}+\boldsymbol{f}_{k-1}\frac{t_{k+1}-t_k}{t_{k-1}-t_k}\\ =&2\boldsymbol{f}_k-\boldsymbol{f}_{k-1}.\end{aligned} \tag{39}$$

which gives us

$$\int_{t_k}^{t_{k+1}}\boldsymbol{f}(\boldsymbol{y}(s),s)ds\simeq\int_{t_k}^{t_{k+1}}\Pi_1\boldsymbol{f}(\boldsymbol{y}(s),s)ds=\frac{\Delta t}{2}\left(3\boldsymbol{f}_k-\boldsymbol{f}_{k-1}\right). \tag{40}$$

Substituting this back into (6) yields the scheme

$$\boldsymbol{u}_{k+1}=\boldsymbol{u}_k+\frac{\Delta t}{2}\left[3\boldsymbol{f}(\boldsymbol{u}_k,t_k)-\boldsymbol{f}(\boldsymbol{u}_{k-1},t_{k-1})\right]\qquad\text{AB2 method} \tag{41}$$

- *Three-step Adams-Bashforth method (AB3)*: With reference to Figure (4)

$$q=3\quad\Rightarrow\quad\Pi_2\boldsymbol{f}(\boldsymbol{y}(s),s)=\boldsymbol{f}_k l_k(s)+\boldsymbol{f}_{k-1}l_{k-1}(s)+\boldsymbol{f}_{k-2}l_{k-2}(s) \tag{42}$$

where $l_k(s)$, $l_{k-1}(s)$ and $l_{k-2}(s)$ are Lagrange characteristic polynomials

$$l_k(s)=\frac{s-t_{k-1}}{t_k-t_{k-1}}\frac{s-t_{k-2}}{t_k-t_{k-2}}, \tag{43}$$

$$l_{k-1}(s)=\frac{s-t_k}{t_{k-1}-t_k}\frac{s-t_{k-2}}{t_{k-1}-t_{k-2}}, \tag{44}$$

$$l_{k-2}(s)=\frac{s-t_k}{t_{k-2}-t_k}\frac{s-t_{k-1}}{t_{k-2}-t_{k-1}}. \tag{45}$$

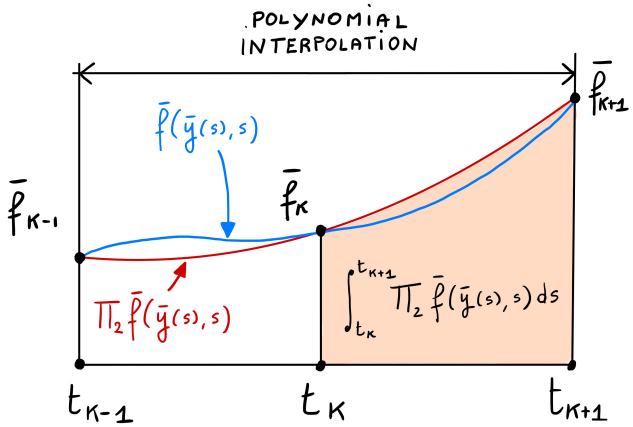


Figure 5: Derivation of the two-step Adams-Moulton scheme (AM2). We first construct the polynomial $\Pi_2\boldsymbol{f}$ that interpolates $\boldsymbol{f}(\boldsymbol{y}(s),s)$ at t_{k-1}, t_k and t_{k+1}. Subsequently, we use $\Pi_2\boldsymbol{f}$ polynomial to approximate the integral in equation (30).

By integrating $\Pi_2\boldsymbol{f}(\boldsymbol{y}(s),s)$ from t_k to t_{k+1} we obtain

$$\int_{t_k}^{t_{k+1}} \boldsymbol{f}(\boldsymbol{y}(s),s)ds \simeq \int_{t_k}^{t_{k+1}} \Pi_2\boldsymbol{f}(\boldsymbol{y}(s),s)ds = \frac{\Delta t}{12}\left(23\boldsymbol{f}_k - 16\boldsymbol{f}_{k-1} + 5\boldsymbol{f}_{k-2}\right). \tag{46}$$

Substituting this back into (6) yields the scheme

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \frac{\Delta t}{12}\left[23\boldsymbol{f}(\boldsymbol{u}_k,t_k) - 16\boldsymbol{f}(\boldsymbol{u}_{k-1},t_{k-1}) + 5\boldsymbol{f}(\boldsymbol{u}_{k-2},t_{k-2})\right] \qquad \text{AB3 method} \tag{47}$$

Higher-order Adams-Bashforth schemes can be obtained similarly. Note that in order to start-up a linear multistep scheme we need to compute the solution at the intermediate steps using different methods. For example, we could start-up the AB2 method with one step of the Heun method (to compute $\boldsymbol{u}_1$) and then carry on the integration using the scheme (41)

Adams-Moulton methods. These are implicit multistep methods in which the time integral at the right hand-side of (6) is approximated by replacing $\boldsymbol{f}(\boldsymbol{y}(s),s)$ by a polynomial $\Pi_q\boldsymbol{f}(\boldsymbol{y}(s),s)$ of degree q interpolating

$$\{\boldsymbol{f}_{k+1},\boldsymbol{f}_k,\ldots,\boldsymbol{f}_{k-q+1}\} \qquad \text{at} \qquad \{t_{k+1},t_{k-1},\ldots,t_{k-q+1}\} \tag{48}$$

The main difference with respect to the Adams-Bashforth method is that there is no extrapolation step, i.e., the point $(t_{k+1},\boldsymbol{f}_{k+1})$ is included in the interpolation (see Figure 5). Let us derive the Adams-Moulton schemes for $q=0,1,2$.

- *One-step Adams-Moulton method (AM0)*:

$$q=0 \quad \Rightarrow \quad \Pi_0\boldsymbol{f}(\boldsymbol{y}(s),s) = \boldsymbol{f}_{k+1} \tag{49}$$

where $\boldsymbol{f}_{k+1} = \boldsymbol{f}(\boldsymbol{y}(t_{k+1}),t_{k+1})$. Hence,

$$\int_{t_k}^{t_{k+1}} \boldsymbol{f}(\boldsymbol{y}(s),s)ds \simeq \int_{t_k}^{t_{k+1}} \Pi_0\boldsymbol{f}(\boldsymbol{y}(s),s)ds = \Delta t\boldsymbol{f}(\boldsymbol{y}(t_{k+1}),t_{k+1}). \tag{50}$$

This yields

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \Delta t\boldsymbol{f}(\boldsymbol{u}_{k+1},t_{k+1}) \qquad \text{AM0 method.} \tag{51}$$

Note that the AM0 method coincides with the Euler backward method.

- *One-step Adams-Moulton method (AM1)*:

$$q = 1 \quad \Rightarrow \quad \Pi_1 \boldsymbol{f}(\boldsymbol{y}(s), s) = \boldsymbol{f}_{k+1} l_{k+1}(s) + \boldsymbol{f}_k l_k(s) \tag{52}$$

where $l_k(s)$ and $l_{k-1}(s)$ are Lagrange characteristic polynomials

$$l_{k+1}(s) = \frac{s - t_k}{t_{k+1} - t_k}, \qquad l_k(s) = \frac{s - t_{k+1}}{t_k - t_{k+1}}. \tag{53}$$

This yields

$$\Pi_1 \boldsymbol{f}(\boldsymbol{y}(s), s) = \boldsymbol{f}_{k+1} \frac{s - t_k}{t_{k+1} - t_k} + \boldsymbol{f}_k \frac{s - t_{k+1}}{t_k - t_{k+1}}. \tag{54}$$

The (linear) polynomial $\Pi_1 \boldsymbol{f}(\boldsymbol{y}(s), s)$ is constructed in $[t_k, t_{k+1}]$ and it can be integrated exactly in the same interval. To this end, we notice that

$$\Pi_1 \boldsymbol{f}(\boldsymbol{y}(t_k), t_k) = \boldsymbol{f}_k, \tag{55}$$

$$\Pi_1 \boldsymbol{f}(\boldsymbol{y}(t_{k+1}), t_{k+1}) = \boldsymbol{f}_{k+1} \tag{56}$$

which imply

$$\int_{t_k}^{t_{k+1}} \boldsymbol{f}(\boldsymbol{y}(s), s) ds \simeq \int_{t_k}^{t_{k+1}} \Pi_1 \boldsymbol{f}(\boldsymbol{y}(s), s) ds = \frac{\Delta t}{2} \left(\boldsymbol{f}_{k+1} + \boldsymbol{f}_k \right). \tag{57}$$

Substituting this back into (6) yields the scheme

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \frac{\Delta t}{2} \left[\boldsymbol{f}(\boldsymbol{u}_{k+1}, t_{k+1}) + \boldsymbol{f}(\boldsymbol{u}_k, t_k) \right] \qquad \text{AM1 method} \tag{58}$$

Hence, the AM1 scheme coincides with the Crank-Nicolson scheme.

- *Two-step Adams-Moulton method (AM2)*:

$$q = 2 \quad \Rightarrow \quad \Pi_2 \boldsymbol{f}(\boldsymbol{y}(s), s) = \boldsymbol{f}_{k+1} l_{k+1}(s) + \boldsymbol{f}_k l_k(s) + \boldsymbol{f}_{k-1} l_{k-1}(s). \tag{59}$$

where $l_{k+1}(s)$, $l_k(s)$ and $l_{k-1}(s)$ are Lagrange characteristic polynomials

$$l_{k+1}(s) = \frac{s - t_k}{t_{k+1} - t_k} \frac{s - t_{k-1}}{t_{k+1} - t_{k-1}}, \tag{60}$$

$$l_k(s) = \frac{s - t_{k+1}}{t_k - t_{k+1}} \frac{s - t_{k-1}}{t_k - t_{k-1}}, \tag{61}$$

$$l_{k-1}(s) = \frac{s - t_{k+1}}{t_{k-1} - t_{k+1}} \frac{s - t_k}{t_{k-1} - t_k}. \tag{62}$$

By integrating $\Pi_2 \boldsymbol{f}(\boldsymbol{y}(s), s)$ from t_k to t_{k+1} we obtain

$$\int_{t_k}^{t_{k+1}} \boldsymbol{f}(\boldsymbol{y}(s), s) ds \simeq \int_{t_k}^{t_{k+1}} \Pi_2 \boldsymbol{f}(\boldsymbol{y}(s), s) ds = \frac{\Delta t}{12} \left(5\boldsymbol{f}_{k+1} + 8\boldsymbol{f}_k - \boldsymbol{f}_{k-1} \right). \tag{63}$$

Substituting this back into (6) yields the scheme

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \frac{\Delta t}{12} \left[5\boldsymbol{f}(\boldsymbol{u}_{k+1}, t_{k+1}) + 8\boldsymbol{f}(\boldsymbol{u}_k, t_k) - \boldsymbol{f}(\boldsymbol{u}_{k-1}, t_{k-1}) \right] \qquad \text{AM2 method.} \tag{64}$$

Backward differentiation formulas (BDF) methods. These are linear implicit multistep methods that perform well for stiff problems. These methods are obtained by approximating $d\boldsymbol{y}(t)/dt$ in (1) using a backward finite-difference formula. These formulas are obtained by interpolating $\boldsymbol{y}(s)$ at $\{t_{k+1}, t_k, \ldots, t_{k-q+1}\}$ with a polynomial of degree q, differentiating the polynomial and evaluating the derivative at $t = t_{k+1}$ (see Figure 6). Let us derive the first few BDF methods.

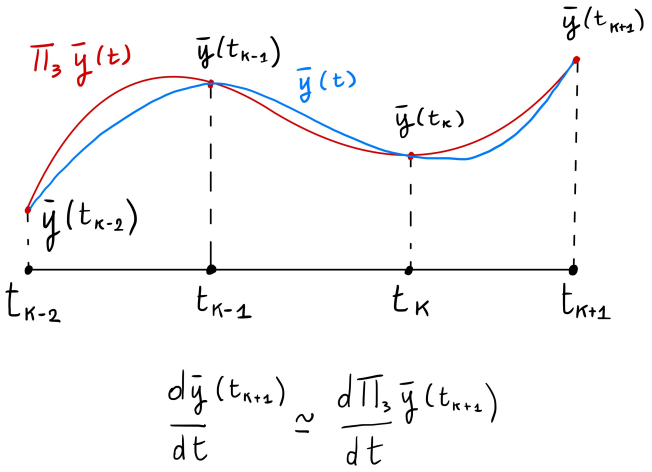


Figure 6: Derivation of the three-step backward differentiation formula (BDF3) method. We first construct the polynomial $\Pi_3 \boldsymbol{f}$ that interpolates $\boldsymbol{y}(t)$ at t_{k-2}, t_{k-1}, t_k, and t_{k+1}. Subsequently, approximate the derivative of $\boldsymbol{y}(t)$ at t_{k+1} with the derivative of the polynomial $\Pi_3 \boldsymbol{y}(t)$ at t_{k+1}.

- *One-step BDF method (BDF1):*

$$\Pi_1 \boldsymbol{y}(s) = \boldsymbol{y}_{k+1} l_{k+1}(s) + \boldsymbol{y}_k l_k(s) \tag{65}$$

where the Lagrange characteristic polynomials are given by

$$l_{k+1}(s) = \frac{s - t_k}{t_{k+1} - t_k}, \qquad l_k(s) = \frac{s - t_{k+1}}{t_k - t_{k+1}} \tag{66}$$

We approximate the derivative of $\boldsymbol{y}(s)$ at t_{k+1} with the derivative of the polynomial $\Pi_1 \boldsymbol{y}(s)$ at t_{k+1}, i.e.,

$$\frac{d\boldsymbol{y}(t_{k+1})}{dt} \simeq \frac{d\Pi_1 \boldsymbol{y}(t_{k+1})}{dt} = \frac{\boldsymbol{y}_{k+1} - \boldsymbol{y}_k}{\Delta t}. \tag{67}$$

Substituting this approximation into the exact equation

$$\frac{d\boldsymbol{y}(t_{k+1})}{dt} = \boldsymbol{f}(\boldsymbol{y}(t_{k+1}), t_{k+1}) \tag{68}$$

yields the scheme

$$\boldsymbol{u}_{k+1} - \boldsymbol{u}_k = \Delta t \boldsymbol{f}(\boldsymbol{u}_{k+1}, t_{k+1}). \tag{69}$$

Note that BDF1 coincides with the Euler backward scheme.

- *Two-step BDF method (BDF2):*

$$\Pi_2 \boldsymbol{y}(s) = \boldsymbol{y}_{k+1} l_{k+1}(s) + \boldsymbol{y}_k l_k(s) + \boldsymbol{y}_{k-1} l_{k-1}(s) \tag{70}$$

where the Lagrange characteristic polynomials are given by

$$l_{k+1}(s) = \frac{s - t_k}{t_{k+1} - t_k} \frac{s - t_{k-1}}{t_{k+1} - t_{k-1}}, \tag{71}$$

$$l_k(s) = \frac{s - t_{k+1}}{t_k - t_{k+1}} \frac{s - t_{k-1}}{t_k - t_{k-1}}, \tag{72}$$

$$l_{k-1}(s) = \frac{s - t_{k+1}}{t_{k-1} - t_{k+1}} \frac{s - t_k}{t_{k-1} - t_k}. \tag{73}$$

We approximate the derivative of $\boldsymbol{y}(s)$ at t_{k+1} with the derivative of the polynomial $\Pi_1\boldsymbol{y}(s)$ at t_{k+1}, i.e.,

$$\frac{d\boldsymbol{y}(t_{k+1})}{dt} \simeq \frac{d\Pi_2\boldsymbol{y}(t_{k+1})}{dt} = \frac{3\boldsymbol{y}_{k+1} - 4\boldsymbol{y}_k + \boldsymbol{y}_{k-1}}{2\Delta t}. \tag{74}$$

Substituting this approximation into the exact equation

$$\frac{d\boldsymbol{y}(t_{k+1})}{dt} = \boldsymbol{f}(\boldsymbol{y}(t_{k+1}), t_{k+1}) \tag{75}$$

yields the scheme

$$\frac{3}{2}\boldsymbol{u}_{k+1} - 2\boldsymbol{u}_k - \frac{1}{2}\boldsymbol{u}_{k-1} = \Delta t\boldsymbol{f}(\boldsymbol{u}_{k+1}, t_{k+1}). \tag{76}$$

- *Three-step BDF method (BDF3):* By following a similar procedure as in BDF2 it is straightforward to show that

$$\frac{11}{6}\boldsymbol{u}_{k+1} - 3\boldsymbol{u}_k + \frac{3}{2}\boldsymbol{u}_{k-1} - \frac{1}{3}\boldsymbol{u}_{k-2} = \Delta t\boldsymbol{f}(\boldsymbol{u}_{k+1}, t_{k+1}). \tag{77}$$

General form of a linear multistep method (LMM). The general form of a linear multistep method is

$$\sum_{j=0}^{q} \alpha_j \boldsymbol{u}_{k+j} = \Delta t \sum_{j=0}^{q} \beta_j \underbrace{\boldsymbol{f}(\boldsymbol{u}_{k+j}, t_{k+j})}_{\boldsymbol{f}_{k+j}}. \tag{78}$$

Note that Adams-Bashforth, Adams-Moulton and BDF methods are all in the form (78). To avoid non-uniqueness of coefficients due to rescaling we set $\alpha_q = 1$. Clearly, if $\beta_q = 0$ and then the method is *explicit*. On the other hand, if $\beta_q \neq 0$ then the method is *implicit*. Let us provide a few examples.

Example: The AB3 method

$$\boldsymbol{u}_{k+3} = \boldsymbol{u}_{k+2} + \frac{\Delta t}{12}\left(23\boldsymbol{f}_{k+2} - 16\boldsymbol{f}_{k+1} + 5\boldsymbol{f}_k\right) \tag{79}$$

can be written in the form (78) by setting

$$\alpha_3 = 1 \quad \alpha_2 = -1, \quad \alpha_1 = 0, \qquad \alpha_0 = 0,$$

$$\beta_3 = 0, \quad \beta_2 = \frac{23}{12}, \quad \beta_1 = -\frac{16}{12}, \quad \beta_0 = \frac{5}{12}.$$

Note that $(\alpha_3, \beta_3) = (1, 0)$ (the method is explicit) and

$$\sum_{j=0}^{3} \beta_j = 1. \tag{80}$$

Example: The BDF2 method

$$\boldsymbol{u}_{k+2} - \frac{4}{3}\boldsymbol{u}_{k+1} + \frac{1}{3}\boldsymbol{u}_k = \frac{2}{3}\Delta t\boldsymbol{f}_{k+2} \tag{81}$$

can be written in the form (78) by setting

$$\alpha_2 = 1, \qquad \alpha_1 = -\frac{4}{3}, \qquad \alpha_0 = \frac{1}{3}, \qquad \beta_2 = \frac{2}{3}, \qquad \beta_1 = 0, \qquad \beta_0 = 0. \tag{82}$$

Note that $(\alpha_2, \beta_2) = (1, 1/3)$ (the method is implicit) and

$$\sum_{j=0}^{2} \alpha_j = 0, \qquad \sum_{j=0}^{2} (\beta_j - j\alpha_j) = 0. \tag{83}$$

As we will see the conditions (80) and (83) guarantee that AB3 and BDF2 are *consistent* methods, i.e., that the truncation error of these methods goes to zero as we send Δt to zero. Roughly speaking this means that the numerical schemes (79) and (81) converge to the ODE (1) as $\Delta t \to 0$. This is necessary but not sufficient for the numerical solution generated by the scheme to converge to the analytical solution. The other element that is needed for convergence is *zero-stability*. The consistency conditions for a general linear q-step method are

$$\sum_{j=0}^{q} \alpha_j = 0, \qquad \sum_{j=0}^{q} (\beta_j - j\alpha_j) = 0. \tag{84}$$

Remark: All linear multistep methods rely on a polynomial interpolation process on an evenly-spaced temporal grid. As is well-known, polynomial interpolation on evenly spaced grids is, in general, ill-conditioned and can undergo a severe Gibbs phenomenon depending on the function. However, the process of interpolating a function with a polynomial of degree $q+1$ within a very small a time interval (equal to $q\Delta t$ for a q-step method) is not ill-conditioned. The reason can be traced back to the fact that we are interpolating on a small time interval $[t - q\Delta t, t]$ in which the function behaves more or less almost like a line. More rigorously, if $g(t)$ is any function of class C^{q+1} in $[t - q\Delta t, t]$ and $\Pi_q g(t)$ is a polynomial of degree q that interpolates $g(t)$ at $\{t, t - \Delta t, \ldots, t - q\Delta t\}$ then we have the error estimate

$$|g(t) - \Pi_q g(t)| = \frac{1}{(q+1)!} \left| \frac{d^{q+1} f(\xi)}{dt^{q+1}} \right| \prod_{j=0}^{q} |(t - t_j)| \leq \frac{(q\Delta t)^{q+1}}{(q+1)!} \left| \frac{d^{q+1} f(\xi)}{dt^{q+1}} \right| \qquad \xi \in [t - q\Delta t, t], \tag{85}$$

which clearly goes to zero for sufficiently small Δt.

Runge-Kutta methods. Runge-Kutta (RK) methods are one-step methods (implicit or explicit) that aim at increasing accuracy by increasing the number of function evaluations within each time step. The general form of a RK method with s stages is

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \Delta t \sum_{i=1}^{s} b_i \boldsymbol{K}_i \tag{86}$$

where

$$\boldsymbol{K}_i = \boldsymbol{f} \left(\boldsymbol{u}_k + \Delta t \sum_{j=1}^{s} a_{ij} \boldsymbol{K}_j, t_k + c_i \Delta t \right). \tag{87}$$

The coefficients of the RK method are usually collected in a table called *Butcher array*

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

The elements a_{ij} in the Butcher table can be positive or negative. Constency of RK methods, i.e., the fact that (86) converge to (1) as $\Delta t \to 0$ implies the following conditions

$$\sum_{j=1}^{s} b_i = 1. \tag{88}$$

Moreover, we assume that

$$c_i = \sum_{j=1}^{s} a_{ij}. \tag{89}$$

Such a "row-sum condition" is not needed for a consistent method.

If $a_{ij} = 0$ for $i \geq j$ then each $\boldsymbol{K}_i$ can be computed recursively from the previous ones and the RK method is *explicit*. Otherwise the RK method is implicit. Let us provide a few examples.

- *Euler forward method*: The Euler forward method can be seen as a one-step explicit RK method. The Butcher array corresponding to such explicit RK1 method is

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

- *Heun method*: The Heun mehod (22) is a two-stage explicit Runge-Kutta method. The Butcher array for the Heun method is:

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

  This table corresponds to the following RK2 method

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \frac{\Delta t}{2} \left(\boldsymbol{K}_1 + \boldsymbol{K}_2 \right), \tag{90}$$

  where

$$\boldsymbol{K}_1 = \boldsymbol{f} \left(\boldsymbol{u}_k, t_k \right), \tag{91}$$
$$\boldsymbol{K}_2 = \boldsymbol{f} \left(\boldsymbol{u}_k + \Delta t \boldsymbol{f}(\boldsymbol{u}_k, t_k), t_k + \Delta t \right). \tag{92}$$

- *Crank-Nicolson method*: The Crank-Nicolson (CN) mehod (12) is a two-stage implicit Runge-Kutta method. The Butcher array corresponding to such method is:

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

  This table corresponds to the following RK2 method

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \frac{\Delta t}{2} \left(\boldsymbol{K}_1 + \boldsymbol{K}_2 \right), \tag{93}$$

  where

$$\boldsymbol{K}_1 = \boldsymbol{f} \left(\boldsymbol{u}_k, t_k \right), \tag{94}$$
$$\boldsymbol{K}_2 = \boldsymbol{f} \left(\boldsymbol{u}_k + \frac{\Delta t}{2} \left(\boldsymbol{K}_1 + \boldsymbol{K}_2 \right), t_{k+1} \right). \tag{95}$$

From equation (93) we see that

$$\frac{\Delta t}{2}\left(\boldsymbol{K}_1+\boldsymbol{K}_2\right)=\boldsymbol{u}_{k+1}-\boldsymbol{u}_k. \tag{96}$$

Substituting this expression into (95) yields

$$\boldsymbol{K}_1=\boldsymbol{f}\left(\boldsymbol{u}_k,t_k\right),\qquad \boldsymbol{K}_2=\boldsymbol{f}\left(\boldsymbol{u}_{k+1},t_{k+1}\right). \tag{97}$$

- *Kutta's method*: This method is an explicit 3-stage method corresponding to the Butcher array:

0	0	0	0
1/2	1/2	0	0
1	−1	2	0
	1/6	2/3	1/6

The method can be written as

$$\boldsymbol{K}_1=\boldsymbol{f}(\boldsymbol{u}_k,t_k), \tag{98}$$
$$\boldsymbol{K}_2=\boldsymbol{f}\left(\boldsymbol{u}_k+\frac{\Delta t}{2}\boldsymbol{K}_1,t_k+\frac{\Delta t}{2}\right), \tag{99}$$
$$\boldsymbol{K}_3=\boldsymbol{f}\left(\boldsymbol{u}_k-\Delta t\boldsymbol{K}_1+2\Delta t\boldsymbol{K}_2,t_k+\Delta t\right), \tag{100}$$

$$\boldsymbol{u}_{k+1}=\boldsymbol{u}_k+\frac{\Delta t}{6}\left(\boldsymbol{K}_1+4\boldsymbol{K}_2+\boldsymbol{K}_3\right). \tag{101}$$

- *Runge-Kutta method (RK4)*: The most famous RK method is perhaps the one proposed in the original paper by Runge and Kutta. Such a method is an explicit 4-stage method corresponding to the Butcher array:

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	1/3	1/3	1/6

The method can be written as

$$\boldsymbol{K}_1=\boldsymbol{f}(\boldsymbol{u}_k,t_k), \tag{102}$$
$$\boldsymbol{K}_2=\boldsymbol{f}\left(\boldsymbol{u}_k+\frac{\Delta t}{2}\boldsymbol{K}_1,t_k+\frac{\Delta t}{2}\right), \tag{103}$$
$$\boldsymbol{K}_3=\boldsymbol{f}\left(\boldsymbol{u}_k+\frac{\Delta t}{2}\boldsymbol{K}_2,t_k+\frac{\Delta t}{2}\right), \tag{104}$$
$$\boldsymbol{K}_4=\boldsymbol{f}\left(\boldsymbol{u}_k+\Delta t\boldsymbol{K}_3,t_k+\Delta t\right), \tag{105}$$

$$\boldsymbol{u}_{k+1}=\boldsymbol{u}_k+\frac{\Delta t}{6}\left(\boldsymbol{K}_1+2\boldsymbol{K}_2+2\boldsymbol{K}_3+\boldsymbol{K}_4\right). \tag{106}$$

$$\begin{array}{c|cc} 0 & 0 & 0 \\ c & c & 0 \\ \hline & b_1 & b_2 \end{array}$$

Derivation of explicit RK2 methods. Let us show how to derive an arbitrary explicit two-stage RK method. To this end we first write the Butcher array:

where $b_1 + b_2 = 1$. The RK2 method corresponding to this table can be written explicitly as

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \Delta t(b_1 \boldsymbol{K}_1 + b_2 \boldsymbol{K}_2) \tag{107}$$

where

$$\boldsymbol{K}_1 = \boldsymbol{f}(\boldsymbol{u}_k, t_k) \tag{108}$$
$$\boldsymbol{K}_2 = \boldsymbol{f}(\boldsymbol{u}_k + c\Delta t \boldsymbol{K}_1, t_k + c\Delta t). \tag{109}$$

Expand $\boldsymbol{K}_2$ in a Taylor series in Δt to obtain

$$\boldsymbol{K}_2 = \boldsymbol{K}_1 + c\Delta t \left(\sum_{j=1}^{n} \frac{\partial \boldsymbol{f}}{\partial y_j} K_{1j} + \frac{\partial \boldsymbol{f}}{\partial t} \right) + \cdots . \tag{110}$$

A substitution of this expression into (107) yields

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \Delta t(b_1 + b_2)\boldsymbol{f}(\boldsymbol{u}_k, t_k) + b_2 c(\Delta t)^2 \left(\sum_{j=1}^{n} \frac{\partial \boldsymbol{f}(\boldsymbol{u}_k, t_k)}{\partial y_j} f_j(\boldsymbol{u}_k, t_k) + \frac{\partial \boldsymbol{f}(\boldsymbol{u}_k, t_k)}{\partial t} \right). \tag{111}$$

Next, we expand the solution to the ODE (1) in a Taylor series at time $t_{k+1} = t_k + \Delta t)$, assuming that $\boldsymbol{y}(t_k) = \boldsymbol{u}_k$. This yields

$$\boldsymbol{y}(t_{k+1}) = \boldsymbol{u}(t_k) + \frac{d\boldsymbol{y}(t_k)}{dt}\Delta t + \frac{(\Delta t)^2}{2}\frac{d^2\boldsymbol{y}(t_k)}{dt^2} + \cdots . \tag{112}$$

By using (1) and the chain rule we obtain:

$$\frac{d\boldsymbol{y}(t_k)}{dt} = \boldsymbol{f}(\boldsymbol{y}_k, t_k), \qquad \frac{d^2\boldsymbol{y}(t_k)}{dt^2} = \sum_{j=1}^{n} \frac{\partial \boldsymbol{f}(\boldsymbol{y}_k, t_k)}{\partial y_j} f_j(\boldsymbol{y}_k, t_k) + \frac{\partial \boldsymbol{f}(\boldsymbol{y}_k, t_k)}{\partial t}. \tag{113}$$

Assuming that $\boldsymbol{y}_k = \boldsymbol{u}_k$ and matching the terms multiplying the same powers of Δt in (111) and (112) we obtain[4]

$$b_1 + b_2 = 1 \qquad \text{and} \qquad cb_2 = \frac{1}{2}. \tag{114}$$

This is a system of 2 equations in 3 unknowns. Thus, there is an *infinite number* of explicit (and consistent) RK2 methods. For example, setting

$$c = 1, \qquad b_1 = b_2 = \frac{1}{2} \qquad \text{yields the } \textit{Heun method } (22). \tag{115}$$

On the other hand, setting

$$c = \frac{1}{2}, \quad b_1 = 0 \quad b_2 = 1 \qquad \text{yields the } \textit{explicit midpoint method } (27). \tag{116}$$

[4]The condition $b_1 + b_2 = 1$ is a consistency condition which guarantees that the scheme (107) converges to the ODE (1) in the limit $\Delta t \to 0$.

By using the Taylor series approach discussed in this section, it is possible to derive conditions on the entries of the Butcher array for RK methods with an arbitrary number of stages. Essentially, we perform a Taylor series of the RK method in Δt and then match it with the Taylor series expansion of the solution up to a given order. The corresponding equations for the coefficients, e.g., (114) are called "stage-order" conditions. This is discussed, e.g., in [2, §5.9]. For example, it can be shown that for a three stage RK method

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ c_2 & a_{21} & 0 & 0 \\ c_3 & a_{31} & a_{32} & 0 \\ \hline & b_1 & b_2 & b_3 \end{array}$$

we obtain the order conditions

$$\begin{cases} b_1 + b_2 + b_3 = 1 \\ b_2 c_2 + b_3 c_3 = \dfrac{1}{2} \\ b_2 c_2^2 + b_3 c_3^2 = \dfrac{1}{3} \\ b_3 a_{32} c_2 = \dfrac{1}{6} \end{cases} \tag{117}$$

which yields to one two-parameter family of solutions and two one-parameter families of solutions [2, p. 178]. As easily seen, the Taylor series approach rapidly become intractable as the number of stages increases, and so does the corresponding set of order conditions. Fortunately, there is a more efficient way to derive conditions such as (114) by using rooted trees (see [2, §5.6]).

Derivation of implicit RK methods. Implicit RK methods can be derived from the integral formulation (6) of the Cauchy problem. In fact, if a Gauss quadrature formula with s nodes in $[tk, t_{k+1}]$ is employed to approximate the integral at the right hand side of (6), we obtain

$$\int_{t_k}^{t_{k+1}} \boldsymbol{f}(\boldsymbol{y}(s), s) ds \simeq \Delta t \sum_{j=1}^{s} b_j \boldsymbol{f}\left(\boldsymbol{y}(t_k + c_j \Delta t), t_k + c_j \Delta t\right). \tag{118}$$

Here, we denoted by b_j the quadrature weights and by $t_k + c_j \Delta t$ the quadrature nodes. It can be proved that for any RK formula (86)-(87), there exists a correspondence between the coefficients b_j , c_j of the formula and the weights and nodes of a Gauss quadrature rule (see, [2, §5.11] for details). For instance, the implicit midpoint method can be seen as a one step implicit RK method based on a 1 point Gauss-Legendre quadrature rule. This corresponds to the Butcher array

$$\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array}$$

and can attain order 2. In general, an implicit s-stage Gauss RK method can achieve order $2s$. Similarly, Gauss-Radau and Gauss-Lobatto RK methods can achieve order $2s-1$ and $2s-2$, respectively.

References

[1] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*. Springer, 2006.

[2] J. D. Lambert. *Numerical methods for ordinary differential systems: the initial value problem*. Wiley, 1991.

[3] R. LeVeque. *Finite difference methods for ordinary and partial differential equations*. SIAM, 2007.

[4] A. Quarteroni, R. Sacco, and F. Salieri. *Numerical mathematics*. Springer, 2007.

Consistency of numerical methods for ODEs

In the previous lecture we provided an overview of several numerical methods to solve an initial value problem for a system of ODEs. In particular, we discussed linear multistep methods (LMM), Runge-Kutta methods (RK), and backward differentiation formulas (BFD) methods. All these methods can be written in the general form (see [1, p. 24])

$$\sum_{j=0}^{q} \alpha_j \boldsymbol{u}_{k+j} = \Delta t \boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{u}_{k+q}, \ldots, \boldsymbol{u}_k, t_k, \Delta t), \tag{1}$$

where $\Phi_{\boldsymbol{f}}$ is an *iteration function* that depends on $\boldsymbol{f}$ (right hand side of the ODE system), as well as on the approximate solution $\boldsymbol{u}_k$ at different times. In (1) we set $\alpha_q = 1$ to avoid non-uniqueness of the scheme, e.g., when we multiply it by a nonzero constant. The iteration function $\Phi_{\boldsymbol{f}}$ satisfies the following conditions

$$\boldsymbol{\Phi}_{\boldsymbol{0}}(\boldsymbol{u}_{k+q}, \ldots, \boldsymbol{u}_k, t_k, \Delta t) = 0, \tag{2}$$

$$\|\boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{z}_{k+q}, \ldots, \boldsymbol{z}_k, t_k, \Delta t) - \boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{u}_{k+q}, \ldots, \boldsymbol{u}_k, t_k, \Delta t)\| \leq M \sum_{j=0}^{q} \|\boldsymbol{u}_{k+j} - \boldsymbol{z}_{k+j}\|. \tag{3}$$

The second condition follows from the Lipshitz continuity of $\boldsymbol{f}$. Let us show how to write a few well-known schemes in the form (1).

- *Crank-Nicolson:*

  The Crank-Nicolson scheme corresponds to $q = 1$ (one step), with $\alpha_1 = 1$, $\alpha_0 = -1$ and iteration function given by

  $$\boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, t_k, \Delta t) = \frac{1}{2}\left[\boldsymbol{f}(\boldsymbol{u}_k, t_k) + \boldsymbol{f}(\boldsymbol{u}_{k+1}, t_k + \Delta t)\right]. \tag{4}$$

- *Heun (RK2):*

  Here we have again $q = 1$ (one step), $\alpha_1 = 1$, $\alpha_0 = -1$ and iteration function given by

  $$\boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{u}_k, t_k, \Delta t) = \frac{1}{2}\left[\boldsymbol{f}(\boldsymbol{u}_k, t_k) + \boldsymbol{f}(\boldsymbol{u}_k + \Delta t \boldsymbol{f}(\boldsymbol{u}_k, t_k), t_k + \Delta t)\right] \tag{5}$$

- *Adams-Bashforth 2 (AB2):*

  Here we have $q = 2$ (two-steps), $\alpha_2 = 1$, $\alpha_1 = -1$, $\alpha_0 = 0$

  $$\boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{u}_{k+1}, \boldsymbol{u}_k, t_k, \Delta t) = \frac{1}{2}\left[3\boldsymbol{f}(\boldsymbol{k}_{n+1}, t_k + \Delta t) - \boldsymbol{f}(\boldsymbol{u}_k, t_k)\right] \tag{6}$$

Local truncation error and consistency. The local truncation error of a numerical scheme is the error arising from the scheme when we perform one step forward from an exact initial condition, i.e., an initial condition defined by the analytical solution of the ODE system

$$\begin{cases} \dfrac{d\boldsymbol{y}}{dt} = \boldsymbol{f}(\boldsymbol{y}, t) \\ \boldsymbol{y}(0) = \boldsymbol{y}_0 \end{cases} \tag{7}$$

Let $\boldsymbol{y}(t)$ be such analytical solution, and denote by $\boldsymbol{y}_k = \boldsymbol{y}(t_k)$. The *local truncation error* $\boldsymbol{\tau}_{k+q}$ of the scheme (1) is defined as

$$\sum_{j=0}^{q} \alpha_j \boldsymbol{y}_{k+j} = \Delta t \boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{y}_{k+q}, \ldots, \boldsymbol{y}_k, t_k, \Delta t) + \Delta t \boldsymbol{\tau}_{k+q}, \tag{8}$$

i.e.,

$$\boldsymbol{\tau}_{k+q} = \frac{1}{\Delta t} \sum_{j=0}^{q} \alpha_j \boldsymbol{y}_{k+j} - \boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{y}_{k+q}, \ldots, \boldsymbol{y}_k, t_k, \Delta t), \qquad \text{(local truncation error)} \tag{9}$$

The *global truncation error* of a numerical scheme that undergoes multiple iterations, say N within a certain time interval $[0, T]$, is defined as

$$\|\boldsymbol{\tau}\| = \max_{k=1,\ldots,N} \|\boldsymbol{\tau}_{k+q}\| . \tag{10}$$

Definition 1 (Consistency). Let $\boldsymbol{\tau}_{n+q}$ be the local truncation error of the scheme (1). If

$$\lim_{\Delta t \to 0} \|\boldsymbol{\tau}_{k+q}\| = 0 \tag{11}$$

then we say that the numerical scheme (1) is *consistent*. If $\|\boldsymbol{\tau}_{k+q}\|$ goes to zero as Δt^p then we say that the numerical scheme is *consistent with order* p.

Stated in simple terms, a consistent numerical scheme converges to the ODE (7) as we send Δt to zero (to some order in Δt). As we will see, this is not alone sufficient to guarantee that the discrete solution we obtain by iterating the scheme, i.e., $\boldsymbol{u}_k$ converges to the solution to (7), i.e.,

$$\lim_{\Delta t \to} \|\boldsymbol{u}_k - \boldsymbol{y}_k\| = 0 \qquad \forall k = 0, \ldots, N. \tag{12}$$

In order for u_k to converge to $\boldsymbol{y}_k = \boldsymbol{y}(t_k)$ as $\Delta t \to 0$ a consistent scheme has to be also *zero-stable*.

Remark: In equation (11) and (12) $\|\cdot\|$ denotes any norm in $\mathbb{R}^n$. Since all norms are equivalent in $\mathbb{R}^n$ we have that consistency in one norm implies consistency in any other norm. Also, the consistency order does not depend on the norm that is used. Let us consider a few examples in which we determine the local truncation error and the consistency order by a direct calculation.

- *Consistency of Euler-Forward:* The Euler forward scheme

  $$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \Delta t \boldsymbol{f}(\boldsymbol{u}_k, t_k) \tag{13}$$

  is consistent with order one. To this end, suppose we have available the analytical solution $\boldsymbol{y}(t)$ to (7), and denote by

  $$\boldsymbol{y}_{k+1} = \boldsymbol{y}(t_{k+1}). \tag{14}$$

  A substitution of this expression into (13) yields

  $$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \Delta t \boldsymbol{f}(\boldsymbol{y}_k, t_k) + \Delta t \boldsymbol{\tau}_{k+1}, \tag{15}$$

  i.e.,

  $$\boldsymbol{\tau}_{k+1} = \frac{\boldsymbol{y}_{k+1} - \boldsymbol{y}_k}{\Delta t} - \boldsymbol{f}(\boldsymbol{y}_k, t_k). \tag{16}$$

By expanding $\boldsymbol{y}_{k+1} = \boldsymbol{y}(t_{k+1})$ in a Taylor series (in time) we obtain[1]

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \Delta t \frac{d\boldsymbol{y}(t_k)}{dt} + \underbrace{\frac{\Delta t^2}{2}\frac{d^2\boldsymbol{y}(\boldsymbol{\xi}_k)}{dt^2}}_{\text{remainder}}, \qquad \boldsymbol{\xi}_k \in [t_k, t_{k+1}]^n. \tag{17}$$

In this equation, $\boldsymbol{y}(\boldsymbol{\xi}_k)$ represents a vector with components $y_j(\xi_{jk})$, i.e.

$$\boldsymbol{y}(\boldsymbol{\xi}_k) = \begin{bmatrix} y_1(\xi_{1k}) \\ y_2(\xi_{2k}) \\ \vdots \\ y_n(\xi_{nk}) \end{bmatrix}. \tag{18}$$

Substituting the series (17) into the expression (16) and computing the norm yields

$$\|\boldsymbol{\tau}_{k+1}\| = \frac{\Delta t}{2}\left\|\frac{d^2\boldsymbol{y}(\boldsymbol{\xi}_k)}{dt^2}\right\|. \tag{19}$$

Hence, the Euler forward method is *consistent with order one.*

- *Consistency of Crank-Nicolson:* The Crank-Nicolson scheme

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \frac{\Delta t}{2}\left[\boldsymbol{f}(\boldsymbol{u}_k, t_k) + \boldsymbol{f}(\boldsymbol{u}_{k+1}, t_{k+1})\right] \tag{20}$$

is consistent with order two. To show this, let us substitute of the analytical solution of (7), denoted as $\boldsymbol{y}_k = \boldsymbol{y}(t_k)$, into (20) to obtain the local truncation error

$$\boldsymbol{\tau}_{k+1} = \frac{\boldsymbol{y}_{k+1} - \boldsymbol{y}_k}{\Delta t} + \frac{1}{2}\left[\boldsymbol{f}(\boldsymbol{y}_k, t_k) + \boldsymbol{f}(\boldsymbol{y}_{k+1}, t_{k+1})\right]. \tag{21}$$

Next, we expand $\boldsymbol{f}(\boldsymbol{y}_{k+1}, t_{k+1})$ in a Taylor series

$$\boldsymbol{f}(\boldsymbol{y}_{k+1}, t_{k+1}) = \boldsymbol{f}(\boldsymbol{y}_k, t_k) + \Delta t \left.\frac{d\boldsymbol{f}(\boldsymbol{y}(t), t)}{dt}\right|_{t=t_k} + \frac{\Delta t^2}{2}\left.\frac{d^2\boldsymbol{f}(\boldsymbol{y}(t), t)}{dt^2}\right|_{t=t_k} + \cdots. \tag{22}$$

Similarly, we expand $\boldsymbol{y}_{k+1}$ is another Taylor series

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \Delta t \frac{d\boldsymbol{y}(t_k)}{dt} + \frac{\Delta t^2}{2}\frac{d^2\boldsymbol{y}(t_k)}{dt^2} + \frac{\Delta t^3}{6}\frac{d^3\boldsymbol{y}(t_k)}{dt^3} + \cdots. \tag{23}$$

A substution of (22)-(23) into (21) yields[2] after simple algebraic simplifications

$$\|\boldsymbol{\tau}_{k+1}\| \leq \frac{2}{3}\Delta t^2 \left\|\frac{d^3\boldsymbol{y}(t_k)}{dt^3}\right\| + o(\Delta t^2) \tag{24}$$

for any vector norm $\|\cdot\|$. Equation (24) shows that the local truncation error $\|\boldsymbol{\tau}_{k+1}\|$ goes to zero as Δt^2 and therefore the Crank-Nicolson method is *consistent with order 2.*

[1] Equation (17) represents simultaneously n different Taylor series, one for each $y_j(t_k + \Delta t)$, $j = 1, \ldots, n$. This yields a remainder for each series in which the second derivative d^2y_j/dt^2 is evaluated at some point ξ_{jk} within the time interval $[t_k, t_{k+1}]$. If we use a vector notation, this yields a vector $\boldsymbol{\xi}_k$ representing a point in the hyper-cube $[t_k, t_{k+1}]^n$.

[2] To derive (24) we recall that

$$\frac{d^2\boldsymbol{y}}{dt^2} = \frac{d\boldsymbol{f}(\boldsymbol{y}, t)}{dt} \qquad \frac{d^3\boldsymbol{y}}{dt^3} = \frac{d^2\boldsymbol{f}(\boldsymbol{y}, t)}{dt^2}$$

General conditions for consistency. Next, we derive a set conditions guaranteeing that the local truncation method of the general method (1) goes to zero as $\Delta t \to 0$. To this end, let us expand $\boldsymbol{y}_{k+j} = \boldsymbol{y}(t_k + j\Delta t)$ in equation (9) in a first-order Taylor series

$$\boldsymbol{y}_{k+j} = \boldsymbol{y}_k + j\Delta t \frac{d\boldsymbol{y}(t_k)}{dt} + \cdots . \tag{25}$$

Substituting (25) into (9) yields

$$\boldsymbol{\tau}_{k+q} = \frac{\boldsymbol{y}_k}{\Delta t}\sum_{j=0}^{q}\alpha_j + \frac{d\boldsymbol{y}(t_k)}{dt}\sum_{j=0}^{q} j\alpha_j - \boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{y}_{k+q},\ldots,\boldsymbol{y}_k,t_k,\Delta t) + \cdots . \tag{26}$$

In the limit $\Delta t \to 0$ we have that $\boldsymbol{y}_{k+j} \to \boldsymbol{y}_k$ for all $j = 0,\ldots,q$. Assuming that $\boldsymbol{y}_k$ is arbitrary, the previous equation yields the consistency conditions

$$1. \qquad \sum_{j=0}^{q}\alpha_j = 0 \tag{27}$$

$$2. \qquad \frac{\boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{y}_k,\ldots,\boldsymbol{y}_k,t_k,0)}{\displaystyle\sum_{j=0}^{q} j\alpha_j} = \boldsymbol{f}(\boldsymbol{y}_k,t_k) \tag{28}$$

At this point it is convenient to define the *first characteristic polynomial* associated with the numerical method (1)

$$\rho(z) = \sum_{j=0}^{q}\alpha_j z^j \qquad \text{(first characteristic polynomial)} \tag{29}$$

By using ρ we can write the *consistency conditions* in a more compact form as

$$1. \qquad \rho(1) = 0 \tag{30}$$

$$2. \qquad \frac{\boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{y}_k,\ldots,\boldsymbol{y}_k,t_k,0)}{\rho'(1)} = \boldsymbol{f}(\boldsymbol{y}_k,t_k) \tag{31}$$

where $\rho'(z) = d\rho(z)/ds$.

We emphasize that the consistency conditions (27)-(28) (or the equivalent ones (30)-(31)) do not provide *any information on the consistency order*, but simply allow us to check whether a numerical scheme is consistent or not. Let us provide a few examples.

- ***One-step methods***: Consider a general one-step method[3] in the form

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \Delta t\boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{u}_{k+1},\boldsymbol{u}_k,t_k,\Delta t). \tag{32}$$

  The first characteristic polynomial associated with (32) is

$$\rho(z) = z - 1 \tag{33}$$

  In fact, the scheme (32) can be written in the form (1) if we set $q = 1$, $\alpha_1 = 1$ and $\alpha_0 = -1$. By evaluating $\rho(z)$ and $\rho'(z)$ at $z = 1$ we obtain

$$\rho(1) = 0, \qquad \rho'(1) = 1 \tag{34}$$

[3]Recall that all Runge-Kutta methods (implicit and explicit) are in the form (32).

Hence, (30) is always satisfied. The second condition, i.e. (31), can be written as

$$\Phi_{\boldsymbol{f}}(\boldsymbol{y}_k, \boldsymbol{y}_k, t_k, 0) = \boldsymbol{f}(\boldsymbol{y}_k, t_k) \quad \text{for all} \quad \boldsymbol{y}_k \in \mathbb{R}^n. \tag{35}$$

This condition is clearly satisfied, e.g., by the Crank-Nicolson method (see the iteration function (4)), and by the implicit midpoint method. We recall that the iteration function for the latter is

$$\Phi_{\boldsymbol{f}}(\boldsymbol{u}_{k+1}, \boldsymbol{u}_k, t_k, \Delta t) = \boldsymbol{f}\left(\frac{\boldsymbol{u}_{k+1} + \boldsymbol{u}_k}{2}, t_k + \Delta t\right) \quad \Rightarrow \quad \Phi_{\boldsymbol{f}}(\boldsymbol{y}_k, \boldsymbol{y}_k, t_k, 0) = \boldsymbol{f}(\boldsymbol{y}_k, t_k). \tag{36}$$

Regarding Runge-Kutta methods, we recall that their iteration function can be written as

$$\boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, t_k, \Delta t) = \sum_{i=1}^{s} b_i \boldsymbol{K}_i \qquad \boldsymbol{K}_i = \boldsymbol{f}\left(\boldsymbol{u}_k + \Delta t \sum_{j=1}^{s} a_{ij} \boldsymbol{K}_j, t_k + c_i \Delta t\right) \tag{37}$$

Hence, condition (35) implies that Runge-Kutta methods are consistent if and only if

$$\sum_{i=1}^{s} b_i = 1. \tag{38}$$

- ***Adams methods***: The general form of a q-step Adams method is

$$\boldsymbol{u}_{k+q} = \boldsymbol{u}_{k+q-1} + \Delta t \sum_{j=0}^{q} \beta_j f(\boldsymbol{u}_{k+j}, t_{k+j}). \tag{39}$$

If $\beta_q = 0$ then the method is explicit (Adams-Bashforth). Otherwise it is implicit (Adams-Moulton). The first characteristic polynomial and the iteration function of a q-step Adams method are, respectively

$$\rho(z) = z^q - z^{q-1}, \qquad \boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{u}_{k+q}, \dots, \boldsymbol{u}_k, t_k, \Delta t) = \sum_{j=0}^{q} \beta_j \boldsymbol{f}(\boldsymbol{u}_{k+j}, t_k + j\Delta t). \tag{40}$$

Clearly,

$$\rho(1) = 0 \qquad \text{and} \qquad \rho'(z) = q z^{q-1} - (q-1) z^{q-2} \quad \Rightarrow \quad \rho'(1) = 1. \tag{41}$$

Hence, the first consistency condition (30) is always satisfied. The second condition (31) can be written as

$$\Phi_{\boldsymbol{f}}(\boldsymbol{y}_k, \dots, \boldsymbol{y}_k, t_k, 0) = \boldsymbol{f}(\boldsymbol{u}_k, t_k) \sum_{j=0}^{q} \beta_j = \boldsymbol{f}(\boldsymbol{y}_k, t_k), \tag{42}$$

and it is satisfied for all $y_k \in \mathbb{R}^n$ if and only if

$$\sum_{j=0}^{q} \beta_j = 1. \tag{43}$$

- ***BDF methods***: The general form of a q-step BDF method is

$$\sum_{j=0}^{q} \alpha_j \boldsymbol{u}_{k+j} = c \Delta t \boldsymbol{f}(\boldsymbol{u}_{k+q}, t_{k+q}). \tag{44}$$

where the constant c takes care of the fact that we set $\alpha_q = 1$. Equivalently, we can say that we set $c = \beta_q$. For BDF methods we have

$$\rho(z) = \sum_{j=0}^{q} \alpha_j z^j, \qquad \boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{u}_{k+q}, t_k, \Delta t) = c \boldsymbol{f}(\boldsymbol{u}_{k+q}, t_k + q\Delta t). \tag{45}$$

Therefore the consistency conditions (30)-(31) reduce to

$$\rho(1) = 0, \qquad \rho'(1) = c. \tag{46}$$

For example, the BDF2 method can be written in the form (44) as

$$\boldsymbol{u}_{k+2} - \frac{4}{3}\boldsymbol{u}_{k+1} + \frac{1}{3}\boldsymbol{u}_k = \frac{2}{3}\Delta t \boldsymbol{f}(\boldsymbol{u}_{k+2}, t_{k+2}), \tag{47}$$

which yields

$$\rho(z) = z^2 - \frac{4}{3}z + \frac{1}{3} \qquad \rho'(z) = 2z - \frac{4}{3}. \tag{48}$$

Clearly, $\rho(1) = 0$ and $\rho'(1) = 2/3$, which implies that BDF2 is consistent.

- ***LMM methods***: We have seen that the general form of a linear q-step method is

$$\sum_{j=0}^{q} \alpha_j \boldsymbol{u}_{k+j} = \Delta t \sum_{j=0}^{q} \beta_j \boldsymbol{f}(\boldsymbol{u}_{k+j}, t_{k+j}). \tag{49}$$

In this case the consistency conditions (30)-(31) can be written, respectively, as

$$\sum_{j=0}^{q} \alpha_j = 0, \qquad \sum_{j=0}^{q} (j\alpha_j - \beta_j) = 0. \tag{50}$$

Order of consistency of linear multistep methods. We have seen in previous section that there is a simple criterion to check whether a numerical scheme of the form (1) is consistent or not. The criterion is summarized by the conditions (27)-(28), or the equivalent ones (30)-(31) involving the first characteristic polynomial of the scheme. The consistency conditions (27)-(28), however, do not provide any indication on the order of consistency. In this section we derive a theory that allows us to determine the order of consistency for general linear multistep methods of the form

$$\sum_{j=0}^{q} \alpha_j \boldsymbol{u}_{k+j} = \Delta t \sum_{j=0}^{q} \beta_j \boldsymbol{f}(\boldsymbol{u}_{k+j}, t_{k+j}). \tag{51}$$

As we know, this class of methods includes Adams-Bashforth methods, Adams-Moulton methods and BDF methods. We define the following polynomials associated with (51)

$$\rho(z) = \sum_{j=0}^{q} \alpha_j z^j \qquad \text{(First characteristic polynomial)}, \tag{52}$$

$$\sigma(z) = \sum_{j=0}^{q} \beta_j z^j \qquad \text{(Second characteristic polynomial)}. \tag{53}$$

The local truncation error of the linear multistep scheme (51) is

$$\boldsymbol{\tau}_{k+q} = \frac{1}{\Delta t}\sum_{j=0}^{q} \alpha_j \boldsymbol{y}_{k+j} - \sum_{j=0}^{q} \beta_j \boldsymbol{f}(\boldsymbol{y}_{k+j}, t_{k+j}), \tag{54}$$

where $\boldsymbol{y}_k = \boldsymbol{y}(t_k)$ denotes the analytical solution to (7) evaluated at t_k. On the other hand, by evaluating the ODE (7) at t_{k+j} we obtain

$$\boldsymbol{f}(\boldsymbol{y}_{k+j}, t_{k+j}) = \frac{d\boldsymbol{y}(t_{k+j})}{dt}. \tag{55}$$

Substituting (55) into (54) yields

$$\boldsymbol{\tau}_{k+q} = \frac{1}{\Delta t} \underbrace{\sum_{j=0}^{q} \left(\alpha_j \boldsymbol{y}_{k+j} - \Delta t \beta_j \frac{d\boldsymbol{y}_{k+j}}{dt} \right)}_{\mathcal{L}\boldsymbol{y}(t_k)}, \tag{56}$$

where we defined the linear difference operator $\mathcal{L}$ as

$$\mathcal{L}\boldsymbol{y}(s) = \sum_{j=0}^{q} \left(\alpha_j \boldsymbol{y}(s + j\Delta t) - \Delta t \beta_j \frac{d\boldsymbol{y}(s + j\Delta t)}{dt} \right). \tag{57}$$

Assuming that $\boldsymbol{y}(t)$ is differentiable with respect to t as many times as we need, we compute the Taylor series

$$\boldsymbol{y}(t_k + j\Delta t) = \boldsymbol{y}(t_k) + j\Delta t \frac{d\boldsymbol{y}(t_k)}{dt} + \frac{(j\Delta t)^2}{2} \frac{d^2\boldsymbol{y}(t_k)}{dt^2} + \cdots \tag{58}$$

$$\frac{d\boldsymbol{y}(t_k + j\Delta t)}{dt} = \frac{d\boldsymbol{y}(t_k)}{dt} + j\Delta t \frac{d^2\boldsymbol{y}(t_k)}{dt^2} + \frac{(j\Delta t)^2}{2} \frac{d^3\boldsymbol{y}(t_k)}{dt^3} + \cdots \tag{59}$$

A substitution of (58)-(59) into (57) yields the following Taylor series

$$\begin{aligned} \mathcal{L}\boldsymbol{y}(t_k) =& \sum_{j=0}^{q} \alpha_j \left(\boldsymbol{y}(t_k) + j\Delta t \frac{d\boldsymbol{y}(t_k)}{dt} + \cdots \right) - \Delta t \beta_j \left(\frac{d\boldsymbol{y}(t_k)}{dt} + j\Delta t \frac{d^2\boldsymbol{y}(t_k)}{dt^2} + \cdots \right) \\ =& C_0 \boldsymbol{y}(t_k) + C_1 \Delta t \frac{d\boldsymbol{y}(t_k)}{dt} + C_2 \Delta t^2 \frac{d^2\boldsymbol{y}(t_k)}{dt^2} + \cdots, \end{aligned} \tag{60}$$

where

$$C_0 = \sum_{j=0}^{q} \alpha_j = \rho(1) \tag{61}$$

$$C_1 = \sum_{j=0}^{q} (j\alpha_j - \beta_j) = \rho'(1) - \sigma(1) \tag{62}$$

$$\vdots$$

$$C_s = \frac{1}{s!} \sum_{j=0}^{q} \left(j^s \alpha_j - s j^{s-1} \beta_j \right) \qquad s = 2, 3, \ldots \tag{63}$$

Dividing $\mathcal{L}\boldsymbol{y}(t_k)$ by Δt (see (56)) finally yields the following series expansion of the truncation error

$$\boldsymbol{\tau}_{k+q} = \frac{C_0}{\Delta t} \boldsymbol{y}(t_k) + C_1 \frac{d\boldsymbol{y}(t_k)}{dt} + C_2 \Delta t \frac{d^2\boldsymbol{y}(t_k)}{dt^2} + C_3 \Delta t^2 \frac{d^3\boldsymbol{y}(t_k)}{dt^3} \cdots. \tag{64}$$

We have seen that a necessary condition for consistency is that $\rho(1) = 0$ (see Eq. (30)), and therefore $C_0 = 0$. For consistency we also need to have $C_1 = 0$ (otherwise the truncation error (64) does not go to zero as $\Delta t \to 0$). Clearly, if for a certain scheme the coefficients $C_1, \ldots, C_p$ are all zero and $C_{p+1} \neq 0$ then we see that the linear multistep method is *consistent with order p*. In fact, in this case we have (to leading order in Δt)

$$\|\boldsymbol{\tau}_{k+q}\| \leq C_{p+1} \Delta t^p \left\| \frac{d^{p+1}\boldsymbol{y}(t_k)}{dt^{p+1}} \right\| + O(\Delta t^{p+1}). \tag{65}$$

- ***Order of consistency of AB3***: The AB3 scheme can be written as

$$\boldsymbol{u}_{k+3} = \boldsymbol{u}_{k+2} + \frac{\Delta t}{12}\left(23\boldsymbol{f}_{k+2} - 16\boldsymbol{f}_{k+1} + 5\boldsymbol{f}_k\right). \tag{66}$$

The characteristic polynomials (52)-(53) associated with (66) are

$$\rho(z) = z^3 - z^2 \qquad \sigma(z) = \frac{23}{12}z^2 - \frac{4}{3}z + \frac{5}{12}. \tag{67}$$

By using (61)-(64) we obtain

$$C_0 = \rho(1) = 0 \tag{68}$$

$$C_1 = \rho'(1) - \sigma(1) = 1 - \frac{23}{12} + \frac{16}{12} - \frac{5}{12} = 0, \tag{69}$$

$$C_2 = \frac{1}{2}\left[3^2 - 2^2 - 2\left(-\frac{16}{12} + 2\frac{23}{12}\right)\right] = 0, \tag{70}$$

$$C_3 = \frac{1}{3}\left[3^3 - 2^3 - 3\left(-\frac{16}{12} + 2^2\frac{23}{12}\right)\right] = 0, \tag{71}$$

$$C_4 = \frac{1}{4}\left[3^4 - 2^4 - 4\left(-\frac{16}{12} + 2^3\frac{23}{12}\right)\right] = \frac{9}{4}. \tag{72}$$

Therefore AB3 is consistent with order 3.

- ***Order of consistency of BDF2***: The BDF2 scheme can be written as

$$\boldsymbol{u}_{k+2} - \frac{4}{3}\boldsymbol{u}_{k+1} + \frac{1}{3}\boldsymbol{u}_k = \frac{2}{3}\Delta t\boldsymbol{f}_{k+2}. \tag{73}$$

The characteristic polynomials (52)-(53) associated with (66) are

$$\rho(z) = z^2 - \frac{4}{3}z + \frac{1}{3} \qquad \sigma(z) = \frac{2}{3}z^2. \tag{74}$$

By using (61)-(64) we obtain

$$C_0 = \rho(1) = 0 \tag{75}$$

$$C_1 = \rho'(1) - \sigma(1) = \frac{2}{3} - \frac{2}{3} = 0, \tag{76}$$

$$C_2 = \frac{1}{2}\left[2^2 - 1^2\frac{4}{3} - 4\frac{2}{3})\right] = 0, \tag{77}$$

$$C_3 = \frac{1}{3}\left[2^3 - 1^3\frac{4}{3} - 12\frac{2}{3})\right] = -\frac{4}{9}. \tag{78}$$

Therefore BDF2 is consistent with order 2.

What is the maximum order of consistency attainable by a q-step linear method? The scheme (51) is fully determined by the by the $2q+1$ coefficients (recall that we set $\alpha_q = 1$)

$$\{\alpha_{q-1}, \ldots, \alpha_0, \beta_q, \ldots, \beta_0\}. \tag{79}$$

If the method is consistent with order p then we $p+1$ conditions (see Eq. (64))

$$C_0 = 0, \quad C_1 = 0, \quad \cdots, \quad C_p = 0. \tag{80}$$

By setting $2q + 1 = p + 1$ we see that

Theorem 1. The maximum order attainable by q-step linear method of the form (51) is

$$p = 2q \quad \text{(implicit LMM methods)}, \qquad p = 2q - 1 \quad \text{(explicit LMM methods)} \tag{81}$$

In particular, for Adams-Bashforth and Adams-Moulton methods we have the following result.

Theorem 2. The maximum order of consistency attainable by a q-step Adams-Bashforth method is $p = q$. Similarly, the maximum order of consistency attainable by a q-step Adams-Moulton method is $p = q + 1$

In fact, the condition $\alpha_q = -\alpha_{q-1}$ automatically guarantees that $\rho(1) = C_0 = 0$. Therefore a q-step Adams-Bashforth has q free parameters $\{\beta_0, \beta_1, \ldots, \beta_{q-1}\}$, which can be chosen to satisfy the conditions $C_1 = 0$, $C_2 = 0$ up to $C_p = 0$ (p equations). This implies that a q-step Adams-Bashforth method has maximal consistency order $p = q$.

As we will see in the next course note, Adams-Bashforth and Adams-Moulton methods are all zero-stable, and therefore $p = q$ and $p = q+1$ is actually the order of convergence of these methods. On the other hand, for general LMM methods, it is possible to prove that LMM methods with consistency order exceeding $p = q + 1$ (q odd) or $p = q + 2$ (q even) are all zero unstable. This result is known as *first Dahlquist barrier*.

Order of consistency of RK methods. The order of an RK method (like the order of any other method) can be determined by using a Taylor series analysis of truncation error (see the examples at the beginning of this note). On the other hand, if we are interested in developing an explicit or implicit RK method with a maximal consistency order, we can just expand the RK method in a Taylor series and then try to match as many powers of Δt as possible relative to a power series expansion of the exact solution. We have already seen one of such calculations when we derived the one-parameter family of explicit two-stage RK methods. Obtaining similar results for RK methods with a larger number of stages is quite cumbersome[4], and also yields surprising results. In general, it can be shown that:

Theorem 3. An s-stage explicit RK method cannot have order greater than s.

This theorem establishes an upper bound for the maximum order attainable by explicit RK methods. However, determining the maximum attainable order for a fixed number of stages is not a trivial problem. Order conditions similar to those derived for LMM methods, i.e., (61)-(63) can be derived for RK methods using Butcher's theory. for For instance, it can be shown that for the three-stage explicit RK method

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ c_2 & a_{21} & 0 & 0 \\ c_3 & a_{31} & a_{32} & 0 \\ \hline & b_1 & b_2 & b_3 \end{array}$$

[4]The Taylor series analysis can be simplified substantially by using by using Butcher's theory (see [1, §5.6]), which relies on graph techniques.

to be of order 3 the following *order conditions* need to be satisfied

$$\begin{cases} b_1 + b_2 + b_3 = 1 \\ b_2 c_2 + b_3 c_3 = \dfrac{1}{2} \\ b_2 c_2^2 + b_3 c_3^2 = \dfrac{1}{3} \\ b_3 a_{32} c_2 = \dfrac{1}{6} \end{cases} \tag{82}$$

The solution to this algebraic nonlinear system yields a one two-parameter family of solutions and two one-parameter families of solutions (see [1, p. 178]). A similar calculation performed on a five-stage explicit RK method yields a system of order conditions that has no solution (see [1, p. 181]). In other words:

Theorem 4. There exist no five-stage explicit RK method of order 5.

The following table summarizes the maximum order attainable by an *explicit* RK method with s stages:

order of consistency	1	2	3	4	5	6	7	8
minimum number of stages	1	2	3	4	6	7	9	11

Regarding *implicit* RK methods, the highest attainable order is $2s$ (Gauss-RK methods). Similarly, Gauss-Radau and Gauss-Lobatto RK methods can attain consistency order $2s-1$ and $2s-2$, respectively.

References

[1] J. D. Lambert. *Numerical methods for ordinary differential systems: the initial value problem*. Wiley, 1991.

Stability and convergence of numerical methods for ODEs

Consider the initial value problem for a system of ODEs

$$\begin{cases} \dfrac{d\boldsymbol{y}}{dt} = \boldsymbol{f}(\boldsymbol{y}, t) \\ \boldsymbol{y}(0) = \boldsymbol{y}_0 \end{cases} \tag{1}$$

and the perturbed problem

$$\begin{cases} \dfrac{d\boldsymbol{z}}{dt} = \boldsymbol{f}(\boldsymbol{z}, t) + \boldsymbol{\delta}(t) \\ \boldsymbol{z}(0) = \boldsymbol{y}_0 + \boldsymbol{\delta}_0 \end{cases} \tag{2}$$

where $\boldsymbol{\delta}(t)$ is an integrable function and $\boldsymbol{\delta}_0 \in \mathbb{R}^n$. Note that we replaced $\boldsymbol{y}(t)$ with $\boldsymbol{z}(t)$ in (2) to emphasize the fact that the solutions to (1) and (2) are (in general) different.

Definition 1 (Stability of the Cauchy problem (see [1, 3]))**.** The Cauchy problem (1) is said to be *stable* within the time interval $[0, T]$ if for any perturbations $\boldsymbol{\delta}_0$ and $\boldsymbol{\delta}(t)$ such that[1]

$$\|\boldsymbol{\delta}_0\| \leq \epsilon, \quad \text{and} \quad \|\boldsymbol{\delta}(t)\| \leq \epsilon \quad \text{for all } t \in [0, T] \tag{3}$$

we have that

$$\|\boldsymbol{z}(t) - \boldsymbol{y}(t)\| \leq C\epsilon, \quad \text{for all } t \in [0, T], \tag{4}$$

where C is a finite constant that does not depend on ϵ.

Based on this definition, the Cauchy problem (1) is "stable" if the difference between the solutions of (1) and (2) is bounded in $[0, T]$ after we introduce a small perturbation $\boldsymbol{\delta}_0$ in the initial condition $\boldsymbol{y}_0$ and a perturbation $\boldsymbol{\delta}(t)$ in $\boldsymbol{f}(\boldsymbol{y}, t)$. The definition of stability also implies that the difference between the solutions of (1) and (2) goes to zero as $\epsilon \to 0$. In fact, from (4) it follows that

$$\lim_{\epsilon \to 0} \|\boldsymbol{z}(t) - \boldsymbol{y}(t)\| \leq C \lim_{\epsilon \to 0} \epsilon = 0. \tag{5}$$

The constant C appearing in (4) may not be small. This is consistent with the fact that a small perturbations in the Cauchy problem (1) can introduce large perturbations in its solution.

Hereafter we show that any well-posed initial value problem (1) is stable, i.e., robust to perturbations in the limit of small perturbations.

Theorem 1. Let $D \subseteq \mathbb{R}^n$ be an open set, $\boldsymbol{y}_0 \in D$. If $\boldsymbol{f}(\boldsymbol{y}, t)$ is Lipschitz continuous in D and $\boldsymbol{\delta}(t)$ is integrable then the initial value problem (1) is stable.

Proof. We need to show that for any $\boldsymbol{\delta}_0$ and $\boldsymbol{\delta}(t)$ the difference between the solutions of (1) and (2) is bounded in some time interval $[0, T]$ and that the difference goes to zero as we send ϵ to zero. First of all, we notice that if D is open and ϵ is small enough then the initial condition $(\boldsymbol{y}_0 + \boldsymbol{\delta}_0)$ is in D. If $\boldsymbol{f}(\boldsymbol{y}, t)$ is Lipschitz continuous

$$\|\boldsymbol{f}(\boldsymbol{z}, t) - \boldsymbol{f}(\boldsymbol{y}, t)\| \leq L \|\boldsymbol{z} - \boldsymbol{y}\| \quad \forall \boldsymbol{z}, \boldsymbol{y} \in D, \tag{6}$$

[1] Note that in (3) we are bounding $\boldsymbol{\delta}_0$ and $\boldsymbol{\delta}(t)$ with the same constant ϵ. Such a constant coincides with the radius of the largest ball centered at zero in $\mathbb{R}^n$ that includes both $\boldsymbol{\delta}_0$ and $\boldsymbol{\delta}(t)$ (for all $t \in [0, T]$).

and $\boldsymbol{\delta}(t)$ is integrable, then we have existence and uniqueness of the solution to both problems (1) and (2). Such problems can be equivalently written as

$$\boldsymbol{y}(t) = \boldsymbol{y}_0 + \int_0^t \boldsymbol{f}(\boldsymbol{y}(s), s)ds \tag{7}$$

$$\boldsymbol{z}(t) = \boldsymbol{y}_0 + \boldsymbol{\delta}_0 + \int_0^t \boldsymbol{f}(\boldsymbol{z}(s), s)ds + \int_0^t \boldsymbol{\delta}(s)ds \tag{8}$$

for all $t \in [0, T]$, where T is is the smallest "exit time", i.e., the time in which either $\boldsymbol{y}(t)$ or $\boldsymbol{z}(t)$ get out of D. Subtracting (7) from (8) and taking the norm yields

$$\begin{aligned}\|\boldsymbol{z}(t) - \boldsymbol{y}(t)\| =& \left\| \boldsymbol{\delta}_0 + \int_0^t \left[\boldsymbol{f}(\boldsymbol{z}(s), s) - \boldsymbol{f}(\boldsymbol{y}(s), s)\right] ds + \int_0^t \boldsymbol{\delta}(s)ds \right\| \\ \leq& \|\boldsymbol{\delta}_0\| + \int_0^t \|\boldsymbol{f}(\boldsymbol{z}(s), s) - \boldsymbol{f}(\boldsymbol{y}(s), s)\| \, ds + \int_0^t \|\boldsymbol{\delta}(s)\| \, ds \\ \leq& (1+t)\epsilon + L \int_0^t \|\boldsymbol{z}(s) - \boldsymbol{y}(s)\| \, ds \end{aligned} \tag{9}$$

where we used the triangle inequality, the inequalities (3), and the definition of Lipschitz continuity (6). At this point we use Grönwall's inequality[2] to conclude that

$$\begin{aligned}\|\boldsymbol{z}(t) - \boldsymbol{y}(t)\| \leq& (1+t)e^{tL}\epsilon \\ \leq& \underbrace{(1+T)e^{TL}}_{C} \epsilon. \end{aligned} \tag{13}$$

This proves that the Cauchy problem (1) is stable. Note that the constant C appearing in (13) does not depend on ϵ and it can be very big, depending on the Lipschitz constant L and the integration time T (integration time).

□

Remark: If we replace the initial value problem (1) by a numerical scheme we introduce errors that can accumulate in time. Such errors can be considered as perturbations in the ODE (1). Just think about representing the discrete numerical solution $\boldsymbol{u}_k$ in terms of some local interpolant $\boldsymbol{z}(t)$ (differentiable in time) and substituting it into (1). This yields an ODE in the form (2). If the initial value problem (1) is not stable, i.e., robust to small perturbations, then there is no hope for any numerical method to approximate the solution.

Stability and zero-stability of numerical methods for ODEs. The concept of stability we discussed in previous section for continuous-time dynamical systems can be extended to discrete-time dynamical

[2]The Grönwall's inequality (see [3, Lemma 11.1]) can be stated as follows. Let $u(t)$, $g(t)$ and $p(t)$ such that

$$u(t) \leq g(t) + \int_0^t p(s)u(s)ds. \tag{10}$$

If $g(t)$ is non-decreasing and $p(s)$ is strictly positive then

$$u(t) \leq g(t) \exp\left[\int_0^t p(s)u(s)ds\right]. \tag{11}$$

In the case of equation (9) we have

$$u(t) = \|\boldsymbol{z}(t) - \boldsymbol{y}(t)\|, \qquad g(t) = (1+t)\epsilon, \qquad p(s) = L. \tag{12}$$

systems, i.e., to numerical schemes aiming at computing the approximate solution of the initial value problem (1). we have seen that such schemes can be written in the general form[3]

$$\begin{cases} \displaystyle\sum_{j=0}^{q} \alpha_j \boldsymbol{u}_{k+j} = \Delta t \boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{u}_{k+q}, \ldots, \boldsymbol{u}_k, t_k, \Delta t), \\ \text{given} \quad \{\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{q-1}\} \end{cases} \tag{14}$$

where $\Phi_{\boldsymbol{f}}$ is some iteration function. By taking a perturbation of the initial condition $\boldsymbol{u}_0$ and a "time-dependent" perturbation of $\Phi_{\boldsymbol{f}}$ in (14) we obtain

$$\begin{cases} \displaystyle\sum_{j=0}^{q} \alpha_j \boldsymbol{z}_{k+j} = \Delta t \left[\boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{z}_{k+q}, \ldots, \boldsymbol{z}_k, t_k, \Delta t) + \boldsymbol{\delta}_{k+q}\right], \\ \text{given} \quad \{\boldsymbol{z}_0 = \boldsymbol{u}_0 + \boldsymbol{\delta}_0, \ldots, \boldsymbol{z}_{q-1} = \boldsymbol{u}_{q-1} + \boldsymbol{\delta}_{q-1}\} \end{cases} \tag{15}$$

where $\{\boldsymbol{\delta}_0, \ldots, \boldsymbol{\delta}_{k+q}, \ldots\}$ is a sequence of vectors in $\mathbb{R}^n$ bounded by some constant ϵ, i.e.,

$$\|\boldsymbol{\delta}_j\| \leq \epsilon \quad \text{for all} \quad j = 0, 1, \ldots. \tag{16}$$

The perturbations $\boldsymbol{\delta}_j$ can arise, e.g., because of round-off or truncation errors when performing floating point operations using double precision arithmetic. Clearly, the orbits generated by (14) and (15), i.e.,

$$\{\boldsymbol{u}_0, \ldots, \boldsymbol{u}_N\} \quad \text{and} \quad \{\boldsymbol{z}_0, \ldots, \boldsymbol{z}_N\} \tag{17}$$

are (in general) different. For any given iteration function $\boldsymbol{\Phi}_{\boldsymbol{f}}$ and any given Δt we can provide a definition of stability for the numerical scheme (14) that closely resembles Definition 1 for continuous time dynamical systems. To this end, let T be the period of integration, and N be the number of time steps, i.e.,

$$\Delta t^* = \frac{T}{N} \tag{18}$$

Of particular interest when performing convergence analysis, is the behavior of the scheme for small Δt, i.e., for all Δt smaller than Δt^*.

Definition 2 (Zero-stability)**.** We say that the numerical scheme (14) is *zero-stable* if there exists a $\Delta t^* > 0$ such that *for all* $\Delta t \leq \Delta t^*$ and for any perturbations $\boldsymbol{\delta}_j$ $(j = 0, \ldots, N)$ such that

$$\|\boldsymbol{\delta}_k\| \leq \epsilon, \quad \text{for all } j = 0, \ldots, N \tag{19}$$

we have that

$$\|\boldsymbol{z}_k - \boldsymbol{u}_k\| \leq C\epsilon, \quad \text{for all } j = 0, \ldots, N, \tag{20}$$

where $\boldsymbol{u}_k$ and $\boldsymbol{z}_k$ are defined by (14) and (15), and C is a finite constant that does not depend on ϵ[4].

The definition "zero-stability" follows from the fact that we require $\|\boldsymbol{z}_k - \boldsymbol{u}_k\| \leq C\epsilon$ for all $\Delta t \leq \Delta t^*$, and in particular for $\Delta t \to 0$. Hence the "zero" part in "zero-stability" refers to the stability of the scheme in the limit $\Delta t \to 0$.

- Zero stability is a property of the numerical scheme, not of the ODE system (1). We have seen, in fact, that a well-posed Cauchy problem is always stable.

[3] In (14) we set $\alpha_q = 1$ to remove the non-uniqueness of α_j and β_j due to possible rescaling by a constant.
[4] The constant C in (20) may depend also on T, Δt or other constants, but it cannot depend on k or ϵ.

- Numerical methods that are not zero-stable have no hope to reliably approximate the solution of (1). In fact, even if the method is consistent, i.e., if the truncation error goes to zero as $\Delta t \to 0$, we have that perturbations due to finite-arithmetic may rapidly propagate in schemes that are not zero-stable, and therefore generate instabilities. In other words, consistent schemes that are not zero-stable may not converge as $\Delta t \to 0$. For example, the numerical scheme in equation (32) hereafter is consistent but not zero stable. Another example of a consistent scheme that is not zero stable is discussed in [2, p. 32].

The root condition and zero-stability. The numerical method (14) is said to satisfy the *root condition* if all roots of the first characteristic polynomial

$$\rho(z) = \sum_{j=0}^{q} \alpha_j z^j \tag{21}$$

are within the unit circle, and those of modulus one (i.e., the ones on the unit circle) are simple. The following fundamental theorem relates zero stability of the numerical method (14) to the root condition.

Theorem 2. The numerical method (14) is zero-stable if and only if it satisfies the root condition.

A detailed proof of this theorem is provided at the end of this note[5]. Recall that a necessary condition for consistency is that $\rho(1) = 0$, i.e., $z = 1$ is a root of (21). Such a root must be simple in order for the method to satisfy the root condition. Let us now study zero-stability of all schemes we have considered so far.

- ***One-step methods:*** The most general form of a one-step method is

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \Delta t \Phi_{\boldsymbol{f}}(\boldsymbol{u}_{k+1}, \boldsymbol{u}_k, t_k, \Delta t). \tag{22}$$

  The characteristic polynomial for this class of methods is

$$\rho(z) = z - 1 \tag{23}$$

  Clearly, $\rho(z)$ has a simple root at $z = 1$ and therefore (22) satisfies the root condition. This implies that *all one-step methods are zero-stable*. Recall that all Runge-Kutta methods are one-step methods.

- ***Adams-Bashforth and Adams-Moulton methods:*** A q-step Adams method can be written in the general form

$$\boldsymbol{u}_{k+q} = \boldsymbol{u}_{k+q-1} + \Delta t \sum_{j=0}^{q} \beta_j \boldsymbol{f}\left(\boldsymbol{u}_{k+j}, t_{k+j}\right). \tag{24}$$

  For Adams-Bashforth methods (explicit) we have $\beta_q = 0$; for Adams-Moulton (implicit) $\beta_q \neq 0$. The characteristic polynomial associated with (24) is

$$\rho(z) = z^q - z^{q-1} = z^{q-1}(z-1). \tag{25}$$

  This polynomial has as a simple root at $z = 1$ and a root with algebraic multiplicity $q - 1$ at $z = 0$. Therefore it satisfies the root condition. By Theorem (2) we have that *all Adams-Bashforth and all Adams-Moulton methods are zero-stable.*

[5] For whatever reason, none of the books I came across in my career provides concise and direct proof of Theorem 2 in the general case we are considering here, i.e., for vector-valued ODEs and numerical methods of the form (14). Hence, I decided to provide my own version of the proof.

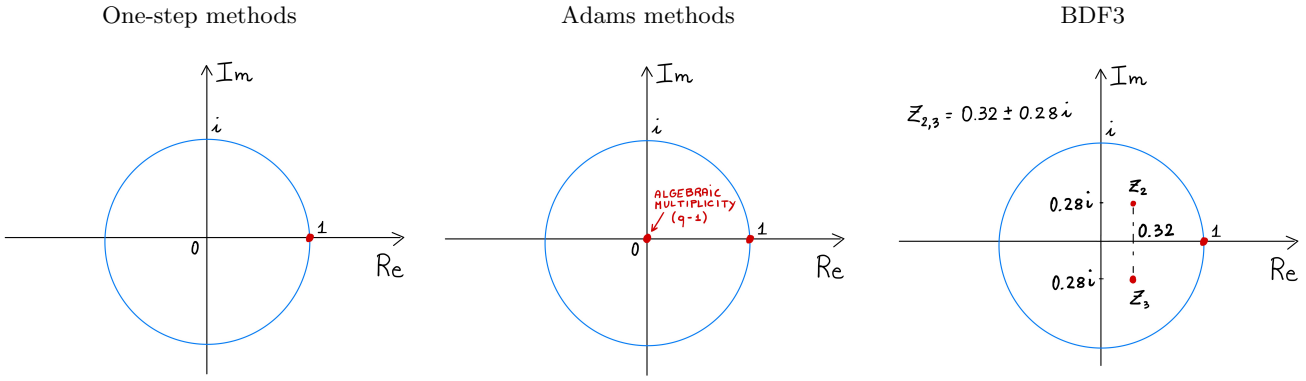


Figure 1: Roots of the characteristic polynomial (21). If all roots are within the unit circle and those modulus one (i.e., the ones on the unit circle) are simple (i.e., they have algebraic multiplicity one) then the method is zero-stable. All methods sketched in this figure are zero-stable.

- ***BDF methods:*** We know that a q-step BDF method can be written in the form

$$\sum_{j=0}^{q} \alpha_j \boldsymbol{u}_{k+j} = c\Delta t \boldsymbol{f}(\boldsymbol{u}_{k+q}, t_{k+q}). \tag{26}$$

  The characteristic polynomial associated with (26) is

$$\rho(z) = z^q + \alpha_{q-1} z^{q-1} + \cdots \alpha_0. \tag{27}$$

  It can be shown that a q-step BDF method satisfies the root condition and therefore it is zero-stable if and only if $q \leq 6$.

- ***2-step midpoint method:*** The 2-step midpoint method

$$\boldsymbol{u}_{k+2} = \boldsymbol{u}_k + 2\Delta t \boldsymbol{f}(\boldsymbol{u}_{k+1}, t_{k+1}) \tag{28}$$

  satisfies the root condition and therefore it is zero-stable. In fact, the characteristic polynomial associated with (28) is

$$\rho(z) = z^2 - 1. \tag{29}$$

  The roots $z = \pm 1$ are both simple and sitting at the boundary of the unit circle in the complex plane. As we will see, a scheme that satisfies the root condition with simple eigenvalues at boundary of the unit circle is theoretically zero-stable, but in practical applications it can generate instabilities.

- ***2-step LMM method:*** The following two-step explicit linear multi-step method[6]

$$\boldsymbol{u}_{k+2} - 4\boldsymbol{u}_{k+1} + 3\boldsymbol{u}_k = -2\Delta t \boldsymbol{f}(\boldsymbol{u}_k, t_k) \tag{32}$$

  is consistent but *not zero-stable*. The characteristic polynomial is

$$\rho(z) = z^2 - 4z + 3. \tag{33}$$

[6]The method (32) is *not* a BDF method, and is obtained by approximating $d\boldsymbol{y}(t_k)/dt$ with a second-order forward finite difference formula:

$$\frac{d\boldsymbol{y}(t_k)}{dt} \simeq \frac{-3\boldsymbol{y}(t_k) + 4\boldsymbol{y}(t_k + 1) - \boldsymbol{y}(t_{k+2})}{2\Delta t}. \tag{30}$$

and setting the equality

$$\frac{-3\boldsymbol{u}_k + 4\boldsymbol{u}_{k+1} - \boldsymbol{u}_{k+2}}{2\Delta t} = f(\boldsymbol{u}_k, t_k). \tag{31}$$

Consistency can be checked immediately, since (see course note number 3)

$$\rho(1) = 0 \qquad \frac{\boldsymbol{\Phi}(\boldsymbol{u}_k, t_k, 0)}{\rho'(z)} = f(\boldsymbol{u}_k, t_k). \tag{34}$$

The polynomial (33) has roots $z = 1$ and $z = 3$. Therefore the method (32) is not zero-stable.

- ***General LMM methods:*** We have seen in the course note 3 that the maximal order of consistency of a linear q-step method of the form

$$\sum_{j=0}^{q} \alpha_j \boldsymbol{u}_{k+j} = \Delta t \sum_{j=0}^{q} \beta_j \boldsymbol{f}(\boldsymbol{u}_{k+j}, t_{k+1}), \tag{35}$$

is $2q$ (implicit methods) or $2q-1$ (explicit methods). At this point we notice that such maximal order LMM methods are, in general, *zero-unstable*, i.e., they do not satisfy the root-condition (see [2, §3.4]). It fact the following theorem holds true.

Theorem 3 (First Dahlquist barrier - 1956)**.** There is no zero-stable linear q-step method with consistency order exceeding $q+1$ (q odd) or $q+2$ (q even).

Zero-stable linear q-step implicit methods with order $q+2$ are called *optimal*. These methods have all roots with algebraic multiplicity one sitting on the boundary of the unit circle. This can yields stability issues.

Convergence. Let $T = N\Delta t$ be period of integration. We say that the scheme (14) is convergent if the error (in any norm)

$$\max_{k \in 0,\dots,N} \|\boldsymbol{u}_k - \boldsymbol{y}_k\| \tag{36}$$

goes to zero as $\Delta t \to 0$. Here $\boldsymbol{y}_k = \boldsymbol{y}(t_k)$ represents the analytical solution of the ODE system (1) evaluated at $t = t_k$, while $\boldsymbol{u}_k$ is the numerical solution produced by the scheme (14). If the error decreases as Δt^p then we say that the scheme converges with order p.

If a numerical scheme is convergent then the order of convergence is the same as the order of consistency (see the proof of theorem 4 at the end of this note). Indeed the error (36) can be bounded by the norm of the global truncation error, which goes to zero to some order in Δt (if the scheme is consistent) The following fundamental theorem provides necessary and sufficient conditions for convergence of numerical method for a system of ODEs.

Theorem 4 (Convergence)**.** The numerical method (14) is convergent if and only if is consistent and zero-stable. In other words,

$$\text{convergence} \;\Leftrightarrow\; \text{consistency} \;+\; \text{zero stability}. \tag{37}$$

Moreover, the convergence order coincides with the consistency order.

The proof of this theorem follows exactly the same steps as the proof of theorem 2, and it is briefly discussed at the end of this note. This theorem has several corollaries. For instance, we have just seen that all one-step methods are zero-stable and therefore we have that:

Corollary 1. A one-step method is convergent if and only if it is consistent.

This means that in order to prove convergence of a one-step method it is necessary and sufficient to prove consistency. Hence, in the case of RK methods a necessary and sufficient condition for convergence is

$$\sum_{i=1}^{s} b_i = 1. \tag{38}$$

Corollary 2. Adams methods are convergent if and only if they are consistent.

In fact, we have seen that Adams methods are always zero-stable and therefore consistency implies convergence. Recall that Adams-Bashforth and Adams-Moulton methods are consistent if and only if

$$\sum_{j=0}^{q} \beta_j = 1. \tag{39}$$

Hence, if (39) is satisfied then Adams-Bashforth ($\beta_q = 0$) or Adams-Moulton ($\beta_q \neq 0$) methods are convergent.

Example: The numerical scheme (32) is not convergent. In fact, it is consistent, but not zero-stable.

Estimating the convergence order of a numerical method. To estimate the convergence order of the scheme (14) numerically it is sufficient to compute the error $\|\boldsymbol{y}(t_k) - \boldsymbol{u}_k\|$ (in any norm) relative to an analytical solution $\boldsymbol{y}(t)$ for various (sufficiently small) Δt, and then plot

$$\max_{k=1,\ldots,N} \|\boldsymbol{y}(t_k) - \boldsymbol{u}_k\|$$

versus Δt in a logarithmic scale. The slope of the line obtained in this way represents the order of the method. In fact, suppose that for sufficiently small Δt we have

$$\max_{k=1,\ldots,N} \|\boldsymbol{y}(t_k) - \boldsymbol{u}_k\| \simeq C\Delta t^p. \tag{40}$$

Taking the logarithm yields

$$\log\left(\max_{k=1,\ldots,N} \|\boldsymbol{y}(t_k) - \boldsymbol{u}_k\|\right) \simeq \log(C) + p\log(\Delta t) \tag{41}$$

which represents a line with slope p in a log-log plot. To compute the error, we need of course the analytical solution to the initial value problem (1), which is not always available. However, it is very easy to manufacture an ODE with a time-dependent right hand side that has any desired solution $\boldsymbol{y}(t)$. To this end, choose any continuously differentiable vector $\boldsymbol{y}(t)$ and any Lipschitz continuous function $\boldsymbol{f}(\boldsymbol{y})$. Compute the time forcing term

$$\boldsymbol{h}(t) = \frac{d\boldsymbol{y}(t)}{dt} - \boldsymbol{f}(\boldsymbol{y}(t)). \tag{42}$$

Then the chosen $\boldsymbol{y}(t)$ is the analytical solution to the initial value problem

$$\begin{cases} \dfrac{d\boldsymbol{y}}{dt} = \boldsymbol{f}(\boldsymbol{y}) + \boldsymbol{h}(t) \\ \boldsymbol{y}(0) = \boldsymbol{y}_0 \end{cases} \tag{43}$$

In this way, for each given Δt we can solve (43) using the numerical method (14) and compute the error $\|\boldsymbol{u}_k - \boldsymbol{y}(t_k)\|$.

Proof of Theorem 2. Let us consider the m-th component of the perturbed scheme (14)

$$\sum_{j=0}^{q} \alpha_j z_{k+j}^m = \Delta t \left[\Phi_{\boldsymbol{f}}^m(\boldsymbol{z}_{k+q}, \ldots, \boldsymbol{z}_k, t_k, \Delta t) + \delta_{q+k}^m \right]. \tag{44}$$

and the unperturbed one

$$\sum_{j=0}^{q} \alpha_j u_{k+j}^m = \Delta t \Phi_{\boldsymbol{f}}^m(\boldsymbol{u}_{k+q}, \ldots, \boldsymbol{u}_k, t_k, \Delta t). \tag{45}$$

Subtracting (44) from (44) yields

$$\sum_{j=0}^{q} \alpha_j e_{k+j}^m = \Delta t \left[\Phi_{\boldsymbol{f}}^m(\boldsymbol{z}_{k+q}, \ldots, \boldsymbol{z}_k, t_k, \Delta t) - \Phi_{\boldsymbol{f}}^m(\boldsymbol{u}_{k+q}, \ldots, \boldsymbol{u}_k, t_k, \Delta t) + \delta_{q+k}^m \right], \tag{46}$$

where

$$e_{k+j}^m = z_{k+j}^m - u_{k+j}^m. \tag{47}$$

Upon definition of

$$\boldsymbol{e}_k^m = \begin{bmatrix} e_k^m \\ e_{k+1}^m \\ \vdots \\ e_{k+q-1}^m \end{bmatrix}, \quad \boldsymbol{b}_k^m = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \Phi_{\boldsymbol{f}}^m(\boldsymbol{z}_{k+q}, \ldots, \boldsymbol{z}_k, t_k, \Delta t) - \Phi_{\boldsymbol{f}}^m(\boldsymbol{u}_{k+q}, \ldots, \boldsymbol{u}_k, t_k, \Delta t) \end{bmatrix}, \quad \boldsymbol{d}_k^m = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \delta_{q+k}^m \end{bmatrix} \tag{48}$$

we see that we can write (46) in a compact form as as[7]

$$\boldsymbol{e}_{k+1}^m = \boldsymbol{A}\boldsymbol{e}_k^m + \Delta t \left(\boldsymbol{b}_k^m + \boldsymbol{d}_k^m \right), \tag{49}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & \cdots & -\alpha_{q-1} \end{bmatrix}. \tag{50}$$

By using the discrete variation of constant formula (in which we treat $\Delta t \left(\boldsymbol{b}_k^m + \boldsymbol{d}_k^m \right)$ as a "forcing term") we write the formal solution of (49) as

$$\boldsymbol{e}_{k+1}^m = \boldsymbol{A}^{k+1} \boldsymbol{e}_0^m + \Delta t \sum_{p=0}^{k} \boldsymbol{A}^{k-p} \left(\boldsymbol{b}_p^m + \boldsymbol{d}_p^m \right), \tag{51}$$

As we shall see hereafter, the zero-stability of the numerical scheme is essentially determined by the properties of the matrix $\boldsymbol{A}$, in particular by the behavior of the matrix powers $\boldsymbol{A}^k$ as k is increased. If the norm of the matrix powers can be bounded by a constant that is independent of k then zero-stability follows rather straightforwardly. The properties of the matrix powers $\boldsymbol{A}^k$ are fully determined by the roots of the first characteristic polynomial (21).

[7] Recall that $\alpha_q = 1$.

Lemma 1. Let $\|\cdot\|$ be any matrix norm compatible with a vector norm. Then $\left\|\boldsymbol{A}^k\right\|$ can be bounded by a quantity M that does not depend on k, i.e.,

$$\left\|\boldsymbol{A}^k\right\| \leq M \quad \text{for all } k \in \mathbb{N} \tag{52}$$

if and only if the root condition is satisfied.

Proof. The matrix $\boldsymbol{A}$ in (50) is the transpose of the companion matrix associated with the characteristic polynomial (21). This means that the eigenvalues of $\boldsymbol{A}$ coincide with the roots of the polynomial (21). Moreover, companion matrices are *non-derogatory*, which means that there exists only one eigenvector corresponding to each eigenvalue λ. Such eigenvector is explicitly obtained as

$$\boldsymbol{h} = \begin{bmatrix} 1 \\ \lambda \\ \lambda^2 \\ \vdots \\ \lambda^{q-1} \end{bmatrix}. \tag{53}$$

The non-derogatory property of $\boldsymbol{A}$ implies that if there exists any eigenvalue with algebraic multiplicity $r_j > 1$, then the corresponding eigenspace has dimension $r_j - 1$. This means that the matrix $\boldsymbol{A}$ is diagonalizable (i.e., similar to a diagonal matrix), if and only of all the eigenvalues are simple. If there exist any eigenvalue with multiplicity larger than one then the matrix $\boldsymbol{A}$ is similar to a (block-diagonal) Jordan matrix $\boldsymbol{J}$

$$\boldsymbol{A} = \boldsymbol{P}\boldsymbol{J}\boldsymbol{P}^{-1} \tag{54}$$

where $\boldsymbol{P}$ is the matrix that has the generalized eigenvectors of $\boldsymbol{A}$ columnwise and

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{J}_l \end{bmatrix}, \qquad \boldsymbol{J}_i = \lambda_i \boldsymbol{I}_{r_i} + \boldsymbol{B}_{r_i}. \tag{55}$$

In this equation, $\boldsymbol{J}_i$ denotes the Jordan block corresponding to the eigenvalue λ_i (which has algebraic multiplicity r_i), $\boldsymbol{I}_{r_i}$ is a $r_i \times r_i$ identity matrix and $\boldsymbol{B}_{r_i}$ is a $r_i \times r_i$ matrix with ones above the main diagonal. For instance, if λ_i has algebraic multiplicity $r_i = 3$ then the geometric multiplicity is 2 and we have

$$\boldsymbol{J}_i = \begin{bmatrix} \lambda_i & 1 & 0 \\ 0 & \lambda_i & 1 \\ 0 & 0 & \lambda_i \end{bmatrix}, \qquad \boldsymbol{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \boldsymbol{B}_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}. \tag{56}$$

The matrix power $\boldsymbol{A}^k$ can be written as

$$\boldsymbol{A}^k = \boldsymbol{P}\boldsymbol{J}^k\boldsymbol{P}^{-1}, \tag{57}$$

where

$$\boldsymbol{J}^k = \begin{bmatrix} \boldsymbol{J}_1^k & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J}_2^k & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{J}_l^k \end{bmatrix}, \qquad \boldsymbol{J}_i^k = (\lambda_i \boldsymbol{I}_{r_i} + \boldsymbol{B}_{r_i})^k \tag{58}$$

Let us compute J_i^k for $r_i = 1$ (simple eigenvalue)

$$J_i^k = \lambda_i^k. \tag{59}$$

On the other hand, for $r_i = 2$, (eigenvalue with algebraic multiplicity 2 and geometric multiplicity 2) we have have

$$J_i = \lambda_i \boldsymbol{I}_2 + \boldsymbol{B}_2, \tag{60}$$

$$J_i^2 = (\lambda_i \boldsymbol{I}_2 + \boldsymbol{B}_2)^2 = \lambda_i^2 \boldsymbol{I}_2 + 2\lambda_i \boldsymbol{B}_2, \tag{61}$$

$$\vdots$$

$$J_i^k = (\lambda_i \boldsymbol{I}_2 + \boldsymbol{B}_2)^k = \lambda_i^k \boldsymbol{I}_2 + k\lambda_i^{k-1} \boldsymbol{B}_2, \tag{62}$$

where we used the fact that $\boldsymbol{B}_2^k = 0$ for all $\boldsymbol{k} \geq 2$. Similarly, for $r_i = 3$ it can be shown that $\boldsymbol{B}_3^k = 0$ for all $k \geq 3$, which yields

$$J_i^k = \lambda_i^k \boldsymbol{I}_3 + k\lambda_i^{k-1} \boldsymbol{B}_3 + k\lambda_i^{k-2} \boldsymbol{B}_3^2 \qquad k \geq 3. \tag{63}$$

By taking the norm of (57) we obtain

$$\left\| \boldsymbol{A}^k \right\| \leq K \left\| \boldsymbol{J}^k \right\|, \qquad K = \|\boldsymbol{P}\| \left\| \boldsymbol{P}^{-1} \right\|. \tag{64}$$

At this point we recall that for any matrix norm compatible with a vector norm and for any block-diagonal matrix such as $\boldsymbol{J}$ or $\boldsymbol{J}^k$ we have

$$\left\| \boldsymbol{J}^k \right\| = \max \left\{ \left\| \boldsymbol{J}_1^k \right\|, \ldots, \left\| \boldsymbol{J}_l^k \right\| \right\}. \tag{65}$$

If the eigenvalues of $\boldsymbol{A}$ sitting at the boundary of the unit circle are *simple* then, by equation (59) we have

$$\left\| \boldsymbol{J}_i^k \right\| = 1. \tag{66}$$

On the other hand, if $|\lambda_i| < 1$ (eigenvalue within the unit circle or arbitrary multiplicity) then by equation (62) or (63) we have that

$$\left\| \boldsymbol{J}_i^k \right\| \to 0 \quad \text{for } k \to \infty. \tag{67}$$

Since $\left\| \boldsymbol{J}_i^k \right\|$ if finite for all k, there exists a finite M such that $\left\| \boldsymbol{J}_i^k \right\| \leq M$ for all k.

Finally, if there exists a non-simple eigenvalue λ_i (eigenvalue with algebraic multiplicity larger than one) at the boundary of the unit circle then we can no longer guarantee that $\left\| \boldsymbol{J}_i^k \right\|$ is bounded independently of k. In fact, suppose that the algebraic multiplicity of the eigenvalue λ_i at the boundary of the unit circle (i.e., $|\lambda_i| = 1$) is $r_i = 2$. Then by using (62) we see that

$$\left\| \boldsymbol{J}_i^k \right\|_1 = |\lambda_i|^k + k\,|\lambda_i|^{k-1} = 1 + k \qquad \text{for all } k \geq 2. \tag{68}$$

In summary, if the root condition is satisfied, i.e., if all the eigenvalues of $\boldsymbol{A}$ are within the unit circle with the exception of a finite number of *simple* eigenvalues sitting at the boundary of the unit circle then

$$\left\| \boldsymbol{A}^k \right\| \leq K \left\| \boldsymbol{J}^k \right\| \leq M \qquad \text{for all } k \in \mathbb{N}, \tag{69}$$

where $M > 0$ is independent of k. This completes the proof of Lemma 1.

□

We now have all elements to show that if a scheme satisfies the root condition then it is zero-stable. To this end, let us take the infinity norm of (51), and use (52) (or (69)) to obtain

$$\left| e_{k+q}^m \right| \leq \left\| \boldsymbol{e}_{k+1}^m \right\|_\infty \leq M \left(\left\| \boldsymbol{e}_0^m \right\|_\infty + \Delta t \sum_{p=0}^{k} \left\| \boldsymbol{b}_p^m \right\|_\infty + \Delta t \sum_{p=0}^{k} \left\| \boldsymbol{d}_p^m \right\|_\infty \right). \tag{70}$$

By using definition (48) and (47) we see that

$$\begin{aligned}\sum_{m=1}^{n}\left\|\boldsymbol{b}_p^m\right\|_\infty &= \sum_{m=1}^{n}\left|\Phi_{\boldsymbol{f}}^m(\boldsymbol{z}_{p+q},\ldots,\boldsymbol{z}_p,t_p,\Delta t)-\Phi_{\boldsymbol{f}}^m(\boldsymbol{u}_{p+q},\ldots,\boldsymbol{u}_p,t_p,\Delta t)\right| \\ &= \left\|\boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{z}_{p+q},\ldots,\boldsymbol{z}_p,t_p,\Delta t)-\boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{u}_{p+q},\ldots,\boldsymbol{u}_p,t_p,\Delta t)\right\|_1 \\ &\leq L\sum_{s=0}^{q}\left\|\boldsymbol{z}_{p+s}-\boldsymbol{u}_{p+s}\right\|_1 \\ &= L\sum_{s=0}^{q}\sum_{m=1}^{n}\left|z_{p+s}^m-u_{p+s}^m\right| \\ &= L\sum_{s=0}^{q}\sum_{m=1}^{n}\left|e_{p+s}^m\right|, \end{aligned} \tag{71}$$

where we assumed that $\boldsymbol{\Phi}_{\boldsymbol{f}}$ is Lischitz continuous. Next, define

$$g_{k+q}=\sum_{m=1}^{n}\left|e_{k+q}^m\right|. \tag{72}$$

Note that g_{k+q} is the 1-norm of the vector $\boldsymbol{z}_{k+q}-\boldsymbol{u}_{k+q}$ (see Eq. (47)). Substituting (71) into the inequality (70) (summed-up in m) yields

$$\begin{aligned} g_{k+q} &\leq M\left(\sum_{m=1}^{n}\left\|\boldsymbol{e}_0^m\right\|_\infty + L\Delta t\sum_{s=0}^{k+q}g_s+\Delta t\sum_{p=0}^{k}\sum_{m=1}^{n}\left\|\boldsymbol{d}_p^m\right\|_\infty\right) \\ &\leq Mn\epsilon(1+k\Delta t)+ML\Delta t\sum_{s=0}^{k+q}g_s. \end{aligned} \tag{73}$$

Now we can use the discrete Grönwall lemma (see, e.g., [3, Lemma 11.2]) to conclude that

$$g_{k+q}\leq\epsilon\left(nM(1+(k+1)\Delta t)e^{ML(k+q+1)\Delta t}\right)\leq\epsilon\underbrace{nM(1+T)e^{MLT}}_{C}, \tag{74}$$

where $T\geq(q+k+1)\Delta t$ is some integration period. Recalling that g_{k+p} is the 1-norm of the vector $\boldsymbol{z}_{k+q}-\boldsymbol{u}_{k+q}$ (see Eq. (72)) we see that (74) can be written as

$$\left\|\boldsymbol{z}_{k+q}-\boldsymbol{u}_{k+q}\right\|_1\leq C\epsilon, \tag{75}$$

for all k such that $(q+k+1)\Delta t\leq T$. Alternatively, if we set a maximum number of time steps steps $N\geq k$ and an integration period T then (75) holds for all $k\leq N$ and for all $\Delta t\leq T/(N+q)=\Delta t^*$. This is were the definition of zero-stability kicks in, i.e., conditions (74) and (75) are satisfied for all $\Delta t\leq T/(N+q)=\Delta t^*$. Based on definition (19) we conclude that the root condition implies zero-stability. The converse statement, i.e., zero-stability implies root condition, is straightforward. Indeed, if the scheme is zero stable then (20) is satisfied for all ϵ. This implies that (see Euation 51)

$$\left\|\boldsymbol{A}^{k+1}\boldsymbol{e}_0^m+\Delta t\sum_{p=0}^{k}\boldsymbol{A}^{k-p}\left(\boldsymbol{b}_p^m+\boldsymbol{d}_p^m\right)\right\|_\infty\leq C\epsilon \tag{76}$$

Recalling that C must be independent of k, this condition can be satisfied for all ϵ if and only if $\left\|\boldsymbol{A}^k\right\|\leq M$.

Proof of theorem 4. Let $\boldsymbol{y}_k = \boldsymbol{y}(t_k)$ be the solution of the ODE (1) evaluated at $t = t_k$. A substitution of such solution into the scheme (14) yields the truncation error $\boldsymbol{\tau}_{k+q}$, hereafter written in a componentwise form $(m = 1, ..., n)$

$$\sum_{j=0}^{q} \alpha_j y_{k+j}^m = \Delta t \left(\Phi_{\boldsymbol{f}}^m(\boldsymbol{y}_{k+q}, \ldots, \boldsymbol{y}_k, t_k, \Delta t) + \tau_{q+k}^m \right). \tag{77}$$

Similarly, the numerical solution $\boldsymbol{u}_k$ satisfies

$$\sum_{j=0}^{q} \alpha_j u_{k+j}^m = \Delta t \Phi_{\boldsymbol{f}}^m(\boldsymbol{u}_{k+q}, \ldots, \boldsymbol{u}_k, t_k, \Delta t). \tag{78}$$

Subtracting (78) from (77) yields

$$\sum_{j=0}^{q} \alpha_j e_{k+j}^m = \Delta t \left[\Phi_{\boldsymbol{f}}^m(\boldsymbol{y}_{k+q}, \ldots, \boldsymbol{y}_k, t_k, \Delta t) - \Phi_{\boldsymbol{f}}^m(\boldsymbol{u}_{k+q}, \ldots, \boldsymbol{u}_k, t_k, \Delta t) + \tau_{q+k}^m \right], \tag{79}$$

where

$$e_{k+j}^m = y_{k+j}^m - u_{k+j}^m. \tag{80}$$

By following exactly same steps that took as from equation (46) to (75) in the proof of theorem 2 we obtain the error bound

$$\|\boldsymbol{y}(t_k) - \boldsymbol{u}_k\|_1 \leq MTe^{MLT} \|\boldsymbol{\tau}(\Delta t)\|_1, \tag{81}$$

where the global truncation error $\|\boldsymbol{\tau}\|$ is a function of Δt. To obtain (81) we replaced $(q + k)\Delta t$ with T, which implies that (81) holds for all $\Delta t \leq T/(N + q)$ (this is where zero stability comes in) where N any fixed number larger or equal than $(k + q)$.

Moreover, for Δt small enough, we have seen that $\|\boldsymbol{\tau}\|_1$ goes to zero as some power of Δt (otherwise the method is not consistent). Equation (81) says that the convergence order of the method is the same as the order of consistency.

To obtain the bound (81) we assumed that the initial condition has no error, and that the numerical computation of $\boldsymbol{\Phi_f}$ and all arithmetic operations in the schemes are exact. Clearly this is not the case in practice. It is possible to repeat the proof above, by assuming that all these numerical inaccuracies are bounded, e.g., as a function of the machine precision ϵ, and develop a more detailed bound that depends on ϵ.

References

[1] W. Hahn. *Stability of motion.* Springer, 1967.

[2] J. D. Lambert. *Numerical methods for ordinary differential systems: the initial value problem.* Wiley, 1991.

[3] A. Quarteroni, R. Sacco, and F. Salieri. *Numerical mathematics.* Springer, 2007.

Absolute stability of numerical methods for ODEs

We have seen in previous lecture notes that if a method is zero-stable then[1]

$$\|\boldsymbol{y}(t_k) - \boldsymbol{u}_k\|_1 \leq MTe^{MLT} \|\boldsymbol{\tau}(\Delta t)\|_1 \quad \text{for all} \quad k = 0, 1, \ldots, N \tag{1}$$

where $\boldsymbol{\tau}(\Delta t)$ is the global truncation error of the scheme[2]. Equation (1) bounds the error between the analytical solution of the initial value problem

$$\begin{cases} \dfrac{d\boldsymbol{y}}{dt} = \boldsymbol{f}(\boldsymbol{y}, t) \\ \boldsymbol{y}(0) = \boldsymbol{y}_0 \end{cases} \tag{4}$$

evaluated at t_k and the numerical solution of (4) computed with the scheme

$$\begin{cases} \displaystyle\sum_{j=0}^{q} \alpha_j \boldsymbol{u}_{k+j} = \Delta t \boldsymbol{\Phi}_{\boldsymbol{f}}(\boldsymbol{u}_{k+q}, \ldots, \boldsymbol{u}_k, t_k, \Delta t), \\ \text{given} \quad \{\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{q-1}\} \end{cases} \tag{5}$$

If we send Δt to zero we have that $\|\boldsymbol{\tau}(\Delta t)\|$ in (1) goes to zero (by consistency) and therefore we can make the error between the analytical solution $\boldsymbol{y}(t_k)$ and the numerical solution $\boldsymbol{u}_k$ as small as we like (modulus errors due to finite machine precision).

However, for *finite* Δt it is possible that the errors due to truncation and finite machine precision propagate form one iteration to then next, and eventually build up in a way that drives the numerical solution away from the exact solution. Such "unstable" dynamics is still going to have an error that is bounded by the right hand side of (1) within the integration period T (if the numerical method is zero-stable).

Prototype problem for absolute stability analysis. To study the way local errors accumulate in time and eventually yield instabilities it is convenient to consider a prototype ODE system that has a well-defined time-asymptotic state. Of course, the simplest system we can think of is a linear system[3] of the form

$$\begin{cases} \dfrac{d\boldsymbol{y}}{dt} = \boldsymbol{B}\boldsymbol{y} \\ \boldsymbol{y}(0) = \boldsymbol{y}_0 \end{cases} \tag{6}$$

where $\boldsymbol{B}$ is a matrix with eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$ having strictly negative real part, i.e.,

$$\text{Re}(\lambda_i) < 0 \quad \text{for all } i = 1, \ldots, n. \tag{7}$$

[1] Recall that all norms in a finite-dimensional vector space are equivalent. Hence we can replace the 1-norm in (1) with any other (equivalent) norm.

[2] Note that the bound at the right hand side of (1) has an amplification factor

$$C = MTe^{MLT} \tag{2}$$

that can be very big. For instance, if $T = 10$ (integration period), $L = 2$ (Lipschitz constant of $\boldsymbol{f}$ in (4)), and $M = 1$ (norm of the matrix $\boldsymbol{A}$ defined in the course note 4, Lemma 1) then we obtain

$$C = 10e^{20} \simeq 4.851 \times 10^9. \tag{3}$$

[3] A numerical method which cannot handle satisfactorily the linear system (6) shall not be considered a good method. Moreover, there is ample computational evidence that methods with ample absolute stability regions (see, e.g., Figure 1) outperform those with small regions.

Hereafter, we also assume that the matrix $\boldsymbol{B}$ is diagonalizable. This simplifies the mathematical derivations and it does not change the conclusions of the analysis, meaning that the same results can obtained for non-diagonalizable matrices using a more involved analysis. As is well known, if the matrix $\boldsymbol{B}$ is diagonalizable then there exists an invertible matrix $\boldsymbol{P}$ such that

$$\boldsymbol{B} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^{-1}, \tag{8}$$

where

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} \qquad \text{(diagonal matrix of eigenvalues)}, \tag{9}$$

and

$$\boldsymbol{P} = \begin{bmatrix} \begin{bmatrix} v_{11} \\ \vdots \\ v_{n1} \end{bmatrix} \cdots \begin{bmatrix} v_{1n} \\ \vdots \\ v_{nn} \end{bmatrix} \end{bmatrix} \qquad \text{(matrix of eigenvectors)}. \tag{10}$$

With the representation (8) available, we can write the analytical solution to (6) as

$$\boldsymbol{y}(t) = \boldsymbol{P}e^{t\boldsymbol{\Lambda}}\boldsymbol{P}^{-1}\boldsymbol{y}_0, \tag{11}$$

where

$$e^{t\boldsymbol{\Lambda}} = \begin{bmatrix} e^{t\lambda_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{t\lambda_n} \end{bmatrix}. \tag{12}$$

The assumption $\mathrm{Re}(\lambda_i) < 0$ implies that

$$\lim_{t\to\infty} \|\boldsymbol{y}(t)\| = 0. \tag{13}$$

Note that the matrix $\boldsymbol{P}$ allows us to fully decouple the system of ODEs (6). In fact, a substitution of (8) into (6) yields

$$\begin{cases} \dfrac{d\boldsymbol{q}}{dt} = \boldsymbol{\Lambda}\boldsymbol{q} \\ \boldsymbol{q}(0) = \boldsymbol{q}_0 \end{cases} \tag{14}$$

where

$$\boldsymbol{q}(t) = \boldsymbol{P}^{-1}\boldsymbol{y}(t), \qquad \boldsymbol{q}_0 = \boldsymbol{P}^{-1}\boldsymbol{y}_0.$$

The matrix $\boldsymbol{\Lambda}$ is diagonal, and therefore the system of ODEs (14) is fully decoupled (meaning that we can solve each ODE independently of the others). Note also that, in general, the matrix of eigenvectors $\boldsymbol{P}$ is complex, i.e., $\boldsymbol{q}(t)$ can be a complex vector.

Remark: If We drop the assumption that $\boldsymbol{B}$ is diagonalizable, then we have that $\boldsymbol{B}$ is similar to a block-diagonal Jordan matrix $\boldsymbol{J}$. Everything we have said so far still holds, with the only difference that matrix $\boldsymbol{\Lambda}$ is replaced by a block-diagonal matrix $\boldsymbol{J}$. In this case the system (14) is not fully decoupled.

Next, we study under which conditions the numerical solution $\{\boldsymbol{u}_k\}$ produced by the scheme (5) applied the linear ODE (6) decays to zero as t_k goes to infinity.

Definition 1 (Absolute stability)**.** The numerical method (5) is said to be absolutely stable if when applied to the linear system (6) generates a numerical solution $\{\boldsymbol{u}_k\}$ that decays to zero as t_k goes to infinity, i.e.,

$$\lim_{k\to\infty} \|\boldsymbol{u}_k\| = 0 \tag{15}$$

As we shall see hereafter, for any given matrix $\boldsymbol{B}$, the absolute stability condition may be satisfied for some Δt but not for others.

Absolute stability analysis of elementary one-step methods. Let us provide a few simple examples of absolute stability analysis for well-known one-step methods.

- ***Euler forward:*** Let us approximate the numerical solution of (6) using the Euler forward scheme

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \Delta t \boldsymbol{B}\boldsymbol{u}_k. \tag{16}$$

By using the similarity transformation $\boldsymbol{P}$ we can decouple this scheme exactly as we did for the system (6). To this end, note that

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \Delta t \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^{-1}\boldsymbol{u}_k \quad \Leftrightarrow \quad \underbrace{\boldsymbol{P}^{-1}\boldsymbol{u}_{k+1}}_{\boldsymbol{w}_{k+1}} = \underbrace{\boldsymbol{P}^{-1}\boldsymbol{u}_k}_{\boldsymbol{w}_k} + \Delta t \boldsymbol{\Lambda} \underbrace{\boldsymbol{P}^{-1}\boldsymbol{u}_k}_{\boldsymbol{w}_k} \tag{17}$$

which, upon definition of[4]

$$\boldsymbol{w}_k = \boldsymbol{P}^{-1}\boldsymbol{u}_k \tag{18}$$

can be written component by component as

$$w_{k+1}^j = w_k^j + \Delta t \lambda_j w_k^j = (1 + \Delta t \lambda_j)\, w_k^j = (1 + \Delta t \lambda_j)^{k+1}\, w_0^j \qquad j = 1, \ldots, n. \tag{19}$$

By taking the modulus we obtain

$$\left| w_{k+1}^j \right| = |1 + \Delta t \lambda_j|^{k+1} \left| w_0^j \right|. \tag{20}$$

Hence, a necessary and sufficient condition for absolute stability of the Euler forward method is

$$|1 + \Delta t \lambda_j| < 1. \tag{21}$$

This condition defines a region of the complex plane, called the *region of absolute stability* in which the Euler forward scheme is absolutely stable (see Figure 1). The region of absolute stability imposes conditions on Δt for a given set of eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$. Such conditions are sketched in Figure 1 and derived analytically hereafter. To this end, note that

$$\begin{aligned} |1 + \Delta t \lambda_j|^2 &= [\mathrm{Re}\,(1 + \Delta t \lambda_j)]^2 + [\mathrm{Im}\,(1 + \Delta t \lambda_j)]^2 \\ &= [1 + \Delta t\, \mathrm{Re}(\lambda_j)]^2 + \Delta t^2\, \mathrm{Im}(\lambda_j)^2 \\ &= 1 + \Delta t^2 \left[\mathrm{Re}(\lambda_j)^2 + \mathrm{Im}(\lambda_j)^2\right] + 2\Delta t\, \mathrm{Re}(\lambda_j) \\ &= 1 + \Delta t^2\, |\lambda_j|^2 + 2\Delta t\, \mathrm{Re}(\lambda_j). \end{aligned} \tag{22}$$

Clearly,

$$|1 + \Delta t \lambda_j|^2 \leq 1 \quad \Leftrightarrow \quad \Delta t\, |\lambda_j|^2 + 2\, \mathrm{Re}(\lambda_j) < 0, \tag{23}$$

i.e.,

$$0 < \Delta t < \max_{j=1,\ldots,n} \left(-\frac{2\, \mathrm{Re}(\lambda_j)}{|\lambda_j|^2} \right). \tag{24}$$

Hence the Euler forward method is *conditionally absolutely stable*, the condition being Δt smaller than the maximum of $-2\, \mathrm{Re}(\lambda_j)/\, |\lambda_j|^2$.

[4]Note that the vector $\boldsymbol{w}_k$ defined in equation (18) has, in general, complex entries. In fact the matrix of eigenvectors $\boldsymbol{P}$ is complex if the eigenvalues are complex.

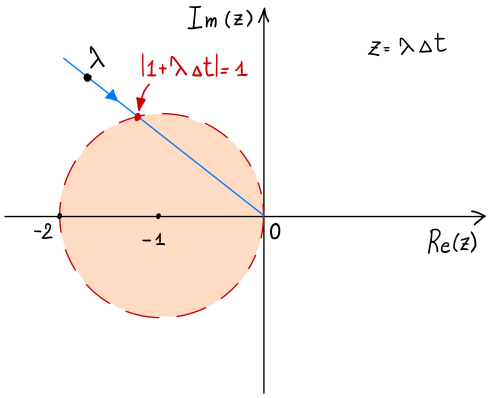


Figure 1: Region of absolute stability of the Euler forward method (shaded unit circle excluding the boundary). The largest Δt that guarantees absolute stability of the Euler Forward method is the one that re-scales the eigenvalues of the matrix $\boldsymbol{B}$ and brings them all within the unit circle (excluding the boundary). In the figure we sketch the re-scaling of one eigenvalue λ by a factor Δt that brings it exactly at the boundary of the circle.

- ***Euler backward:*** Let us approximate the numerical solution of (6) using the Euler backward scheme

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \Delta t \boldsymbol{B} \boldsymbol{u}_{k+1}. \tag{25}$$

By using the similarity transformation defined by $\boldsymbol{P}$ we decouple this scheme exactly as we did for the ODE system (6) and for the Euler forward method. To this end, substitute (18) into (25) to obtain

$$\boldsymbol{w}_{k+1} - \Delta t \boldsymbol{\Lambda} \boldsymbol{w}_{k+1} = \boldsymbol{w}_k. \tag{26}$$

By writing (26) component by component we obtain

$$(1 - \Delta t \lambda_j)\, w_{k+1}^j = w_k^j \quad \Rightarrow \quad w_{k+1}^j = \frac{1}{(1 - \Delta t \lambda_j)^{k+1}} w_0^j. \tag{27}$$

Therefore, the Euler backward method is absolutely stable if and only if for all $j = 1, \ldots, n$ we have

$$\frac{1}{|1 - \Delta t \lambda_j|} < 1 \quad \text{i.e.} \quad |1 - \Delta t \lambda_j| > 1. \tag{28}$$

The inequality $|1 - z| > 1$ with $z \in \mathbb{C}$ defines the region outside a unit circle centered at 1 (see Figure 2). In terms of restrictions on Δt, a substitution of (22) into (28) yields

$$\Delta t \underbrace{\left(\Delta t \, |\lambda_j|^2 - 2 \, \mathrm{Re}(\lambda_j) \right)}_{>0} > 0 \quad \Leftrightarrow \quad \Delta t > 0 \tag{29}$$

Since this condition is satisfied by any $\Delta t > 0$ we say that Euler Backward is *unconditionally absolutely stable*.

- ***Crank-Nicolson:*** Let us approximate the numerical solution of (6) using the Crank-Nicolson scheme

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \frac{\Delta t}{2} \left[\boldsymbol{B} \boldsymbol{u}_{k+1} + \boldsymbol{B} \boldsymbol{u}_k \right]. \tag{30}$$

As before, we decouple this scheme by using the similarity transformation defined by $\boldsymbol{P}$. This yields,

$$\boldsymbol{w}_{k+1} - \frac{\Delta t}{2} \boldsymbol{\Lambda} \boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \frac{\Delta t}{2} \boldsymbol{\Lambda} \boldsymbol{w}_k, \tag{31}$$

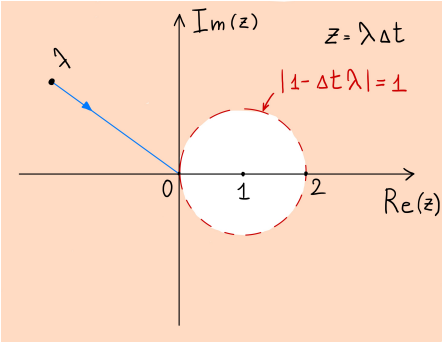


Figure 2: Region of absolute stability of the Euler backward method (shaded area outside the unit circle centered at $z = 1$ excluding the boundary of the circle). The Euler backward method is unconditionally absolutely stable (A-stable) since any eigenvalue with negative real part is in the region of absolute stability.

which can be written component by component as

$$\left(1 - \frac{\Delta t \lambda_j}{2}\right) w_{k+1}^j = \left(1 + \frac{\Delta t \lambda_j}{2}\right) w_k^j \quad \Rightarrow \quad w_{k+1}^j = \left|\frac{1 + \frac{\Delta t \lambda_j}{2}}{1 - \frac{\Delta t \lambda_j}{2}}\right|^{k+1} w_0^j. \tag{32}$$

Hence, the Crank-Nicolson method is absolutely stable if and only if

$$\left|1 + \frac{\Delta t \lambda_j}{2}\right| < \left|1 - \frac{\Delta t \lambda_j}{2}\right| \quad \Leftrightarrow \quad \mathrm{Re}(\lambda_j \Delta t) < 0. \tag{33}$$

The last condition follows from the following simple calculation. Set $z = \Delta t \lambda_j / 2$. Then we have[5]

$$|1 + z|^2 < |1 - z|^2 \quad \Leftrightarrow \quad 1 + 2\,\mathrm{Re}(z) + |z|^2 < 1 - 2\,\mathrm{Re}(z) + |z|^2 \quad \Leftrightarrow \quad \mathrm{Re}(z) < 0. \tag{34}$$

Since $\mathrm{Re}(\lambda_j) < 0$ we conclude from (33) that the Crank-Nicolson method is absolutely stable for all $\Delta t > 0$. In other words it is *unconditionally absolutely stable*. The region of absolute stability of the Crank-Nicolson method is sketched in Figure 3

- ***Heun:*** Let us approximate the numerical solution of (6) using the Heun method

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \frac{\Delta t}{2}\left[\boldsymbol{B}\left(\boldsymbol{u}_k + \Delta t \boldsymbol{B}\boldsymbol{u}_k\right) + \boldsymbol{B}\boldsymbol{u}_k\right] = \boldsymbol{u}_k + \Delta t \boldsymbol{B}\boldsymbol{u}_k + \frac{\Delta t^2}{2}\boldsymbol{B}^2\boldsymbol{u}_k. \tag{35}$$

As before, we decouple the scheme by using the similarity transformation defined by $\boldsymbol{P}$ to obtain

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \Delta t \boldsymbol{\Lambda}\boldsymbol{w}_k + \frac{\Delta t^2}{2}\boldsymbol{\Lambda}^2\boldsymbol{w}_k. \tag{36}$$

This can be written component by component as

$$w_{k+1}^j = \left(1 + \Delta t \lambda_j + \frac{\Delta t^2 \lambda_j^2}{2}\right)^{k+1} w_0^j, \tag{37}$$

[5]Recall that for any $z \in \mathbb{C}$ we have:

$$|1+z|^2 = (1+z)(1+z^*) = 1 + (z+z^*) + zz^* = 1 + 2\,\mathrm{Re}(z) + |z|^2,$$
$$|1-z|^2 = (1-z)(1-z^*) = 1 - (z+z^*) + zz^* = 1 - 2\,\mathrm{Re}(z) + |z|^2.$$

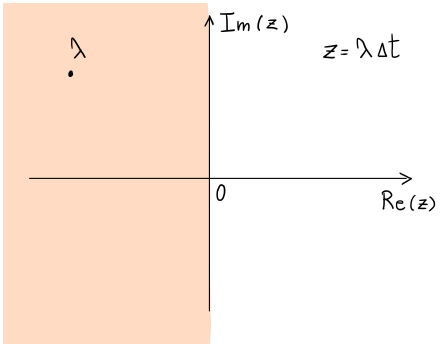


Figure 3: Region of absolute stability of the Crank-Nicolson method (shaded area representing half of the complex plane). The Crank-Nicolson method is unconditionally absolutely stable (A-stable) since any eigenvalue with negative real part is in the region of absolute stability.

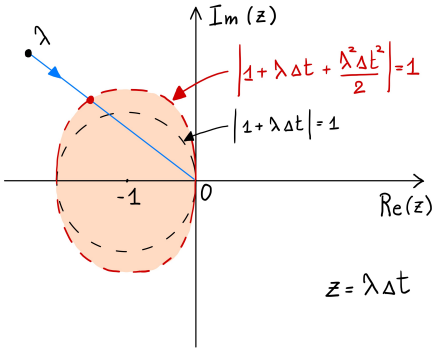


Figure 4: Region of absolute stability of the Heun method (shaded area). The largest Δt that guarantees absolute stability of the Heun method is the one that re-scales the eigenvalues of the matrix $\boldsymbol{B}$ and brings them all within the shaded area sketched in the Figure (excluding the boundary). In the figure we sketch the re-scaling of one eigenvalue λ by a factor Δt that brings it exactly at the boundary of the area. Note that the region of absolute stability of the Heun method is larger than the one of Euler forward, and therefore allows for slightly larger Δt (if the eigenvalues of the matrix $\boldsymbol{B}$ are complex).

which yields the absolute stability condition

$$\left|1 + \Delta t\lambda_j + \frac{\Delta t^2 \lambda_j^2}{2}\right| < 1 \quad \text{for all } j = 1, \ldots, n. \tag{38}$$

The region of absolute stability of the Heun method is sketched in Figure 4. The boundary of stability region is the *one level set* of the real-valued function

$$b(z) = \left|1 + z + \frac{z^2}{2}\right| \qquad z \in \mathbb{C}. \tag{39}$$

Similarly to the Euler forward method, the Heun method is conditionally absolutely stable.

At this point we provide a more rigorous definition of unconditional absolute stability. To this end, let

$$\mathbb{C}^- = \{z \in \mathbb{C} : \text{Re}(z) < 0\}. \tag{40}$$

Definition 2 (A-stability)**.** Let R be the region of absolute stability of the numerical method (5). We say that the method is A-stable if

$$R \cap \mathbb{C}^- = \mathbb{C}^- \tag{41}$$

In other words, if the R includes $\mathbb{C}^-$ then the method is A-stable (or unconditionally absolutely stable).

Clearly, Euler backward and Crank-Nicolson methods are both A-stable, while Euler forward and Heun methods are all conditionally stable. More generally, one can prove that

Theorem 1. There is no explicit A-stable numerical method.

This theorem states that all explicit methods are conditionally absolutely stable. On the other hand, implicit methods can be A-stable (e.g., Crank-Nicolson) or conditionally stable (e.g., BDF methods with three or more steps, or Adams-Moulton methods with two or more steps). As we shall see hereafter, there is in fact no A-stable implicit linear multistep method of order greater than 2.

Absolute stability analysis of linear multistep methods. Consider a general linear q-step method applied to the linear ODE system (6)

$$\sum_{j=0}^{q}\alpha_j\boldsymbol{u}_{k+j}=\Delta t\sum_{j=0}^{q}\beta_j\boldsymbol{B}\boldsymbol{u}_{k+j}. \tag{42}$$

We decouple the system by using the similarity transformation $\boldsymbol{P}$ defined in (8). To this end, define

$$\boldsymbol{w}_k=\boldsymbol{P}^{-1}\boldsymbol{u}_k, \tag{43}$$

and substitute it into (42) to obtain

$$\sum_{j=0}^{q}\alpha_j\boldsymbol{w}_{k+j}=\Delta t\sum_{j=0}^{q}\beta_j\boldsymbol{\Lambda}\boldsymbol{w}_{k+j}, \tag{44}$$

where $\boldsymbol{\Lambda}$ is the diagonal matrix (9). It is convenient to write (44) component by component as

$$\sum_{j=0}^{q}\underbrace{(\alpha_j-\Delta t\lambda_m\beta_j)}_{c_j}w_{k+j}^m=0\qquad m=1,\ldots,n. \tag{45}$$

At this point we follow the same mathematical technique we used in the proof of Theorem 2 in the course note 4 (i.e., root condition implies zero-stability). To this end, we define[6]

$$\boldsymbol{z}_k^m=\begin{bmatrix}w_k^m\\ w_{k+1}^m\\ \vdots\\ w_{k+q-1}^m\end{bmatrix},\qquad \boldsymbol{C}=\begin{bmatrix}0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \cdots & 1\\ -c_0/c_q & -c_1/c_q & -c_2/c_q & \cdots & -c_{q-1}/c_q\end{bmatrix}. \tag{46}$$

and write (45) as a recurrence (for a complex vector)

$$\boldsymbol{z}_{k+1}^m=\boldsymbol{C}\boldsymbol{z}_k^m. \tag{47}$$

The matrix $\boldsymbol{C}$ is the companion matrix of the characteristic polynomial

$$\pi(z)=\rho(z)-\lambda_j\Delta t\sigma(z)\qquad\text{(stability polynomial)}, \tag{48}$$

where

$$\rho(z)=\sum_{j=0}^{q}\alpha_jz^j\quad\text{(first characteristic polynomial)}, \tag{49}$$

$$\sigma(z)=\sum_{j=0}^{q}\beta_jz^j\quad\text{(second characteristic polynomial)}. \tag{50}$$

[6]Note that the vectors $\boldsymbol{z}_k^m$ and the matrix $\boldsymbol{A}$ have (in general) complex entries.

The recurrence (47) can be easily solved to obtain

$$\boldsymbol{z}_{k+1}^m = \boldsymbol{C}^{k+1}\boldsymbol{z}_0^m. \tag{51}$$

Clearly, a necessary and sufficient condition for $\|\boldsymbol{z}_k^m\| \to 0$ as $k \to \infty$ is that the matrix $\boldsymbol{C}$ is a contraction. This happens if and only if the eigenvalues of $\boldsymbol{C}$, i.e., the roots of the polynomial (48), are within the unit disk (excluding the boundary). We can summarize these results as follows.

Theorem 2. The linear multistep method (42) is absolutely stable if and only if the roots of the stability polynomial (48) are within the unit disk (excluding the boundary of the disk).

Note that for $\Delta t \to 0$ the polynomial (48) tends to the first characteristic polynomial (49). Hence, in the limit of small Δt the condition for absolute stability tends to be the same as the root condition. This means that there exists a simple root of $\pi(z)$, say z^*, that approaches 1 for $\Delta t \to 0$. This is necessary for the consistency of the method. However, it should be kept in mind that zero-stability and absolute stability are different concepts. Indeed there exist convergent methods that are not absolutely stable. Let us provide an example

- ***Leapfrog method:*** Let us study absolute stability of the Leapfrog method

$$\boldsymbol{u}_{k+2} = \boldsymbol{u}_k + 2\Delta t \boldsymbol{f}(\boldsymbol{u}_{k+1}, t_k). \tag{52}$$

  The first and second characteristic polynomials associated with the scheme are

$$\rho(z) = z^2 - 1, \qquad \sigma(z) = 2z. \tag{53}$$

  This gives us the following stability polynomial (see (48))

$$\pi(z) = z^2 - 2\lambda_j \Delta t z - 1. \tag{54}$$

  This is a polynomial with (in general) complex coefficients. To find the boundary of the region of absolute stability we look for all roots of $\pi(z)$ with modulus one, that is set[7]

$$z = e^{i\vartheta}, \tag{55}$$

  substitute it into (54) and set the equation to zero

$$e^{2i\vartheta} - 2\lambda_j \Delta t e^{i\vartheta} - 1 = 0 \quad \Leftrightarrow \quad \lambda_j \Delta t = \frac{e^{2i\vartheta} - 1}{2e^{i\vartheta}} = \frac{e^{i\vartheta} - e^{-i\vartheta}}{2} = i\sin(\vartheta) \tag{56}$$

  As shown in Figure 5 the region of absolute stability in this case collapses to the interval $[-i, i]$ on the imaginary axis. Hence, the leapfrog method is always absolutely unstable. This means that there is no hope for the method (52) to simulate accurately a linear system that has an attractor at the origin. The method is convergent through. Therefore as $\Delta t \to 0$ the global error becomes smaller and smaller (see Eq. (1)).

The technique we used to compute the boundary of the absolute stability region of the leapfrog method can be generalized to arbitrary linear multistep methods. To this end, we just need to look for all roots of modulus one of the stability polynomial (48), that is plot the set of complex numbers

$$\lambda_j \Delta t = \frac{\rho\left(e^{i\vartheta}\right)}{\sigma\left(e^{i\vartheta}\right)} = \frac{\displaystyle\sum_{j=0}^{q} \alpha_j e^{ij\vartheta}}{\displaystyle\sum_{j=0}^{q} \beta_j e^{ij\vartheta}}, \qquad \vartheta \in [0, 2\pi]. \tag{57}$$

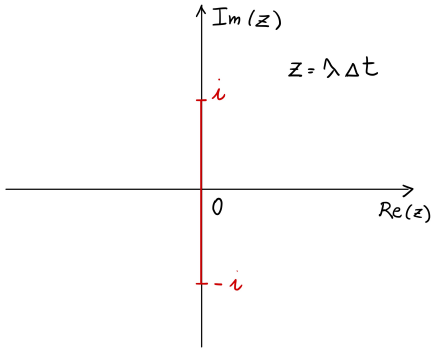


Figure 5: Region of absolute stability of the leapfrog method (52). Note that region of stability in the case collapses to the interval $[-i, i]$ on the imaginary axis. Hence, the leapfrog method is always absolutely unstable. In other words, there is no hope for the method (52) to simulate accurately a linear dynamical system that has an attractor at the origin.

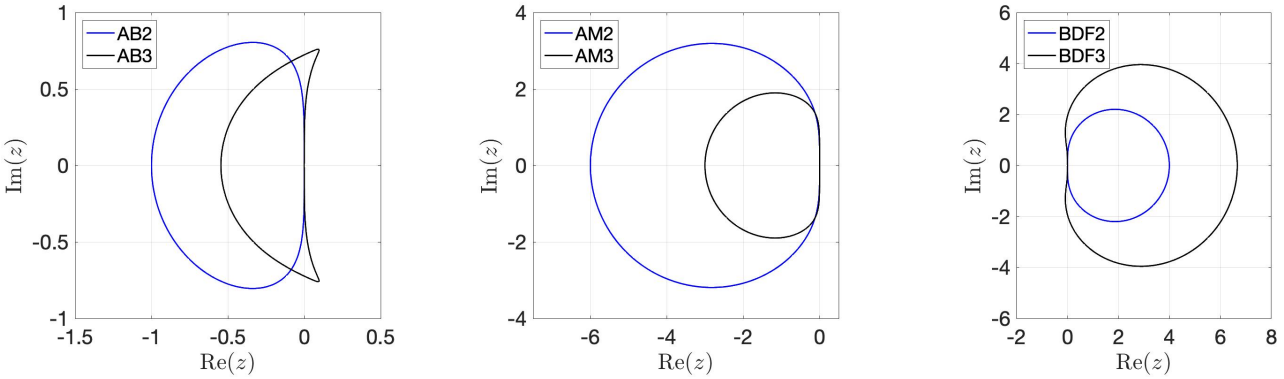


Figure 6: Boundary of the absolute stability region for various linear multistep methods. For Adams-Bashforth (AM) and Adams-Moulton (AM) methods, the region of absolute stability is the area inside the closed curve, while for BDF method is the area outside the curve. Note that the region of absolute stability of AM methods is larger than that of AB methods.

In Figure 6 we provide a few plots of the boundary of the absolute stability region for a Adams-Bashforth, Adams-Moulton and BDF methods.

Remark: It is important to emphasize that the curves plotted in Figure 6 represent the set of values $\lambda\Delta t$ for which the stability polynomial (48) has at least one root with modulus one. As is well known, the roots of a polynomial are continuous functions of the coefficients of the polynomial. In the case of (48) we have one parameter, i.e., $\lambda\Delta t$, which multiplies all coefficients of $\sigma(z)$, hence affecting simultaneously multiple coefficients. To figure out which region of the complex plane is absolutely stable, e.g., the inner or the outer part of the curve defined in (57), it is sufficient to compute the roots of (48) for $\lambda\Delta t$ inside or outside the region defined by the curve. If such roots are within the unit disk, then the method is absolutely stable.

[7] Recall that the set of complex numbers with modulus one sits on the unit circle in the complex plane and can be represented in therm of the complex exponential function $e^{i\theta} = \cos(\theta) + i\sin(\theta)$.

Remark: Zero-unstable linear multistep methods are necessarily absolutely unstable. To show this, we notice that in the limit $\Delta t \to 0$ we have

$$\pi(z) = \rho(z) - \lambda \Delta t \sigma(z) \to \rho(z). \tag{58}$$

If the method is zero-unstable then $\rho(z)$ has roots outside the unit disk. By continuity of polynomial roots as a function of $\Delta t \lambda$, we have that for all $\Delta t \lambda$ in a small neighborhood of 0 the polynomial (48) has roots outside the unit disk. If a method is consistent then the curve (57) passes through the origin (since $\rho(1) = 0$). Recalling that such curve represent the set of points $\lambda \Delta t$ for which at least one root of (48) has modulus one, we conclude by the continuity of the roots if $\pi(z)$ as a function of $\lambda \Delta t$ at $\lambda \Delta t = 0$ that both inner and outer regions of the curve are absolutely unstable. This proves the following lemma:

Lemma 1. A zero-unstable consistent linear multistep method is absolutely unstable.

At this point we recall that no explict scheme can be A-stable. This implies, in particular, that there is no A-stable explicit linear multistep method. What can we say about implicit LMM methods?

Theorem 3 (Second Dahlquist barrier – 1963)**.** There is no A-stable LMM method with order greater than 2.

Recall that AM2 and BDF3 are both methods of order 3. It is seen in Figure 6 that these methods are in fact not A-stable.

Absolute stability analysis of Runge-Kutta methods. The absolute stability analysis we performed for one-step and LMM methods clearly shows that in order to compute the region of absolute stability of a numerical method it is sufficient to consider only one complex ODE of the form

$$\begin{cases} \dfrac{dq}{dt} = \lambda q \\ q(0) = q_0 \end{cases} \tag{59}$$

This ODE can be any in the decoupled system (14) corresponding to an arbitrary eigenvalue λ. Let us discretize (59) with the s-stage RK method

$$w_{k+1} = w_k + \Delta t \sum_{i=1}^{s} b_i K_i, \tag{60}$$

where

$$K_i = \lambda w_k + \lambda \Delta t \sum_{j=1}^{s} a_{ij} K_j \qquad i = 1, \dots, s. \tag{61}$$

At this point it is convenient to define

$$\boldsymbol{K} = \begin{bmatrix} K_1 \\ K_2 \\ \vdots \\ K_s \end{bmatrix}, \qquad \boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1s} \\ a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ a_{s1} & a_{s2} & \cdots & a_{ss} \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_s \end{bmatrix}, \qquad \boldsymbol{h} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{62}$$

and write (61) in a matrix-vector form as

$$(\boldsymbol{I} - \lambda \Delta t \boldsymbol{A}) \boldsymbol{K} = \lambda w_k \boldsymbol{h} \quad \Leftrightarrow \quad \boldsymbol{K} = (\boldsymbol{I} - \lambda \Delta t \boldsymbol{A})^{-1} \boldsymbol{h} \lambda w_k. \tag{63}$$

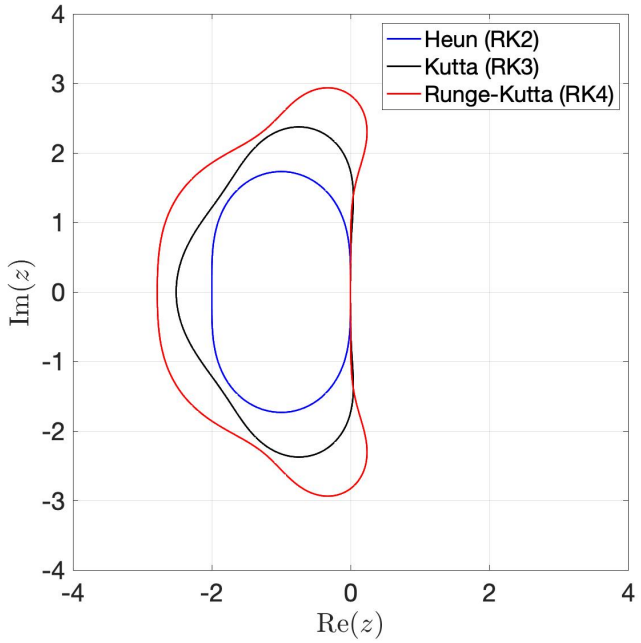


Figure 7: Boundary of the absolute stability region for various explicit RK methods. The region of stability is the interior of each closed curve.

Next, substitute expression we derived for $\boldsymbol{K}$ into (60) to obtain

$$w_{k+1} = w_k + \Delta t \boldsymbol{b}^T \boldsymbol{K} = \left[1 + \lambda \Delta t \boldsymbol{b}^T \left(\boldsymbol{I} - \lambda \Delta t \boldsymbol{A}\right)^{-1} \boldsymbol{h}\right] w_k. \tag{64}$$

At this point we define the *stability function*

$$S(z) = 1 + z\boldsymbol{b}^T \left(\boldsymbol{I} - z\boldsymbol{A}\right)^{-1} \boldsymbol{h}, \tag{65}$$

and iterate (64) to obtain

$$w_{k+1} = S(\lambda \Delta t)^{k+1} w_0. \tag{66}$$

Hence a necessary and sufficient condition for absolute stability of RK methods is that

$$|S(\lambda \Delta t)| < 1. \tag{67}$$

As shown in [1, p. 200], by using the Cramer's rule we can write the stability function (65) as

$$S(z) = \frac{\det\left(\boldsymbol{I} - z\boldsymbol{A} + z\boldsymbol{h}\boldsymbol{b}^T\right)}{\det(\boldsymbol{I} - z\boldsymbol{A})}. \tag{68}$$

Note that, in general $S(z)$ is a rational function, i.e., the ratio between two polynomials in z. In the particular case of explicit RK methods we have that the matrix $\boldsymbol{A}$ is strictly lower triangular. This yields $\det(\boldsymbol{I} - z\boldsymbol{A}) = 1$, which results in

$$S(z) = \det\left(\boldsymbol{I} - z\boldsymbol{A} + z\boldsymbol{h}\boldsymbol{b}^T\right) \qquad \text{(stability function for explicit RK methods)}. \tag{69}$$

In Figure 7 we plot the boundary of the absolute stability region for the explicit RK methods corresponding to the following Butcher arrays[8]:

[8] The boundary of the stability regions are computed as *zero-level set* of $|S(z)| - 1$.

0	0	0
1	1	0
	1/2	1/2

Heun's method (RK2)

0	0	0	0
1/2	1/2	0	0
1	−1	2	0
	1/6	2/3	1/6

Kutta's method (RK3)

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	1/3	1/3	1/6

Runge-Kutta's method (RK4)

References

[1] J. D. Lambert. *Numerical methods for ordinary differential systems: the initial value problem.* Wiley, 1991.

Boundary value problems for ODEs

A boundary value problem (BVP) for an ODE is a problem in which we set conditions on the solution to the ODE corresponding to different values in the independent variable. Such conditions can be on the solution, on the derivatives of the solution, or more general conditions. Perhaps the simplest boundary value problem for an ODE is[1]

$$\begin{cases} \dfrac{d^2u(x)}{dx^2} = f(x) \qquad x \in [0,1] \\ u(0) = \alpha \\ u(1) = \beta \end{cases} \tag{2}$$

in which we set conditions on the value of the solution at $x = 0$ and $x = 1$. Such conditions are called *Dirichlet boundary conditions*. The general solution to (2) can be written as

$$u(x) = c_1 + c_2 x + \int_0^x F(s)ds \qquad \text{where} \qquad F(s) = \int_0^s f(y)dy. \tag{3}$$

By using integration by parts

$$\int_0^x F(s)ds = [sF(s)ds]_{s=0}^{s=x} - \int_0^x sf(s)ds = \int_0^x (x-s)f(s)ds. \tag{4}$$

Substituting this expression into (3) yields

$$u(x) = c_1 + c_2 x + \int_0^x (x-s)f(s)ds. \tag{5}$$

At this point we enforce the boundary conditions to obtain

$$\alpha = c_1 \qquad \beta = c_1 + c_2 + \int_0^1 (1-s)f(s)ds, \tag{6}$$

which gives the following unique solution to (2)

$$u(x) = \alpha + x\left(\beta - \alpha - \int_0^1 (1-s)f(s)ds\right) + \int_0^x (x-s)f(s)ds. \tag{7}$$

Lemma 1. For every $f \in C^0([0,1])$ there exists a unique solution $u \in C^2([0,1])$ to the boundary value problem (2). Moreover, if $f \in C^k([0,1])$ then $u \in C^{k+2}([0,1])$.

Green function and maximum principle. The solution (7) corresponding to zero Dirichlet conditions can be conveniently written in terms of an integral involving a Green function. Setting $\alpha = \beta = 0$ in (7)

[1] From a physical viewpoint, the BVP (2) defines a steady state heat conduction problem in a one-dimensional slab with uniform conductivity, heat generation, and fixed temperature conditions at the boundary. In fact (2) can be derived from the Fourier equation [1]

$$\frac{\partial u}{\partial t} = \frac{\lambda}{\rho c_p}\nabla^2 u + \frac{1}{\lambda} f(\boldsymbol{x}). \tag{1}$$

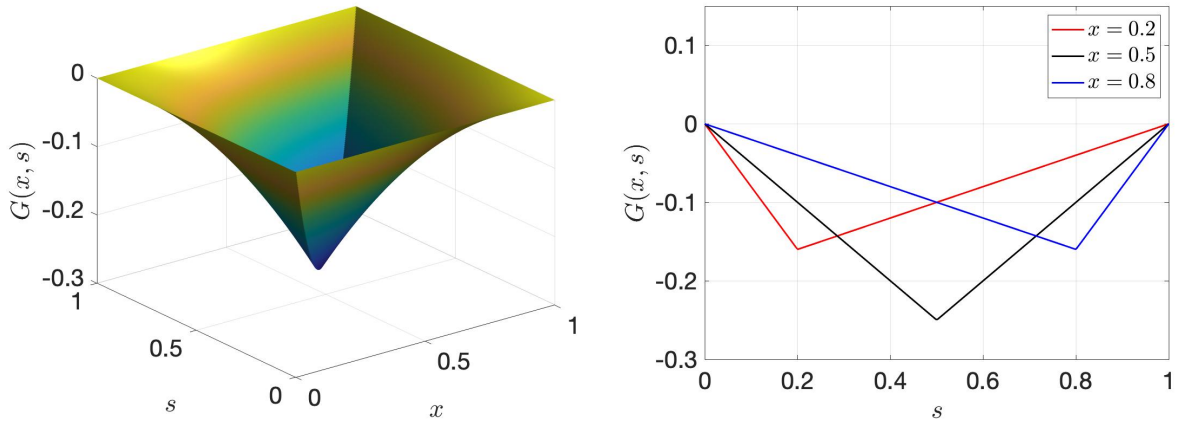


Figure 1: Green function $G(s,x)$ defined in equations (8)-(9).

yields

$$\begin{aligned} u(x) &= -x\int_0^1 (1-s)f(s)ds + \int_0^x (x-s)f(s)ds \\ &= \int_0^x \left[(x-s)-x(1-s)\right] f(s)ds - x\int_x^1 (1-s)f(s)ds \\ &= \int_0^x s(x-1)f(s)ds + \int_x^1 x(s-1)f(s)ds \\ &= \int_0^1 G(x,s)f(s)ds, \end{aligned} \tag{8}$$

where we defined

$$G(x,s) = \begin{cases} s(1-x) & 0 \le s \le x \\ x(1-s) & x \le s \le 1 \end{cases} \qquad \text{(Green function)}. \tag{9}$$

The Green function $G(x,s)$ is the kernel of the integral operator (8), and it represents the "response" of the system corresponding to any forcing function $f(x)$. The Green function satisfies (in a distributional sense, and for all $s \in [0,1]$) the boundary value problem

$$\begin{cases} \dfrac{d^2G(x,s)}{dx^2} = \delta(x-s) \\ G(0,s) = 0 \\ G(1,s) = 0 \end{cases} \tag{10}$$

With the Green function available, it is straightforward to obtain the following bound for (8)

$$\|u\|_\infty \le \frac{1}{8}\|f\|_\infty \qquad \text{(maximum principle)}, \tag{11}$$

where $\|\cdot\|_\infty$ here denotes the uniform norm of a function, i.e.,

$$\|u\|_\infty = \sup_{x\in[0,1]} |u(x)|\,, \qquad \|f\|_\infty = \sup_{x\in[0,1]} |f(x)|\,. \tag{12}$$

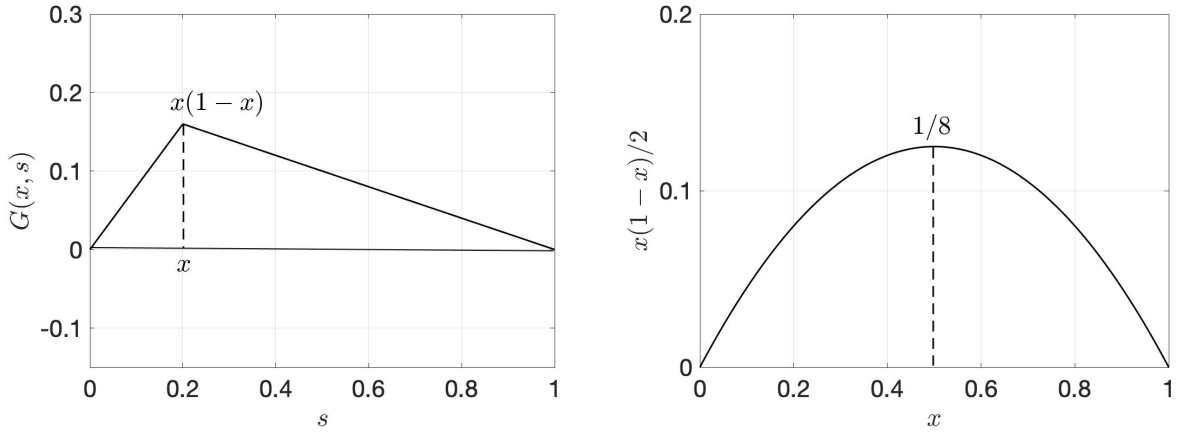


Figure 2: Evaluation of the integral appearing in (13).

The inequality (11), states that the solution of the boundary value problem (2) with homogeneous Dirichlet conditions ($\alpha = \beta = 0$) is always smaller that $1/8$ of the maximum value of $f(x)$ in the domain $[0, 1]$. To prove (11) we observe that

$$|u(x)| \leq \int_0^1 |G(x,s)|\,|f(s)|\,ds \leq \|f\|_\infty \int_0^1 |G(x,s)|\,ds. \tag{13}$$

The Green function $G(x, s)$ is always negative, and for each fixed it is the union of two triangular functions joining at the point $(x, x(1, x))$ (see Figure 2). Therefore,

$$\int_0^1 |G(x,s)|\,ds = x\frac{x(1-x)}{2} + (1-x)\frac{x(1-x)}{2} = \frac{x(1-x)}{2}. \tag{14}$$

Substituting this result into (13) yiels

$$|u(x)| \leq \|f\|_\infty \frac{x(1-x)}{2}. \tag{15}$$

Finally, by taking the maximum over all $x \in [0, 1]$ we obtain[2]

$$\max_{x\in[0,1]} |u(x)| \leq \|f\|_\infty \max_{x\in[0,1]} \frac{x(1-x)}{2} = \frac{1}{8}\|f\|_\infty \tag{17}$$

which coincides with (11).

Ill-posed linear boundary value problems. If we replace the Dirichlet boundary conditions in (2) with two *Neumann boundary conditions* (i.e., we set the value of the derivative of $u(x)$ at $x = 0$ and $x = 1$ instead of the value of the function) then the problem can have either no solution or an infinite number of

[2]The maximum of the function $x(x-1)/2$ is $1/8$ and it is attained at $x = 1/2$ (see Figure 2), i.e., we have

$$\max_{x\in[0,1]} \int_0^1 |G(x,s)|\,ds = \max_{x\in[0,1]} \frac{x(1-x)}{2} = \frac{1}{8}. \tag{16}$$

.

solutions. To show this, let us consider the BVP

$$\begin{cases} \dfrac{d^2u(x)}{dx^2} = f(x) \qquad x \in [0,1] \\ \dfrac{du(0)}{dx} = \alpha \\ \dfrac{du(1)}{dx} = \beta \end{cases} \tag{18}$$

By integrating the ODE once, we obtain

$$\frac{du(x)}{dx} = c_1 + \int_0^x f(s)ds \tag{19}$$

which shows that the derivative of u depends only on one arbitrary constant of integration. Clearly, we do not have enough degrees of freedom to satisfy (in general) both boundary conditions in (18). By enforcing $du(0)/dx = \alpha$ we obtain $c_1 = \alpha$, i.e.,

$$\frac{du(x)}{dx} = \alpha + \int_0^x f(s)ds. \tag{20}$$

If we now try to enforce $du(1)/dx = \beta$ in (20) we obtain the equation

$$\beta - \alpha = \int_0^1 f(s)ds. \tag{21}$$

If $f(x)$ satisfies (21) then the problem (18) has an infinite number of solutions. In fact, by integrating (20) we see that there exits a one-parameter family of solutions (with parameter c_2) of the form

$$u(x) = c_2 + \alpha x + \int_0^x \left(\int_0^y f(s)ds \right) dy. \tag{22}$$

Clearly, the solution (22) satisfies (18) for all $c_2 \in \mathbb{R}$, provided (21) holds. On the other hand, if $f(x)$ does not satisfy (21) then the boundary value problem (18) has no solution.

Exercise: By using a physical argument based on the interpretation of (18) as a model of heat conduction in a one-dimensional slab with heat generation, justify the infinite multiplicity of solutions or the lack of a solution.

Example: It is straightforward to show that the linear BVP

$$\frac{d^2y}{dt^2} + y = 0, \qquad y(0) = 0, \qquad y(\pi) = 0. \tag{23}$$

has no solution. In fact, the flow generated by the corresponding first-order system is a center. There are in principle infinite trajectories that start from $y = 0$ and end at $y = 0$. None of them though makes the trip exactly in π time units.

Ill-posed nonlinear boundary value problems. Next, consider a nonlinear boundary value problem of the form

$$\begin{cases} \dfrac{d^2y}{dt^2} = f\left(\dfrac{dy}{dt}, y, t\right) \qquad t \in [0,T] \\ y(0) = \alpha \\ y(T) = \beta \end{cases} \tag{24}$$

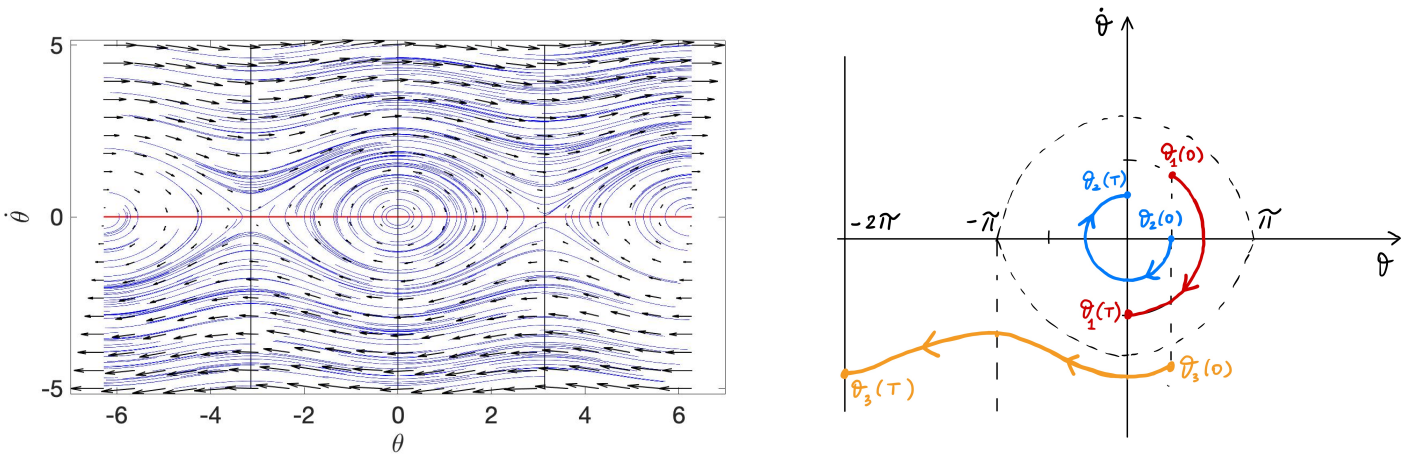


Figure 3: Phase portrait of the pendulum equation $\ddot{\theta} = -\sin(\theta)$ and sketch of two solutions (ϑ_1 and ϑ_2) of the BVP (25). The third solution technically does not satisfy $\theta(T) = 0$ but rather $\theta(T) = -2\pi$ which is physically equivalent, but mathematically different.

It is easy to show by a simple physical example that this problem can have an infinite number of solutions (all of which make sense). To this end, consider the pendulum equations

$$\begin{cases} \dfrac{d^2\theta}{dt^2} = -\sin(\theta) \\ \theta(0) = \dfrac{\pi}{2} \\ \theta(T) = 0 \end{cases} \tag{25}$$

where T is the time that it takes to the pendulum to reach the vertical position after swinging from right to left only once from a zero velocity initial condition. It is clear that there are multiple solutions to this problem. In Figure 3 we sketch two of such initial velocities, and corresponding trajectories.

Exercise: What's the motion of the pendulum corresponding to the paths $(\theta_i, \dot{\theta}_i)$ sketched in Figure 3 for $i = 1, 2, 3$? Interpret the infinite (countable) number of solutions of (25) physically. How many solutions are there within the initial velocity interval $\dot{\theta}(0) \in [-v, v]$, for a given v?

Existence and uniqueness of solutions. There is no general theory for existence and uniqueness of the solution to nonlinear two-point boundary value problems with arbitrary boundary conditions. However, a lot can be said in specific cases. For example, it is straightforward to show that the two-point boundary value problems for the linear system of ODEs

$$\frac{d^2\boldsymbol{y}}{dt^2} = \boldsymbol{A}\boldsymbol{y}, \tag{26}$$

with diagonalizable $\boldsymbol{A}$ and Dirichlet boundary conditions $\boldsymbol{y}(0) = \boldsymbol{\alpha}$ and $\boldsymbol{y}(1) = \boldsymbol{\beta}$ has a unique solution. In fact, upon definition of $\boldsymbol{z} = d\boldsymbol{y}/dt$ we can write (26) as

$$\frac{d}{dt}\begin{bmatrix}\boldsymbol{z}\\ \boldsymbol{y}\end{bmatrix} = \underbrace{\begin{bmatrix}\boldsymbol{0} & \boldsymbol{A}\\ \boldsymbol{I} & \boldsymbol{0}\end{bmatrix}}_{C}\begin{bmatrix}\boldsymbol{z}\\ \boldsymbol{y}\end{bmatrix}. \tag{27}$$

Let $\boldsymbol{P}$ and $\boldsymbol{\Lambda}$ be the matrix of eigenvectors and the diagonal matrix of eigenvalues of $\boldsymbol{A}$, i.e.,

$$\boldsymbol{A} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^{-1}, \tag{28}$$

and consider the transformation induced by the invertible block matrix

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{P} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P} \end{bmatrix}. \tag{29}$$

Clearly,

$$\underbrace{\begin{bmatrix} \boldsymbol{P} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P} \end{bmatrix}}_{\boldsymbol{H}} \underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Lambda} \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{C}} \underbrace{\begin{bmatrix} \boldsymbol{P}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}^{-1} \end{bmatrix}}_{\boldsymbol{H}^{-1}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{A} \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \tag{30}$$

By applying $\boldsymbol{H}^{-1}$ to the system (27) we obtain

$$\frac{d}{dt}\begin{bmatrix} \tilde{\boldsymbol{z}} \\ \tilde{\boldsymbol{y}} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Lambda} \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{C}} \begin{bmatrix} \tilde{\boldsymbol{z}} \\ \tilde{\boldsymbol{y}} \end{bmatrix}, \qquad \text{where} \qquad \begin{bmatrix} \tilde{\boldsymbol{z}} \\ \tilde{\boldsymbol{y}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{y} \end{bmatrix} \tag{31}$$

The solution to this ODE (treated as initial value problem with unknown $\tilde{\boldsymbol{z}}(0)$) is

$$\begin{bmatrix} \tilde{\boldsymbol{z}}(t) \\ \tilde{\boldsymbol{y}}(t) \end{bmatrix} = e^{t\boldsymbol{C}} \begin{bmatrix} \tilde{\boldsymbol{z}}(0) \\ \tilde{\boldsymbol{y}}(0) \end{bmatrix}. \tag{32}$$

Setting the boundary conditions $\boldsymbol{y}(0) = \boldsymbol{\alpha}$ and $\boldsymbol{y}(1) = \boldsymbol{\beta}$ yields

$$\begin{bmatrix} \tilde{\boldsymbol{z}}(1) \\ \boldsymbol{P}^{-1}\boldsymbol{\beta} \end{bmatrix} = e^{\boldsymbol{C}} \begin{bmatrix} \tilde{\boldsymbol{z}}(0) \\ \boldsymbol{P}^{-1}\boldsymbol{\alpha} \end{bmatrix}. \tag{33}$$

The exponential matrix $e^{\boldsymbol{C}}$ has the following structure

$$e^{\boldsymbol{C}} = \begin{bmatrix} \boldsymbol{D}_1 & \boldsymbol{D}_2 \\ \boldsymbol{D}_3 & \boldsymbol{D}_1 \end{bmatrix}, \tag{34}$$

where $\boldsymbol{D}_1$, $\boldsymbol{D}_2$ and $\boldsymbol{D}_3$ are diagonal matrices. Moreover, $\boldsymbol{D}_1$ and $\boldsymbol{D}_3$ are invertible. Substituting (34) into (33) gives

$$\begin{bmatrix} \tilde{\boldsymbol{z}}(1) \\ \boldsymbol{P}^{-1}\boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}_1 & \boldsymbol{D}_2 \\ \boldsymbol{D}_3 & \boldsymbol{D}_1 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{z}}(0) \\ \boldsymbol{P}^{-1}\boldsymbol{\alpha} \end{bmatrix}. \tag{35}$$

This equation allows us to determine $\tilde{\boldsymbol{z}}(0)$ uniquely for any given $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ In fact, the second equation in (35) can be written as

$$\boldsymbol{D}_3\tilde{\boldsymbol{z}}(0) = \boldsymbol{P}^{-1}\boldsymbol{\beta} - \boldsymbol{D}_1\boldsymbol{P}^{-1}\boldsymbol{\alpha} \qquad \Leftrightarrow \qquad \tilde{\boldsymbol{z}}(0) = \boldsymbol{D}_3^{-1}\boldsymbol{P}^{-1}\left(\boldsymbol{\beta} - \boldsymbol{D}_1\boldsymbol{\alpha}\right). \tag{36}$$

Hence, we proved that for every given $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ there exists a unique initial state

$$\begin{bmatrix} \tilde{\boldsymbol{y}}(0) \\ \tilde{\boldsymbol{z}}(0) \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}^{-1}\boldsymbol{\alpha} \\ \boldsymbol{D}_3^{-1}\boldsymbol{P}^{-1}\left(\boldsymbol{\beta} - \boldsymbol{D}_1\boldsymbol{\alpha}\right) \end{bmatrix}. \tag{37}$$

By leveraging the existence and uniqueness of solutions to the initial value problem (32) we conclude that the two-point boundary value problem for the ODE (26) with Dirichlet boundary conditions has a unique solution.

Remark: If we drop the assumption of diagonalizability of $\boldsymbol{A}$ and replace the diagonal matrix $\boldsymbol{\Lambda}$ with its block diagonal Jordan form $\boldsymbol{J}$, then the ODE (26) with Dirichlet boundary conditions still has a unique solution. In fact, in this case the matrix exponential $e^{\boldsymbol{C}}$ is still a block matrix in the form (34), but with upper triangular $\boldsymbol{D}_1$, $\boldsymbol{D}_2$ and $\boldsymbol{D}_3$. Moreover, $\boldsymbol{D}_1$ and $\boldsymbol{D}_3$ are invertible. Hence (36) still holds.

General form of two-point boundary value problems. A two-point boundary value problem for a system of n-dimensional nonlinear ODEs can be written in the general form

$$\begin{cases} \dfrac{d\boldsymbol{y}}{dt} = \boldsymbol{f}\left(\boldsymbol{y}, t\right) \qquad t \in [0, T] \\ \boldsymbol{g}(\boldsymbol{y}(0), \boldsymbol{y}(T)) = \boldsymbol{0} \end{cases} \tag{38}$$

where $\boldsymbol{g} \in \mathbb{R}^n$ is nonlinear function. All two-point boundary value problem we studied so far can be written in this form, provided we define appropriate phase variables $\boldsymbol{y}$, the right hand side $\boldsymbol{f}(\boldsymbol{y}, t)$, and the boundary function $\boldsymbol{g}$.

References

[1] D. W. Hahn and M. N. Özisik. *Heat Conduction*. Wiley, third edition, 2012.

Numerical methods to solve boundary value problems for ODEs

In this note we provide a brief overview of the most common numerical methods to approximate the solution of a boundary value problem for an ODE or a system of ODEs. In particular,

- Shooting method;
- Methods based on finite-differences or collocation;
- Methods based on weighted residuals (Galerkin and least squares).

When applying these methods to a boundary value problem, we will always assume that the problem has at least one solution[1].

Shooting method. The shooting method is a method for solving a boundary value problem by reducing it an to initial value problem which is then solved multiple times until the boundary condition is met. To describe the method let us first consider the following two-point boundary value problem for a second-order nonlinear ODE with Dirichlet boundary conditions

$$\begin{cases} \dfrac{d^2y}{dt^2} = f\left(\dfrac{dy}{dt}, y, t\right) \qquad t \in [0,T] \\ y(0) = \alpha \\ y(T) = \beta \end{cases} \tag{1}$$

We have seen in previous lecture that this problem can have an infinite number of solutions (e.g., the pendulum problem). The shooting method replaces the boundary condition $y(T) = \beta$ in (1) with the initial condition $dy(0)/dt = v$, for an unknown "slope" v, and attempts to find v by using an iterative root-finding algorithm or an optimization method so that the quantity

$$E(v) = y(T; v) - \beta \tag{2}$$

(or $E(v)^2$ in the case of optimization) is equal to zero[2]. In equation (2) $y(T; v)$ represents the solution (flow) to the initial value problem

$$\begin{cases} \dfrac{dy}{dt} = z \\ \dfrac{dz}{dt} = f\left(z, y, t\right) \\ y(0) = \alpha \\ z(0) = v \end{cases} \tag{4}$$

at time T. The notation $y(T; v)$ we used in (2) emphasizes that the solution of (4) depends on v. Recall, in fact, that if the right hand side of the system (4), i.e., $(z, f(y, z, t))$ is of class C^k then the solution $(y(t; \alpha, v), z(t; \alpha, v))$ is of class C^k in the initial condition (α, v). This implies the following lemma.

Lemma 1. If the initial value problem (4) is well-posed then the error function (2) is at least continuous in v. Moreover, if f is of class C^k then the error function (2) is of class C^k (differentiable k times with continuous derivative).

In Figure 1 we provide a sketch of the shooting method.

[1] Recall that a boundary value problem can have a unique solution, an infinite number of solution or no solution at all.

[2] Note that the shooting method is essentially a *control problem* of the form

$$\min_{v \in \mathbb{R}} |y(T; v) - \beta| \quad \text{subject to (4).} \tag{3}$$

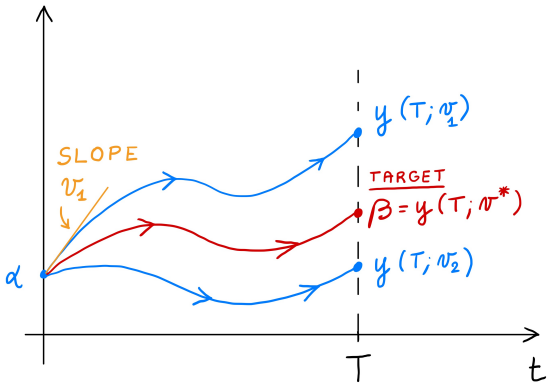


Figure 1: Sketch of the shooting method to solve the two-point boundary value problem (1). Basically, we look for the slope v of the solution at initial time ("shoot" with elevation v), so that we "hit" the target $y(T) = \beta$ at time $t = T$.

With the error function (2) available, we can construct an iterative procedure that generates a sequence of slopes $\{v_0, v_1, v_2, \ldots\}$ with the property

$$\lim_{k\to\infty} E(v_k) = 0 \qquad \text{(rootfinding methods)}. \tag{5}$$

To generate a sequence $\{v_k\}$ satisfying (5) we can use any rootfinding method for scalar equations, such as the bisection method, the secant method or the Newton's method. Note that the function $E(v)$ is not explicitly known, but it is certainly continuous (or smoother depending on the regularity of f), and it can be sampled at any point v we like. To this end, we just need to integrate (4) forward in time up to $t = T$ for the initial condition $z(0) = v$ and then evaluate (2). Let us briefly describe three rootfinding methods we can use to generate a sequence $\{v_0, v_1, v_2, \ldots\}$ with the property (5). These methods are sketched in Figure 2

- ***Bisection method:*** Given any two initial guesses v_0 and v_1 we solve the initial value problem (4) to obtain

$$E(v_0) = y(T; v_0) - \beta, \qquad \text{and} \qquad E(v_1) = y(T; v_1) - \beta. \tag{6}$$

  If the sign of the product $E(v_0)E(v_1)$ is strictly positive, then we cannot claim that there exists a zero v^* of the function $E(v)$ in the interval $[v_0, v_1]$ (although there maybe actually one). On the other hand, if the sign of the product $E(v_0)E(v_1)$ is strictly negative then there exists a point v^* within the interval $[v_0, v_1]$ such that $E(v^*) = 0$. To find v^* we split the interval $[v_0, v_1]$ in half (hence the name "bisection method"), and evaluate $E(v)$ at $v_2 = (v_1 + v_0)/2$. At this point we proceed as before, i.e., if $E(v_0)E(v_2) > 0$ then we forget about the interval $[v_0, v_2]$ and split $[v_2, v_1]$ in half, evaluate $E(v)$ at $(v_2 + v_1)/2$ and so on so forth. On the other hand, if $E(v_0)E(v_2) < 0$ then we forget about the interval $[v_2, v_1]$ and split $[v_0, v_2]$ in half, evaluate $E(v)$ at $(v_2 + v_0)/2$, etc. The bisection procedure until either the function value $E(v_k)$ or the difference between two subsequent iterates $|v_{k+1} - v_k|$ or both is smaller than a prescribed tolerance. For more details on the bisection method see [2, §6.2.1]. Convergence of the bisection method is, on average, linear with the interation number.

- ***Secant method:*** Similarly to the bisection method, we start with two initial guesses v_0 and v_1 (hopefully close enough to the zero v^* we are interested in), and evaluate $E(v_0)$ and $E(v_1)$. Then we

construct the line passing through $(v_0, E(v_0))$ and $(v_1, E(v_1))$ and extrapolate such a line onto the x axis (see Figure 2). By doing this iteratively, we obtain the sequence (see [2, p. 254]

$$v_{k+1} = v_k - \frac{v_k - v_{k-1}}{E(v_k) - E(v_{k-1})} E(v_k) \qquad k = 1, 2, \dots \tag{7}$$

The convergence order of the secant method is $(\sqrt{5}+1)/2$.

- ***Newton method:*** To determine a zero of (2) with the Newton method, we need an initial guess v_0, the corresponding $E(v_0)$ and also $E'(v_0)$, i.e., the first derivative of the error function (2) evaluated at v_0. This allows us to initialize the iterative formula [2, p. 255]

$$v_{k+1} = v_k - \frac{E(v_k)}{E'(v_k)} \tag{8}$$

The first derivative of $E(v)$ is usually not available, but it can be estimated numerically based on samples of v that are sufficiently close, e.g., by using a finite-difference formula. A better (more accurate) approach relies on deriving an evolution equation for $dy(t;v)/dv$, solve such such equation, and evaluate the solution at final time to obtain

$$E'(v) = \frac{dy(T;v)}{dt}. \tag{9}$$

The evolution equation for $dy(t;v)/dv$ can be determined by differentiating (4) with respect to v. This yields the linear initial value problem

$$\begin{cases} \dfrac{d^2\eta}{dt^2} = \dfrac{\partial f}{\partial y'}\dfrac{d\eta}{dt} + \dfrac{\partial f}{\partial y}\eta \\ \eta(0;v) = 0 \\ \dfrac{d\eta(0;v)}{dt} = 1 \end{cases} \tag{10}$$

where

$$\eta(t;v) = \frac{dy(t;v)}{dv}. \tag{11}$$

Note, in fact, that if differentiate the ODE in (1) with respect to v we obtain

$$\frac{d^2y}{dt^2} = f\left(\frac{dy}{dt}, y, t\right) \qquad \Rightarrow \qquad \frac{d^2}{dt^2}\left(\frac{dy}{dv}\right) = \frac{\partial f}{\partial y'}\frac{d}{dt}\left(\frac{dy}{dv}\right) + \frac{\partial f}{\partial y}\frac{dy}{dv}. \tag{12}$$

The system (10) depends on the solution of (4), i.e., and it can be solved only if the solution to (4) is available[3]. In summary, to solve (1) with the Newton method we proceed as follows:

1. Choose v_0.
2. Solve the initial value problem

[3] If the ODE is linear then (10) can be solved independently of (4).

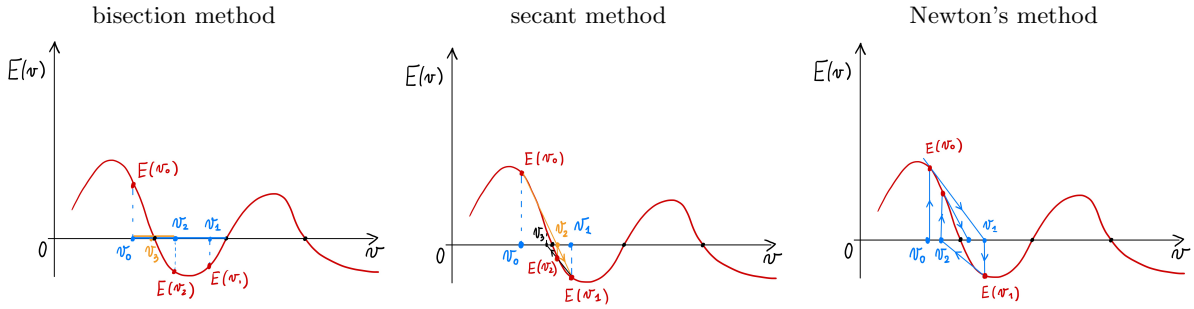


Figure 2: Sketch of the most common rootfinding methods applied to equation (2).

$$
\begin{cases}
\dfrac{d^2y}{dt^2} = f\left(\dfrac{dy}{dt}, y, t\right) \\
y(0;v) = \alpha \\
\dfrac{dy(0;v)}{dt} = v \\
\dfrac{d^2\eta}{dt^2} = \dfrac{\partial f}{\partial y'}\dfrac{d\eta}{dt} + \dfrac{\partial f}{\partial y}\eta \\
\eta(0;v) = 0 \\
\dfrac{d\eta(0;v)}{dt} = 1
\end{cases}
\tag{13}
$$

3. Evaluate $y(T;v_0)$ and $\eta(T;v) = dy(T;v)/dv$.

4. Update the initial guess v_0 as

$$
v_1 = v_0 - \frac{y(T;v_0) - \beta}{\eta(T;v_0)}. \tag{14}
$$

5. Go to point 2. and repeat the calculation with the updated initial condition $dy/dt(0;v) = v_1$.

Example (pendulum equations): Consider the two-point boundary value problem for the pendulum equation

$$
\begin{cases}
\dfrac{d^2\theta}{dt^2} = -\sin(\theta) \\
\theta(0) = \alpha \\
\theta(T) = \beta
\end{cases}
\tag{15}
$$

The system of equations (13) corresponding to the pendulum BVP (15) is

$$\begin{cases} \dfrac{d^2\theta}{dt^2} = -\sin(\theta) \\ \theta(0) = \alpha \\ \dfrac{d\theta(0)}{dt} = v_k \\ \dfrac{d^2\eta}{dt^2} = \cos(\theta)\eta \\ \eta(0;v) = 0 \\ \dfrac{d\eta(0;v)}{dt} = 1 \end{cases} \tag{16}$$

By solving this system and updating v_k according to

$$v_{k+1} = v_k - \frac{y(T;v_k) - \beta}{\eta(T;v_k)}, \tag{17}$$

for a properly chosen v_0, we eventually converge to one of the solutions of (15).

Remark (Optimization methods): A different class of techniques that can be used in the shooting method relies on *optimization*. In the optimization setting, we seek for a minimizer of the function

$$C(v) = (y(T;v) - \beta)^2, \tag{18}$$

at or nearby $C(v) = 0$. For example, we can use a descent method [2, p. 305] to minimize (18), e.g. the classical gradient descent scheme

$$v_{v+1} = v_k - \gamma_k C'(v_k), \tag{19}$$

where

$$\gamma_k = \left| \frac{v_k - v_{k-1}}{C'(v_k) - C'(v_{k-1})} \right|, \qquad C'(v_k) = 2E(v_k)E'(v_k). \tag{20}$$

The function $C'(v_k)$ is not known analytically, but needs to be evaluated as in the Newtown's method. In particular, each step of gradient descent requires the evaluation of both $E(v_k)$ and $E'(v_k)$.

It can be shown that the shooting method can be very sensitive to the coice of initial condition v_0. Indeed the set of initial conditions for which the method converges is often concentrated in a small neighborhood of the exact solution.

Shooting method for higher-order ODEs and system of ODEs. A two-point boundary value problem for a system of n-dimensional ODEs can be written in the abstract form

$$\begin{cases} \dfrac{d\boldsymbol{y}}{dt} = \boldsymbol{f}(\boldsymbol{y}, t) \qquad t \in [0, T] \\ \boldsymbol{g}(\boldsymbol{y}(0), \boldsymbol{y}(T)) = \boldsymbol{0} \end{cases} \tag{21}$$

where $\boldsymbol{g} \in \mathbb{R}^n$ is, in general, a nonlinear function. Every two point boundary value problem we considered so far can be written in this form, upon definition of appropriate phase variables and boundary function $\boldsymbol{g}$. To show this, let us provide an example of a boundary value problem involving a fourth-order ODE.

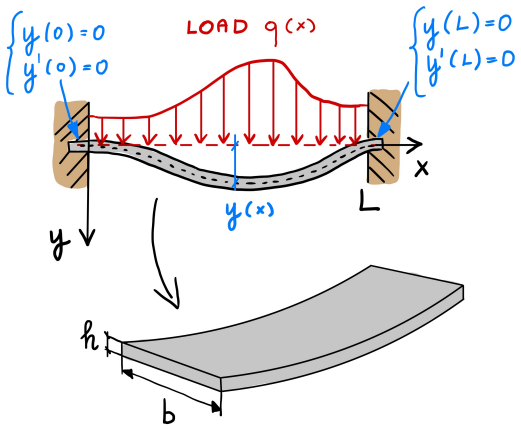


Figure 3: Sketch of the fully clamped Euler-Bernoulli beam modeled by the two-point boundary value problem (21).

Example (Euler-Bernoulli beam equations): An example of a boundary value problem in the form (21) is the equation describing the displacement of a fully clamped Euler-Bernoulli beam subject to a load $q(x)$

$$\begin{cases} EI\dfrac{d^4y}{dx^4} = q(x) \\ y(0) = 0 \\ y(L) = 0 \\ \dfrac{dy(0)}{dx} = 0 \\ \dfrac{dy(L)}{dx} = 0 \end{cases} \tag{22}$$

Here, E is the modulus of elasticity[4] of the beam, and I is the flexural moment of inertia. For a square section of thickness h and width b (see Figure 3) we have

$$I = \frac{bh^3}{12}. \tag{23}$$

Upon definition of

$$z_0(x) = y(x), \qquad z_1(x) = \frac{dz_0(x)}{dx} \qquad z_2(x) = \frac{dz_1(x)}{dx}, \qquad z_3(x) = \frac{dz_2(x)}{dx} \tag{24}$$

we can rewrite (22) as

$$\begin{cases} \dfrac{dz_3}{dx} = \dfrac{q(x)}{EI}, \qquad \dfrac{dz_2}{dx} = z_3(x), \qquad \dfrac{dz_1}{dx} = z_2(x), \qquad \dfrac{dz_0}{dx} = z_1(x) \\ z_0(0) = 0 \\ z_0(L) = 0 \\ z_1(0) = 0 \\ z_1(L) = 0 \end{cases} \tag{25}$$

[4]For stainless steel we have $E \simeq 200$ GPa.

i.e., as a system of four first-order ODEs with four simple boundary condition conditions involving z_0 and z_1 at $x = 0$ and $x = L$. To solve (25) with the shooting method, e.g., by using Newton's iterations, we proceed as follows. We first replace the boundary value problem with the initial value problem

$$\begin{cases} \dfrac{dz_3}{dx} = \dfrac{q(x)}{EI}, \qquad \dfrac{dz_2}{dx} = z_3(x), \qquad \dfrac{dz_1}{dx} = z_2(x), \qquad \dfrac{dz_0}{dx} = z_1(x) \\ z_0(0) = 0 \\ z_1(0) = 0 \\ z_2(0) = v_1 \\ z_3(0) = v_2 \end{cases} \tag{26}$$

depending on two unknown parameters v_1 and v_2. To determine these parameters, we define the vector-valued error function

$$\boldsymbol{E}(\boldsymbol{v}) = \begin{bmatrix} z_0(T;\boldsymbol{v}) - 0 \\ z_1(T;\boldsymbol{v}) - 0 \end{bmatrix}, \qquad \boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}. \tag{27}$$

Clearly, the dependence of $z_0(T;\boldsymbol{v})$ and $z_1(T;\boldsymbol{v})$ on $\boldsymbol{v}$ is affine (flow map generated by a linear system driven by $q(x)$). This allows us to avoid Newton's iterations and solve the linear system

$$\boldsymbol{E}(\boldsymbol{v}) = \boldsymbol{0} \tag{28}$$

for $\boldsymbol{v}$. Note that $z_0(T;\boldsymbol{v})$ and $z_1(T;\boldsymbol{v})$ involve integrals of $q(x)$, and therefore the solution to (28) is not the trivial vector $\boldsymbol{v} = 0$.

More generally, if the system is nonlinear, then we can use Newton's iterations to compute the solution to the problem. To this end, suppose we are given a boundary value problem for a fourth-order system of the form

$$\begin{cases} \dfrac{dz_3}{dx} = f(z_2, z_1, z_0), \qquad \dfrac{dz_2}{dx} = z_3(x), \qquad \dfrac{dz_1}{dx} = z_2(x), \qquad \dfrac{dz_0}{dx} = z_1(x) \\ z_0(0) = 0 \\ z_0(L) = 0 \\ z_1(0) = 0 \\ z_1(L) = 0 \end{cases} \tag{29}$$

We rewrite this system as an initial value problem with unknown v_1 and v_2

$$\begin{cases} \dfrac{dz_3}{dx} = f(z_2, z_1, z_0), \qquad \dfrac{dz_2}{dx} = z_3(x), \qquad \dfrac{dz_1}{dx} = z_2(x), \qquad \dfrac{dz_0}{dx} = z_1(x) \\ z_0(0) = 0 \\ z_1(0) = 0 \\ z_2(0) = v_1 \\ z_3(0) = v_2 \end{cases} \tag{30}$$

Given an initial guess $\boldsymbol{v}_0 = \begin{bmatrix} v_{01} & v_{02} \end{bmatrix}^T$ we construct the sequence of iterates $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots\}$ satisfying

$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k - \boldsymbol{J}^{-1}(\boldsymbol{v}_k)\boldsymbol{E}(\boldsymbol{v}_k), \tag{31}$$

where $\boldsymbol{J}(\boldsymbol{v}_k)$ is the Jacobian of the error function (27)

$$\boldsymbol{J}(\boldsymbol{v}) = \begin{bmatrix} \dfrac{\partial E_1(\boldsymbol{v})}{\partial v_1} & \dfrac{\partial E_1(\boldsymbol{v})}{\partial v_2} \\ \dfrac{\partial E_2(\boldsymbol{v})}{\partial v_1} & \dfrac{\partial E_2(\boldsymbol{v})}{\partial v_2} \end{bmatrix} \tag{32}$$

evaluated at $\boldsymbol{v}_k$. As before, it is possible to derive evolution equations for the components of the Jacobian.

Example (Euler-Bernoulli beam equations in the framework of Newton's iterations): The Jacobian of the error function (27) for the Euler-Bernoulli beam model (26) has the form

$$\boldsymbol{J}(\boldsymbol{v}) = \begin{bmatrix} \dfrac{\partial z_0(L;\boldsymbol{v})}{\partial v_1} & \dfrac{\partial z_0(L;\boldsymbol{v})}{\partial v_2} \\ \dfrac{\partial z_1(L;\boldsymbol{v})}{\partial v_1} & \dfrac{\partial z_1(L;\boldsymbol{v})}{\partial v_2} \end{bmatrix}. \tag{33}$$

As shown hereafter, $\boldsymbol{J}(\boldsymbol{v})$ does not depend on $\boldsymbol{v}$. This means that with just one Newton's iteration we can compute the correct initial condition for the system, and therefore solve the problem with the shooting method. The evolution equations for the components of the Jacobian (33) are obtained by differentiating the system (26) with respect to v_1 and v_2. This yields

$$\frac{d}{dx}\frac{\partial z_3}{\partial v_i} = 0 \quad \text{and} \quad \frac{d}{dx}\frac{\partial z_j}{\partial v_i} = \frac{\partial z_{j+1}}{\partial v_i} \qquad j = 0,1,2 \qquad i = 1,2, \tag{34}$$

with initial conditions

$$\frac{\partial z_3}{\partial v_1} = 0, \qquad \frac{\partial z_3}{\partial v_2} = 1, \qquad \frac{\partial z_2}{\partial v_1} = 1, \qquad \frac{\partial z_2}{\partial v_2} = 0, \tag{35}$$

and

$$\frac{\partial z_1}{\partial v_1} = 0, \qquad \frac{\partial z_1}{\partial v_2} = 0, \qquad \frac{\partial z_0}{\partial v_1} = 0, \qquad \frac{\partial z_0}{\partial v_2} = 0. \tag{36}$$

Clearly, the solution to the system (34)-(36) does not depend on $\boldsymbol{v}$ and therefore the Jacobian (33) does not depend on $\boldsymbol{v}$. This is just another way to say that we can solve the shooting problem for the linear Euler-Bernoulli beam by just one Newton iteration as

$$\boldsymbol{v}_1 = \boldsymbol{v}_0 - \boldsymbol{J}^{-1}\boldsymbol{E}(\boldsymbol{v}_0), \tag{37}$$

where $\boldsymbol{v}_0$ is any initial guess, $E(\boldsymbol{v}_0)$ is defined in (27) and $\boldsymbol{J}$ is the Jacobian (33).

Finite difference methods for BVP. To solve a two-point boundary value problem with finite difference methods we simply discretize its solution on a grid and replace the derivatives appearing in the ODE and the boundary conditions with appropriate finite-difference formulas. To illustrate this process let us first consider the simple prototype problem

$$\begin{cases} \dfrac{d^2y(x)}{dx^2} = f(x) \qquad x \in [0,1] \\ y(0) = \alpha \\ y(1) = \beta \end{cases} \tag{38}$$

Let $\{x_0,\ldots,x_{N+1}\}$ be $N+2$ evenly-spaced grid points in the interval $[0,1]$, i.e.,

$$x_j = j\Delta x, \qquad \Delta x = \frac{1}{N+1}, \qquad j=0,\ldots,N+1. \tag{39}$$

We approximate the second derivative d^2y/dx^2 in (38), e.g., by using the second-order finite difference formula

$$\frac{d^2y(x_j)}{dx^2} \simeq \frac{y_{j-1}-2y_j+y_{j+1}}{\Delta x^2}, \qquad u_j = u(x_j). \tag{40}$$

A substitution of (40) into (38) yields the system of equations[5]

$$\begin{cases} \dfrac{u_{j-1}-2u_j+u_{j+1}}{\Delta x^2} = f_j \qquad j=1,\ldots,N \\ u_0 = \alpha \\ u_{N+1} = \beta \end{cases} \tag{41}$$

where we defined $f_j = f(x_j)$. The system (41) can be written compactly as

$$\boldsymbol{D}_{\text{FD}}^2 \boldsymbol{u} = \boldsymbol{f}, \tag{42}$$

where

$$\boldsymbol{D}_{\text{FD}}^2 = \frac{1}{\Delta x^2}\begin{bmatrix} -2 & 1 & 0 & 0 & \cdots & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & 1 & -2 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 & -2 \end{bmatrix}, \qquad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix}, \qquad \boldsymbol{f} = \begin{bmatrix} f_1 - \alpha/\Delta x^2 \\ f_2 \\ f_3 \\ \vdots \\ \vdots \\ f_{N-1} \\ f_N - \beta/\Delta x^2 \end{bmatrix}. \tag{43}$$

The differentiation matrix $\boldsymbol{D}_{\text{FD}}^2$ corresponding to the second-order finite difference approximation is tridiagonal, diagonally dominant and negative definite. In fact, the eigenvalues of $\boldsymbol{D}_{\text{FD}}^2$ are

$$\lambda_k = \frac{2}{\Delta x^2}\left(\cos(k\pi\Delta x) - 1\right). \tag{44}$$

Clearly, $\lambda_k < 0$ for all $k = 1,\ldots,N$. This implies that matrix $\boldsymbol{D}_{\text{FD}}^2$ is invertible[6] and therefore the system (42) has a unique solution.

Remark: To prove that (44) are indeed the eigenvalues of $\boldsymbol{D}_{\text{FD}}^2$, consider the eigenvalue problem

$$\boldsymbol{D}_{\text{FD}}^2 \boldsymbol{u} = \lambda \boldsymbol{u}, \tag{45}$$

i.e.,

$$\frac{u_{j-1}-2u_j+u_{j+1}}{\Delta x^2} = \lambda u_j \quad \Rightarrow \quad u_{j+1} = (2+\Delta x^2\lambda)u_j - u_{j-1} \quad \text{with} \quad u_0 = u_{N+1} = 0 \tag{46}$$

Setting $2Q = (2+\Delta x^2\lambda)$ and rescaling all equations so that $u_1 = 1$ yields

$$\begin{cases} u_0 = 0 \\ u_1 = 1 \\ u_{j+1} = 2Qu_j - u_{j-1} \end{cases} \tag{47}$$

[5] In equation (41) u_j represents the numerical approximation of $y_j = y(x_j)$.

[6] Recall that the determinant of a matrix is the product of the eigenvalues.

Equation (47) is the three-term recurrence relation satisfied by the Chebyshev polynomials of the second kind $U_j(Q)$ in the variable Q. Setting $u_{j+1} = U_j(Q)$ and using the boundary condition $u_{N+1} = 0$ we have that $U_N(Q) = 0$. Hence, Q must be a root of the N-th degree Chebyshev polynomial of the second kind. Such roots are

$$Q_k = \cos\left(\frac{k\pi}{N+1}\right) = \cos\left(k\pi\Delta x\right), \qquad k = 1\ldots, N \tag{48}$$

Recalling that $2Q_k = (2 + \Delta x^2\lambda_k)$ yields

$$\lambda_k = \frac{2}{\Delta x^2}\left(\cos\left(k\pi\Delta x\right) - 1\right). \tag{49}$$

- ***Convergence Analysis:*** We now perform the convergence analysis of the finite difference approximation (42). To this end, we need to show that the error between the analytical solution of (38) and the numerical solution goes to zero as we send Δx to zero, i.e., as we consider more and more evenly-spaced grid points in the interval $[0, 1]$. Let $y_j = y(x_j)$ and u_j be, respectively, the analytical solution and the finite-differences solution of (38). We define the error

$$\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{u}, \tag{50}$$

where

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \qquad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix}. \tag{51}$$

By applying $\boldsymbol{D}_{\text{FD}}^2$ to $\boldsymbol{e}$ we obtain

$$\boldsymbol{D}_{\text{FD}}^2\boldsymbol{e} = \boldsymbol{\tau}, \tag{52}$$

where

$$\tau_j = \frac{y_{j-1} - 2y_j + 2y_{j+1}}{\Delta x^2} - f_j \qquad j = 1, \ldots, N \tag{53}$$

is the *local truncation error* (LTE) associated with the finite-difference approximation under consideration[7]. At this point we recall that the matrix $\boldsymbol{D}_{\text{FD}}^2$ is symmetric and invertible. This allows us to express the error $\boldsymbol{e}$ in equation (52) explicitly in terms of the truncation error $\boldsymbol{\tau}$ as

$$\boldsymbol{e} = \left(\boldsymbol{D}_{\text{FD}}^2\right)^{-1}\boldsymbol{\tau}. \tag{55}$$

By taking the 2-norm of this expression we obtain

$$\|\boldsymbol{e}\|_2 \leq \left\|\left(\boldsymbol{D}_{\text{FD}}^2\right)^{-1}\right\|_2 \|\boldsymbol{\tau}\|_2, \tag{56}$$

where the matrix 2-norm $\left\|\left(\boldsymbol{D}_{\text{FD}}^2\right)^{-1}\right\|_2$ induced by the vector 2-norm coincides with the largest singular value of the matrix $\left(\boldsymbol{D}_{\text{FD}}^2\right)^{-1}$. Recall that the inverse of a symmetric matrix is symmetric. This implies that the square root of the singular values of $\left(\boldsymbol{D}_{\text{FD}}^2\right)^{-1}$ matrix coincide with the absolute

[7] By using Taylor series we obtain

$$\begin{aligned} \tau_j =& \frac{y_{j-1} - 2y_j + y_{j+1}}{\Delta x^2} - f_j \\ =& \frac{d^2y(x_j)}{dx^2} - f_j + \frac{\Delta x^2}{12}\frac{d^4y(x_j)}{dx^4} + \cdots \\ =& \frac{\Delta x^2}{12}\frac{d^4y(x_j)}{dx^4} + \cdots. \end{aligned} \tag{54}$$

Therefore the local truncation error goes to zero as Δx^2.

values of the eigenvalues[8] of $\left(\boldsymbol{D}^2_{\text{FD}}\right)^{-1}$. Moreover the eigenvalues of the inverse matrix are the inverses of the eigenvalues of the matrix. This proves the following equality

$$\left\|\left(\boldsymbol{D}^2_{\text{FD}}\right)^{-1}\right\|_2 = \max_{k=1,\dots,N}\left|\frac{1}{\lambda_k}\right| = \frac{1}{\min\limits_{k=1,\dots,N}|\lambda_k|}, \tag{58}$$

where λ_k are the eigenvalues of $\boldsymbol{D}^2_{\text{FD}}$. By using equation (44) we see that

$$\min_{k=1,\dots,N}|\lambda_k| = |\lambda_1| = \frac{2}{\Delta x^2}\left|\cos(\pi\Delta x) - 1\right|. \tag{59}$$

Moreover, for sufficiently small Δx we can expand (59) in a Taylor series to obtain

$$|\lambda_1| = \frac{2}{\Delta x^2}\left|1 - \frac{\pi^2\Delta x^2}{2} + \frac{\pi^4\Delta x^4}{24}\cdots - 1\right| = \pi^2\left|1 - \frac{\pi^2}{12}\Delta x^2 + \cdots\right|. \tag{60}$$

Therefore, in the limit $\Delta x \to 0$ we have

$$\|\boldsymbol{e}\|_2 \leq \frac{1}{\pi^2}\|\boldsymbol{\tau}\|_2 = \frac{\Delta x^2}{12\pi^2}\sqrt{\sum_{k=1}^{N}\left[\frac{d^4y(x_k)}{dx^4}\right]^2} = \frac{\Delta x^{3/2}}{12\pi^2}\sqrt{\sum_{k=1}^{N}\Delta x\left[\frac{d^2f(x_k)}{dx^2}\right]^2}, \tag{61}$$

where we used equation (54) for the 2-norm of the local truncation error. Note that the quantity under the square root at the right hand side of (61) converges to the integral of the square of the second derivative of $f(x)$ in the limit $N \to \infty$, i.e.,

$$\lim_{N\to\infty}\sum_{k=1}^{N}\Delta x\left[\frac{d^2f(x_k)}{dx^2}\right]^2 = \int_0^1\left[\frac{d^2f(x)}{dx^2}\right]^2 dx. \tag{62}$$

Assuming that such an integral is finite, i.e., that the second derivative of f is square integrable in $[0,1]$, we conclude that the second-order finite difference approximation (41) of the boundary value problem (38) is convergent with order $3/2$ in Δx. Similarly, in the uniform norm we obtain

$$\|\boldsymbol{e}\|_\infty \leq \|\boldsymbol{e}\|_2 \leq \frac{1}{\pi^2}\|\boldsymbol{\tau}\|_2 \leq \sqrt{N}\|\boldsymbol{\tau}\|_\infty \simeq \frac{\sqrt{N}}{12\pi^2(N+1)^2}\left\|\frac{d^2f(x)}{dx^2}\right\|_\infty. \tag{63}$$

Clearly, in the limit $N \to \infty$ we have that $\|\boldsymbol{e}\|_\infty$ goes to zero as $1/N^{3/2}$.

Remark (Neumann boundary conditions): Consider the boundary value problem

$$\begin{cases}\dfrac{d^2y(x)}{dx^2} = f(x) \qquad x \in [0,1]\\[2ex] \dfrac{dy(0)}{dx} = \alpha\\[2ex] y(1) = \beta\end{cases} \tag{64}$$

How do we impose the Neumann boundary condition $dy(0)/dx = \alpha$ in a finite difference setting? The simplest way is to use forward finite differences, e.g.,

$$\frac{dy(0)}{dx} \simeq \frac{-3y_0 + 4y_1 - y_2}{2\Delta x} = \alpha. \tag{65}$$

[8] Recall that for any symmetric matrix

$$\|\boldsymbol{A}\|_2 = \max_j\sqrt{\lambda_j(\boldsymbol{A}^T\boldsymbol{A})} = \max_j\sqrt{\lambda_j(\boldsymbol{A}^2)} = \max|\lambda_j(\boldsymbol{A})|. \tag{57}$$

In this way, we can write the fully discrete finite difference system as

$$\frac{1}{\Delta x^2}\begin{bmatrix} -3\Delta x/2 & 2\Delta x & -\Delta x/2 & 0 & \cdots & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & 1 & -2 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 & -2 \end{bmatrix}\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} = \begin{bmatrix} \alpha \\ f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_{N-1} \\ f_N - \beta/\Delta x^2 \end{bmatrix}. \tag{66}$$

Remark (nonlinear BVP): Consider the nonlinear boundary value problem

$$\begin{cases} \dfrac{d^2y(x)}{dx^2} = f\left(\dfrac{dy}{dx}, y, x\right) \qquad x \in [0,1] \\ y(0) = \alpha \\ y(1) = \beta \end{cases} \tag{67}$$

A second-order finite difference approximation of (67) is

$$\begin{cases} \dfrac{u_{j-1} - 2u_j + u_{j+1}}{\Delta x^2} = f\left(\dfrac{u_{j+1} - u_{j-1}}{2\Delta x}, u_j, x_j\right) \qquad j = 1, \ldots, N \\ u_0 = \alpha \\ u_{N+1} = \beta \end{cases} \tag{68}$$

This is a system of N nonlinear equations in N unknowns $\{u_1, \ldots u_N\}$ which can be solved, e.g., with the Newton's method.

We conclude this section by emphasizing that we could have used also higher-order finite difference formulas to solve the problem (67) or (38). For instance, we could have used a fourth-order formula based on a stencil with 5 points, with forward and backward representation at the left and the right boundary, respectively, to accommodate Dirichlet or Neumann boundary conditions.

Method of weighted residuals for BVP. The method of weighted residuals for BVP is based on the so-called weak (or variational) formulation of the problem. To describe the method, consider the following prototype boundary value problem

$$\begin{cases} -\dfrac{d}{dx}\left(a(x)\dfrac{dy(x)}{dx}\right) + b(x)(x) = f(x) \qquad x \in [0,1] \\ y(0) = 0 \\ y(1) = 0 \end{cases} \tag{69}$$

where $a(x)$ is a strictly positive function, i.e., $a(x) > 0$ for all $x \in [0,1]$. Multiply the differential equation in (69) by a test function $v(x)$ and integrate over $[0,1]$ to obtain

$$-\int_0^1 \frac{d}{dx}\left(a(x)\frac{dy(x)}{dx}\right)v(x)dx + \int_0^1 b(x)y(x)v(x)dx = \int_0^1 f(x)v(x)dx. \tag{70}$$

By integrating the first term by parts and assuming that $v(x)$ satisfies the boundary condition in (69) we obtain

$$\int_0^1 a(x)\frac{dy(x)}{dx}\frac{dv(x)}{dx}dx + \int_0^1 b(x)y(x)v(x)dx = \int_0^1 f(x)v(x)dx. \tag{71}$$

Clearly, if $y(x)$ is of class $C^2([0,1])$ and it satisfies (69) (strong solution) then $y(x)$ is also a solution to the following *weak formulation* of the BVP:

Find $y \in H_0^1([0,1])$ such that (71) is satisfied for all $v \in H_0^1([0,1])$. Here $H_0^1([0,1])$ denotes the Sobolev space square integrable functions vanishing at $x = 0$ and $x = 1$, with square integrable first-order derivatives, i.e.,

$$H_0^1([0,1]) = \{v \in L^2([0,1]) \quad \text{such that} \quad \frac{dv}{dx} \in L^2([0,1]) \quad \text{and} \quad v(0) = v(1) = 0\}. \tag{72}$$

Note that the weak formulation (71) involves only the first derivative of $y(x)$. Therefore a weak solution to the BVP (69), i.e., a solution to (71), may not actually satisfy (69). This means that the weak formulation of a BVP might have a solution even when the strong formulation does not.

With the weak formulation (71) available, we look for a finite-dimensional approximation of $y(x)$. To this end, suppose that $y(x)$ can be accurately represented in a finite-dimensional subspace of $V_N \subset H_0^1([0,1])$, i.e.,

$$y_N(x) = \sum_{k=1}^N a_k\varphi_k(x), \qquad \varphi_k \in H_0^1([0,1]). \tag{73}$$

A substitution of (73) into (71) yields a residual $R_N(x)$

$$\int_0^1 a(x)\frac{dy_N(x)}{dx}\frac{dv(x)}{dx}dx + \int_0^1 b(x)y_N(x)v(x)dx = \int_0^1 f(x)v(x)dx + \int_0^1 R_N(x)v(x)dx. \tag{74}$$

Depending on the way we handle the residual we can have different classes of methods:

- **Galerkin method:** We set the residual orthogonal to the span of $\{\varphi_1, \ldots, \varphi_N\}$, i.e.,

$$\int_0^1 R_N(x)\varphi_j(x)dx = 0 \qquad j = 1, \ldots, N. \tag{75}$$

  This yields a system of N equations in the N unknowns $\{a_1, \ldots, a_N\}$ (see equation (73)).

- **Collocation method:** We set the residual $R_N(x)$ equal to zero at a set of collocation nodes $\{x_1, \ldots, x_N\}$, i.e.,

$$R_N(x_j) = 0 \qquad j = 1, \ldots, N. \tag{76}$$

  In this way, the differential equation is satisfied exactly at the collocation nodes.

- **Least-squares method:** We minimize the L^2 norm of the residual $R_N(x)$ over the parameters $\{a_1, \ldots, a_N\}$ in equation (73))

$$\min_{a_1,\ldots,a_N} \int_0^1 R_N^2(x)dx \tag{77}$$

As an example, let us apply all these methods to the simple BVP (38). To this end, we look for a representation of the solution in the form

$$y_N(x) = (1-x)\alpha + x\beta + \sum_{k=1}^N a_k\varphi_k(x) \tag{78}$$

where

$$\varphi_k(x) = x(1-x)T_k(2x-1) \qquad x \in [0,1] \tag{79}$$

and $T_k(x)$ are, e.g., Chebyshev polynomials of the first kind or Legendre polynomials. The linear functions $(1-x)$ and x are called *boundary modes* in finite-element analysis, while $\varphi_k(x)$ are called *interior modes*[9]. A substitution of (78) into (38) yields

$$\frac{d^2 y_N(x)}{dx^2} = f(x) + R_N(x). \tag{81}$$

Clearly, the boundary conditions are automatically satisfied by (78)-(79).

Galerkin method. In the Galerkin method we impose that the residual $R_N(x)$ is orthogonal (in the L^2 sense) to the span of $\{\varphi_1, \ldots, \varphi_N\}$. To impose such orthogonality, we first multiply (81) by $\varphi_j(x)$ and integrate over $[0,1]$ to obtain

$$\int_0^1 \frac{d^2 y_N(x)}{dx^2}\varphi_j(x)dx = \int_0^1 f(x)\varphi_j(x)dx + \int_0^1 R_N(x)\varphi_j(x)dx \qquad j = 1, \ldots, N. \tag{82}$$

Setting

$$\int_0^1 R_N(x)\varphi_j(x)dx = 0 \qquad j = 1, \ldots, N \tag{83}$$

yields the system of equations

$$-\int_0^1 \frac{dy_N(x)}{dx}\frac{d\varphi_j(x)}{dx}dx = \int_0^1 f(x)\varphi_j(x)dx \qquad j = 1, \ldots, N. \tag{84}$$

where we integrated by parts the first term in (82). Substituting (78) and

$$\frac{dy_N(x)}{dx} = \beta - \alpha + \sum_{k=1}^{b} a_k \frac{d\varphi_k(x)}{dx} \tag{85}$$

into (84) yields

$$-(\beta-\alpha)\int_0^1 \frac{d\varphi_j(x)}{dx}dx - \sum_{k=1}^{N} a_k \underbrace{\int_0^1 \frac{d\varphi_j(x)}{dx}\frac{d\varphi_k(x)}{dx}dx}_{\text{stiffness matrix } S_{jk}} = \int_0^1 f(x)\varphi_j(x)dx \qquad j = 1, \ldots, N. \tag{86}$$

i.e.,

$$\sum_{k=1}^{N} S_{jk}a_k = -\int_0^1 f(x)\varphi_j(x)dx - (\beta-\alpha)\int_0^1 \frac{d\varphi_j(x)}{dx}dx \qquad j = 1, \ldots, N. \tag{87}$$

Upon definition of

$$\boldsymbol{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix}, \qquad \boldsymbol{h} = \begin{bmatrix} \int_0^1 f(x)\varphi_1(x)dx \\ \vdots \\ \int_0^1 f(x)\varphi_N(x)dx \end{bmatrix}, \qquad \boldsymbol{b} = (\beta-\alpha)\begin{bmatrix} \int_0^1 \frac{d\varphi_1(x)}{dx}dx \\ \vdots \\ \int_0^1 \frac{d\varphi_N(x)}{dx}dx \end{bmatrix}, \tag{88}$$

[9] In the *spectral method* the basis fuctions $\varphi_k(x)$ are chosen to be Lagrange characteristic polynomials at Gauss-Lobatto nodes in $[0,1]$. This yields the following expansion of the solution

$$y_N(x) = \varphi_0(x)\alpha + \varphi_{N+1}(x)\beta + \sum_{k=1}^{N} a_k\varphi_k(x), \tag{80}$$

where $\varphi_0(x)$ and $\varphi_{N+1}(x)$ are the boundary modes and $\varphi_k(x)$ are the interior modes. The integrals in (82) can be computed using Gauss-Lobatto quadrature.

we can write the system (87) in a matrix-vector form as

$$\boldsymbol{S}\boldsymbol{a} = -(\boldsymbol{h} + \boldsymbol{b}) \tag{89}$$

Inverting the (positive definite) stiffness matrix $\boldsymbol{S}$ yields the solution $\boldsymbol{a} = -\boldsymbol{S}^{-1}(\boldsymbol{f} + \boldsymbol{b})$ which can be then substituted back into (78).

Collocation method. In the collocation method we impose that the residual $R_N(x)$ is equal to zero at N distinct (interior) nodes $\{x_1, \ldots, x_N\}$. A substitution of (73) into (69) yields (81). By imposing $R_N(x_i) = 0$ in (81) we obtain

$$\sum_{k=2}^{N-1} a_k \frac{d^2\varphi_k(x_j)}{dx^2} = \underbrace{f(x_j) - \alpha\frac{d^2\varphi_0(x_j)}{dx^2} - \beta\frac{d^2\varphi_{N+1}(x_j)}{dx^2}}_{g_j} \qquad j = 2, \ldots, N-1 \tag{90}$$

Upon definition of the differentiation matrix $D^2_{jk} = d^2\varphi_k(x_j)/dx^2$ we can write the linear system (90) as

$$\boldsymbol{D}^2_{\text{Coll}}\boldsymbol{a} = \boldsymbol{g}, \tag{91}$$

which resembles very much the system (42). Indeed the finite-difference methods is a particular type of collocation method. Convergence analysis of spectral collocation methods for BVP follows the analysis we have done for second order finite differences. In particular, the eigenvalues of second-order spectral collocation matrices are discussed in [4].

Least squares method. Finally, we set up the BVP (38) as a least-squares problem. To this end, we consider the residual

$$R_N(x) = \sum_{k=1}^{N} a_k \frac{d^2\varphi_k(x)}{dx^2} - f(x) \tag{92}$$

and its L^2 norm

$$\|R_N(x)\|^2_{L^2([0,1]} = \int_0^1 \left(\sum_{k=1}^{N} a_k \frac{d^2\varphi_k(x)}{dx^2} - f(x) \right)^2 dx. \tag{93}$$

By expanding integrand we see that minimization of (93) is essentially a quadratic programming problem with two linear constraints (boundary conditions) which can be solved, for example, using high-performance solvers such as OSQP [3] (`https://osqp.org/`).

Some results on polynomial approximation theory. Collocation, Galerkin and Least Squares methods are based on functional series expansions of the form

$$y_N = \sum_{k=0}^{N} a_k \varphi_k(x). \tag{94}$$

For simplicity we consider $x \in [-1, 1]$ (any bounded interval can be rescaled to $[-1, 1]$). The functions $\varphi_k(x)$ here are orthogonal polynomials relative to a weight function $w(x)$, i.e.,

$$\int_{-1}^{1} \varphi_k(x)\varphi_j(x)w(x)dx = \delta_{kj} \left\|\varphi_k\right\|^2_{L^2_w([-1,1])}. \tag{95}$$

or Lagrange polynomials associated to a properly chosen set of notes, e.g., zeros of orthogonal polynomials. Polynomial approximation theory theory is thoroughly discussed in [1, Chapter 6]. Hereafter we briefly summarize some of the main results.

Theorem 1. Let $\{\varphi_k(x)\}$ in (94) be a set polynomials orthogonal in $[-1,1]$ relative to the weight function $w(x)$. Then for any $y(x) \in H^p_w([-1,1])$ $p \geq 0$, there exists a constant C, independent of N, such that

$$\|y(x) - y_N(x)\|_{L^2_w([-1,1])} \leq CN^{-p} \|y(x)\|_{H^p_w([-1,1])} \tag{96}$$

Theorem 1 demonstrates that the error between the function $y(x)$ and the approximation (94) decays spectrally with the number of basis functions. If the function $y(x)$ is infinitely smooth, then the error decays exponentially fast with the number of basis functions. Similar results can be obtained the approximation of the derivatives of $y(x)$ (see [1, Theorem 6.2 and Theorem 6.3]).

The next theorem summarizes the approximation properties of *spectral collocation* representations, i.e., series expansions of the form (94) where $a_k = y(x_k)$ and $\varphi_k(x)$ are Lagrangian characteristic polynomials associated with a set of Gauss or Gauss-Lobatto nodes[10] $\{x_0, \ldots, x_N\}$ in $[-1,1]$.

Theorem 2. Let $\varphi_k(x)$ in (94) be Lagrange characteristic polynomials associated with a set of Gauss or Gauss-Lobatto nodes in $[-1,1]$. Suppose that such Gauss or Gauss Lobatto nodes are defined by a polynomial orthogonal in $[-1,1]$ relative to the weight function $w(x)$. Then for any $y(x) \in H^p_w([-1,1])$ $p \geq 1$, there exists a constant C, independent of N, such that (96) holds.

Theorem 2 demonstrates that contrary to finite difference methods, the error of spectral collocation methods does not decay as a fixed power of $1/N$ but rather as a power that depends on the smoothness of the function we are approximating. For infinitely differentiable function functions, the error decreases exponentially fast with N. Hence, spectral collocation methods are, in a certain sense, *methods of infinite order* when applied to smooth problems.

Gauss-Chebyshev-Lobatto spectral collocation method for BVP. Let briefly review the main ingredients of the Gauss-Lobatto Chebyshev expansion, and its usage for boundary value problems. For more details we refer to [1]. We first recall that the Chebyshev polynomials of the first kind are defined as

$$T_k(x) = \cos(k \arccos(x)) \qquad x \in [-1,1] \qquad \text{(trigonometric representation)}. \tag{97}$$

It can be shown that $T_k(x)$ (like any other orthogonal polynomial) satisfies a three-term recurrence relation

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_{n+1}(x) &= 2x\, T_n(x) - T_{n-1}(x). \end{aligned} \tag{98}$$

which gives

$$T_2(x) = 2x^2 - 1, \qquad T_3(x) = 4x^3 - 3x \qquad T_4(x) = 8x^4 - 8x^2 + 1, \ldots. \tag{99}$$

The Gauss-Chebyshev-Lobatto nodes are zeros of the polynomial

$$Q_{N+1}(x) = (1 - x^2)\frac{dT_N(x)}{dx}, \tag{100}$$

i.e., $x_0 = -1$, $x_N = 1$ and all maxima and minima of $T_N(x)$. By differentiating (97) with respect to x we obtain

$$\frac{dT_N(x)}{dx} = \frac{\sin(N \arccos(x))}{\sqrt{1 - x^2}}. \tag{101}$$

Hence $Q_{N+1}(x) = 0$ implies that

$$x_j = \cos\left(\frac{k\pi}{N}\right) \qquad j = 0, \ldots, N \qquad \text{(Gauss-Chebyshev-Lobatto points)}. \tag{102}$$

[10] Recall that *Gauss nodes* in $[-1,1]$ are the zeros of an orthogonal polynomial $P_{N+1}(x)$ of degree $N+1$ defined in $[-1,1]$. Orthogonality is relative to some weight function $w(x)$. If $w(x) = 1$ then $P_k(x)$ are Lagrange polynomials. On the other hand, *Gauss-Lobatto nodes* are zeros of the polynomial $(1 - x^2)dP_N(x)/dx$.

These points are obtained by dividing half unit circle in evenly spaced parts and projecting them onto the x-axis. It can be shown that the Lagrange characteristic polynomials associated with the Gauss-Chebyshev-Lobatto nodes are

$$\varphi_j(x) = \frac{(-1)^{N+j+1}(1-x^2)}{d_j N^2 (x-x_j)} \frac{dT_N(x)}{dx} = \frac{(-1)^{N+j+1}\sqrt{(1-x^2)}}{d_j N^2 (x-x_j)} \sin(N \arccos(x)), \tag{103}$$

where x_j is given in (102) and

$$d_0 = d_N = 2 \qquad d_1 = d_2 = \cdots = d_{N-1} = 1. \tag{104}$$

A substitution of (103) into (94) yields the series expansion[11]

$$y_N(x) = \sum_{k=0}^{N} y(x_k)\varphi_k(x). \tag{105}$$

At this point we can differentiate (105) with respect to x and evaluate the derivative at $x = x_j$. This yields the expressions

$$\frac{dy_N(x_j)}{dx} = \sum_{k=0}^{N} y(x_k) \underbrace{\frac{d\varphi_k(x_j)}{dx}}_{D^1_{jk}}, \qquad \frac{d^2y_N(x_j)}{dx^2} = \sum_{k=0}^{N} y(x_k) \underbrace{\frac{d^2\varphi_k(x_j)}{dx^2}}_{D^2_{jk}}, \tag{106}$$

where $\boldsymbol{D}^1$ and $\boldsymbol{D}^2$ are, respectively, first- and second-order Gauss-Chebyshev-Lobatto differentiation matrices. A direct calculation shows that

$$D^1_{ij} = \begin{cases} -\dfrac{2N^2+1}{6} & i = j = 0 \\ \dfrac{d_i}{d_j}\dfrac{(-1)^{i+j}}{x_i - x_j} & i \neq j \\ -\dfrac{x_i}{2(1-x_i^2)} & i = j \\ \dfrac{2N^2+1}{6} & i = j = N \end{cases} \qquad \text{(first-order differentiation matrix)} \tag{107}$$

where d_i are defined in (104), and

$$D^2_{ij} = \begin{cases} \dfrac{(-1)^{i+j}}{d_i}\dfrac{x_i^2 + x_i x_j - 2}{\left(1-x_i^2\right)\left(x_i - x_j\right)^2} & 1 \leq i \leq N-1, \quad 0 \leq j \leq N, \qquad j \neq i \\ -\dfrac{(N^2-1)(1-x_i^2)+3}{3\left(1-x_i^2\right)^2} & 1 \leq i = j \leq N-1 \\ \dfrac{2(-1)^j}{3d_j}\left[\dfrac{(2N^2+1)(1-x_j)-6}{\left(1-x_j\right)^2}\right] & i = 0, \quad 1 \leq j \leq N \\ \dfrac{2(-1)^{N+j}}{3d_j}\left[\dfrac{(2N^2+1)(1+x_j)-6}{\left(1+x_j\right)^2}\right] & i = N, \quad 0 \leq j \leq N-1 \\ \dfrac{(N^4-1)}{15} & i = j = 0, N \end{cases} \tag{108}$$

[11] Note that the series expansion (105), is the Lagrange interpolant of $y(x)$ through the Gauss-Chebyshev-Lobatto points (102).

The matrix $\boldsymbol{D}^2$ can be also approximated by a product of two matrices $\boldsymbol{D}^1$, i.e.,

$$\boldsymbol{D}^2 \simeq \boldsymbol{D}^1\boldsymbol{D}^1, \tag{109}$$

although $\boldsymbol{D}^2$ is obviously more accurate than $\boldsymbol{D}^1\boldsymbol{D}^1$.

Example: Chebyshev-Gauss-Lobatto nodes are defined in $[-1,1]$. If we are given a VBP on the interval $[a,b]$ then we can transform it to $[-1,1]$ by using the following elementary coordinate transformation

$$x = \frac{b-a}{2}z + \frac{b+a}{2} \qquad z \in [-1,1]. \tag{110}$$

This yields the following transformation for the derivatives

$$y(x) = y\left(\frac{b-a}{2}z + \frac{b+a}{2}\right) \qquad \Rightarrow \qquad \frac{dy}{dx} = \frac{dy}{dz}\frac{dz}{dx} = \frac{dy}{dz}\frac{2}{b-a}, \tag{111}$$

and

$$\frac{d^2y}{dx^2} = \frac{d^2y}{dz^2}\left(\frac{2}{b-a}\right)^2. \tag{112}$$

The last equation implies that the differentiation matrix for a function defined in $[a,b]$ is simply a re-scaled version of the differentiation matrix in $[-1,1]$, the rescaling factor being some power of $2/(b-a)$. Substituting (105) into (69) (mapped from $x \in [0,1]$ to $z \in [-1,1]$ using the simple transformation $z = 2x-1$) and setting the residual equal to zero at the nodes (102) yields the system of equations

$$\begin{cases} 2\displaystyle\sum_{k=0}^{N} D_{jk}^2 u_k = f\left(\frac{z_j+1}{2}\right) \qquad j = 1,\ldots,N-1 \\ u_0 = \alpha \\ u_N = \beta \end{cases} \tag{113}$$

where z_j are the Chebyshev-Gauss-Lobatto nodes (102). This system allows us to compute the numerical solution to the BVP (69) using the Chebyshev-Gauss-Lobatto spectral method.

References

[1] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb. *Spectral methods for time-dependent problems*, volume 21 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2007.

[2] A. Quarteroni, R. Sacco, and F. Salieri. *Numerical mathematics*. Springer, 2007.

[3] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020.

[4] J. A. C. Weideman and L. N. Trefethen. The eigenvalues of second-order spectral differentiation matrices. *SIAM Journal on Numerical Analysis*, 25(6):1279–1298, 1988.

Numerical methods for the heat equation

Consider the following initial-boundary value problem (IBVP) for the one-dimensional heat equation

$$\begin{cases} \dfrac{\partial U}{\partial t} = \alpha \dfrac{\partial^2 U}{\partial x^2} + q(x) \qquad t \geq 0 \qquad x \in [0, L] \\ U(x,0) = U_0(x) \\ U(0,t) = g_0(t) \\ U(L,t) = g_L(t) \end{cases} \tag{1}$$

where $q(x)$ is the internal heat generation and α the thermal diffusivity.

The IBVP (1) describes the propagation of temperature in a one-dimensional slab of width L initially at temperature $U_0(x)$ with Dirichlet boundary conditions $U(0,t) = g_0(t)$ and $U(L,t) = g_L(t)$. You have learned in AM 212A that it is possible to compute the analytical solution of the problem (1) using many different techniques. For example, if we set $q(x) = 0$, and $g_0(t) = g_L(t) = 0$ then it is easy to show that

$$U(x,t) = \frac{2}{L} \sum_{k=1}^{\infty} e^{-\alpha k^2 \pi^2 t / L^2} \sin\left(\frac{k\pi}{L} x\right) \int_0^L U_0(x) \sin\left(\frac{k\pi}{L} x\right) dx, \tag{2}$$

where $\sin(k\pi x/L)$ are eigenfunctions of the eigenvalue problem (see [1, p. 48])

$$\frac{d^2 X(x)}{dx^2} + \beta X(x) = 0, \qquad X(0) = 0, \qquad X(L) = 0, \tag{3}$$

with eigenvalues $\beta_k = k^2 \pi^2 / L^2$.

- ***Energy decay:*** It is straightforward to show that in the case of no heat generation and zero Dirichlet boundary conditions the $L^2([0,L])$ norm of the solution to (1) i.e.,

$$\|U\|^2_{L^2([0,L])} = \int_0^L U(x,t)^2 dx \tag{4}$$

decays monotonically to zero as time increases. This can be seen directly from the analytical solution (2). Alternatively, we can derive an evolution equation for (4) and solve it. To this end, let us multiply the heat equation by $U(x,t)$ and integrate it over the spatial domain $[0,L]$. This yields

$$\int_0^L U(x,t) \frac{\partial U(x,t)}{\partial t} dx = \alpha \int_0^L U(x,t) \frac{\partial^2 U(x,t)}{\partial x^2} dx. \tag{5}$$

By integrating by parts and recalling (4) we obtain

$$\frac{d}{dt} \|U\|^2_{L^2([0,L])} = -2\alpha \underbrace{\int_0^L \left(\frac{\partial U}{\partial x}\right)^2 dx}_{\left\|\frac{\partial U}{\partial x}\right\|^2_{L^2([0,L])}} + 2\alpha \underbrace{\left[U \frac{\partial U}{\partial x}\right]_{x=0}^{x=L}}_{=0}. \tag{6}$$

At this point we use the Poincaré inequality[1]

$$\left\|\frac{\partial U}{\partial x}\right\|^2_{L^2([0,L])} \geq C \|U\|^2_{L^2([0,L])} \quad \Leftrightarrow \quad -\left\|\frac{\partial U}{\partial x}\right\|^2_{L^2([0,L])} \leq -C \|U\|^2_{L^2([0,L])} \tag{7}$$

[1]The Poincaré inequality holds for all differentiable functions u with zero boundary conditions.

to obtain

$$\frac{d}{dt}\left\|U\right\|^2_{L^2([0,L])} + 2\alpha \left\|U\right\|^2_{L^2([0,L])} \leq 0 \qquad \Rightarrow \qquad \left\|U\right\|^2_{L^2([0,L])} \leq \left\|U_0\right\|^2_{L^2([0,L])} e^{-2\alpha C t}. \tag{8}$$

Hence the "energy" of the solution, i.e., the L^2 norm (4) decays to zero as $t \to \infty$.

Finite-difference methods. To solve the IBVP (1) with finite differences, let us consider the following an evenly-spaced grid in $[0, L]$, i.e.,

$$x_j = j\Delta x \qquad \Delta x = \frac{L}{N+1} \qquad j = 0, \ldots, N+1. \tag{9}$$

On this grid, we approximate the second derivative in (1) by using, e.g., the second-order finite difference formula

$$\left.\frac{\partial^2 U(x,t)}{\partial x^2}\right|_{x=x_j} \simeq \frac{U(x_{j-1},t) - 2U(x_j,t) + U(x_{j+1},t)}{\Delta x^2} \qquad j = 1, \ldots, N. \tag{10}$$

A substitution of (10) into (1) yields the so-called *semi-discrete* form[2]

$$\begin{cases} \dfrac{du_j}{dt} = \alpha \dfrac{u_{j-1}(t) - 2u_j(t) + u_{j+1}(t)}{\Delta x^2} + q(x_j) \qquad j = 1, \ldots, N \\ u_j(0) = U_0(x_j) \qquad j = 1, \ldots, N \\ u_0(t) = g_0(t) \\ U_{N+1}(t) = g_L(t) \end{cases} \tag{11}$$

where $u_j(t)$ represents an approximation of the exact solution $U(x_j, t)$, i.e., the exact solution evaluated at the grid point x_j. The system (11) can be written in a matrix-vector form as

$$\begin{cases} \dfrac{d\boldsymbol{u}}{dt} = \alpha \boldsymbol{D}^2_{\text{FD}} \boldsymbol{u} + \boldsymbol{h}(t) \\ \boldsymbol{u}(0) = \boldsymbol{U}_0 \end{cases} \tag{12}$$

where[3]

$$\boldsymbol{D}^2_{\text{FD}} = \frac{1}{\Delta x^2} \begin{bmatrix} -2 & 1 & 0 & 0 & \cdots & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & 1 & -2 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 & -2 \end{bmatrix}, \qquad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix}, \qquad \boldsymbol{h}(t) = \begin{bmatrix} q(x_1) + \alpha g_0(t)/\Delta x^2 \\ q(x_2) \\ q(x_3) \\ \vdots \\ \vdots \\ q(x_{N-1}) \\ q(x_N) + \alpha g_L(t)/\Delta x^2 \end{bmatrix}. \tag{13}$$

[2]The system (11) is called "semi-discrete" form of the IBVP (1) because we discretized only the dependence of the solution on the spatial variable x. If, in addition, we discretize (10) time using a time-stepping scheme then we obtain the so-called "fully discrete form" of the IBVP (1). The semi-discrete form (10) is also known as *method of lines* (MOL). The reason for such a definition is that the finite-difference solution of the heat equation is computed by solving a finite-dimensional system of ODEs, each one of which represents the dynamics of $U(x,t)$ at a particular grid point x_j. This corresponds to a "line" emanating from $U(x_j, 0)$.

[3]Recall that the differentiation matrix $\boldsymbol{D}^2_{\text{FD}}$ corresponding to the second-order finite difference discretization is tridiagonal and negative definite.

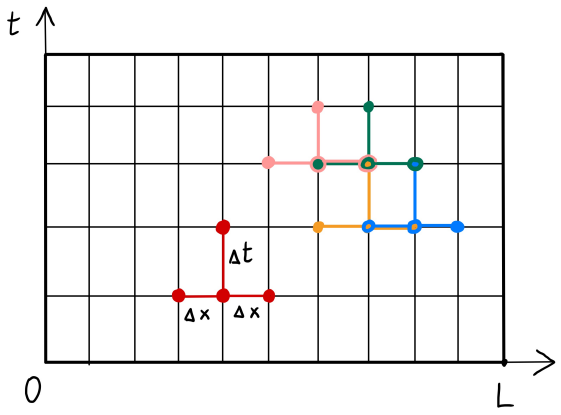


Figure 1: Finite-difference stencil corresponding to the forward-in-time centered-in-space discretization (17). We sketch the coupling of the system as we march forward in time by a few time steps.

In this way, we reduced the IBVP (1) to an *initial value problem* for a linear ODE, i.e., equation (12). Such an initial value problem can be solved using any time-stepping method we studied for initial value problems. For example, if we use the Euler forward scheme we obtain the *fully discrete form*

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \alpha \Delta t \boldsymbol{D}_{\text{FD}}^2 \boldsymbol{u}^k + \Delta t \boldsymbol{h}(t_k), \tag{14}$$

where

$$\boldsymbol{u}^k = \boldsymbol{u}(t_k). \tag{15}$$

On the other hand, if we use the two-step Adams-Bashforth method we obtain

$$\boldsymbol{u}^{k+2} = \boldsymbol{u}^{k+1} + \frac{\alpha \Delta t}{2} \left[3 \left(\boldsymbol{D}_{\text{FD}}^2 \boldsymbol{u}^{k+1} + \boldsymbol{h}^{k+1} \right) - \left(\boldsymbol{D}_{\text{FD}}^2 \boldsymbol{u}^k + \boldsymbol{h}^k \right) \right]. \tag{16}$$

Remark: Clearly, one could use higher-order finite-difference formulas to approximate the second-order derivative $\partial^2 U(x,t)/\partial x^2$. This yields other differentiation matrices, and requires some care when handling boundary conditions.

Local truncation error. The local truncation error (LTE) of a finite difference scheme is the residual arising when we ideally insert the exact solution to the problem into the fully discrete form. For illustration purposes let us compute the local truncation error of the so-called "centered in space forward in time" finite-difference scheme (see Figure 1)

$$\frac{u_j^{k+1} - u_j^k}{\Delta t} = \alpha \frac{u_{j-1}^k - 2u_j^k + u_{j+1}^k}{\Delta x^2} + q(x_j), \tag{17}$$

where u_k^j is an approximation of $U(x_j, t_k)$.

By plugging in the exact solution $U(x,t)$ into (17) we obtain the LTE

$$\tau(x_j, t_{k+1}) = \frac{U(x_j, t_{k+1}) - U(x_j, t_k)}{\Delta t} - \alpha \frac{U(x_{j-1}, t_k) - 2U(x_j, t_k) + U(x_{j+1}, t_k)}{\Delta x^2} - q(x_j). \tag{18}$$

Let us now define $U_j^k = U(x_j, t_k)$ and expand

$$U_j^{k+1} = U(x_j, t_k + \Delta t) \qquad \text{and} \qquad U_{j\pm 1}^k = U(x_j \pm \Delta x, t_k) \tag{19}$$

in Taylor series in Δx and Δt. This yields

$$\frac{U_j^{k+1}-U_j^k}{\Delta t}=\frac{1}{\Delta t}\left(\Delta t\frac{\partial U_j^k}{\partial t}+\frac{\Delta t^2}{2}\frac{\partial^2 U_j^k}{\partial t^2}+\cdots\right)=\frac{\partial U_j^k}{\partial t}+\frac{\Delta t}{2}\frac{\partial^2 U_j^k}{\partial t^2}+\cdots. \tag{20}$$

Similarly,

$$\begin{aligned}\frac{U_{j-1}^k-2U_j^k+U_{j+1}^k}{\Delta x^2}=&\frac{1}{\Delta x^2}\left(-\Delta x\frac{\partial U_j^k}{\partial x}+\frac{\Delta x^2}{2}\frac{\partial^2 U_j^k}{\partial x^2}-\frac{\Delta x^3}{6}\frac{\partial^3 U_j^k}{\partial x^3}+\frac{\Delta x^4}{24}\frac{\partial^4 U_j^k}{\partial x^4}+\cdots\right.\\&\left.\Delta x\frac{\partial U_j^k}{\partial x}+\frac{\Delta x^2}{2}\frac{\partial^2 U_j^k}{\partial x^2}+\frac{\Delta x^3}{6}\frac{\partial^3 U_j^k}{\partial x^3}+\frac{\Delta x^4}{24}\frac{\partial^4 U_j^k}{\partial x^4}+\cdots\right)\\=&\frac{\partial^2 U_j^k}{\partial x^2}+\frac{\Delta x^2}{12}\frac{\partial^4 U_j^k}{\partial x^4}+\cdots.\end{aligned} \tag{21}$$

Substituting (20)-(21) into (18), and using the PDE (1) yields[4]

$$\begin{aligned}\tau(x_j,t_{k+1})=&\underbrace{\frac{\partial U_j^k}{\partial t}-\alpha\frac{\partial^2 U_j^k}{\partial x^2}}_{q(x_j)}-q(x_j)+\frac{\Delta t}{2}\frac{\partial^2 U_j^k}{\partial t^2}-\alpha\frac{\Delta x^2}{12}\frac{\partial^4 U_j^k}{\partial x^4}+\cdots\\=&\left(\alpha\frac{\Delta t}{2}-\frac{\Delta x^2}{12}\right)\alpha\frac{\partial^4 U_j^k}{\partial x^4}+\cdots\end{aligned} \tag{23}$$

where we replaced $\partial^2 U_j^k/\partial t^2$ with $\alpha^2\partial^4 U_j^k/\partial x^4$ using the equation

$$\frac{\partial U}{\partial t}=\alpha\frac{\partial^2 U}{\partial x^2}+q(x)\quad\Rightarrow\quad\frac{\partial U^2}{\partial t^2}=\alpha\frac{\partial^2}{\partial x^2}\left(\frac{\partial U}{\partial t}\right)=\alpha^2\frac{\partial^4 U}{\partial x^4}. \tag{24}$$

The local truncation error goes to zero linearly in Δt and quadratically in Δx. Therefore the centered in space forward in time scheme (17) is consistent with order one in Δt and order two in Δx.

By following exactly the same steps it is possible to derive an expression for the local truncation error of finite-difference schemes involving different spatial and temporal discretizations. For example, we could have used a stencil with 5 points in space and the BDF3 method in time.

Absolute stability analysis of finite-difference methods. Consider the IBVP (1) with $q(x)=0$ and zero Dirichlet boundary conditions. The second-order finite-differences discretization of such problem is given by the system (12) with $\boldsymbol{h}(t)=\boldsymbol{0}$, i.e.,

$$\begin{cases}\dfrac{d\boldsymbol{u}}{dt}=\alpha\boldsymbol{D}_{\rm FD}^2\boldsymbol{u}\\ \boldsymbol{u}(0)=\boldsymbol{U}_0\end{cases} \tag{25}$$

Recall that the matrix $\boldsymbol{D}_{\rm FD}^2$ is negative definite with simple (real) eigenvalues

$$\lambda_k=\frac{2}{\Delta x^2}\left(\cos(k\pi\Delta x)-1\right)\qquad k=1,\ldots,N \tag{26}$$

[4] In (23) we replaced $\partial^2 U_j^k/\partial t^2$ with $\alpha^2\partial^4 U_j^k/\partial x^4$. Such equality follows from the identity

$$\frac{\partial U}{\partial t}=\alpha\frac{\partial^2 U}{\partial x^2}+q(x)\quad\Rightarrow\quad\frac{\partial^2 U}{\partial t^2}=\alpha\frac{\partial^2}{\partial x^2}\frac{\partial U}{\partial t}=\alpha^2\frac{\partial^4 U}{\partial x^4}. \tag{22}$$

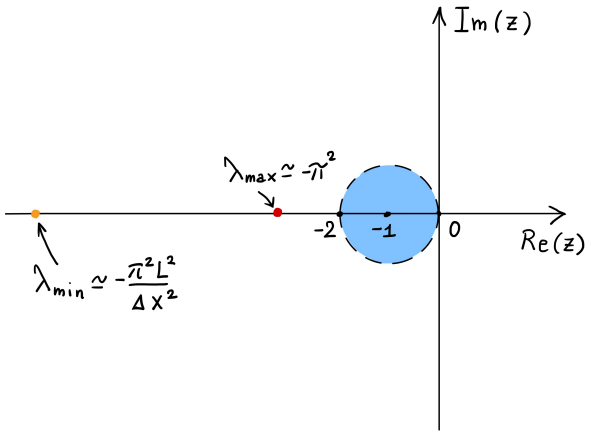


Figure 2: Absolute stability analysis of second-order finite-differences to solve the heat equation (1) with $q(x) = 0$ and zero Dirichlet boundary conditions. Shown are the smallest and largest eigenvalues of the differentiation matrix defined in (13) and the region of absolute stability of the Euler forward method. The fully discrete form of the heat equation (31) is absolutely stable if and only if $\Delta t < 2\Delta x^2/(\alpha\pi^2 L^2)$.

Since $\lambda_k < 0$ we have that the linear dynamical system (25) has a globally attracting stable node at the origin $\boldsymbol{u} = \boldsymbol{0}$. For small Δx (i.e., large number of spatial points) we obtain

$$\lambda_k \simeq \frac{2}{\Delta x^2}\left(1 - \frac{1}{2}k^2\pi^2\Delta x^2 + \cdots - 1\right) \qquad k = 1, \ldots, N \tag{27}$$

Therefore, the smallest and largest eigenvalues of the matrix $\boldsymbol{D}_{\text{FD}}^2$ for sufficiently small Δx are[5]

$$\lambda_{\min} = \lambda_L \simeq -\frac{\pi^2 L^2}{\Delta x^2}\left(\frac{N}{N+1}\right)^2 \simeq -\frac{\pi^2 L^2}{\Delta x^2}, \tag{29}$$

$$\lambda_{\max} = \lambda_1 = -\pi^2. \tag{30}$$

These equations show that as we increase the number of points in $[0, L]$ the system (25) becomes stiffer and stiffer, since there $\lambda_{\min} \to -\infty$ and $\lambda_{\max} \simeq -\pi^2$.

- ***Euler forward time integration:*** If we integrate the system (25) in time with the Euler Forward scheme we obtain the *fully discrete scheme*

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \alpha\Delta t\boldsymbol{D}_{\text{FD}}^2\boldsymbol{u}^k, \tag{31}$$

  where we denoted by $\boldsymbol{u}^k = \boldsymbol{u}(t_k)$. Clearly, the absolute stability condition for Euler forward is satisfied if (see Figure 2)

$$\lambda_{\min}\alpha\Delta t \geq -2 \qquad \text{i.e.} \qquad \Delta t \leq \frac{2\Delta x^2}{\alpha\pi^2 L^2} = \frac{2}{\alpha\pi^2(N+1)^2} \tag{32}$$

  This result holds for a large number of points, i.e., for small Δx. For a small number of points we can still compute the smallest eigenvalue with (26) and use exactly the same reasoning. The condition

$$\Delta t \leq \frac{2}{\alpha\pi^2(N+1)^2} \tag{33}$$

[5]Recall that

$$\Delta x = \frac{L}{N+1}. \tag{28}$$

clearly imposes severe restrictions on the maximum time step we can use in scheme (31). For instance, if $N = 2000$ and $\alpha = 1$ we have

$$\Delta t \leq 5.061 \times 10^{-8}. \tag{34}$$

- ***Three-step Adams-Bashforth time integration (AB3):*** If we integrate the system (25) in time with the three-step Adams-Bashforth method we obtain the fully discrete scheme

$$\boldsymbol{u}^{k+3} = \boldsymbol{u}^{k+2} + \frac{\alpha\Delta t}{12}\boldsymbol{D}_{\text{FD}}^2\left(23\boldsymbol{u}^{k+2} - 16\boldsymbol{u}^{k+1} + 5\boldsymbol{u}^k\right). \tag{35}$$

As we know, the region of absolute stability of AB3 intersects the real axis at $-6/11$. If the number of spatial points is large enough, then we obtain the absolute stability requirement

$$\Delta t \leq \frac{6\Delta x^2}{11\alpha\pi^2 L^2} = \frac{6}{11\alpha\pi^2(N+1)^2}, \tag{36}$$

which is even more restrictive than the condition (33) we obtained for the Euler-forward time integrator.

- ***Crank-Nicolson time integration:*** If we discretize the system (25) in time using the Crank-Nicolson method or any other A-stable time stepping scheme then we do not have any time step restrictions. As is well-known the Crank-Nicolson method

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \frac{\alpha\Delta t}{2}\boldsymbol{D}_{\text{FD}}^2\left(\boldsymbol{u}^{k+1} + \boldsymbol{u}^k\right). \tag{37}$$

can be conveniently written as

$$\left(\boldsymbol{I} - \frac{\alpha\Delta t}{2}\boldsymbol{D}_{\text{FD}}^2\right)\boldsymbol{u}^{k+1} = \left(\boldsymbol{I} + \frac{\alpha\Delta t}{2}\boldsymbol{D}_{\text{FD}}^2\right)\boldsymbol{u}^k. \tag{38}$$

The matrix

$$\boldsymbol{K} = \boldsymbol{I} - \frac{\alpha\Delta t}{2}\boldsymbol{D}_{\text{FD}}^2 \tag{39}$$

is symmetric and positive-definite[6]. Therefore we can perform a Cholesky decomposition $\boldsymbol{K} = \boldsymbol{R}^T\boldsymbol{R}$, where $\boldsymbol{R}$ upper-triangular, in a *pre-processing stage* and write the system (38) as

$$\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{u}^{k+1} = \left(\boldsymbol{I} + \frac{\alpha\Delta t}{2}\boldsymbol{D}_{\text{FD}}^2\right)\boldsymbol{u}^k. \tag{40}$$

This system can be decomposed as a hierarchy of two triangular systems

$$\begin{cases} \boldsymbol{R}^T\boldsymbol{q}^{k+1} = \left(\boldsymbol{I} + \frac{\alpha\Delta t}{2}\boldsymbol{D}_{\text{FD}}^2\right)\boldsymbol{u}^k & \text{(lower triangular system)} \\ \boldsymbol{R}\boldsymbol{u}^{k+1} = \boldsymbol{q}^{k+1} & \text{(upper triangular system)} \end{cases} \tag{41}$$

which can be solved by using forward/backward substitution at a cost of $O(N^2)$ operations.

Absolute stability analysis can be generalized to higher-order finite difference schemes and other time integrators, e.g., RK or BDF methods.

[6]For second-order finite differences the matrix $\boldsymbol{K}$ is actually tridiagonal. This means that it can be inverted at a linear cost in N using Thomas' algorithm [3, p. 93].

Finite-difference methods for nonlinear PDEs. Consider the following initial-boundary value problem for a fourth-order nonlinear PDE (Kuramoto-Sivashinsky equation)

$$\begin{cases} \dfrac{\partial U}{\partial t} + u\dfrac{\partial U}{\partial x} + \dfrac{\partial^2 U}{\partial x^2} + \dfrac{\partial^4 U}{\partial x^4} = 0 \qquad t \geq 0 \qquad x \in [-L, L] \\ U(x,0) = U_0(x) \\ \text{Periodic B.C.} \end{cases} \tag{42}$$

The Kuramoto-Sivashinsky equation models the diffusive instabilities in a laminar flame front. Its solution can exhibit chaotic space-time dynamics. We discretize the IBVP with a second-order (in space) finite-difference method. To this end, we first approximate the derivatives $\partial U/\partial x$, $\partial^2 U/\partial x^2$ and $\partial^4 U/\partial x^4$ with fourth-order centered finite formulas on the grid

$$x_j = j\Delta x - L \qquad \Delta x = \frac{2L}{N} \qquad j = 0, \ldots, N. \tag{43}$$

Upon definition of $U_j(t) = U(x_j, t)$ such derivatives can be expressed as

$$\frac{\partial U(x_j,t)}{\partial x} \simeq \frac{U_{j+1}(t) - U_{j-1}(t)}{2\Delta x}, \tag{44}$$

$$\frac{\partial^2 U(x_j,t)}{\partial x^2} \simeq \frac{U_{j-1}(t) - 2U_j(t) + U_{j+1}(t)}{\Delta x^2}, \tag{45}$$

$$\frac{\partial^4 U(x_j,t)}{\partial x^4} \simeq \frac{U_{j-2}(t) - 4U_{j-1}(t) + 6U_j(t) - 4U_{j+1}(t) + U_{j+2}(t)}{\Delta x^4}. \tag{46}$$

A substitution of (44)-(46) into (42) yields the semi-discrete form

$$\frac{du_j}{dt} = -\underbrace{u_j \frac{u_{j+1} - u_{j-1}}{2\Delta x}}_{\text{nonlinear term}} - \frac{u_{j-1} - 2u_j + u_{j+1}}{\Delta x^2} - \frac{u_{j-2} - 4u_{j-1} + 6u_j - 4u_{j+1} + u_{j+2}}{\Delta x^4}, \tag{47}$$

for $j = 0, \ldots, N-1$. Here $u_j(t)$ denotes the finite-difference approximation of the solution to (42). The system (47) is supplemented with the periodic conditions

$$u_{j+N}(t) = u_j(t) \quad \text{for all } j \tag{48}$$

and with the initial condition

$$u_j(0) = U_0(x_j) \quad \text{for all} \quad j = 0, \ldots, N-1. \tag{49}$$

Note that the second-order discretization (47) involves stencils with different number of points, i.e., three points for the first- and the second-order derivatives, and five points for the fourth-order derivative. The system (47) can be discretized in time with any time-stepping, e.g., with the AB2 method.

Remark: The stability of the fully discrete scheme may depend on the PDE being discretized and on the type of spatial and temporal discretization, in particular for hyperbolic IBVP problems.

Finite difference methods in two-dimensional spatial domains. Consider the following initial-boundary value problem for the two-dimensional heat equation

$$\begin{cases} \dfrac{\partial U}{\partial t} = \alpha\left(\dfrac{\partial^2 U}{\partial x^2} + \dfrac{\partial^2 U}{\partial y^2}\right) + q(x,y) \qquad t \geq 0 \qquad (x,y) \in \Omega \\ U(x,y,0) = U_0(x,y) \quad (x,y) \in \Omega \\ \text{Periodic B.C.} \end{cases} \tag{50}$$

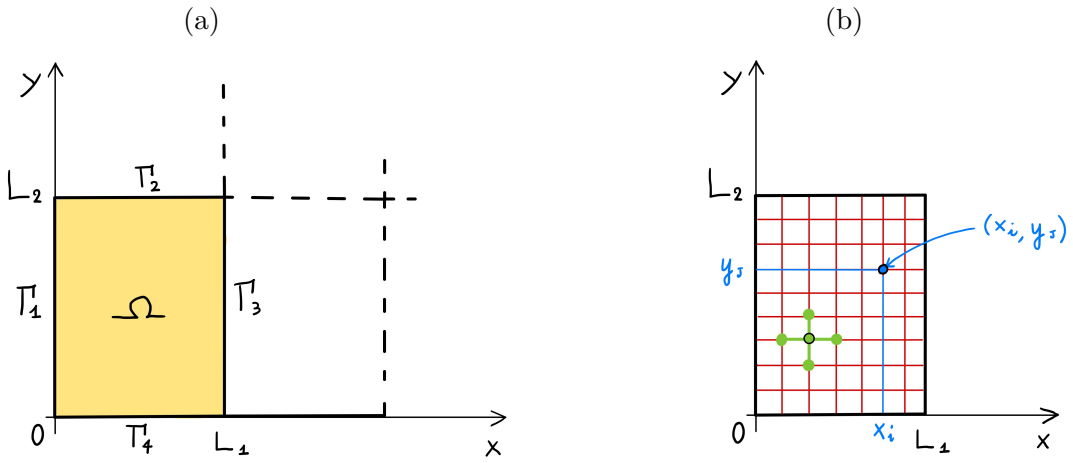


Figure 3: (a) Sketch of the spatial domain for the IBVP (50). The boundary of the domain Ω is the union between Γ_1, Γ_2, Γ_3 and Γ_4. The solution is assumed to be periodic in x and y. (b) Two-dimensional grid and stencil (green cross) used to approximate the Laplacian $\nabla^2 U = U_{xx} + U_{yy}$.
.

where Ω is a spatial domain defined as the Cartesian product of two intervals $[0, L_1]$ and $[0, L_2]$, i.e.,

$$\Omega = [0, L_1] \times [0, L_2]. \tag{51}$$

Periodic boundary conditions are set as

$$U(0, y) = U(L_1, y), \qquad \frac{\partial U(0, y)}{\partial x} = \frac{\partial U(L_1, y)}{\partial x}, \tag{52}$$

$$U(x, 0) = U(x, L_2), \qquad \frac{\partial U(x, 0)}{\partial y} = \frac{\partial U(x, L_2)}{\partial x}. \tag{53}$$

We discretize Ω in terms of the dimensional grid (see Figure 3(b))

$$(x_i, y_j) = \begin{cases} x_i = i\Delta x \qquad \Delta x = \dfrac{L_1}{N} \qquad i = 0, \ldots, N, \\ \\ y_j = j\Delta y \qquad \Delta y = \dfrac{L_2}{M} \qquad j = 0, \ldots, M. \end{cases} \tag{54}$$

By using second-order (in space) centered finite differences, we approximate the partial derivatives $\partial^2 U/\partial x^2$ and $\partial^2 U/\partial y^2$ at (x_i, y_j) as

$$\frac{\partial^2 U(x_i, y_j, t)}{\partial x^2} \simeq \frac{U_{i-1,j} - 2U_{i,j} + U_{i+1,j}}{\Delta x^2} \tag{55}$$

$$\frac{\partial^2 U(x_i, y_j, t)}{\partial x^2} \simeq \frac{U_{i,j-1} - 2U_{i,j} + U_{i,j+1}}{\Delta y^2}, \tag{56}$$

where we denoted by $U_{i,j}(t) = U(x_i, y_j, t)$. A substitution of (55)-(56) into (50) yields

$$\frac{du_{i,j}(t)}{dt} = \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\Delta y^2} + q(x_i, y_j), \tag{57}$$

with boundary conditions

$$u_{i+N,j}(t) = u_{i,j}(t), \qquad u_{i,j+M}(t) = u_{i,j}(t), \tag{58}$$

and initial condition

$$u_{i,j}(0) = U_0(x_i, y_j). \tag{59}$$

The system (57) can written in terms of differentiation matrices applied to the solution matrix $u_{i,j}(t)$. Alternatively, we can reshape the solution matrix into a column vector and construct appropriate differentiation matrices. The third option is to just write a function that takes in the matrix $u_{i,j}(t)$ and returns the right hand side of the system (57) at each time. This is usually the best option for practical implementation, especially for nonlinear system, or systems with space-dependent coefficients.

Galerkin method for the heat equation. Let us consider the IBVP problem

$$\begin{cases} \dfrac{\partial U}{\partial t} = \alpha \dfrac{\partial^2 U}{\partial x^2} + q(x) \qquad t \geq 0 \qquad x \in [0, L] \\ U(x, 0) = U_0(x) \\ U(0, t) = g_0 \\ U(L, t) = g_L \end{cases} \tag{60}$$

To solve this problem using the Galerkin method, let us first consider the function space

$$V = \{v \in L^2 \quad \text{such that} \quad \frac{\partial v}{\partial x} \in L^2, \quad v(0,t) = g_0 \quad \text{and} \quad v(L,t) = g_L\}. \tag{61}$$

where L^2 is the space of square integrable functions in $x \in [0, L]$. The function space V can be approximated by the finite-dimensional space

$$V_N = \text{span}\{\varphi_0, \ldots, \varphi_{N+1}\} \tag{62}$$

where $\varphi_k(x)$ can be, e.g., Lagrange characteristic polynomials associated with a set of Gauss-Lobatto nodes in $[0, L]$, e.g., Gauss-Lobatto-Legendre nodes. Alternatively, φ_0 and φ_{N+1} can be linear boundary modes, i.e.,

$$\varphi_0(x) = 1 - \frac{x}{L} \qquad \varphi_{N+1}(x) = \frac{x}{L} \tag{63}$$

while φ_k can be eigenfunctions of a Sturm-Liouville problem with zero boundary conditions, i.e.,

$$\varphi_k(x) = \sin\left(\frac{k\pi}{L} x\right), \qquad k = 1, 2, \ldots, N, \tag{64}$$

or shifted Chebyshev polynomials

$$\varphi_k(x) = x(L - x) T_{k-1}\left(\frac{2}{L} x - 1\right), \qquad k = 1, 2, \ldots, N. \tag{65}$$

In any case, a representation of the solution $U(x, t)$ in V_N takes the form

$$U_N(x, t) = \underbrace{g_0 \varphi_0(x) + g_L \varphi_{N+1}(x)}_{\text{boundary modes}} + \sum_{k=1}^{N} a_k(t) \varphi_k(x). \tag{66}$$

Substituting (66) into (1) and projecting the resulting equation onto $\varphi_j(x)$ ($j = 1, \ldots, N$) yields

$$\sum_{k=1}^{N} \frac{da_k(t)}{dt} \int_0^L \varphi_j(x) \varphi_k(x) dx = \alpha g_0 \int_0^L \frac{d^2 \varphi_0(x)}{dx^2} \varphi_j(x) dx + \alpha g_L \int_0^L \frac{d^2 \varphi_{N+1}(x)}{dx^2} \varphi_j(x) dx + \alpha \sum_{k=1}^{N} a_k(t) \int_0^L \frac{d^2 \varphi_k(x)}{dx^2} \varphi_j(x) dx + \int_0^L q(x) \varphi_j(x) dx + \int_0^L R_N(x) \varphi_j(x) dx. \tag{67}$$

By integrating by parts the terms at the right hand side involving second derivatives and imposing that the residual $R_N(x)$ is orthogonal to the span of $\{\varphi_1, \ldots, \varphi_N\}$ (Galerkin method) we obtain

$$\sum_{k=1}^{N} M_{jk} \frac{da_k(t)}{dt} = -\alpha g_0 S_{0j} - \alpha g_L S_{N+1j} - \alpha \sum_{k=1}^{N} S_{jk} a_k(t) + \int_0^L q(x)\varphi_j(x)dx \qquad j = 1, \ldots, N \tag{68}$$

where we defined

$$M_{jk} = \int_0^L \varphi_j(x)\varphi_k(x)dx \qquad \text{(mass matrix)}, \tag{69}$$

$$S_{jk} = \int_0^L \frac{d\varphi_j(x)}{dx}\frac{d\varphi_k(x)}{dx}dx \qquad \text{(stiffness matrix)}. \tag{70}$$

The system (68) can be written as

$$\boldsymbol{M}\frac{d\boldsymbol{a}(t)}{dt} = -\alpha \boldsymbol{S}\boldsymbol{a} + \boldsymbol{q}, \tag{71}$$

where

$$\boldsymbol{q} = \begin{bmatrix} \displaystyle\int_0^L q(x)\varphi_1(x)dx - \alpha\left(g_0 S_{01} + g_L S_{01}\right) \\ \vdots \\ \displaystyle\int_0^L q(x)\varphi_N(x)dx - \alpha\left(g_0 S_{0N} + g_L S_{N+1,N}\right) \end{bmatrix}. \tag{72}$$

If we use the interior modes (64) (basis functions) then the mass matrix and the stiffness matrix are both diagonal matrices. In particular,

$$M_{ij} = \frac{L}{2}\delta_{ij} \qquad \text{and} \qquad S_{ij} = \frac{\pi^2 j^2}{2L}\delta_{ij}. \tag{73}$$

This implies that the initial condition for the ODE (71) is

$$a_k(0) = \frac{1}{\|\varphi_k\|_{L^2}^2}\int_0^L U_0(x)\varphi_k(x)dx = \frac{2}{L}\int_0^L U_0(x)\varphi_k(x)dx \tag{74}$$

To study absolute stability of the Galerkin method, let us set $\boldsymbol{q} = \boldsymbol{0}$ in (71). In this way the solution certainly decays to zero. By using the matrices (73), we rewrite the system (71) as

$$\frac{da_k(t)}{dt} = -\frac{\alpha\pi^2 k^2}{L^2}a_k(t). \tag{75}$$

If we use the Euler forward time integration scheme is we obtain the absolute stability condition

$$\Delta t \leq -\frac{2}{\lambda_N} = \frac{2L^2}{\alpha\pi^2 N^2}. \tag{76}$$

This implies that as we add more and more modes the Galerkin system becomes stiffer and stiffer, which result is a smaller and smaller Δt if we use an explicit method.

Collocation method for the heat equation. In the Gauss-Legendre-Lobatto collocation method [2, p.132] we seek solutions to (60) in the form

$$U_N(x,t) = \sum_{k=0}^{N} U_N(x_j,t)l_j(x), \tag{77}$$

where $l_j(x)$ are the Lagrange characteristic polynomials corresponding to the Legendre-Gauss-Lobatto quadrature points. A substitution of (77) into (60) yields

$$\frac{\partial U_N}{\partial t} = \alpha \frac{\partial^2 U_N}{\partial x^2} + q(x) + R_N(x,t). \tag{78}$$

By requiring that the residual $R_N(x,t)$ vanish at the interior points yields the $N-1$ equations

$$\frac{dU_N(x_j,t)}{dt} = \alpha \sum_{k=0}^{N} D_{jk}^2 U_N(x_k,t) + q(x_j) \qquad j = 1,\ldots,N-1. \tag{79}$$

Here, D_{ij}^2 is the second-order differentiation matrix corresponding to the Gauss-Legendre-Lobatto quadrature points (see [2, §5.4.1]). We close the system by using the boundary conditions

$$U_N(0,t) = g_0, \quad U_N(L,t) = g_L. \tag{80}$$

Of course, we can replace the Gauss-Legendre-Lobatto expansion with the Gauss-Chebyshev-Lobatto expansion described at the end of Chapter 7 of the course notes (see also [2, §5.4.2]). This yields easily computable collocation points and differentiation matrices.

References

[1] D. W. Hahn and M. N. Özisik. *Heat Conduction*. Wiley, third edition, 2012.

[2] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb. *Spectral methods for time-dependent problems*, volume 21 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2007.

[3] A. Quarteroni, R. Sacco, and F. Salieri. *Numerical mathematics*. Springer, 2007.

Convergence analysis of finite difference methods for PDEs

Consider the following initial/boundary value problem for a system of linear PDEs

$$\begin{cases} \dfrac{\partial \boldsymbol{U}(\boldsymbol{x},t)}{\partial t} = \boldsymbol{L}(\boldsymbol{x},t)\boldsymbol{U}(\boldsymbol{x},t) + \boldsymbol{f}(\boldsymbol{x},t) \qquad t \geq 0 \qquad \boldsymbol{x} \in \Omega \\ \boldsymbol{S}\boldsymbol{U}(\boldsymbol{x},t) = 0 \qquad \boldsymbol{x} \in \partial\Omega \\ \boldsymbol{U}(\boldsymbol{x},0) = \boldsymbol{U}_0(\boldsymbol{x}) \end{cases} \tag{1}$$

Here $\boldsymbol{U}(\boldsymbol{x},t)$ denotes a vector field defined in a compact domain $\Omega \subseteq \mathbb{R}^d$, $\partial\Omega$ is the boundary of Ω, $\boldsymbol{L}$ is a linear operator that can depend on $\boldsymbol{x} = (x_1, \ldots, x_d)$ and t, and $\boldsymbol{S}$ is a (linear/affine) boundary operator enforcing Dirichlet, Neumann, Robin or mixed boundary conditions. We assume that the IBVP (1) is well-posed, i.e., that it admits a unique solution. Let us provide a few simple examples of PDEs that can be written in the form (1)

- **Liouville equation:** Consider a dynamical system

$$\begin{cases} \dfrac{d\boldsymbol{y}}{dt} = \boldsymbol{F}(\boldsymbol{y},t) \\ \boldsymbol{y}(0) = \boldsymbol{y}_0 \end{cases} \tag{2}$$

  evolving from a random initial state $\boldsymbol{y}_0$ with probability density function $p_0(\boldsymbol{y})$. The PDE governing the evolution equation of the joint probability density function of $\boldsymbol{y}(t)$ is

$$\frac{\partial p(\boldsymbol{y},t)}{\partial t} + \nabla \cdot [\boldsymbol{F}(\boldsymbol{y},t)p(\boldsymbol{y},t)] = 0. \tag{3}$$

  Clearly, this PDE can written in the form $\partial p/\partial t = L(\boldsymbol{x},t)p$, where

$$L(\boldsymbol{x},t)p = -\nabla \cdot \boldsymbol{F}(\boldsymbol{x},t) - \boldsymbol{F}(\boldsymbol{x},t) \cdot \nabla p \tag{4}$$

  is a first-order differential operator that depends on the phase variables $\boldsymbol{y}$ as well as on time.

- **Wave equation:** Consider the wave equation

$$\frac{\partial^2 \psi(\boldsymbol{x},t)}{\partial t^2} = c^2 \nabla^2 \psi(\boldsymbol{x},t) \tag{5}$$

  and the equivalent system of two first-order PDEs as

$$\begin{cases} \dfrac{\partial \psi(\boldsymbol{x},t)}{\partial t} = \eta(\boldsymbol{x},t), \\ \dfrac{\partial \eta(\boldsymbol{x},t)}{\partial t} = c^2 \nabla^2 \psi(\boldsymbol{x},t). \end{cases} \tag{6}$$

  Clearly, the system (6) can be written in the form (1) as

$$\frac{\partial}{\partial t} \underbrace{\begin{bmatrix} \psi(\boldsymbol{x},t) \\ \eta(\boldsymbol{x},t) \end{bmatrix}}_{\boldsymbol{U}(\boldsymbol{x},t)} = \underbrace{\begin{bmatrix} 0 & 1 \\ c^2\nabla^2 & 0 \end{bmatrix}}_{\boldsymbol{L}(\boldsymbol{x},t)} \underbrace{\begin{bmatrix} \psi(\boldsymbol{x},t) \\ \eta(\boldsymbol{x},t) \end{bmatrix}}_{\boldsymbol{U}(\boldsymbol{x},t)} \tag{7}$$

  Note that the linear operator $\boldsymbol{L}(\boldsymbol{x},t)$ in this case does not depend on $\boldsymbol{x}$ and t.

Lax-Richtmyer stability theory. In this section we provide necessary and sufficient conditions for convergence of finite-difference schemes to approximate the solution of the IBVP (1). In the interest of simplicity we consider the case where the linear operator $\boldsymbol{L}(\boldsymbol{x},t)$ in (1) is time-independent, although all consideration in the present discussion apply as well when $\boldsymbol{L}$ is time-dependent. The fully discrete finite-difference form of the IBVP (1) with time-independent linear operator $\boldsymbol{L}$ can always be written in the form

$$\boldsymbol{u}^{k+1} = \boldsymbol{B}\boldsymbol{u}^k + \boldsymbol{b}^k, \tag{8}$$

where $\boldsymbol{u}^k$ is the vector representing the approximation of the solution $\boldsymbol{U}(x,t)$ at *all* grid points[1] and time t_k, or (more generally) a vector representing solution at all grid points and multiple time instants (see the AB2 method described below).

The matrix $\boldsymbol{B}$ usually depends on Δt, Δx_1, Δx_2, etc., and also on the spatial discretization of the functions and operators appearing in $\boldsymbol{L}(\boldsymbol{x})$, while $\boldsymbol{b}^k$ takes cares of external forcing terms and/or the boundary conditions. The vector $\boldsymbol{b}^k$ may also depend of Δt, Δx_1, Δx_2, etc.

Example: Consider the one-dimensional heat-equation

$$\frac{\partial U}{\partial t} = \alpha \frac{\partial^2 U}{\partial x^2}, \tag{9}$$

with Dirichlet boundary conditions. Discretize the second derivative $\partial^2 U/\partial x^2$ by, e.g., second-order centered finite differences. This yields the semi-discrete form

$$\frac{d\boldsymbol{u}(t)}{dt} = \alpha \boldsymbol{D}^2_{\text{FD}} \boldsymbol{u}. \tag{10}$$

We have seen that we can discretize (10) in time using many different schemes,e.g.,

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \alpha \Delta t \boldsymbol{D}^2_{\text{FD}} \boldsymbol{u}^k \qquad \text{(Euler forward)}, \tag{11}$$

$$\left(\boldsymbol{I} - \frac{\alpha \Delta t}{2} \boldsymbol{D}^2_{\text{FD}}\right) \boldsymbol{u}^{k+1} = \left(\boldsymbol{I} + \frac{\alpha \Delta t}{2} \boldsymbol{D}^2_{\text{FD}}\right) \boldsymbol{u}^k \qquad \text{(Crank-Nicolson)}. \tag{12}$$

These schemes can be written in the form (8) provided we define

$$\boldsymbol{B} = \boldsymbol{I} + \alpha \Delta t \boldsymbol{D}^2_{\text{FD}} \qquad \text{(Euler forward)}, \tag{13}$$

$$\boldsymbol{B} = \left(\boldsymbol{I} - \frac{\alpha \Delta t}{2} \boldsymbol{D}^2_{\text{FD}}\right)^{-1} \left(\boldsymbol{I} + \frac{\alpha \Delta t}{2} \boldsymbol{D}^2_{\text{FD}}\right) \qquad \text{(Crank-Nicolson)}. \tag{14}$$

Similarly, if we discretize (9) in time with the two-step Adams-Bashforth method we obtain

$$\boldsymbol{u}^{k+2} = \boldsymbol{u}^{k+1} + \alpha \Delta t \boldsymbol{D}^2_{\text{FD}} \left(\frac{3}{2}\boldsymbol{u}^{k+1} - \frac{1}{2}\boldsymbol{u}^k\right). \tag{15}$$

We can always write a two-step method as a one-step method in a higher-dimensional space. To this end, define

$$\boldsymbol{v}^{k+1} = \boldsymbol{u}^k \tag{16}$$

[1]The numerical solution $\boldsymbol{u}^k$ in (8) can be arranged as a vector, a matrix, or a multi-dimensional array. Correspondingly, $\boldsymbol{B}$ can be a matrix, a tensor or a more general linear operator in the space in which $\boldsymbol{u}^k$ is defined.

and rewrite (15) as

$$\begin{cases} \boldsymbol{u}^{k+2} = \boldsymbol{u}^{k+1} + \alpha\Delta t \boldsymbol{D}_{\text{FD}}^2 \left(\frac{3}{2}\boldsymbol{u}^{k+1} - \frac{1}{2}\boldsymbol{v}^{k+1} \right) \\ \boldsymbol{v}^{k+2} = \boldsymbol{u}^{k+1} \end{cases} \tag{17}$$

i.e.,

$$\boldsymbol{z}^{k+2} = \boldsymbol{B}\boldsymbol{z}^{k+1} \tag{18}$$

where

$$\boldsymbol{z}^{k+2} = \begin{bmatrix} \boldsymbol{u}^{k+2} \\ \boldsymbol{v}^{k+2} \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{I} + \frac{3}{2}\alpha\Delta t \boldsymbol{D}_{\text{FD}}^2 & -\frac{1}{2}\alpha\Delta t \boldsymbol{D}_{\text{FD}}^2 \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}. \tag{19}$$

Note that (18) is in the form (8). However, in this case the vector of unknowns $\boldsymbol{z}^k$ is not just the solution $\boldsymbol{u}^k$ at time t_k but rather a concatenation of the solution at time t_k and t_{k-1}.

Definition 1 (Lax-Richtmyer stability)**.** The finite difference scheme (8) is *stable* if there exists a constant C_T independent of k, Δt, Δx_1, Δx_2 etc such that

$$\left\| \boldsymbol{B}^k \right\| \leq C_T \quad \text{for all } k \text{ such that} \quad k\Delta t \leq T. \tag{20}$$

Here, $\|\cdot\|$ denotes any matrix norm induced by a vector norm. In other words, we require that the matrix powers $\boldsymbol{B}$ are uniformly bounded[2] by a constant C_T for all $k \leq T/\Delta t$, where the integration period T is fixed and is chosen arbitrarily Δt.

When studying convergence of (8) we are interested, in particular, in the behavior of $\left\|B^k\right\|$ when Δt and Δx_i are sent to zero.

Theorem 1 (Lax-Richtmyer equivalence theorem [1])**.** *Given a properly posed initial-boundary value problem* (1) *and a consistent*[3] *finite-difference approximation* (8)*, stability is a necessary and sufficient condition for convergence.*

Proof. For simplicity we consider the case where time integration is defined by a one-step scheme although all consideration in the present proof apply as well for multistep schemes. A substitution of the exact solution $\boldsymbol{U}(\boldsymbol{x}, t)$ of the IBVP (1) into the fully discrete scheme (8) yields the local truncation error (LTE) $\boldsymbol{\tau}^k$, defined by the equation

$$\boldsymbol{U}^{k+1} = \boldsymbol{B}\boldsymbol{U}^k + \boldsymbol{b}^k + \Delta t \boldsymbol{\tau}^k. \tag{22}$$

Here, we denoted by

$$\boldsymbol{U}^k = \begin{bmatrix} U(\boldsymbol{x}_1, t_k) \\ U(\boldsymbol{x}_2, t_k) \\ \vdots \\ U(\boldsymbol{x}_N, t_k) \end{bmatrix} \tag{23}$$

where N denotes the total number of spatial grid points. For example, in 3D we have

$$\boldsymbol{x}_i = (x_{l(i)}, y_{m(i)}, z_{n(i)}). \tag{24}$$

[2]Uniformly bounded means that the bound

$$\left\| \boldsymbol{B}\left(\Delta t, \Delta x_1, \Delta x_2, \cdots\right)^k \right\| \leq C_T \tag{21}$$

holds for every Δt, Δx_1, Δx_2, etc., and every $k \leq T\Delta t$ (fixed T, and any chosen Δt).

[3]Recall that a finite-difference approximation is said to be consistent if the local truncation error goes to zero as we send Δt and Δx_i $(i = 1, \ldots, d)$ to zero.

i.e, we have $N = n^3$ points, where n is the number of points in each variable z, y or z. Note that for multi-step time integration methods we just need to replace $\boldsymbol{U}^k$ by $\boldsymbol{Z}^k = [U^k, \quad U^{k-1}, \quad U^{k-2}, \ldots]^T$, i.e., a vector collecting the solution vector at times t_k, t_{k-1}, etc. (see, e.g., Eqs. (17)-(19)). Subtracting (8) from (22) yields

$$\boldsymbol{e}^{k+1} = \boldsymbol{B}\boldsymbol{e}^k + \Delta t \boldsymbol{\tau}^k, \tag{25}$$

where

$$\boldsymbol{e}^k = \boldsymbol{U}^k - \boldsymbol{u}^k \qquad \text{(error)} \tag{26}$$

The recursion (25) can be iterated back to the error $\boldsymbol{e}^0$. To this end,

$$\begin{aligned}
\boldsymbol{e}^k =& \boldsymbol{B}\boldsymbol{e}^{k-1} + \Delta t \boldsymbol{\tau}^{k-1} \\
=& \boldsymbol{B}\left(\boldsymbol{B}\boldsymbol{e}^{k-2} + \Delta t \boldsymbol{\tau}^{k-2}\right) + \Delta t \boldsymbol{\tau}^{k-1} \\
=& \boldsymbol{B}^2\boldsymbol{e}^{k-2} + \Delta t \boldsymbol{B}\boldsymbol{\tau}^{k-2} + \Delta t \boldsymbol{\tau}^{k-1} \\
& \vdots \\
=& \boldsymbol{B}^k\boldsymbol{e}^0 + \Delta t \sum_{j=1}^{k} \boldsymbol{B}^{k-j}\boldsymbol{\tau}^{j-1}.
\end{aligned} \tag{27}$$

At this point we take any vector norm of $\boldsymbol{e}^k$ (and corresponding induced matrix norm), and use the stability assumption (20) to obtain

$$\begin{aligned}
\left\|\boldsymbol{e}^k\right\| =& \left\| \boldsymbol{B}^k\boldsymbol{e}^0 + \Delta t \sum_{j=1}^{k} \boldsymbol{B}^{k-j}\boldsymbol{\tau}^{j-1} \right\| \\
\leq& \left\|\boldsymbol{B}^k\right\| \left\|\boldsymbol{e}^0\right\| + \Delta t \sum_{j=1}^{k} \left\|\boldsymbol{B}^{k-j}\right\| \left\|\boldsymbol{\tau}^{j-1}\right\| \\
\leq& C_T \left\|\boldsymbol{e}^0\right\| + k\Delta t C_T \max_{j=1,\ldots,k} \left\|\boldsymbol{\tau}^{j-1}\right\| \\
\leq& C_T \left\|\boldsymbol{e}^0\right\| + T C_T \max_{j=1,\ldots,k} \left\|\boldsymbol{\tau}^{j-1}\right\|,
\end{aligned} \tag{28}$$

where T is period of integration, and C_T is the the uniform bound in (20). The upper bound in (28) goes to zero if the method is consistent, i.e., if

$$\max_{j=1,\ldots,k} \left\|\boldsymbol{\tau}^{j-1}\right\| \to 0 \quad \text{for} \quad \Delta t, \Delta x_i \to 0, \tag{29}$$

and if the error at initial time $\left\|\boldsymbol{e}^0\right\|$ is either zero or goes to zero as we send Δt and Δx_1, Δx_2, etc., to zero. This proves that consistency plus Lax-Richtmyer stability implies convergence.

□

At this point a few remarks are in order.

- **Sufficient condition for stability:** Recall that for any matrix norm and any $k \in \mathbb{N}$

$$\left\|\boldsymbol{B}^k\right\| \leq \|\boldsymbol{B}\|^k. \tag{30}$$

  Therefore, to prove stability it is sufficient to show that for sufficiently small Δt

$$\|\boldsymbol{B}\| \leq 1 + \beta \Delta t \quad \text{for some } \beta \in \mathbb{R}. \tag{31}$$

In fact[4],

$$\left\|\boldsymbol{B}^k\right\| \leq \|\boldsymbol{B}\|^k \leq (1+\beta\Delta t)^k \leq e^{k\Delta t\beta} \leq e^{T\beta}. \tag{33}$$

- **Necessary and sufficient conditions for stability:** The spectral radius of a matrix is a lower bound for any matrix sub-multiplicative matrix norm. This implies that

$$\rho(\boldsymbol{B}) \leq \|\boldsymbol{B}\| \quad \Rightarrow \quad \rho(\boldsymbol{B})^k \leq \left\|\boldsymbol{B}^k\right\| \leq C_T \quad \Leftrightarrow \quad \rho(\boldsymbol{B})^k \leq C_T. \tag{34}$$

From this equation it follows that

$$\rho(\boldsymbol{B}) \leq 1+\beta\Delta t \tag{35}$$

is necessary for stability. If the matrix $\boldsymbol{B}$ is normal (i.e. $\boldsymbol{B}\boldsymbol{B}^T = \boldsymbol{B}^T\boldsymbol{B}$) then the 2-norm coincides with the spectral radius, i.e.

$$\rho(\boldsymbol{B}) = \|\boldsymbol{B}\|_2 \tag{36}$$

and (35) is *necessary and sufficient for stability.*

Stability analysis of forward-in-time centered-in-space scheme for the heat equation: It is important to remark that the stability condition may depend on the way we send Δt and Δx_i to zero. To show this, consider the one-dimensional heat equation (9) with zero Dirichlet boundary conditions, and the matrix $\boldsymbol{B}$ corresponding to the centered-in-space forward-in-time finite-difference discretization (13). The matrix $\boldsymbol{B}$ is symmetric, and therefore the 2-norm coincides with the spectral radius. This yields[5],

$$\|\boldsymbol{B}\|_2 = \max_{i=1,\dots,N} |\alpha\Delta t\lambda_i + 1|. \tag{37}$$

At this point we recall that, for large N (number of spatial grid points), we have

$$\min_{i=1,\dots,N} \lambda_i = \lambda_N = \frac{2}{\Delta x^2}\left(\cos(N\pi\Delta x) - 1\right) \simeq -\frac{4}{\Delta x^2}. \tag{38}$$

Hence,

$$\|\boldsymbol{B}\|_2 \simeq \left|\frac{4\alpha\Delta t}{\Delta x^2} - 1\right| \tag{39}$$

Recalling equation (31), we conclude that a necessary and sufficient condition for stability is

$$\left|\frac{4\alpha\Delta t}{\Delta x^2} - 1\right| \leq 1+\beta\Delta t. \tag{40}$$

This equation defines a *stability region* in the $(\Delta t, \Delta x)$-plane for each β (see Figure 20). Such a stability region can be computed analytically, although the computation is a bit cumbersome (except for the case $\beta = 0$). In fact, define

$$\eta = \frac{\Delta t}{\Delta x^2}. \tag{41}$$

The inequality (40) can be split into the following two inequalities

$$\begin{cases} 4\alpha\eta - \beta\eta\Delta x^2 \leq 2 & \Leftrightarrow \quad \eta\left(4\alpha - \beta\Delta x^2\right) \leq 2 \quad \text{for} \quad \Delta x^2 \leq 4\alpha\Delta t, \\ -4\alpha\eta - \beta\eta\Delta x^2 \leq 0 & \Leftrightarrow \quad 4\alpha + \beta\Delta x^2 \geq 0 \quad\quad\quad\; \text{for} \quad \Delta x^2 \geq 4\alpha\Delta t. \end{cases} \tag{42}$$

[4]The inequalities (33) follow from the basic inequality

$$\log(1+x) \leq x \quad \Leftrightarrow \quad \log(1+x)^k = k\log(1+x) \leq kx \quad \Leftrightarrow \quad (1+x)^k \leq e^{kx} \tag{32}$$

[5]The eigenvalues of the matrix $\boldsymbol{B} = \boldsymbol{I} + \alpha\Delta t\boldsymbol{D}_{\text{FD}}^2$ are $1+\alpha\Delta t\lambda_i$, where λ_i are the eigenvalues of $\boldsymbol{D}_{\text{FD}}^2$.

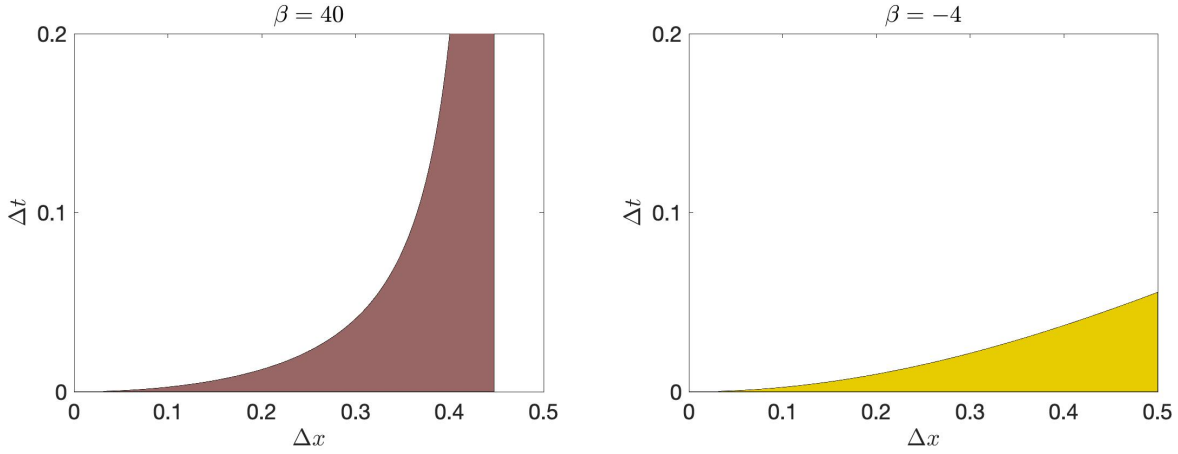


Figure 1: Lax-Richtmyer stability regions for the forward-in-time (Euler) centered-in-space (second-order) discretization of the heat equation (9) with $\alpha = 2$ and zero Dirichlet boundary conditions. The regions of stability are computed numerically using (40). Note the vertical asymptote for $\beta = 40$ at $\Delta x = 2\sqrt{\alpha/\beta} = 0.4472$ (see Eq. (43))

The first one can be written as

$$\frac{\Delta t}{\Delta x^2} \leq \frac{2}{4\alpha - \beta \Delta x^2}. \tag{43}$$

For $\beta > 0$ this yields the additional condition $\Delta x \leq 2\sqrt{\alpha/\beta}$ (see Figure 20). On the other hand, for $\beta = 0$ everything simplifies substantially. In particular, the second inequality in (42) yields the trivial condition $\alpha \geq 0$, while the first inequality yields

$$\frac{\Delta t}{\Delta x^2} \leq \frac{1}{2\alpha}. \tag{44}$$

The condition (44) also shows that Δt and Δx cannot be sent to zero at arbitrary rates. Indeed, we must have $\Delta t \sim \kappa \Delta x^2$ for (44) (or (43)) to hold in the limits $\Delta t, \Delta x \to 0$.

Lax-Richtmyer stability analysis applies to both implicit and explicit temporal integration schemes. However, for implicit schemes the matrix $\boldsymbol{B}$ involves the inverse of some other matrix (see, e.g., equation (14)). This makes the stability analysis of implicit schemes not straightforward nor practical using the matrix $\boldsymbol{B}$. We will see hereafter that this issue can be mitigated (at least for linear PDEs) by using discrete Fourier series.

Convergence analysis for nonlinear PDEs. The fully discrete finite-difference formulation of a one-dimensional nonlinear PDE can be written as

$$\sum_{k=0}^{q} \alpha_q u_j^{k+q} = \Delta t \Phi_j \left(\boldsymbol{u}^{k+q}, \ldots, \boldsymbol{u}^k, \Delta t, \Delta x \right). \tag{45}$$

For instance, the second-order central finite-difference discretization of the Kuramoto-Sivashinsky equation with Euler forward time stepping can be written as

$$u_j^{k+1} - u_j^k = \Delta t \left(-u_j^k \frac{u_{j+1}^k - u_{j-1}^k}{2\Delta x} - \frac{u_{j-1}^k - 2u_j^k + u_{j+1}^k}{\Delta x^2} - \frac{u_{j-2}^k - 3u_{j-1}^k + 6u_j^k - 3u_{j+1}^k + u_{j+2}^k}{\Delta x^4} \right), \tag{46}$$

To study convergence of this scheme for Δt and Δx going to zero, we can use methods similar to the those we used in the convergence analysis of numerical schemes for ODEs, in particular the convergence proof

in the course note 4. To use such a proof in the context of finite-difference approximations of PDEs, we need to make sure that the Lipshitz constant of $\mathbf{\Phi}$ in (45) can be bounded by some constant when we send Δt and Δx to zero at an appropriate rate. Under this assumption, it is rather straightforward to show that method is convergent (provided the method it is consistent). To this end, just follow the proof in the appendix of the course note 4.

Von-Neumann stability theory. Stability analysis of finite-difference schemes can be simplified substantially if the PDE is defined in a periodic domain. The key idea is to use discrete Fourier series applied to the finite-difference discretization of the PDE and determine under which conditions on Δt, Δx_1, Δx_2, etc., the scheme is stable. One reason for the Fourier series analysis is that it allows us to determine stability conditions for both implicit and explicit schemes is a rather straightforward way. To illustrate the method, let us consider the prototype IBVP

$$\begin{cases} \dfrac{\partial U(x,t)}{\partial t} = \alpha \dfrac{\partial^2 U}{\partial x^2} \qquad t \geq 0 \qquad x \in [0, L] \\ U(x,0) = U_0(x) \\ \text{Periodic B.C.} \end{cases} \tag{47}$$

We have seen that (47) can be discretized with second-order finite-differences in space and Euler-forward time integration as

$$u_j^{k+1} = u_j^k + \frac{\alpha \Delta t}{\Delta x^2}\left(u_{j-1}^k - 2u_j^k + u_{j+1}^k\right) \qquad j = 0, \ldots, N-1, \tag{48}$$

where u_j^k is the approximation of $U(x_j, t_k)$, and

$$x_j = j\frac{L}{N} \qquad j = 0, \ldots, N. \tag{49}$$

The scheme (48) is supplemented with periodic boundary conditions

$$u_j^k = u_{j+N}^k \quad \text{for all } j \in \mathbb{Z}. \tag{50}$$

and the initial condition

$$u_j^0 = U_0(x_j) \qquad j = 0, \ldots, N-1. \tag{51}$$

Let us now expand the numerical solution u_j^k in a discrete Fourier series[6]

[6] As is well known, the solution to (47) can be expanded in a Fourier series as

$$U(x,t) = \sum_{k=-\infty}^{\infty} C_k(t) e^{2\pi i k x/L} \tag{52}$$

Evaluating $U(x,t)$ on the grid (49) yields the discrete Fourier series

$$U(x_j,t) = \sum_{k=-\infty}^{\infty} C_k(t) e^{2\pi i k x_j/L} = \sum_{k=-\infty}^{\infty} C_k(t) e^{2\pi i k j/N} \qquad j = 0, \ldots, N-1. \tag{53}$$

Moreover,

$$U(x_j,t) = \sum_{k=-\infty}^{\infty} C_k(t) e^{2ik j\pi/N} = \sum_{k=0}^{N-1} \sum_{p=-\infty}^{\infty} C_{k+pN}(t) e^{2\pi i (k+pN) j/N} = \sum_{k=0}^{N-1} e^{2\pi i k j/N} \underbrace{\sum_{p=-\infty}^{\infty} C_{k+pN}(t)}_{c_k(t)/N}. \tag{54}$$

This can be written as

$$U(x_j,t) = \frac{1}{N}\sum_{k=0}^{N-1} c_k(t) e^{2\pi i k j/N} = \frac{\Delta x}{L}\sum_{k=0}^{N-1} c_k(t) e^{ikj\xi}, \qquad \xi = \frac{2\pi \Delta x}{L}. \tag{55}$$

$$u_j^k = \frac{1}{N}\sum_{p=0}^{N-1} c_p^k e^{ipj\xi}, \qquad \text{where} \qquad \xi = \frac{2\pi\Delta x}{L}, \tag{56}$$

and substitute it into (48) to obtain

$$\begin{aligned}\sum_{p=0}^{N-1} c_p^{k+1} e^{ipj\xi} &= \sum_{p=0}^{N-1} c_p^k e^{ipj\xi}\left[1+\frac{\alpha\Delta t}{\Delta x^2}\left(e^{-ip\xi}-2+e^{ip\xi}\right)\right] \\ &= \sum_{p=0}^{N-1} c_p^k e^{ipj\xi}\left[1+\frac{\alpha\Delta t}{\Delta x^2}\left(2\cos\left(p\xi\right)-2\right)\right],\end{aligned} \tag{57}$$

i.e.,

$$c_p^{k+1} = \underbrace{c_p^k\left[1+\frac{2\alpha\Delta t}{\Delta x^2}\left(\cos\left(2\pi p\frac{\Delta x}{L}\right)-1\right)\right]}_{\text{amplification factor } G_p(\Delta t,\Delta x)}. \tag{58}$$

Upon definition of

$$\boldsymbol{c}^k = \begin{bmatrix} c_0^k \\ c_0^k \\ \vdots \\ c_{N-1}^k \end{bmatrix}, \qquad \boldsymbol{G}(\Delta t,\Delta x) = \begin{bmatrix} G_0(\Delta t,\Delta x) & 0 & \cdots & 0 \\ 0 & G_1(\Delta t,\Delta x) & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & G_{N-1}(\Delta t,\Delta x) \end{bmatrix} \tag{59}$$

we can write (58) as

$$\boldsymbol{c}^{k+1} = \boldsymbol{G}(\Delta t,\Delta x)\boldsymbol{c}^k. \tag{60}$$

The matrix $\boldsymbol{G}(\Delta t,\Delta x)$ in (60) plays the same role in Fourier space as the matrix $\boldsymbol{B}$ in (8) does in physical space. In other words, for the scheme (48) to be stable we must have

$$\left\|\boldsymbol{G}^k\right\| \leq H_T \quad \text{for all } k \text{ such that } k\Delta t \leq T. \tag{61}$$

where H_T is a constant that does not depend on Δx or on Δt. In (61) $\|\cdot\|$ denotes any matrix norm induced by a vector norm.

Remark: Clearly, if we compute the inverse Fourier transform of (60) we obtain

$$\boldsymbol{u}^{k+1} = \boldsymbol{F}\boldsymbol{G}\boldsymbol{F}^{-1}\boldsymbol{u}^k, \tag{62}$$

where $\boldsymbol{F}$ is the Fourier transform matrix such that

$$\boldsymbol{u}^k = \boldsymbol{F}\boldsymbol{c}^k, \qquad \boldsymbol{F} = \frac{1}{N}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{i\xi} & e^{2i\xi} & \cdots & e^{i(N-1)\xi} \\ 1 & e^{2i\xi} & e^{4i\xi} & \cdots & e^{2i(N-1)\xi} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & e^{(N-1)i\xi} & e^{2(N-1)\xi} & \cdots & e^{i(N-1)^2\xi}. \end{bmatrix} \tag{63}$$

A comparison between (62) and (8) shows that

$$\boldsymbol{B} = \boldsymbol{F}\boldsymbol{G}\boldsymbol{F}^{-1}. \tag{64}$$

This equation justifies why stability can be equivalently studied in Fourier space by studying the norm of $\boldsymbol{G}^k$. In fact,

$$\left\|\boldsymbol{B}^k\right\| = \left\|\boldsymbol{F}\boldsymbol{G}^k\boldsymbol{F}^{-1}\right\|. \tag{65}$$

The Fourier transform matrix $\boldsymbol{F}$ plays no role in the stability properties of the scheme.

Necessary and sufficient conditions for Von-Neumann stability. Let us recall that the spectral radius of a matrix $\boldsymbol{G}$, i.e.,

$$\rho(\boldsymbol{G}) = \max_i |\lambda_i| \tag{66}$$

where λ_i are the eigenvalues of $\boldsymbol{G}$, is a lower bound for any (sub-multiplicative) matrix norm[7] of $\boldsymbol{G}$, i.e.,

$$\rho(\boldsymbol{G}) \leq \|\boldsymbol{G}\| \quad \text{for every sub-multiplicative matrix norm } \|\cdot\|. \tag{69}$$

Moreover, the spectral radius of the matrix power $\boldsymbol{G}^k$ is equal to $\rho(\boldsymbol{G})^k$ (recall that the eigenvalues of $\boldsymbol{G}^k$ are λ_i^k). By using the stability condition (61) we obtain

$$\rho(\boldsymbol{G})^k \leq \left\|\boldsymbol{G}^k\right\| \leq H_T \tag{70}$$

i.e.,

$$\rho(\boldsymbol{G})^k \leq H_T. \tag{71}$$

As before, this implies that for sufficiently small Δt the spectral radius of $\boldsymbol{G}$ must satisfy (see Eq. (31))

$$\rho(\boldsymbol{G}) \leq 1 + \gamma\Delta t \tag{72}$$

This is a *necessary condition*, not a sufficient condition. In fact, it is possible that $\rho(\boldsymbol{G})^k \leq H_T$ even though $\left\|\boldsymbol{G}^k\right\|$ grows unboundedly as we send Δt, Δx_1, Δx_2, etc., to zero. In other words,

$$\rho(\boldsymbol{G})^k \leq H_T \quad \text{does not imply} \quad \left\|\boldsymbol{G}^k\right\| \leq H_T. \tag{73}$$

However, if the matrix $\boldsymbol{G}$ is normal, i.e., if $\boldsymbol{G}\boldsymbol{G}^* = \boldsymbol{G}\boldsymbol{G}^*$ (where * denotes the conjugate transpose) then it is easy to show that Von-Neumann stability condition (72) is sufficient.

Lemma 1. The Von-Neumann stability condition (72) is sufficient if the amplification matrix $\boldsymbol{G}$ is normal.

Proof. The spectral radius of normal matrices is equal to the matrix 2-norm

$$\rho(\boldsymbol{G}) = \sqrt{\rho(\boldsymbol{G}\boldsymbol{G}^*)} = \|\boldsymbol{G}\|_2. \tag{74}$$

This allows us to write (70) as

$$\rho(\boldsymbol{G})^k = \left\|\boldsymbol{G}^k\right\|_2 \leq H_T. \tag{75}$$

Hence, for normal matrices $\boldsymbol{G}$ we have that (72) implies

$$\left\|\boldsymbol{G}^k\right\|_2 \leq H_T \tag{76}$$

[7] A sub-multiplicative matrix norm is a norm satisfying

$$\|\boldsymbol{A}\boldsymbol{B}\| \leq \|\boldsymbol{A}\|\,\|\boldsymbol{B}\| \tag{67}$$

for all matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. All matrix norms induces by vector norms are sub-multiplicative. To prove (69) it is sufficient to consider one eigenvalue λ_i of $\boldsymbol{G}$ and the corresponding eigenvector $\boldsymbol{v}$. Construct the matrix $\boldsymbol{V} = [\boldsymbol{v} \quad \cdots \quad \boldsymbol{v}]$, and note that

$$\|\boldsymbol{G}\boldsymbol{V}\| = |\lambda_i|\,\|\boldsymbol{V}\| \leq \|\boldsymbol{G}\|\,\|\boldsymbol{V}\| \quad \Rightarrow \quad \|\boldsymbol{G}\| \geq \max_i |\lambda_i| = \rho(\boldsymbol{G}). \tag{68}$$

the matrix

i.e., that the scheme is stable. Recall that stability in one norm imples stability in any other norm.

□

Fast computation of the amplification factors. The Fourier series of the solution of linear PDEs with constant coefficients can be always decoupled into a system of equations involving one Fourier mode at a time. Hence, to determine the amplification factors of the Fourier coefficients it is sufficient to consider only one wave number. In practice, we can simply substitute

$$u_j^k = c_p^k e^{ijp\xi} \quad \text{where} \quad \xi = \frac{2\pi\Delta x}{L} \tag{77}$$

into the numerical scheme and compute the amplification factors for the p-th mode. Let us show how to perform this calculation for second-order centered finite-difference discretization of the heat equation with Crank-Nicolson time-integration.

- **Von-Neumann stability analysis of the heat equation (Euler-forward time integration).** We have seen that the Fourier transform of finite-difference scheme (48) yields the diagonal matrix of amplification factors defined in (59). The diagonal entries of $\boldsymbol{G}$ are the eigenvalues of $\boldsymbol{G}$. Hence, the spectral radius of $\boldsymbol{G}$ is

$$\rho(\boldsymbol{G}) = \max_{p=0,\dots,N-1} \left| 1 + \frac{2\alpha\Delta t}{\Delta x^2}\left(\cos\left(2\pi p\frac{\Delta x}{L}\right) - 1\right)\right| \simeq \left|\frac{4\alpha\Delta t}{\Delta x^2} - 1\right| \qquad \text{(for large } N\text{)}. \tag{78}$$

  By using the Von-Neumann condition (72) we conclude that the scheme (48) is stable if and only if

$$\left|\frac{4\alpha\Delta t}{\Delta x^2} - 1\right| \leq 1 + \gamma\Delta t. \tag{79}$$

  This is exactly the same condition we obtained in (40) (see the discussion thereafter).

- **Von-Neumann stability analysis of the heat equation (Crank-Nicolson time integration).** Consider the fully discrete finite-difference scheme

$$u_j^{k+1} - \frac{\alpha\Delta t}{2\Delta x^2}\left(u_{j-1}^{k+1} - 2u_j^{k+1} + u_{j+1}^{k+1}\right) = u_j^k + \frac{\alpha\Delta t}{2\Delta x^2}\left(u_{j-1}^k - 2u_j^k + u_{j+1}^k\right). \tag{80}$$

  Substitute (79) into (80) to obtain

$$c_p^{k+1}\left[1 - \frac{\alpha\Delta t}{\Delta x^2}\left(\cos(p\xi) - 1\right)\right] = c_p^k\left[1 + \frac{\alpha\Delta t}{\Delta x^2}\left(\cos(p\xi) - 1\right)\right], \tag{81}$$

  i.e.,

$$c_p^{k+1} = \frac{1 + \frac{\alpha\Delta t}{\Delta x^2}\left(\cos(p\xi) - 1\right)}{1 - \frac{\alpha\Delta t}{\Delta x^2}\left(\cos(p\xi) - 1\right)} c_p^k. \tag{82}$$

  Again, the amplification matrix $\boldsymbol{G}$ is diagonal with spectral radius

$$\rho(\boldsymbol{G}) = \max_{p=0,\dots,N-1} \left|\frac{1 + \frac{\alpha\Delta t}{\Delta x^2}\left(\cos(p\xi) - 1\right)}{1 - \frac{\alpha\Delta t}{\Delta x^2}\left(\cos(p\xi) - 1\right)}\right|. \tag{83}$$

  At this point we notice that $\cos(p\xi) - 1 \leq 0$ for any p. This implies that

$$\left|1 + \frac{\alpha\Delta t}{\Delta x^2}\left(\cos(p\xi) - 1\right)\right| \leq \left|1 - \frac{\alpha\Delta t}{\Delta x^2}\left(\cos(p\xi) - 1\right)\right| \tag{84}$$

and

$$\rho(\boldsymbol{G}) \leq 1. \tag{85}$$

Recalling that the Von-Neumann stability condition (72) we conclude that the second-order centered finite-difference scheme with Crank-Nicolson time integration is unconditionally stable. Moreover, the scheme is consistent, and therefore convergent.

Clearly, by following the same steps that lead us to (31) we see that (61) yields the following *sufficient condition* for stability

$$\|\boldsymbol{G}\| \leq 1 + \delta\Delta t \quad \text{as} \quad \Delta t \to 0, \tag{86}$$

where $\|\boldsymbol{G}\|$ denotes any matrix norm compatible with a vector norm.

References

[1] P. D. Lax and R. D. Richtmyer. Survey of the stability of linear finite-difference equations. *Communications on Pure and Applied Mathematics*, 9:267–293, 1956.

Finite-difference methods for the advection equation

In this course note we study stability and convergence of various finite-difference schemes for simple hyperbolic PDEs (conservation laws) of the form

$$\frac{\partial U(x,t)}{\partial t} + \frac{\partial \left(F(U(x,t))\right)}{\partial x} = 0, \tag{1}$$

where F is a continuously differentiable nonlinear function. The material in this note is discussed in [3, Ch. 10]. More generally numerical methods nonlinear conservation laws, or systems of nonlinear conservation laws[1] are discussed in, e.g., in [1, 2]. Let us begin with the simple prototype linear initial/boundary value problem[2]

$$\begin{cases} \dfrac{\partial U(x,t)}{\partial t} + a\dfrac{\partial U(x,t)}{\partial x} = 0 \qquad x \in [0,L] \\ U(x,0) = U_0(x) \\ \text{Periodic B.C.} \end{cases} \tag{2}$$

As is well-known, this PDE can be solved with the *method of characteristics*, by essentially transforming it into an ODE along the flow generated by the dynamical system (see Appendix at the end of this note)

$$\frac{dx(t)}{dt} = a \qquad x(0) = x_0. \tag{3}$$

In the case of (2) the ODE is $dz/dt = 0$, with initial condition $z(0) = U_0(x_0)$. This yields the analytical solution[3]

$$U(x,t) = U_0(x - at). \tag{4}$$

This is traveling wave moving with velocity a. If a is positive the wave moves to the right, while preserving entirely its structure. Once the wave reaches the periodic boundary, it comes back from the other side.

Finite-difference discretization. We discretize the IBVP (2) with second-order centered finite-differences. To this end, consider the following grid

$$x_j = j\Delta x, \qquad \Delta x = \frac{L}{N}, \qquad j = 0, \ldots, N \tag{5}$$

and approximate the first derivative $\partial U/\partial x$ as

$$\frac{\partial U(x_j,t)}{\partial x} \simeq \frac{U(x_{j+1},t) - U(x_{j-1},t)}{2\Delta x}. \tag{6}$$

A substitution of (6) into (2) yields the semi-discrete form

$$\frac{du_j(t)}{dt} = -a\frac{u_{j+1}(t) - u_{j-1}(t)}{2\Delta x} \qquad j = 0, \ldots, N-1 \tag{7}$$

with periodic boundary conditions

$$u_N(t) = u_0(t) \qquad u_{-1}(t) = u_{N-1}(t). \tag{8}$$

[1] A conservation law is an expression in mathematical terms of the balance within a physical system. It is a statement that the production of a physical quantity such as mass, energy or charge in a closed volume is exactly equal to the flux of that quantity across the boundary of that volume. Such conservation laws often take the form of partial differential equations with appropriate boundary conditions or equivalent integral forms.

[2] The IBVP is ill-posed if $a > 0$ and we set the boundary Dirichlet boundary condition $U(L,t) = g(t)$ where g is a continuous time-dependent function.

[3] To compute the solution of (2) we can of course also use other techniques such as Fourier series and Laplace transforms.

The system (7)-(8) can be written in a matrix-vector form as

$$\begin{cases} \dfrac{d\boldsymbol{u}}{dt} = -a\boldsymbol{D}^1_{\text{FD}}\boldsymbol{u} \\ \boldsymbol{u}(0) = \boldsymbol{U}_0 \end{cases} \tag{9}$$

where

$$\boldsymbol{D}^1_{\text{FD}} = \frac{1}{2\Delta x}\begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & \cdots & -1 \\ -1 & 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & -1 & 0 & 1 & \cdots & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & -1 & 0 & 1 \\ 1 & \cdots & \cdots & \cdots & 0 & 1 & 0 \end{bmatrix}, \qquad \boldsymbol{u} = \begin{bmatrix} u_0 \\ u_2 \\ u_3 \\ \vdots \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix}, \tag{10}$$

The matrix $\boldsymbol{D}^1_{\text{FD}}$ is clearly skew-symmetric and therefore it has purely imaginary eigenvalues. It can be shown that the eigenvalues of $\boldsymbol{D}^1_{\text{FD}}$ are

$$\lambda_k = \frac{i}{\Delta x}\sin\left(\frac{2\pi}{L}k\Delta x\right) \qquad k = 1, \ldots, N. \tag{11}$$

Recall also that skew-symmetric matrices are normal. This implies that the spectral radius of the matrix $\boldsymbol{D}^1_{\text{FD}}$ coincides with its 2-norm, i.e., we have

$$\left\|\boldsymbol{D}^1_{\text{FD}}\right\|_2 = \rho\left(\boldsymbol{D}^1_{\text{FD}}\right). \tag{12}$$

Euler-forward time integration. Let us discretize the ODE system (7) in using the Euler-forward method. This yields the fully discrete scheme

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k - a\Delta t\boldsymbol{D}^1_{\text{FD}}\boldsymbol{u}^k, \tag{13}$$

where $\boldsymbol{u}^k$ denotes the approximation of the solution of (7) at time t_k. It is straightforward to show that the local truncation error (LTE) of (13) goes to zero linearly in Δt and quadratically in Δx. To this end, let us first write (13) component-by-component as

$$u^{k+1}_j = u^k_j - a\Delta t\frac{u^k_{j+1} - u^k_{j-1}}{2\Delta x}. \tag{14}$$

A substitution of the exact solution $U(x,t)$ of (2) into (13) gives the LTE

$$\tau^k_j = \frac{U^{k+1}_j - U^k_j}{\Delta t} + a\frac{U^k_{j+1} - U^k_{j-1}}{2\Delta x}, \tag{15}$$

where we denoted by U^k_j the exact solution evaluated at x_j and t_k, i.e., $U^k_j = U(x_j, t_k)$. Using Taylor series expansions yields

$$\tau^k_j = \frac{\Delta t}{2}\frac{d^2U^h_j}{dt^2} + \frac{a\Delta x^2}{12}\frac{d^3U^h_j}{dx^3} + \text{higher order terms}. \tag{16}$$

Hence, the method is consistent with order one in time and order two in space. Regarding stability, let us write the scheme (13) as

$$\boldsymbol{u}^{k+1} = \boldsymbol{B}\boldsymbol{u}^k \tag{17}$$

where

$$\boldsymbol{B} = \boldsymbol{I} - a\Delta t \boldsymbol{D}^1_{\text{FD}} \tag{18}$$

and $\boldsymbol{D}^1_{\text{FD}}$ is defined in (10). Recall that for normal matrices $\boldsymbol{B}$, a necessary and sufficient condition for Lax-Richtmyer stability[4] is

$$\rho(\boldsymbol{B}) \leq 1 + \beta\Delta t. \tag{20}$$

The spectral radius of the matrix $\boldsymbol{B}$ is easily obtained by shifting and rescaling the eigenvalues of $\boldsymbol{D}^1_{\text{FD}}$, i.e.,

$$\begin{aligned}
\rho(\boldsymbol{B}) &= \max_{p=1,\ldots,N} \left| 1 - i\frac{a\Delta t}{\Delta x} \sin\left(\frac{2\pi}{L} p\Delta x\right)\right| \\
&= \max_{p=1,\ldots,N} \sqrt{1 + \frac{a^2\Delta t^2}{\Delta x^2} \sin\left(\frac{2\pi}{L} p\Delta x\right)^2} \qquad (21)\\
&\leq \sqrt{1 + \frac{a^2\Delta t^2}{\Delta x^2}}. \qquad (22)
\end{aligned}$$

Taking the k-th power yields

$$\left\|\boldsymbol{B}^k\right\| = \rho(\boldsymbol{B})^k \leq \left(1 + \frac{a^2\Delta t^2}{\Delta x^2}\right)^{k/2} \leq \exp\left(a^2 \frac{k\Delta t^2}{2\Delta x^2}\right) \leq \exp\left(a^2 \frac{T\Delta t}{2\Delta x^2}\right). \tag{23}$$

If we choose Δt and Δx such that

$$\frac{\Delta t}{\Delta x^2} \leq b \tag{24}$$

for arbitrary (finite) b, then we see that the scheme (13) is Lax-Richtmyer stable. In fact, substituting (24) into (23) yields

$$\left\|\boldsymbol{B}^k\right\|_2 \leq \exp\left(a^2 \frac{Tb}{2}\right) \quad \text{for all } k \text{ such that } k\Delta t \leq T. \tag{25}$$

By using the Lax equivalence theorem we conclude that the method is convergent, since it is consistent and stable (under the condition (24))

The stability analysis that lead us to (23) is based on the knowledge of the spectral radius of the matrix $\boldsymbol{B}$ which, in turn, is based on the knowledge of the eigenvalues of $\boldsymbol{D}^1_{\text{FD}}$. A more direct method to obtain a stability inequality is based on Von-Neumann analysis. To this end, we study the dynamics of an arbitrary Fourier mode[5]

$$\widehat{u}^k_p = c^k_p e^{ijp\xi} \quad \text{where} \quad \xi = \frac{2\pi\Delta x}{L}. \tag{26}$$

Substituting (26) into (14) yields

$$\begin{aligned}
c^k_p &= c^k_p \left[1 - \frac{a\Delta t}{2\Delta x}\left(e^{ip\xi} - e^{-ip\xi}\right)\right] \\
&= \underbrace{c^k_p \left[1 - i\frac{a\Delta t}{\Delta x}\sin(\xi p)\right]}_{G_p(\Delta t,\Delta x)}. \qquad (27)
\end{aligned}$$

[4]The 2-norm of a normal matrix $\boldsymbol{B}$ coincides with the spectral radius $\rho(\boldsymbol{B})$, i.e.,

$$\rho(\boldsymbol{B}) = \|\boldsymbol{B}\|_2. \tag{19}$$

[5]Since the PDE (2) is linear with constant coefficients it is sufficient to consider one Fourier mode to perform stability analysis.

The amplification matrix $\boldsymbol{G}$ is diagonal

$$\boldsymbol{G}(\Delta t,\Delta x)=\begin{bmatrix} G_0(\Delta t,\Delta x) & 0 & \cdots & 0 \\ 0 & G_1(\Delta t,\Delta x) & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & G_{N-1}(\Delta t,\Delta x) \end{bmatrix} \tag{28}$$

Since $\boldsymbol{G}$ is normal, we have that the following Von-Neumann condition

$$\|\boldsymbol{G}\|_2=\rho(\boldsymbol{G})\leq 1+\gamma\Delta t \tag{29}$$

is necessary and sufficient for stability. The spectral radius of $\boldsymbol{G}$ is the same as the spectral radius of the matrix $\boldsymbol{B}$, i.e.,

$$\begin{aligned}\rho(\boldsymbol{G})&=\max_{p=0,\ldots,N-1}|G_p(\Delta t,\Delta x)|\\ &=\max_{p=0,\ldots,N-1}\left|1-i\frac{a\Delta t}{\Delta x}\sin\left(\frac{2\pi}{L}p\Delta x\right)\right|\\ &=\max_{p=0,\ldots,N-1}\sqrt{1+\frac{a^2\Delta t^2}{\Delta x^2}\sin\left(\frac{2\pi}{L}p\Delta x\right)^2}.\end{aligned} \tag{30}$$

As before (see Eq. (23)),

$$\rho(\boldsymbol{G})^k\leq\left(1+\frac{a^2\Delta t^2}{\Delta x^2}\right)^{k/2}\leq e^{Tba^2/2} \tag{31}$$

provided we select $\Delta t\leq b\Delta x^2$, for any finite $b>0$. Under this condition we have that the scheme (14) is Lax-Richtmyer stable, and therefore convergent.

- **Remark:** Although the scheme (14) is provably convergent (it is consistent and Lax-Richtmyer stable) it is easy to see that the method is in practice "unstable" for every finite Δt and Δx. In fact, by taking the modulus of (27) we obtain

$$\begin{aligned}\left|c_p^{k+1}\right|&=\left|c_p^k\right|\left|1-i\frac{a\Delta t}{\Delta x}\sin\left(\frac{2\pi}{L}p\Delta x\right)\right|\\ &=\left|c_p^k\right|\sqrt{1+\frac{a^2\Delta t^2}{\Delta x^2}\sin\left(\frac{2\pi}{L}p\Delta x\right)^2}\\ &\geq\left|c_p^k\right|.\end{aligned} \tag{32}$$

  This shows that the amplitude of each discrete Fourier mode is *always amplified as time integration proceeds*, no matter how we pick Δt and Δx.

  On the other hand, the analytical solution of (2) in Fourier space suggests that

$$c_p(t)=e^{-iat}c_p(0)\quad\Rightarrow\quad|c_p(t)|=\left|e^{-2\pi iapt/L}\right||c_p(0)|\quad\Rightarrow\quad|c_p(t)|=|c_p(0)|, \tag{33}$$

  i.e., the amplitude of each Fourier mode must be preserved. That's why the scheme (14) is often designated as "unstable". The situation here has similarities to the one we have seen when studying convergence of the leapfrog method applied to ODEs. In fact, such method is zero stable and consistent, and therefore convergent. However, the method is unconditionally absolutely unstable. Note, that here the solution doesn't really go to zero though.

Leapfrog time integration. Let us discretize (7) in time with the leapfrog method

$$u_j^{k+2} = u_j^k - a\frac{\Delta t}{\Delta x}\left(u_{j+1}^{k+1} - u_{j-1}^{k+1}\right). \tag{34}$$

We know that (34) is consistent with order two in both Δx and Δt. Let us perform a Von-Neumann stability analysis. To this end, we first substitute (26) into (34) to obtain

$$c_p^{k+2} = c_p^k - a\frac{\Delta t}{\Delta x}\left(e^{ip\xi} - e^{-ip\xi}\right)c_p^{k+1}. \tag{35}$$

At this point, we write the two-step method (35) as a one step method

$$\begin{bmatrix} c_p^{k+2} \\ d_p^{k+2} \end{bmatrix} = \underbrace{\begin{bmatrix} -2ai\Delta t\sin(p\xi)/\Delta x & 1 \\ 1 & 0 \end{bmatrix}}_{\boldsymbol{G}_p} \begin{bmatrix} c_p^{k+1} \\ d_p^{k+1} \end{bmatrix}. \tag{36}$$

In this case, the amplification matrix $\boldsymbol{G}$ is block-diagonal and symmetric, hence normal. The eigenvalues of $\boldsymbol{G}$ are easily obtained by computing the eigenvalues of each block. The characteristic polynomial corresponding to $\boldsymbol{G}_p$ is

$$\lambda^2 + \frac{2ia\Delta t\sin(\xi p)}{\Delta x}\lambda - 1 = 0. \tag{37}$$

The eigenvalues are

$$\lambda_{1,2}(p) = -\frac{ia\Delta t\sin(\xi p)}{\Delta x} \pm \sqrt{1 - \frac{a^2\Delta t^2\sin(\xi p)^2}{\Delta x^2}}. \tag{38}$$

At this point we notice that for all Δt and Δx such that

$$\left|a\frac{\Delta t}{\Delta x}\right| \leq 1 \tag{39}$$

we have that quantity within the square root in (38) is real. In this assumption, the modulus of the eigenvalues can be computed as

$$|\lambda_{1,2}(p)|^2 = \frac{a^2\Delta t^2\sin(\xi p)^2}{\Delta x^2} + 1 - \frac{a^2\Delta t^2\sin(\xi p)^2}{\Delta x^2} = 1. \tag{40}$$

This implies that the spectral radius of all $\boldsymbol{G}_p$ is equal to one, and therefore

$$\left\|\boldsymbol{G}^k\right\|_2 = \rho(\boldsymbol{G})^k = 1. \tag{41}$$

This proves that the leapfrog method is Lax-Richtmyer stable (provided (39) is satisfied), and therefore convergent. The condition (39) is called Courant-Friedrichs-Levy (CFL) condition, and it is described in more detail hereafter.

Remark: The calculation of the spectral radius of $\boldsymbol{G}$ is more involved when $|a\Delta t/\Delta x| \geq 1$. In fact, in this case we have that the square root in (38) is imaginary.

Lax-Friedrichs method. The Lax-Friedrichs scheme is obtained by replacing u_j^k in the Euler-Forward method (14) with the average over neighboring nodes, i.e.,

$$u_j^{k+1} = \frac{u_{j-1}^k + u_{j+1}^k}{2} - a\Delta t\frac{u_{j+1}^k - u_{j-1}^k}{2\Delta x}. \tag{42}$$

The reason for the modification can be explained as follows. By adding and subtracting u_j^k from the right hand side of (42) we can write the scheme as

$$u_j^{k+1} = u_j^k - a\Delta t \frac{u_{j+1}^k - u_{j-1}^k}{2\Delta x} + \frac{\Delta x^2}{2}\left(\frac{u_{j-1}^k - 2u_j^k + u_{j+1}^k}{\Delta x^2}\right). \tag{43}$$

In this form, we recognize that the scheme introduces a numerical diffusion term with amplitude proportional to the square of the grid spacing. The diffusion term is meant to counterbalance the numerical amplification of Fourier modes induced by the Euler-Forward scheme, see Eq. (32). It is straightforward to show that the Lax-Friedrichs method is consistent with order one in Δt and order two in Δx. Let us now perform a Von-Neumann stability analysis of the scheme (43) (assuming we are considering periodic boundary conditions in $[0, L]$). To this end, we substitute (26) into (43) to obtain

$$\begin{aligned} c_p^{k+1} =& c_p^k\left[1 - \frac{a\Delta t}{2\Delta x}\left(e^{ip\xi} - e^{-ip\xi}\right) + \frac{1}{2}\left(e^{ip\xi} + e^{-ip\xi} - 2\right)\right] \\ =& c_p^k\left[1 - i\frac{a\Delta t}{\Delta x}\sin(p\xi) + \cos(p\xi) - 1\right] \\ =& c_p^k\left[\cos(p\xi) - i\frac{a\Delta t}{\Delta x}\sin(p\xi)\right]. \end{aligned} \tag{44}$$

Again, we have a diagonal amplification matrix $\boldsymbol{G}$ with diagonal entries

$$G_p(\Delta t, \Delta x) = \cos(p\xi) - i\frac{a\Delta t}{\Delta x}\sin(p\xi). \tag{45}$$

Since $\boldsymbol{G}$ is diagonal, the Von-Neumann condition

$$\rho(\boldsymbol{G}) \leq 1 + \gamma\Delta t \tag{46}$$

is necessary and sufficient for stability. We have

$$\begin{aligned} \rho(\boldsymbol{G}) =& \max_{p=0,\ldots,N-1} |G_p(\Delta t, \Delta x)| \\ =& \max_{p=0,\ldots,N-1} \sqrt{\cos(p\xi)^2 + \frac{a^2\Delta t^2}{\Delta x^2}\sin(p\xi)^2}. \end{aligned} \tag{47}$$

Clearly, if

$$\left|a\frac{\Delta t}{\Delta x}\right| \leq 1 \tag{48}$$

then $\rho(\boldsymbol{G}) \leq 1$, and the Lax-Friedrichs method is stable. The condition (48) is known as *Courant-Friedrichs-Levy (CFL) condition*, and the number

$$\nu = |a|\frac{\Delta t}{\Delta x} \tag{49}$$

is known as *Courant number*. The CFL condition is a general statement that the domain of dependence of the numerical scheme must contain the domain of dependence of the physical problem (see Figure 1). For the particular case of a the linear PDEs we are studying in this section, the physical domain of dependence a point (the root of the characteristic curve), while the domain of dependence of the numerical scheme is an interval.

- **Analysis of the Lax-Friedrichs scheme with the method of modified equations:** To analyze the scheme (42) it is possible to use another method based on the so-called "modified equation". Such equation represents the PDE that governs a smooth function $v(x, t)$ that satisfies the numerical scheme (42) exactly, i.e.,

$$v(x_j, t_{k+1}) = \frac{v(x_{j+1}, t_k) + v(x_{j-1}, t_k)}{2} - a\Delta t\frac{v(x_{j+1}, t_k) - v(x_{j-1}, t_k)}{2\Delta x}. \tag{50}$$

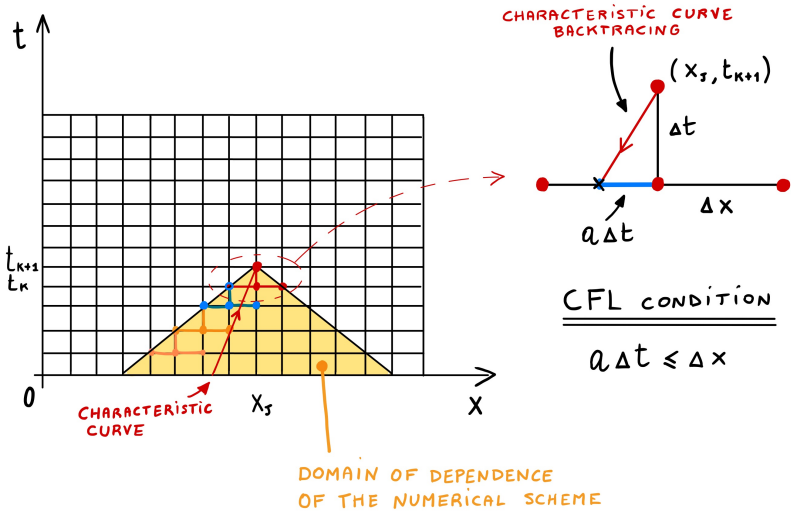


Figure 1: Illustration of the CFL condition, highlighting that the domain of dependence of the numerical scheme must contain that of the physical problem.

By using Taylor series we obtain

$$\frac{\partial v}{\partial t} + a\frac{\partial v}{\partial x} = \frac{\Delta t}{2}\left(\frac{\Delta x^2}{\Delta t^2}\frac{\partial^2 v}{\partial x^2} - \frac{\partial v^2}{\partial t^2}\right) + O(\Delta t^2). \tag{51}$$

This equation can be written as

$$\begin{aligned}\frac{\partial v}{\partial t} + a\frac{\partial v}{\partial x} =& \frac{\Delta t}{2}\left(\frac{\Delta x^2}{\Delta t^2} - a^2\right)\frac{\partial^2 v}{\partial x^2} + O(\Delta t^2) \\ =& \frac{a^2\Delta t}{2\nu^2}\left(1-\nu^2\right)\frac{\partial^2 v}{\partial x^2} + O(\Delta t^2).\end{aligned} \tag{52}$$

Hence, if the Courant number (49) satisfies $\nu \leq 1$ then we see that the numerical solution satisfies an advection-diffusion equation, which is known to have smooth solutions. On the other hand, if $\nu > 1$ the modified equation has negative diffusion

Remark: Note that if

$$a\frac{\Delta t}{\Delta x} = 1 \tag{53}$$

then the amplitude of the Fourier modes c_p^k in (44) is preserved in time, i.e., we have

$$\left|c_p^{k+1}\right| = \left|c_p^k\right| \quad \text{for all} \quad p = 0, \ldots, N-1. \tag{54}$$

It is easy to see that with condition (53) the Lax-Friedrichs scheme (43) is actually exact for the linear advection equation (2). In fact, if we substitute (53) into (43) we obtain

$$u_j^{k+1} = u_j^k, \tag{55}$$

which is what the exact solution does along the characteristics curves evaluated on the grid. In other words, (53) sets up the (space-time) grid in a way that a characteristic passing through one point x_j lands at another grid point (either x_{j+1} or x_{j-1} after Δt time units).

Lax-Wendroff method. Consider the formal solution of the semi-discrete scheme (9)

$$\boldsymbol{u}(t_k+\Delta t)=e^{-a\Delta t\boldsymbol{D}^1_{\rm FD}}\boldsymbol{u}(t_k) \tag{56}$$

and expand it to second-order in Δt. This yields

$$\boldsymbol{u}(t_k+\Delta t)\simeq\left(\boldsymbol{U}-a\Delta t\boldsymbol{D}^1_{\rm FD}+\frac{1}{2}a^2\Delta t^2\boldsymbol{D}^1_{\rm FD}\boldsymbol{D}^1_{\rm FD}\right)\boldsymbol{u}(t_k). \tag{57}$$

Replacing $\boldsymbol{D}^1_{\rm FD}\boldsymbol{D}^1_{\rm FD}$ with the second-order differentiation matrix based on a stencil with three points, yields the *Lax-Wendroff method*

$$u_j^{k+1}=u_j^k-\frac{a\Delta t}{2\Delta x}\left(u_{j+1}^k-u_{j-1}^k\right)+\frac{a^2\Delta t^2}{2\Delta x^2}\left(u_{j-1}^k-2u_j^k+u_{j+1}^k\right). \tag{58}$$

It is straightforward to show that the method is consistent with order two in both Δt and Δx. Regarding stability, a substitution of (26) into (58) yields the following equation for the amplification factors of the discrete Fourier modes

$$c_p^{k+1}=c_p^k\underbrace{\left[1-i\frac{a\Delta t}{\Delta x}\sin\left(p\xi\right)+\frac{a^2\Delta t^2}{\Delta x^2}\left(\cos(p\xi)-1\right)\right]}_{G_p(\Delta t,\Delta x)}. \tag{59}$$

As before the amplification matrix $\boldsymbol{G}$ is diagonal, with diagonal entries G_p. By using the trigonometric identities

$$\cos(p\xi)-1=-2\sin^2\left(\frac{p\xi}{2}\right),\qquad \sin\left(p\xi\right)=2\sin\left(\frac{p\xi}{2}\right)\sin\left(\frac{p\xi}{2}\right) \tag{60}$$

we can rewrite G_p in (59) as

$$G_p(\Delta t,\Delta x)=1-2i\frac{a\Delta t}{\Delta x}\sin\left(\frac{p\xi}{2}\right)\cos\left(\frac{p\xi}{2}\right)-2\frac{a^2\Delta t^2}{\Delta x^2}\sin^2\left(\frac{p\xi}{2}\right). \tag{61}$$

Taking the modulus yields

$$\begin{aligned}|G_p(\Delta t,\Delta x)|^2=&\left[1-2\frac{a^2\Delta t^2}{\Delta x^2}\sin^2\left(\frac{p\xi}{2}\right)\right]^2+4\frac{a^2\Delta t^2}{\Delta x^2}\sin^2\left(\frac{p\xi}{2}\right)\cos^2\left(\frac{p\xi}{2}\right)\\ =&1+4\frac{a^4\Delta t^4}{\Delta x^4}\sin^4\left(\frac{p\xi}{2}\right)-4\frac{a^2\Delta t^2}{\Delta x^2}\sin^2\left(\frac{p\xi}{2}\right)+4\frac{a^2\Delta t^2}{\Delta x^2}\sin^2\left(\frac{p\xi}{2}\right)\cos^2\left(\frac{p\xi}{2}\right)\\ =&1+4\frac{a^4\Delta t^4}{\Delta x^4}\sin^4\left(\frac{p\xi}{2}\right)-4\frac{a^2\Delta t^2}{\Delta x^2}\sin^4\left(\frac{p\xi}{2}\right)\\ =&1-4\frac{a^2\Delta t^2}{\Delta x^2}\left(1-\frac{a^2\Delta t^2}{\Delta x^2}\right)\sin^4\left(\frac{p\xi}{2}\right).\end{aligned} \tag{62}$$

With this expression, we can easily bound the spectral radius of the amplification matrix $\boldsymbol{G}$ as

$$\begin{aligned}\rho(\boldsymbol{G})^2=&\max_p|G_p(\Delta t,\Delta x)|^2\\ \leq&1-4\frac{a^2\Delta t^2}{\Delta x^2}\left(1-\frac{a^2\Delta t^2}{\Delta x^2}\right).\end{aligned} \tag{63}$$

In Figure 2 we plot the function

$$f(\alpha)=1-4\alpha^2(1-\alpha^2)\quad\text{versus}\quad|\alpha|=|a|\frac{\Delta t}{\Delta x}. \tag{64}$$

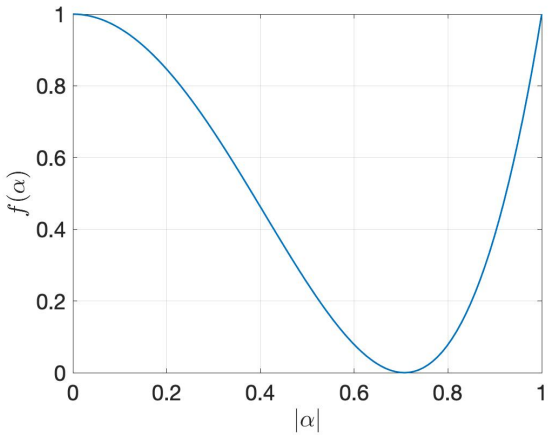


Figure 2: Graph of the function (64) characterizing the square of spectral radius of the amplification matrix $\boldsymbol{G}$ for the Lax-Wendroff method. It is seen that for $|\alpha| \leq 1$ the method is stable.

It is seen that $f(\beta) \leq 1$ for all $|\beta| \leq 1$. This allows us to conclude that the Lax-Wendroff method is stable (and therefore convergent) if

$$\left| a \frac{\Delta t}{\Delta x} \right| \leq 1. \tag{65}$$

From equation (62) we also see that if $|a\Delta t/\Delta x| = 1$ then the amplitude of each discrete Fourier mode is preserved. As before, the condition $|a\Delta t/\Delta x| = 1$ implies that we are working with a space-time grid that is defined by the discrete characteristic curves of (2).

Appendix. The method of characteristics. Consider the semi-linear scalar first-order PDE

$$\begin{cases} \dfrac{\partial U(\boldsymbol{x},t)}{\partial t} + \boldsymbol{a}(\boldsymbol{x},t) \cdot \nabla U(\boldsymbol{x},t) = f(\boldsymbol{x},t,U(\boldsymbol{x},t)) \qquad \boldsymbol{x} \in \mathbb{R} \qquad t \geq 0 \\ U(\boldsymbol{x},0) = U_0(\boldsymbol{x}) \end{cases} \tag{66}$$

This equation can be transformed into an ODE on the flow generated by the nonlinear dynamical system

$$\frac{\boldsymbol{X}(t,\boldsymbol{x}_0)}{dt} = \boldsymbol{a}(\boldsymbol{X}(t,\boldsymbol{x}_0),t) \qquad \boldsymbol{X}(0,\boldsymbol{x}_0) = \boldsymbol{x}_0. \tag{67}$$

The ODE is[6]

$$\frac{dz}{dt} = f(\boldsymbol{X}(t,\boldsymbol{x}_0),t,z(t)) \qquad z(0) = U_0(\boldsymbol{x}_0) \tag{70}$$

The meaning of $\boldsymbol{X}(t,\boldsymbol{x}_0)$ and $z(t)$ is summarized in Figure 3.

If we are interested in the solution of (66) at a particular point in space, say $\boldsymbol{x}^*$ (e.g., a point on a grid) and a particular time say t^* then we proceed as follows:

[6]Equation (70) is easily derived by defining

$$z(t) = U(\boldsymbol{X}(t,\boldsymbol{x}_0),t) \qquad \text{(solution along the flow)} \tag{68}$$

and noting that

$$\frac{dz(t)}{dt} = \frac{dU(\boldsymbol{X}(t,\boldsymbol{x}_0),t)}{dt} = \frac{\partial U(\boldsymbol{X}(t,\boldsymbol{x}_0),t)}{\partial t} + \boldsymbol{a}(\boldsymbol{x},t) \cdot \nabla U(\boldsymbol{X}(t,\boldsymbol{x}_0),t) = f(\boldsymbol{X}(t,\boldsymbol{x}_0),t,z(t)). \tag{69}$$

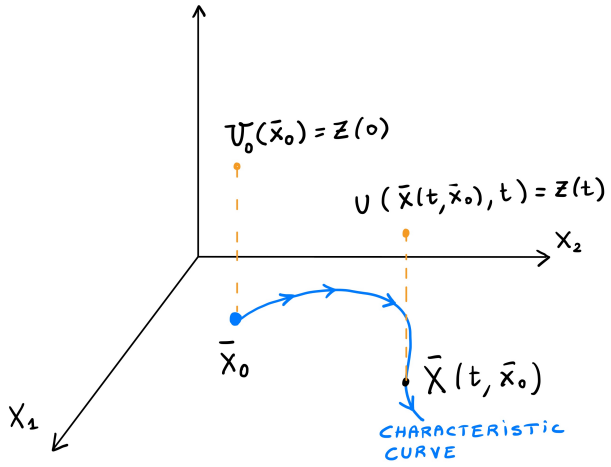


Figure 3: Sketch of the method of characteristics.

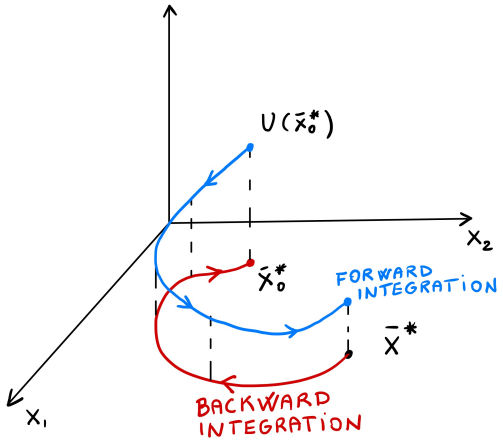


Figure 4: Sketch of the process used to compute the solution of the PDE (66) at a particular point $\boldsymbol{x}^*$ and particular time t^*. Essentially, we integrate the characteristic system (67) backward in time from $t = t^*$ and position $\boldsymbol{x}^*$ to $t = 0$. This gives us the point $\boldsymbol{x}_0^*$. Next we integrate (70) forward in time with initial condition $z(0) = U(\boldsymbol{x}_0^*)$ along the same characteristic curve.

1. Integrate (67) backward in time from $t = t^*$ to $t = 0$ with initial condition $\boldsymbol{x}^*$. That gives us the point $\boldsymbol{x}_0^*$ shown in Figure (4)
2. with $\boldsymbol{x}_0^*$ available, we integrate (70) forward in time from $t = 0$ to $t = t^*$.

We can use this method to compute the solution of (66) at time $t = t^*$ at all spatial grid points of a given grid. To do so, we simply need to solve (67) backward and (70) forward at each grid point.

More generally, the method of characteristics can be applied to solve first-order nonlinear PDEs of the

form[7]

$$\frac{\partial U(\boldsymbol{x},t)}{\partial t}+\boldsymbol{a}(\boldsymbol{x},t,U(\boldsymbol{x},t))\cdot\nabla U(\boldsymbol{x},t)=f(\boldsymbol{x},t,U(\boldsymbol{x},t)). \tag{72}$$

In this case the characteristic system is

$$\begin{cases}\dfrac{\boldsymbol{X}(t)}{dt}=\boldsymbol{a}(\boldsymbol{X}(t),t,z(t)), & \boldsymbol{X}(0,\boldsymbol{x}_0)=\boldsymbol{x}_0,\\ \dfrac{z(t)}{dt}=f(\boldsymbol{X}(t),t,z(t)), & z(0)=U_0(\boldsymbol{x}_0).\end{cases} \tag{73}$$

Note that, in this case,computing the solution at a specific point in space and time is not as easy as before since (73) is coupled. In other words, when we integrate (73) backward in time we need to guess $z(t)$. Long story short, to compute the solution of (71) at a specific point in space and time, we could use the shooting method applied to (73), the control variable being $z(t)$ at $\boldsymbol{x}^*$.

References

[1] J. S. Hesthaven. *Numerical methods for conservation laws: from analysis to algorithms.* SIAM, 2018.

[2] R. J. LeVeque. *Numerical methods for conservation laws.* Birkhäuser, 1992.

[3] R. J. LeVeque. *Finite difference methods for ordinary and partial differential equations.* SIAM, 2007.

[7] A particular case of (72) is the scalar conservation law (1). In fact, we see that (1) can be written as

$$\frac{\partial U}{\partial t}+\underbrace{\frac{\partial F(U)}{\partial U}}_{a(U)}\frac{\partial U}{\partial x}=0. \tag{71}$$