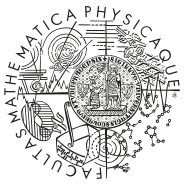


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

Numerical Solution of ODEs

Lecture notes for the course NMNV539

Scott Congreve;
based on original notes by Vladimír Janovský

Charles University,
Faculty of Mathematics and Physics,
Department of Numerical Mathematics,
Sokolovská 83, 18675 Prague

Contents

List of Figures

List of Examples

List of Algorithms

CHAPTER 1

Mathematical modelling of evolution

1.1 Motivation examples

Example 1.1 (Logistic equation). Consider differential equation

$$x' = (a - bx)x - c \tag{1.1}$$

with initial condition

$$x(t_0) = x_0, \tag{1.2}$$

and parameters $a \geq 0$, $b \geq 0$, $c \geq 0$.

Solution of the problem (1.1)–(1.2) is a twice continuously differentiable function $u : \mathbb{R} \to \mathbb{R}$ which satisfies the identity

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} = (a - bu(t))u(t) - c$$

for each $t \in \mathbb{R}$, where $u(t_0) = x_0$.

We define an operator $\phi : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ which maps the initial condition (1.2) to the solution of the problem (1.1)–(1.2) at time t; e.g.,

$$\phi : (t, t_0, x_0) \longmapsto u(t). \tag{1.3}$$

The operator ϕ defines the evolution of the given initial condition in time.

The problem (1.1)–(1.2) is a biological model, which models the density x of a population in a fixed volume over time. This model can be used to predict the future of the population density ($t \geq t_0$), or even used to reconstruct the past ($t \leq t_0$). Figure 1.1 displays *trajectories* initialized with the initial conditions $(t_0, x_0) = (0, 3/2), (0, 1), (0, 1/2), (0, 0)$ and $(0, -1/20)$ for the parameters $a = 1$, $b = 1$ and $c = 0$ of this problem, where the solid lines correspond to $t \geq 0$ and the dotted lines to $t \leq 0$. In general, we call a trajectory a mapping $t \in I \longmapsto (t, \phi(t, t_0, x_0))$, for some interval I containing t_0.

The right-hand side of the differential equation (1.1) can be defined as a mapping $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, defined as

$$(t, x) \longmapsto f(t, x) \equiv (a - bx)x - c; \tag{1.4}$$

here, we note that the right-hand side does not depend on t; however, in future examples it may (cf. Example 1.7)

Using the mapping f we can define a mapping

$$\begin{bmatrix} t \\ x \end{bmatrix} \in \mathbb{R} \times \mathbb{R} \longmapsto \begin{bmatrix} 1 \\ f(t, x) \end{bmatrix} \in \mathbb{R} \times \mathbb{R}. \tag{1.5}$$

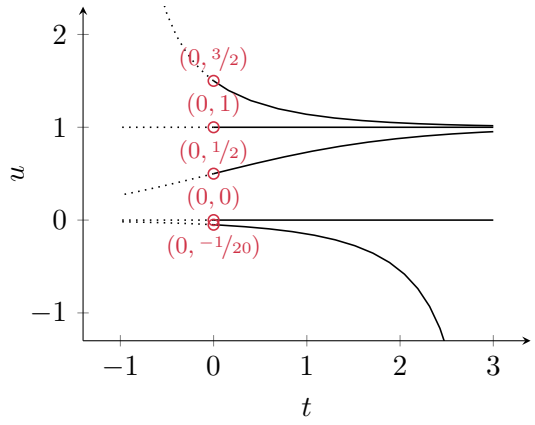


Figure 1.1: Logistic equation — five different trajectories related to the indicated initial conditions with parameters $a = 1$, $b = 1$, $c = 0$.

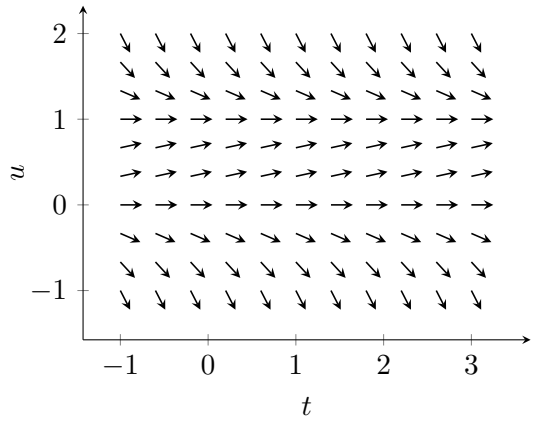


Figure 1.2: Logistic equation — direction field. Parameters $a = 1$, $b = 1$, $c = 0$.

The graph of the mapping (1.5) is called the *direction field*. It describes the displacement of a given vector (t, x) to a position $(t, x) + (1, f(t, x))$. The new position is the superposition of the given vector (t, x) and direction vector $(1, f(t, x))$. Note that the direction vector has normalized the first component; alternatively, we can choose another normalisation: Let $K > 0$ be a given constant; then, the direction field is a graph of the mapping

$$\begin{bmatrix} t \\ x \end{bmatrix} \in \mathbb{R} \times \mathbb{R} \longmapsto \frac{K}{\sqrt{1 + (f(t, x))^2}} \begin{bmatrix} 1 \\ f(t, x) \end{bmatrix} \in \mathbb{R} \times \mathbb{R}. \tag{1.6}$$

Both definitions (1.5) and (1.6) are equivalent, and yield the same information. The direction field is displayed in Figure 1.2, where the direction field is evaluated at fixed points and displayed as arrows of fixed length $K = 0.25$. We observe the trajectories corresponding to initial conditions $(0, 1)$ and $(0, 0)$, see Figure 1.2, do not change in time. These are *stationary solutions* of the problem (1.1)–(1.2).

We now formulate a geometric interpretation of the solution of (1.1)–(1.2): We seek a trajectory in $\mathbb{R}^2$ such that

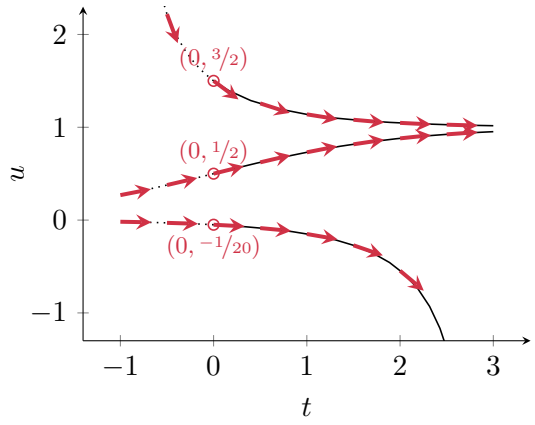


Figure 1.3: Logistic equation — selected tangents to trajectories. Parameters $a = 1$, $b = 1$, $c = 0$.

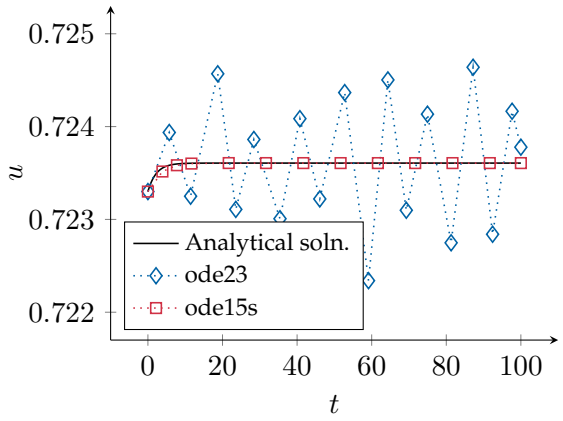


Figure 1.4: Logistic equation — trajectory with the initial condition $(0, 0.7233)$ compared to solution of `ode23` and `ode15s`. Parameters $a = 1$, $b = 1$, $c = 1/5$.

a) it satisfies the initial condition, and

b) at each point the tangent of the trajectory corresponds to the given direction field;

see Figure 1.3. The trajectories in Figure 1.1 were computed numerically; in particular, they were approximated in MATLAB using the function `ode23` (with default parameters for the solver); cf., Shampine and Reichelt (1997). We note that all numerical methods investigated in these notes (including `ode23`) defines a sequence of *discrete* times and solution values. These sequences are processed in such a way that plotting interpolates the output; consequently, the trajectories appear continuous.

Figure 1.4 displays the trajectory corresponding to the initial condition $t_0 = 0$, $x_0 = 0.7233$ and parameters $a = 1$, $b = 1$, $c = 1/5$, which is given by the explicit formula (analytical solution)

$$u(t) = \frac{\sqrt{5}}{10} \tanh\left((t - t_0)\frac{\sqrt{5}}{10} + \operatorname{arctanh}\left((2x_0 - 1)\sqrt{5}\right)\right) + \frac{1}{2}.$$

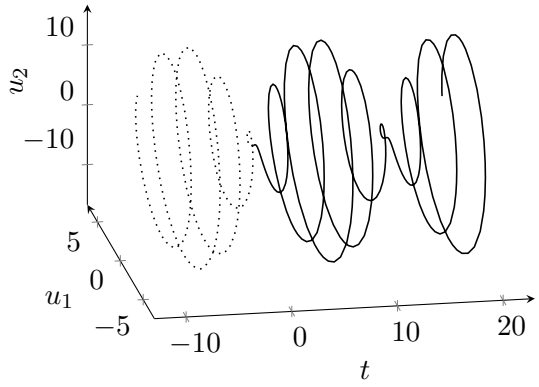


Figure 1.5: Linear oscillator — trajectory for initial condition $t_0 = 0$, $x_0 = (1, 0)$ with parameters $b = 9$, $c = 10$, $\omega = 2.5$. The motion is periodic with a period T, $T \sim 12.5664$.

Additionally, we numerically approximated the trajectory using the Matlab functions **ode23** and **ode15s** (cf., Shampine and Reichelt, 1997), with the linear interpolations of the numerical solutions displayed with dotted lines. We see that **ode23** distorts the reality while **ode15s** yields, at least qualitatively, a correct solution. It is possible to explain the issue: the initial condition is close to one of the stationary solutions. Later, in Section 5.3, we will talk about so-called *stiff problems*.

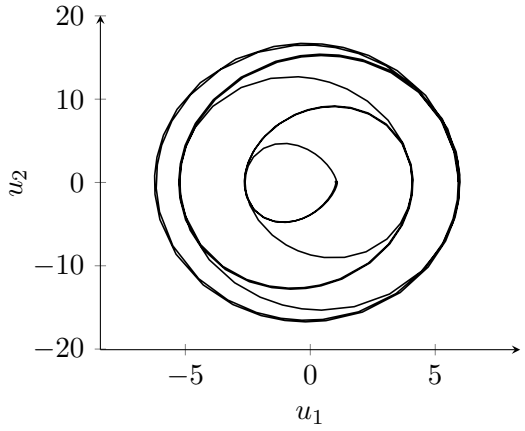


Figure 1.6: Linear oscillator — phase curve with parameters $b = 9$, $c = 10$, $\omega = 2.5$; i.e., the projection of the trajectory on state space $\mathbb{R}^2$.

Example 1.2 (Linear oscillator). Consider a system of two differential equations

$$\begin{aligned} x_1' &= x_2, \\ x_2' &= -bx_1 + c\cos(\omega t). \end{aligned}$$

where $b \geq 0$, $c \geq 0$ and $\omega \in \mathbb{R}$ are parameters. In vector notation, we have

$$x' = f(t, x) \equiv \begin{bmatrix} x_2 \\ -bx_1 + c\cos(\omega t) \end{bmatrix}. \tag{1.7}$$

We complete the system (1.7) with the initial condition

$$x(t_0) = x_0 \in \mathbb{R}^2. \tag{1.8}$$

The solution of the problem (1.7)–(1.8) is a continuously differentiable vector function $t \mapsto u(t) \in \mathbb{R}^2$ such that

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} = f(t, u(t)) \equiv \begin{bmatrix} u_2(t) \\ -bu_1(t) + c\cos(\omega t) \end{bmatrix} \tag{1.9}$$

for all $t \in \mathbb{R}$. Moreover, we require $u(t_0) = x_0 \in \mathbb{R}^2$.

If we choose the initial condition $(t_0, x_0) \in \mathbb{R} \times \mathbb{R}^2$, then for each time $t \in \mathbb{R}$ there exists a unique solution vector $u(t) \in \mathbb{R}^2$ of the problem (1.7)–(1.8). Therefore, there exists an operator

$$(t, t_0, x_0) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^2 \longmapsto u(t) \equiv \phi(t, t_0, x_0) \in \mathbb{R}^2. \tag{1.10}$$

For this particular problem we can construct the operator (1.10) explicitly; however, it is sufficient just to prove the *existence* of this operator. If the operator exists then we can approximate it numerically; this is the concept of the numerical solution of the problem (1.7)–(1.8).

The problem (1.7)–(1.8) models the oscillation of an elastic spring, where the components $u_1(t)$ and $u_2(t)$ of the vector $u(t) = \phi(t, t_0, x_0) \in \mathbb{R}^2$ are interpreted as the deflection and speed in time t, respectively.

The variable $x = (x_1, x_2) \in \mathbb{R}^2$ in (1.7) is called the *state variable*. In particular, x_1 and x_2 are deflection and speed, respectively. In relation to problem(1.7)–(1.8), the space $\mathbb{R}^2$ is called the *state space*. Parameters b, c and ω are fixed; in particular, b is the elasticity modulus, c is the amplitude of the oscillations and ω is the frequency of the acting force.

The problem of the linear oscillator is usually formulated via a linear second order differential equation

$$x'' + bx = c\cos(\omega t) \tag{1.11}$$

with the initial condition $(x(t_0), x'(t_0)) = x_0 \in \mathbb{R}^2$. The mentioned initial value problem for (1.11) is equivalent to the problem (1.7)–(1.8). In general, differential equations of higher orders can be transformed to first order systems.

The *trajectory*

$$t \longmapsto (t, \phi(t, t_0, x_0)) \in \mathbb{R} \times \mathbb{R}^2,$$

in Figure 1.5, was computed numerically with `ode23`, and period T of this motion was estimated as $T = 12.5664$. In Figure Figure 1.6 the trajectory is projected onto the state space

$$t \longmapsto \phi(t, t_0, x_0) \in \mathbb{R}^2.$$

The resulting object is called the *phase curve*.

1.2 Formulation of the problem

We are going to formulate the *initial value problem* (Cauchy problem) for a system of *Ordinary Differential Equations* (ODE). Without loss of generality, we consider systems of the first order. The initial value problem models evolution in a finite dimensional *state space*. We will identify the state space with the linear space $\mathbb{R}^n$, with time as a scalar parameter t.

Data of the problem:

1. the initial condition; i.e. a given state $x_0 \in \mathbb{R}^n$ at a particular time t_0.
2. a mapping
$$f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n. \tag{1.12}$$

The mapping (1.12) is called the *right-hand side* of the ODE. We assume that

- the domain of the right-hand side f is defined on an open set $J \times D \subset \mathbb{R} \times \mathbb{R}^n$, where J is an interval,
- $t_0 \in J$ and $x_0 \in D$, i.e., the initial condition $(t_0, x_0) \in \mathbb{R} \times \mathbb{R}^n$ belongs to the domain of the right-hand side, and
- the right-hand side is continuous, i.e.,
$$f \in C(J \times D, \mathbb{R}^n). \tag{1.13}$$

We formally define the *initial value problem* (IVP) for the ODE as:

$$x' = f(t, x), \quad x(t_0) = x_0. \tag{IVP}$$

We have to specify the meaning of the problem (IVP):

Definition 1.1 (Solution of the initial value problem)**.** Let there exist

1. an open interval I, $I \subset J$, containing t_0
2. a vector function $u : \mathbb{R} \to \mathbb{R}^n$, continuously differentiable on I, i.e., $u \in C^1(I, \mathbb{R}^n)$.

such that
$$u'(t) = f(t, u(t)) \tag{1.14}$$
for each $t \in I$. Moreover, let the initial value condition
$$u(t_0) = x_0 \tag{1.15}$$
be satisfied. Then we say that the function u is the solution of the initial value problem on the interval I.

Remark 1.2 (Integral formulation). If $f \in C(J \times D, \mathbb{R}^n)$ then a function u satisfies (1.14)–(1.15) if and only if
$$u(t) = x_0 + \int_{t_0}^{t} f(s, u(s))\,\mathrm{d}s \tag{1.16}$$
for each $t \in I$.

The right-hand side in the initial value problem (IVP) can be interpreted as the *direction field* (or *slope field*) and the *vector field*.

Definition 1.3 (Direction field or Slope field)**.** Let $J \times D$ be the domain of the right-hand side f; then, the direction field is a graph of the mapping
$$\begin{bmatrix} t \\ x \end{bmatrix} \in J \times D \longmapsto \begin{bmatrix} 1 \\ f(t, x) \end{bmatrix} \in \mathbb{R} \times \mathbb{R}^n. \tag{1.17}$$

The direction field describes the displacement of a given vector $(t, x) \in \mathbb{R} \times \mathbb{R}^n$ to the position $(t, x)+(1, f(t, x)) \in \mathbb{R}\times\mathbb{R}^n$. The new position is the superposition of the given vector (t, x) and direction vector $(1, f(t, x))$. Note that the direction vector has been normalized in the first component. We can choose another normalisation: Let $K > 0$ be a given constant;, then the direction field is a graph of the mapping

$$\begin{bmatrix} t \\ x \end{bmatrix} \in J \times D \longmapsto \frac{K}{\sqrt{(1 + (f(t, x))^2)}} \begin{bmatrix} 1 \\ f(t, x) \end{bmatrix} \in \mathbb{R} \times \mathbb{R}^n. \tag{1.18}$$

Both definitions (1.17) and (1.18) are equivalent.

Definition 1.4 (Vector field)**.** Let $J \times D$ be the domain of the right-hand side f. For each fixed $t \in J$ we define the vector field as a graph of the mapping

$$x \in D \longmapsto f(t, x) \in \mathbb{R}^n. \tag{1.19}$$

Let time $t \in J$ be fixed; then, the vector field describes the displacement of a given point $x \in D$ to a new position $x + f(t, x) \in \mathbb{R}^n$. This position is a superposition of x and an increment $f(t, x)$. This increment is interpreted as the immediate *velocity* $x' = f(t, x)$ at time $t \in J$ in the point $x \in D$ of the state space.

An important class of ODE are *autonomous* ODEs:

Definition 1.5 (Autonomous ODE)**.** Let $J \times D$ be the domain of the right-hand side f . If $f(t, x) = f(x)$ for each $(t, x) \in J \times D$, i.e., independent of t, then we say that the ODE is *autonomous*.

Without loss of generality we assume that $J \equiv (-\infty, +\infty)$.

1.3 Flow of a vector field

We consider the initial value problem (IVP). So far we assume continuity of the right-hand side; i.e. $f \in C(J \times D, \mathbb{R}^n)$. In order to proof the existence and *uniqueness* of the solution we need a stronger assumption then just continuity of the right-hand side:

Definition 1.6 (Local Lipschitz continuity)**.** Let $f : J \times D \to \mathbb{R}^n$ and $f \in C(J \times D, \mathbb{R}^n)$. We say that f is *locally Lipschitz continuous* on $J \times D$ provided that the following holds: For each $(t_0, x_0) \in J \times D$ there exists an open neighbourhood $\widetilde{J} \times \widetilde{D}$ of the point (t_0, x_0) such that $f : \widetilde{J} \times \widetilde{D} \to \mathbb{R}^n$ is Lipschitz continuous; i.e., there exists a constant $L \geq 0$ such that

$$\|f(t, x) - f(t, y)\| \leq L \|x - y\| \tag{1.20}$$

for all $t \in \widetilde{J}$ and $x, y \in \widetilde{D}$.

Theorem 1.7 (Picard-Lindelöf — Local existence and uniqueness)**.** *Let the right-hand side f be locally Lipschitz continuous on $J \times D$. Then the problem* (IVP) *is locally uniquely solvable; i.e., for each initial condition $(t_0, x_0) \in J \times D$ it holds that there exists an open interval $I \subset J$ containing $t_0 \in I$ and a function $u \in C^1(I, \mathbb{R}^n)$ such that a vector function $t \mapsto u(t)$ is the unique solution of the equation* (1.14) *on the interval I that satisfies the initial condition* (1.15)*.*

Proof. See Kurzweil (1973, 1986) □

Remark 1.8 (a sufficient condition). Let $f \in C(J \times D, \mathbb{R}^n)$, $\frac{\partial f}{\partial x_i} \in C(J \times D, \mathbb{R}^n)$, $i = 1, \ldots, n$; then, f is locally Lipschitz continuous. Therefore, the initial value problem (IVP) is locally uniquely solvable.

Consider a solution due to Theorem 1.7 and let $(t_0, x_0) \in J \times D$ be an initial condition; then, the solution $t \mapsto u(t)$ exists on the interval I. Employing the axiom of choice (AC) we can extend the existing solution to a larger open interval. Assume that there exists an open interval $\hat{I}$, $I \subset \hat{I} \subset J$, and a solution $t \longmapsto \hat{u}(t)$ of (IVP) on the interval $\hat{I}$. Note that $\hat{u}(t) = u(t)$ for $t \in I$. We say that the solution $t \longmapsto \hat{u}(t)$ is the *extension* of the solution on $\hat{I}$. We naturally define a *trivial extension* which is related to the case $I = \hat{I}$. The *maximal solution* of (IVP) is the solution $t \longmapsto u(t)$ on an interval $\mathcal{J}$ to which there does not exist a non-trivial extension.

Theorem 1.9 (Global solution = Maximal solution)**.** *Let the right-hand side f be locally Lipschitz continuous on $J \times D$. Given the initial condition $(t_0, x_0) \in J \times D$, the corresponding* (IVP) *has a maximal solution on an open interval $\mathcal{J} = \mathcal{J}(t_0, x_0)$.*

Proof. See Deuflhard and Bornemann (2012, Theorem 2.9, p. 39). For additional detail, see Kurzweil (1973, 1986). □

Definition 1.10 (Maximal solution interval)**.** We denote the limits of $\mathcal{J}(t_0, x_0)$ as

$$\mathcal{J}(t_0, x_0) = \left(t^-(t_0, x_0), t^+(t_0, x_0)\right). \tag{1.21}$$

We now introduce an operator related to (IVP).

Definition 1.11 (Flow of a vector field)**.** Let f be locally Lipschitz continuous on $J \times D$. Given an initial condition $(t_0, x_0) \in J \times D$ we consider the global solution of (IVP); i.e., the vector function $t \mapsto u(t)$ defined for $t \in \mathcal{J}(t_0, x_0)$. We define an operator $\phi : \mathbb{R} \times \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ as follows:

$$t \in \mathcal{J}(t_0, x_0), (t_0, x_0) \in J \times D \longmapsto \phi(t, t_0, x_0) = u(t) \in \mathbb{R}^n. \tag{1.22}$$

The operator ϕ is called the *flow* of the vector field f.

Under the assumption of Theorem 1.9 we know that the flow ϕ exists. In Chapters 2 and 3 we will show how to approximate the flow ϕ numerically. However, in rare cases we can construct the flow ϕ explicitly.

Example 1.3 (Explicit constructions of ϕ). Consider the following scalar ODEs:

1. $x' = ax$, $x(t_0) = x_0$, where a is a parameter; then, $J \times D = \mathbb{R} \times \mathbb{R}$ and

$$u(t) = \phi(t, t_0, x_0) = \mathrm{e}^{a(t-t_0)}\, x_0, \quad \mathcal{J}(t_0, x_0) = (-\infty, \infty).$$

2. $x' = x^2$, $x(t_0) = x_0 > 0$; then, $J \times D = \mathbb{R} \times (0, \infty)$,

$$u(t) = \phi(t, t_0, x_0) = \frac{x_0}{1 - (t - t_0)x_0}, \quad \mathcal{J}(t_0, x_0) = \left(-\infty, \frac{1}{x_0} + t_0\right).$$

3. $x' = -x^{-1/2}$, $x(t_0) = x_0 > 0$; then, $J \times D = \mathbb{R} \times (0, \infty)$,

$$u(t) = \phi(t, t_0, x_0) = \left(x_0^{3/2} - \frac{3(t - t_0)}{2} \right)^{2/3}, \quad \mathcal{J}(t_0, x_0) = \left(-\infty, \frac{2}{3} x_0^{3/2} + t_0 \right).$$

It holds that $\lim_{t \to \left(\frac{2}{3} x_0^{3/2} + t_0\right)^-} x'(t) = -\infty$.

The second and the third examples *blow up* and *collapse*, respectively.

Definition 1.12 (Trajectory & Phase curve)**.** Consider the initial value problem (IVP); then, the curve

$$t \in \mathcal{J}(t_0, x_0) \longmapsto (t, \phi(t, t_0, x_0)) \in \mathbb{R} \times \mathbb{R}^n \tag{1.23}$$

is called the *trajectory*. The projection of a trajectory onto the state space $\mathbb{R}^n$

$$t \in \mathcal{J}(t_0, x_0) \longmapsto \phi(t, t_0, x_0) \in \mathbb{R}^n \tag{1.24}$$

is called the *phase curve*.

For the linear oscillator (Example 1.2) Figure 1.5 and Figure 1.6 show examples of a trajectory and the relevant phase curve, respectively.

Remark 1.13. Let f be locally Lipschitz continuous and $(t, x) \in J \times D$. It can be verified that

1. $\phi(t, t, x) = x$
2. for each pair $t_1, t_2 \in \mathcal{J}(t, x)$,

$$\phi(t_2, t_1, \phi(t_1, t, x)) = \phi(t_2, t, x). \tag{1.25}$$

Definition 1.14 (Immediate velocity)**.** Let f be locally Lipschitz continuous; then, for a given $(t, x) \in J \times D$ we define a vector $x' \in \mathbb{R}^n$ by setting

$$x' = \lim_{\tau \to 0} \frac{1}{\tau} \left(\phi(t + \tau, t, x) - x \right). \tag{1.26}$$

We say that $x' \in \mathbb{R}^n$ is the *immediate velocity* at the point $(t, x) \in J \times D$.

Remark 1.15. It holds for the immediate velocity that $x' = f(t, x)$.

We now consider the autonomous ODE, see Definition 1.5. As right-hand side of f does not depend on time, we formally set $J \equiv (-\infty, +\infty)$. For local Lipschitz continuity of f on domain $J \times D$, we ignore J and instead just consider local Lipschitz continuity of f on the domain D.

For an autonomous ODE the flow ϕ has specific properties.

Theorem 1.16. *Let $f(t, x) \equiv f(x)$ be locally Lipschitz continuous on D; then,*

1. *for each $x_0 \in D$ and for each $t_0 \in \mathbb{R}$ it holds that*

$$t^-(0, x_0) + t_0 = t^-(t_0, x_0), \quad t^+(0, x_0) + t_0 = t^+(t_0, x_0),$$

2. *for each* $\tau \in (t^-(0,x_0), t^+(0,x_0))$,

$$\phi(t_0+\tau, t_0, x_0) = \phi(\tau, 0, x_0). \tag{1.27}$$

The result from Theorem 1.16 justifies the following definition.

Definition 1.17 (Flow of an autonomous ODE)**.** Let $f(t,x) \equiv f(x)$ be locally Lipschitz continuous on D; then, for $t \in \mathcal{J}(0,x_0)$ and $x_0 \in D$ we set

$$\phi(t, x_0) \equiv \phi(t, 0, x_0).$$

The operator $\phi : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ acting as

$$t \in \mathcal{J}(0,x_0), x_0 \in D \longmapsto \phi(t,x_0) \in \mathbb{R}^n$$

is called the *flow* of the autonomous vector field f.

Every ODE can be converted to an autonomous ODE using a simple trick. Instead of (IVP) we consider the following initial value problem:

$$\begin{bmatrix} t' \\ x' \end{bmatrix} = \begin{bmatrix} 1 \\ f(t,x) \end{bmatrix}, \quad \begin{bmatrix} t(t_0) \\ x(t_0) \end{bmatrix} = \begin{bmatrix} t_0 \\ x_0 \end{bmatrix}. \tag{1.28}$$

This initial value problem (1.28) is equivalent to (IVP).

Remark 1.18. Note that the variable t in the problem (1.28) does not equate to *time* (cf. (IVP)) but is just the first component of the state variable. For *time* in (1.28) we instead choose a different variable, such as $\tau \in \mathbb{R}$. The problem (1.28) is an autonomous ODE since the right-hand side *does not* depend on τ.

We define a vector field

$$z \longmapsto F(z) \equiv \begin{bmatrix} 1 \\ f(z) \end{bmatrix} \in \mathbb{R}^{n+1} \quad z = \begin{bmatrix} t \\ x \end{bmatrix} \in J \times D; \tag{1.29}$$

then, the corresponding initial value problem for the system (1.29)

$$z' = F(z), \quad z(t_0) = z_0 \equiv \begin{bmatrix} t_0 \\ x_0 \end{bmatrix} \tag{1.30}$$

has a solution on the interval $I \subset J$ if and only if the initial value problem (IVP) has a solution on I. As system (1.30) is autonomous we can shift t_0 to the origin 0; i.e., we consider

$$z' = F(z), \quad z(0) = z_0. \tag{1.31}$$

The solutions of the problems (1.30) and (1.31) correspond up to the *phase shift*.

The flow of autonomous vector field F is the operator

$$\tau, z_0 \longmapsto \Phi(\tau, z_0) \equiv \begin{bmatrix} t_0 + \tau \\ \phi(t_0+\tau, t_0, x_0) \end{bmatrix} \in \mathbb{R} \times \mathbb{R}^n, \tag{1.32}$$

where $\tau \in \mathcal{J}(t_0,x_0) - t_0$ and $(t_0,x_0) \in J \times D$.

1.4 Taylor expansion of the flow

We assume that $f \in C^k(J \times D, \mathbb{R}^n)$, $k \geq 1$. The choice of k specifies the smoothness of the vector field f. The existence and uniqueness of (IVP) is guaranteed due to Remark 1.8.

Let us fix $(t, x) \in J \times D$ and consider the solution u of (IVP) with initial conditions $(t, x) \in J \times D$. We define $u(t + \tau) \equiv \phi(t + \tau, t, x)$ for sufficiently small τ. Due to (1.16), this is equivalent to

$$u(t + \tau) = x + \int_t^{t+\tau} f(t + s, u(t + s)) \, \mathrm{d}s. \tag{1.33}$$

Due to our assumption on the smoothness of f, the function $\tau \longmapsto u(t + \tau)$ is $(k + 1)$-times continuously differentiable at the origin $\tau = 0$; hence, there exists an Taylor expansion

$$u(t + \tau) = u(t) + \tau u'(t) + \frac{\tau^2}{2} u''(t) + \cdots + \frac{\tau^j}{j!} u^{(j)}(t) + \cdots + \frac{\tau^k}{k!} u^{(k)}(t) + \mathcal{O}(\tau^{k+1}). \tag{1.34}$$

Note that $u(t) = x$. We say that the expansion is of the *k-th order*. The coefficients $u^{(j)}(t)$ can be interpreted as

$$u^{(j)}(t) = \frac{\partial^j}{\partial \tau^j} \phi(t + \tau, t, x)\Big|_{\tau=0}, \quad j \geq 0. \tag{1.35}$$

Our aim is to express the Taylor coefficients $u^{(j)}(t)$ by means of data of the problem; i.e., differentials of the right-hand side f.

Let us compute four terms of the expansion. To this end, we assume that $f \in C^3(J \times D, \mathbb{R}^n)$. In order to simplify the computation we consider a scalar ODE; i.e., $n = 1$. Differentiating the equation (1.14) with respect to $\frac{\mathrm{d}}{\mathrm{d}t}$ we get

$$\begin{aligned} u'(t) &= f(t, u(t)) = f(t, x) = f \\ u''(t) &= f_t(t, x) + f_x(t, x)u'(t) = f_t + f_x f \\ u'''(t) &= f_{tt} + 2f_{tx}f + f_{xx}f^2 + f_x(f_t + f_x f) \end{aligned} \tag{1.36}$$

Here, we skip the argument (t, x) of the function f in order to simplify the notation and similarly for its partial derivatives f_x, f_{tx}, etc.. We then substitute (1.36) into (1.34). Additionally, we assume an autonomous ODE which gives that $f_t = f_{tt} = f_{tx} = 0$ and, without loss of generality, we set $t = 0$. Recalling Theorem 1.16 and Definition 1.17 we obtain the simplified expansion:

$$u(\tau) \equiv \phi(\tau, x) = x + \tau f + \frac{\tau^2}{2} f_x f + \frac{\tau^3}{6} f_{xx} f^2 + \frac{\tau^3}{6} f_x f_x f + \mathcal{O}(\tau^4).$$

We now generalize the above formula to an arbitrary dimension n using an analogy in our reasoning. The term $f_x f \in \mathbb{R}$ corresponds in the vector analogy to the term $f^{(1)}[f] \in \mathbb{R}^n$, where $f^{(1)} \in \mathbb{R}^{n \times n}$ is the differential computed in the direction of $f \in \mathbb{R}^n$, and the functions f and $f^{(1)}$ are evaluated at x; hence, by definition,

$$f^{(1)}[f] = \sum_{i=1}^n \frac{\partial f}{\partial x_i} f_i.$$

Similarly, the term $f_{xx} f^2 \in \mathbb{R}$ corresponds to the term

$$f^{(2)}[f, f] = \sum_{i,j=1}^n \frac{\partial^2 f}{\partial x_i \partial x_j} f_i f_j \in \mathbb{R}^n.$$

and the term $f_x f_x f = f_x^2 f \in \mathbb{R}$ to the term

$$f^{(1)}[f^{(1)}[f]] = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} \sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j} f_j = \sum_{i,j=1}^{n} \frac{\partial f}{\partial x_i} \frac{\partial f_i}{\partial x_j} f_j \in \mathbb{R}^n.$$

Lemma 1.19. *Consider an autonomous ODE, assume $f \in C^3(D, \mathbb{R}^n)$ and choose $x \in D$; then,*

$$\phi(\tau, x) = x + \tau f + \frac{\tau^2}{2} f^{(1)}[f] + \frac{\tau^3}{6}(f^{(2)}[f, f] + f^{(1)}[f^{(1)}[f]]) + \mathcal{O}(\tau^4). \tag{1.37}$$

The functions $f^{(i)}$, $i = 0, \ldots, 2$, are evaluated at the point x.

Proof. By computing the derivatives $u'(0)$, $u''(0)$ and $u'''(0)$ in the expansion (1.34) we obtain that $u'(0) = f$, $u''(0) = \frac{\mathrm{d}f}{\mathrm{d}t} = f^{(1)}[f]$, $u'''(0) = \frac{\mathrm{d}}{\mathrm{d}t} f^{(1)}[f] = f^{(2)}[f, f] + f^{(1)}[f^{(1)}[f]]$. Substituting these into the expansion (1.34) completes the proof. □

Corollary 1.20. *Assume $f \in C^3(J \times D, \mathbb{R}^n)$ and choose $(t, x) \in J \times D$; then,*

$$\phi(t + \tau, t, x) = x + \tau f + \frac{\tau^2}{2} f^{(1)}[F] + \frac{\tau^3}{6}(f^{(2)}[F, F] + f^{(1)}[f^{(1)}[F]]) + \mathcal{O}(\tau^4), \tag{1.38}$$

where $F \equiv (1, f) \in \mathbb{R} \times \mathbb{R}^n$. The functions $f^{(i)}$, $i = 0, \ldots, 2$, are evaluated at (t, x).

Proof. We define the autonomous vector field F according to (1.29) and the relevant flow $\Phi(\tau, z)$; cf., (1.32). According to Lemma 1.19,

$$\Phi(\tau, z) = z + \tau F + \frac{\tau^2}{2} F^{(1)}[F] + \frac{\tau^3}{6}(F^{(2)}[F, F] + F^{(1)}[F^{(1)}[F]]) + \mathcal{O}(\tau^4) \in \mathbb{R} \times \mathbb{R}^n,$$

where $F^{(1)}[F] = (0, f^{(1)}[F])$, $F^{(2)}[F, F] = (0, f^{(2)}[F, F])$, and $F^{(1)}[F^{(1)}[F]] = (0, f^{(1)}[f^{(1)}[F]])$. The last n components of the vector $\Phi(\tau, z) \in \mathbb{R} \times \mathbb{R}^n$ are equal to $\phi(t + \tau, t, x) \in \mathbb{R}^n$. □

We state the following two results without proof.

Lemma 1.21. *Consider an autonomous ODE, assume that $f \in C^4(D, \mathbb{R}^n)$ and choose $x \in D$; then,*

$$\begin{aligned}\phi(\tau, x) = x + \tau f + \frac{\tau^2}{2} f^{(1)}[f] + \frac{\tau^3}{6}(f^{(2)}[f, f] + f^{(1)}[f^{(1)}[f]]) \\ + \frac{\tau^4}{24}(f^{(3)}[f, f, f] + 3f^{(2)}[f^{(1)}[f], f] + f^{(1)}[f^{(2)}[f, f]] + f^{(1)}[f^{(1)}[f^{(1)}[f]]]) + \mathcal{O}(\tau^5).\end{aligned}$$

The functions $f^{(i)}$, $i = 0, \ldots, 3$, are evaluated at x.

Corollary 1.22. *Assume $f \in C^4(J \times D, \mathbb{R}^n)$ and choose $(t, x) \in J \times D$; then,*

$$\begin{aligned}\phi(t + \tau, t, x) = x + \tau f + \frac{\tau^2}{2} f^{(1)}[F] + \frac{\tau^3}{6}(f^{(2)}[F, F] + f^{(1)}[f^{(1)}[F]]) \\ + \frac{\tau^4}{24}(f^{(3)}[F, F, F] + 3f^{(2)}[f^{(1)}[F], F] + f^{(1)}[f^{(2)}[F, F]] + f^{(1)}[f^{(1)}[f^{(1)}[F]]]) \\ + \mathcal{O}(\tau^5),\end{aligned}$$

where $F \equiv (1, f) \in \mathbb{R} \times \mathbb{R}^n$. The functions $f^{(i)}$, $i = 0, \ldots, 3$, are evaluated at (t, x).

Remark 1.23. Assume that $f \in C^k(J \times D, \mathbb{R}^n)$, $k \geq 1$; then, for each $(t, x) \in J \times D$ there exists a Taylor expansion of the k-th order. This expansion is expressed by differentials of the right-hand side, which are called the *elementary* differentials. The number of these elementary differentials for various order k is shown below:

k	1	2	3	4	5	6	7	8	9	10
# differentials	2	3	5	9	18	38	86	201	487	1206

We can see that the number of these elementary differentials explodes with k.

CHAPTER 2

One-step methods

Our aim is to compute the *numerical solution* of the initial value problem (IVP). We search for the trajectory $u(t) = \phi(t, t_0, x_0)$ on a *finite* closed interval $t \in [t_0, T]$, where we assume that $t_0 < T < t^+(t_0, x_0)$. Additionally, we assume that the right-hand side is sufficiently smooth.

In case that we need to solve (IVP) on an interval $t^-(t_0, x_0) < T < t_0$, we change the sign of the vector field, i.e., we set $f := -f$ in (IVP), and consider the problem on the interval $[t_0, 2t_0 - T]$.

EulerWe first study the the *Euler method* (1768), which will serve as a prototype to all the methods we shall study in this chapter. To this end, we first define a partition

$$\{t_j\}_{j=0}^N, \qquad t_{j+1} > t_j, \qquad t_N = T, \tag{2.1}$$

of the interval $[t_0, T]$ into N intervals, where t_0 is defined by the initial condition (1.15).

We define the recursive sequence $\{u_j\}_{j=0}^N$

$$u_{j+1} = u_j + (t_{j+1} - t_j) f(t_j, u_j). \tag{2.2}$$

The i-th point $u_j \in \mathbb{R}^n$ is interpreted as an approximation of the state $u(t_j) \in \mathbb{R}^n$ at the time t_j.

Let us consider the scalar equation $x' = 0.3x \sin(t - 4/3)$ with initial condition $x(1) = 2$. We search for the solution $u(t)$ on the interval $[1, 3]$, see Figure 2.1.

We consider a *equidistant* (*uniform*) partition of the interval $[t_0, T]$; i.e., $t_{j+1} - t_j \equiv \tau = \frac{T - t_0}{N}$, $j = 1, \dots, N$. The numerical solution are the couples $(t_j, u_j) \in \mathbb{R} \times \mathbb{R}^n$, $j = 0, \dots, N$, $n = 1$, which are generated by the sequence (2.2). In the case we need to approximate the solution at a given time t where $t_j < t < t_{j+1}$, we use, e.g., the linear interpolation:

$$\begin{bmatrix} t \\ u(t) \end{bmatrix} \approx \frac{t_{j+1} - t}{t_{j+1} - t_j} \begin{bmatrix} t_j \\ u_j \end{bmatrix} + \frac{t - t_j}{t_{j+1} - t_j} \begin{bmatrix} t_{j+1} \\ u_{j+1} \end{bmatrix}.$$

This interpolation for $N = 4 \implies \tau = 1/2$ is shown in Figure 2.1. The numerical procedure is called *Euler's polyhedron formula*. We can control the approximation quality by choosing larger N and, consequently, choosing finer *step size* τ.

We can, alternatively, choose a *non-equidistant* (*non-uniform*) partition (2.1) defining the step size $\tau = t_{j+1} - t_j$ at each time instant t_j in accordance with some prescribed rules.

The iteration (2.2) in the state space $\mathbb{R}^n$ can be equivalently formulated in the *time-space* $\mathbb{R} \times \mathbb{R}^n$:

$$\begin{bmatrix} t_j \\ u_j \end{bmatrix} \longmapsto \begin{bmatrix} t_j \\ u_j \end{bmatrix} + (t_{j+1} - t_j) \begin{bmatrix} 1 \\ f(t_j, u_j) \end{bmatrix} \equiv \begin{bmatrix} t_{j+1} \\ u_{j+1} \end{bmatrix}. \tag{2.3}$$

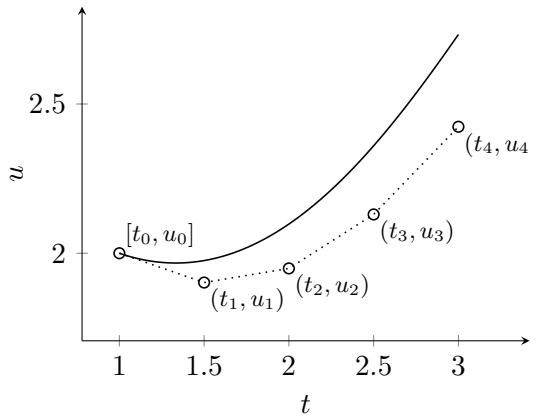


Figure 2.1: Euler method with step size $\tau = 1/2$ — Comparison of actual solution vs. Euler's polyhedron formula

The iteration (2.3) has the following geometrical meaning: Let us recall the notion of the direction field, Definition 1.3; then, we consider the straight line in $\mathbb{R} \times \mathbb{R}^n$ starting at the point (t_j, u_j) in direction $(1, f(t_j, u_j))$, see (1.17).

The straight line has the parametric form

$$\tau \in \mathbb{R} \longmapsto \begin{bmatrix} t_j \\ u_j \end{bmatrix} + \tau \begin{bmatrix} 1 \\ f(t_j, u_j) \end{bmatrix} \in \mathbb{R} \times \mathbb{R}^n.$$

Hence, the $(j+1)-$th step of the iterations (2.3) corresponds to the choice $\tau \equiv t_{j+1} - t_j$.

2.1 Discretisation of the vector field

In this section, we introduce the definitions of a *one-step method*, the *discrete flow of the vector field*, the *local discretisation error* and define the *order of the method*. We will give several examples of one-step methods, which are based on numerical integration; in particular, we define the Euler method, the Runge method, the Implicit Euler method, the Implicit Trapezoidal method, the Heun method, and the Runge-Kutta method.

One-step methods are defined via *one-step recursions*. The objective is to approximate the flow $u(t) = \phi(t, t_0, x_0)$ of a given vector field f via a *time discretisation*. We first give the formal definition of this discretisation, understood as one step of the recursion:

Definition 2.1 (One-step method $\equiv$ the discrete flow of the vector field)**.** Let f be locally Lipschitz continuous on $J \times D$ and the mapping $\psi : J \times D \times \mathbb{R} \to \mathbb{R}^n$,

$$t \in J, x \in D, \tau \geq 0 \quad \longmapsto \quad \psi(t + \tau, t, x) \in \mathbb{R}^n, \tag{2.4}$$

satisfies at each point $(t, x) \in J \times D$ the consistency condition

$$\psi(t, t, x) = \lim_{\tau \to 0^+} \frac{\psi(t + \tau, t, x) - x}{\tau} = f(t, x). \tag{2.5}$$

We say that the operator ψ is the *discrete flow of the vector field* f and the parameter τ is called the *time step*.

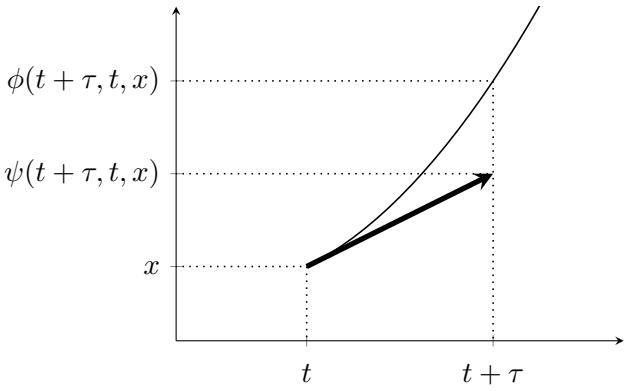


Figure 2.2: One-step method — The discrete flow $\psi(t+\tau,t,x)$ vs. the exact solution $\phi(t+\tau,t,x)$

Remark 2.2. The requirement (2.5) can be rephrased as the existence of the immediate velocity, see (1.26).

Definition 2.1 is a recipe defining how to obtain the numerical solution $\psi(t+\tau,t,x)$ at time $t+\tau$, starting at the initial condition (t,x):

$$\begin{bmatrix} t \\ x \end{bmatrix} \longmapsto \begin{bmatrix} t+\tau \\ \psi(t+\tau,t,x) \end{bmatrix} \in \mathbb{R}\times\mathbb{R}^n. \tag{2.6}$$

Intuitively, we expect that $\psi(t+\tau,t,x)$ yields a good approximation of $\phi(t+\tau,t,x)$ for small time steps $\tau \geq 0$; cf. Figure 2.2.

Consider the initial value problem (IVP), we now explain how to solve the problem numerically by means of a chosen one-step method.

For a given partition (2.1) of the interval $[t_0,T]$, we construct the sequence $\{u_j\}_{j=0}^N$ as the iterations of the mapping ψ,

$$u_j \longmapsto u_{j+1} = \psi(t_{j+1},t_j,u_j) \qquad j=0,\dots,N \tag{2.7}$$

or, equivalently,

$$\begin{bmatrix} t_j \\ u_j \end{bmatrix} \longmapsto \begin{bmatrix} t_{j+1} \\ u_{j+1} \end{bmatrix} = \begin{bmatrix} t_{j+1} \\ \psi(t_{j+1},t_j,u_j) \end{bmatrix}, \qquad j=0,\dots,N. \tag{2.8}$$

In the case of the Euler method, the operator ψ is defined as follows.

Definition 2.3 (Euler method, 1768)**.** Let f be locally Lipschitz continuous on $J\times D$, assume that $(t,x)\in J\times D$, and set $\kappa_1 = f(t,x)$. We define

$$\psi(t+\tau,t,x) \equiv x+\tau\kappa_1 \tag{2.9}$$

for a given $\tau \geq 0$.

Remark 2.4. The Euler method formulated via Definition 2.9 yields the iterations (2.1) and (2.2).

Given a one-step method, we want to be able to measure the accuracy of the given solution.

Definition 2.5 (Local discretisation error. Order of the method)**.** Let f be locally Lipschitz continuous on $J \times D$, ϕ be the flow of the vector field f, choose $(t, x) \in J \times D$ and a time step $\tau > 0$. Consider a particular one-step method; i.e., let $\psi(t + \tau, t, x)$ be the discrete flow of the vector field f, see (2.4). Then the *local discretisation error* of the method at the point (t, x) and the chosen τ is defined as

$$d(t + \tau, t, x) \equiv \|\phi(t + \tau, t, x) - \psi(t + \tau, t, x)\| . \tag{2.10}$$

If there exists positive integer p such that

$$d(t + \tau, t, x) = \mathcal{O}(\tau^{p+1}) \qquad \text{for } \tau \to 0^+; \tag{2.11}$$

then, the method is of the *order* p at the point (t, x).

Remark 2.6. Note that the property (2.5) implies

$$\lim_{\tau \to 0^+} \frac{d(t + \tau, t, x)}{\tau} = 0;$$

i.e., $d(t + \tau, t, x) = \mathcal{O}(\tau)$. Hence, this is the required property of the function d. The order p measures the accuracy.

Remark 2.7 (Autonomous ODE: Discrete flow). If $f(t, x) \equiv f(x)$ then

$$\psi(t + \tau, t, x) = \psi(\tau, 0, x), \qquad \tau \geq 0, x \in D, t \in \mathbb{R}^n. \tag{2.12}$$

Definition 2.8 (Autonomous ODE: Local discretisation error. Order of the method)**.** Let $f(t, x) \equiv f(x)$ for $t \in (-\infty, +\infty)$, f be locally Lipschitz continuous on D, ϕ denote the flow of the vector field f and ψ the discrete flow of the vector field f. Then, the local discretisation error of the method ψ at the point $x \in D$ is a function of τ:

$$d(\tau, x) \equiv \|\phi(\tau, 0, x) - \psi(\tau, 0, x)\| . \tag{2.13}$$

If there exists a positive integer p such that

$$d(\tau, x) = \mathcal{O}(\tau^{p+1}) \quad \text{for } \tau \to 0^+.$$

We say that the method ψ is of the order p at the point $x \in D$.

Corollary 2.9 (Order of the Euler method)**.** *Assume that $f \in C^1(J \times D, \mathbb{R}^n)$, and apply one step of the Euler method (2.9) at an arbitrary $(t, x) \in J \times D$; then, $d(t + \tau, t, x) = \mathcal{O}(\tau^2)$. Hence, the order of the Euler method is $p = 1$.*

Proof. According to (1.38), $\phi(t + \tau, t, x) = x + \tau f + \mathcal{O}(\tau^2)$. Due to the definition of the method (2.9), the discrete flow $\psi(t + \tau, t, x) = x + \tau\kappa_1 = x + \tau f$; hence, by definition (2.10), $d(t + \tau, t, x) = \mathcal{O}(\tau^2)$. □

The one-step method (2.4) may be interpreted via *numerical quadrature*. Recall Definition 2.5; namely, the flow ϕ and the discrete flow ψ. According to the integral identity (1.16)

$$\phi(t+\tau,t,x)-x=\int_t^{t+\tau} f(s,u(s))\,\mathrm{d}s=\int_t^{t+\tau} u'(s)\,\mathrm{d}s. \tag{2.14}$$

This identity can be approximated by the numerical quadrature

$$\int_t^{t+\tau} u'(s)\,\mathrm{d}s=\sum_{j=1}^{N} b_j u'(s_j)+E(\tau) \tag{2.15}$$

with *coefficients* $b_j\in\mathbb{R}$ and *nodes* $s_j\in[t,t+\tau]$ for $j=1,\dots,N$, where $E(\tau)\in\mathbb{R}^n$ is the *error* of the quadrature. Note that $u'(s_j)=f(s_j,u(s_j))\in\mathbb{R}^n$.

Example 2.1 (Quadrature formulas). Let $g:[a,b]\to\mathbb{R}^n$ be a sufficiently smooth vector function. We will only consider *Lagrange quadrature formulas* here (Quarteroni et al., 2010, p. 372). The integral of a vector function $g=g(t)$ can be approximated by the finite sum

$$\int_a^b g(t)\,\mathrm{d}t\approx\sum_{j=1}^{N} b_j g(s_j)$$

with nodes $s_j\in[a,b]$ and coefficients (*weights*) $b_j\in\mathbb{R}$, $j=1,\dots N$; in particular, we consider the following quadrature rules

rectangle rule:

$$\int_a^b g(t)\,\mathrm{d}t=(b-a)g(a)+\mathcal{O}(b-a)^2, \tag{2.16}$$

implicit rectangle rule:

$$\int_a^b g(t)\,\mathrm{d}t=(b-a)g(b)+\mathcal{O}(b-a)^2, \tag{2.17}$$

mid-point rule:

$$\int_a^b g(t)\,\mathrm{d}t=(b-a)g\left(\frac{a+b}{2}\right)+\mathcal{O}(b-a)^3, \tag{2.18}$$

trapezoidal rule:

$$\int_a^b g(t)\,\mathrm{d}t=\frac{b-a}{2}\left(g(a)+g(b)\right)+\mathcal{O}(b-a)^3, \tag{2.19}$$

Simpson rule:

$$\int_a^b g(t)\,\mathrm{d}t=\frac{b-a}{6}\left(g(a)+4g\left(\frac{a+b}{2}\right)+g(b)\right)+\mathcal{O}(b-a)^4, \tag{2.20}$$

$3/8$-rule:

$$\int_a^b g(t)\,\mathrm{d}t=\frac{b-a}{8}\left(g(a)+3g\left(\frac{2a+b}{3}\right)+3g\left(\frac{a+2b}{3}\right)+g(b)\right)+\mathcal{O}(b-a)^4. \tag{2.21}$$

We now analyse the general quadrature formula (2.14)–(2.15). Consider the simplest quadrature formula (2.16); hence, $N = 1$, $s_1 = t$, $b_1 = \tau$ and

$$\sum_{j=1}^{N} b_j u'(s_j) + E(\tau) = \tau u'(t) + \mathcal{O}(\tau^2) = \tau f(t,x) + \mathcal{O}(\tau^2). \tag{2.22}$$

Hence, from (2.14)–(2.15),

$$\phi(t+\tau,t,x) - x = \tau f(t,x) + \mathcal{O}(\tau^2). \tag{2.23}$$

We define the one-step method (2.4) neglecting the error $\mathcal{O}(\tau^2)$ in (2.23); i.e., we define

$$\psi(t+\tau,t,x) - x = \tau f(t,x). \tag{2.24}$$

We note that this corresponds to the formula for the Euler method (2.9).

We now consider the *mid-point rule* (2.18); i.e., $N = 1$, $s_1 = t + \tau/2$, $b_1 = \tau$.

$$\sum_{j=1}^{N} b_j u'(s_j) + E(\tau) = \tau u'(t+\tau/2) + \mathcal{O}(\tau^3) = \tau f(t+\tau/2, u(t+\tau/2)) + \mathcal{O}(\tau^3). \tag{2.25}$$

We need to approximate the unknown value $u(t+\tau/2)$; to this end, we use the Euler method

$$u(t+\frac{\tau}{2}) = u(t) + \frac{\tau}{2} f(t,u(t)) + \mathcal{O}(\tau^2) = x + \frac{\tau}{2} f + \mathcal{O}(\tau^2).$$

We conclude that

$$\phi(t+\tau,t,x) - x = \tau f(t+\frac{\tau}{2}, x+\frac{\tau}{2}f) + \mathcal{O}(\tau^3); \tag{2.26}$$

neglecting the error of order $\mathcal{O}(\tau^3)$ we get the following discrete flow

$$\psi(t+\tau,t,x) - x = \tau f(t+\frac{\tau}{2}, x+\frac{\tau}{2}f). \tag{2.27}$$

The above derived method is called the *Runge* method.

Definition 2.10 (Runge method, 1895)**.** Let f be locally Lipschitz continuous on $J \times D$, $(t,x) \in J \times D$. We set $\kappa_1 = f(t,x)$, $\kappa_2 = f(t+\frac{\tau}{2}, x+\frac{\tau}{2}\kappa_1)$, and define

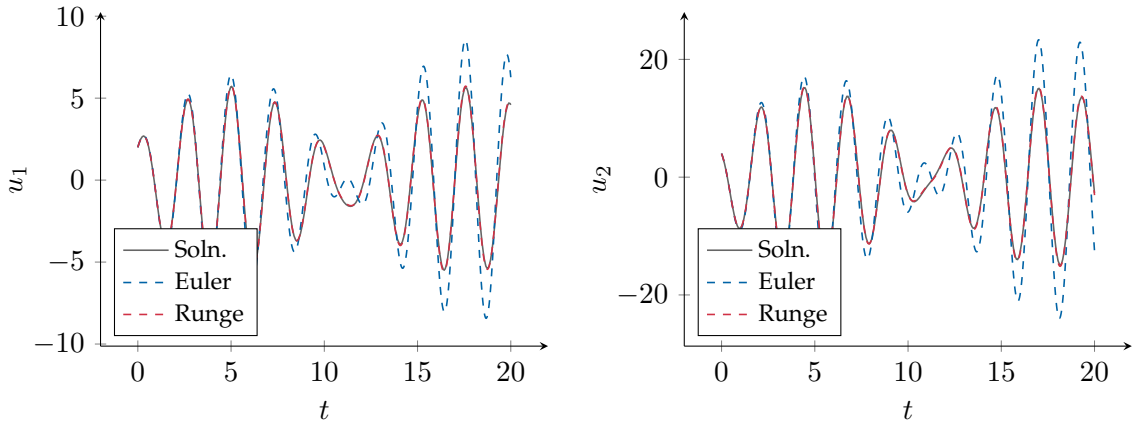
$$\psi(t+\tau,t,x) \equiv x + \tau\kappa_2 \tag{2.28}$$

for a given $\tau \geq 0$.

Corollary 2.11 (Order of the Runge method)**.** *Assume that $f \in C^2(J \times D, \mathbb{R}^n)$ and let us apply one step of the Runge method (2.28) at an arbitrary $(t,x) \in J \times D$. Then $d(t+\tau,t,x) = \mathcal{O}(\tau^3)$; therefore, the order of the Runge method is $p = 2$.*

Proof. The statement follows from (2.26) and (2.27). □

Remark 2.12. We compare the expenses (i.e. the overhead) of the Euler method (see Definition 2.9) and the Runge method (see Definition 2.10). Note that the cost can be estimated by the number of evaluations of f; hence, we compare the function evaluations:



(a) u_1 (b) u_2

Figure 2.3: Linear oscillator — comparison of Euler and Runge

Euler	κ_1	1
Runge	κ_1, κ_2	2

From this we would conclude that the Runge method is twice more expensive then the Euler method. We consider the linear oscillator Example 1.2 with different equidistant partitions of the same interval $[0, 20]$ with different time step size

$$\tau = \begin{cases} 0.01 & \text{for Euler,} \\ 0.02 & \text{for Runge,} \end{cases}$$

such that the computational cost should be the same for both methods. The experimental evidence, cf. Figure 2.3, shows the higher precision of the Runge method compared to the Euler method. We will consider details about the error analysis in Section 2.2, where the order of a particular method will play a crucial role.

We now consider the implicit rectangle quadrature rule (2.17); i.e., $N = 1$, $s_1 = t + \tau$, $b_1 = \tau$:

$$\sum_{j=1}^{N} b_j u'(s_j) + E(\tau) = \tau u'(t+\tau) + \mathcal{O}(\tau^2) = \tau f(t+\tau, u(t+\tau)) + \mathcal{O}(\tau^2). \tag{2.29}$$

Then

$$\phi(t+\tau, t, x) - x = \tau f(t+\tau, \phi(t+\tau, t, x)) + \mathcal{O}(\tau^2); \tag{2.30}$$

hence, neglecting the error, we define one-step method

$$\psi(t+\tau, t, x) - x = \tau f(t+\tau, \psi(t+\tau, t, x)). \tag{2.31}$$

Let us define $\kappa_1 \in \mathbb{R}^n$ such that

$$\tau\kappa_1 = \psi(t+\tau, t, x) - x;$$

then, from (2.31), we get

$$\kappa_1 = f(t+\tau, x+\tau\kappa_1).$$

Definition 2.13 (Implicit Euler method)**.** Let f be locally Lipschitz continuous on $J \times D$, assume that $(t,x) \in J \times D$, set $\kappa_1 = f(t+\tau, x+\tau\kappa_1)$; then, we define

$$\psi(t+\tau, t, x) \equiv x + \tau\kappa_1 \tag{2.32}$$

for a given $\tau \geq 0$.

The formula (2.32) can be considered as a recipe of the type (2.6). However, we note that the vector $\kappa_1 \in \mathbb{R}^n$ is not explicitly defined (like in the Euler method, cf. $\kappa_1 = f(t,x)$). The vector $\kappa_1 \in \mathbb{R}^n$ is defined implicitly, as a fixed point of the mapping

$$\kappa_1 \longmapsto f(t+\tau, x+\tau\kappa_1)$$

for a given $\tau \geq 0$. We will appreciate the above method (in general, the class of the implicit one-step methods) in Chapter 5 when solving *stiff problems*, see Section 5.3. for now, we will just discuss whether the vector κ_1 is defined correctly and how to compute it.

Remark 2.14. In order to show the existence of the fixed point, we make use of the Implicit Function Theorem. Assume that $f \in C^1(J \times D, \mathbb{R}^n)$, and let $(t,x) \in J \times D$ be fixed; then, we define the mapping

$$\kappa_1 \in \mathbb{R}^n, \tau \geq 0 \longmapsto G(\kappa_1, \tau) \equiv f(t+\tau, x+\tau\kappa_1) - \kappa_1 \in \mathbb{R}^n.$$

Since $f \in C^1$ then $G \in C^1$. Note that

1. $G(\kappa_1, \tau) = 0$ for $\kappa_1 = f(t,x) \in \mathbb{R}^n$ and $\tau = 0$
2. $\frac{\partial G}{\partial \kappa_1}(f(t,x), 0) = -I$ where $I \in \mathbb{R}^{n\times n}$ is the identity matrix.

There exists a vector function $\kappa_1 : \mathbb{R} \longmapsto \mathbb{R}^n$, $\kappa_1 = \kappa_1(\tau)$, $\kappa \in C^1(\Delta, \mathbb{R}^n)$, where Δ is an open interval containing $\tau = 0$, such that

$$f(t+\tau, x+\tau\kappa_1(\tau)) - \kappa_1(\tau) = 0 \in \mathbb{R}^n, \qquad \kappa_1(0) = f(t,x). \tag{2.33}$$

Corollary 2.15 (Order of the Implicit Euler method)**.** *Assume that $f \in C^1(J\times D, \mathbb{R}^n)$, and let us apply one step of the Implicit Euler method (2.32) at an arbitrary $(t,x) \in J \times D$; then $d(t+\tau, t, x) = \mathcal{O}(\tau^2)$. Therefore, the order of the Implicit Euler method is $p = 1$.*

Proof. Follows from Remark 2.14. □

Remark 2.16. We give a constructive proof of the existence of the fixed point. Consider the iterations $\kappa_1 \longmapsto f(t+\tau, x+\tau\kappa_1)$. We exploit the local Lipschitz continuity of f with a constant $L > 0$, see (1.20). If $\vartheta \in \mathbb{R}^n$ and $\eta \in \mathbb{R}^n$ then

$$\|f(t+\tau, x+\tau\vartheta) - f(t+\tau, x+\tau\eta)\| < L\|\tau\vartheta - \tau\eta\| = \tau L\|\vartheta - \eta\|. \tag{2.34}$$

If $\tau L < 1$ then the mapping $\kappa_1 \longmapsto f(t+\tau, x+\tau\kappa_1)$ is a *contraction mapping*. It can be shown that the sequence $\{\kappa_1^{(k)}\}_{k=0}^{\infty}$ defined by the iteration

$$\kappa_1^{(k+1)} = f(t+\tau, x+\tau\kappa_1^{(k)}) \tag{2.35}$$

converges to the fixed point $\kappa_1 = f(t+\tau, x+\tau\kappa_1)$ provided that the initial approximation $\kappa_1^{(0)}$ is sufficiently close to κ_1. This is called the *local convergence*.

The next two methods exploit the trapezoidal rule (2.19); i.e, $N = 2$, $s_1 = t$, $s_1 = t + \tau$, $b_1 = \tau/2$, $b_1 = \tau/2$. We derive one implicit (Definition 2.17) and one explicit method (Definition 2.19) from this quadrature rule.

From the quadrature formula we derive the following approximation of the flow

$$\phi(t+\tau,t,x) - x = \frac{\tau}{2}\left(f(t,x) + f(t+\tau,\phi(t+\tau,t,x))\right) + \mathcal{O}(\tau^3). \tag{2.36}$$

Neglecting the terms of order $\mathcal{O}(\tau^3)$, we derive the next formula for the discrete flow

$$\psi(t+\tau,t,x) - x = \frac{\tau}{2}\left(f(t,x) + f(t+\tau,\psi(t+\tau,t,x))\right). \tag{2.37}$$

We set $\kappa_1 = f(t,x)$ and define κ_2 such that

$$\psi(t+\tau,t,x) - x = \frac{\tau}{2}\kappa_1 + \frac{\tau}{2}\kappa_2;$$

then,

$$\kappa_2 = f(t+\tau, x + \frac{\tau}{2}\kappa_1 + \frac{\tau}{2}\kappa_2).$$

Definition 2.17 (Implicit Trapezoidal method)**.** Let f be locally Lipschitz continuous on $J \times D$, assume that $(t,x) \in J \times D$, and set $\kappa_1 = f(t,x)$ and $\kappa_2 = f(t+\tau, x + \frac{\tau}{2}\kappa_1 + \frac{\tau}{2}\kappa_2)$; then, we define

$$\psi(t+\tau,t,x) \equiv x + \frac{\tau}{2}\left(\kappa_1 + \kappa_2\right). \tag{2.38}$$

Hence, the vector $\kappa_2 \in \mathbb{R}^n$ is defined as the fixed point $\kappa_2 = f(t+\tau, x + \frac{\tau}{2}\kappa_1 + \frac{\tau}{2}\kappa_2)$. The existence and the uniqueness of this fixed point is guaranteed for sufficiently small $\tau \geq 0$. If $f \in C^2(J \times D, \mathbb{R}^n)$ then it can be shown that the Implicit Trapezoidal method is of the order $p = 2$.

Remark 2.18 (Crank-Nicholson, 1947). The Implicit Trapezoidal method (Definition 2.17) is often called the *Crank-Nicholson* method.

We now derive an explicit version of the Implicit Trapezoidal method. We start with (2.36) and consider the right-hand side. We note that the value of $\phi(t+\tau,t,x)$ is not known explicitly; therefore, we have to approximate it using the (explicit) Euler method:

$$\phi(t+\tau,t,x) = u(t) + \tau f(t,u(t)) + \mathcal{O}(\tau^2).$$

Hence,

$$\phi(t+\tau,t,x) - x = \frac{\tau}{2}\left(f(t,x) + f(t+\tau,x+\tau f(t,x))\right) + \mathcal{O}(\tau^3). \tag{2.39}$$

Neglecting the terms of order $\mathcal{O}(\tau^3)$ we get an *explicit* formula for the discrete flow.

Definition 2.19 (Heun method, 1900)**.** Let f be locally Lipschitz continuous on $J \times D$, assume that $(t,x) \in J \times D$, and set $\kappa_1 = f(t,x)$ and $\kappa_2 = f(t+\tau, x+\tau\kappa_1)$; then, we define

$$\psi(t+\tau,t,x) \equiv x + \frac{\tau}{2}\left(\kappa_1 + \kappa_2\right). \tag{2.40}$$

Let $f \in C^2(J \times D, \mathbb{R}^n)$; then, the Heun method is of the order $p = 2$.

At the end we introduce a crucial method.

Definition 2.20 (Runge-Kutta method, 1901)**.** Let f be locally Lipschitz continuous on $J \times D$, assume that $(t, x) \in J \times D$, and set

$$\begin{aligned}
\kappa_1 &= f(t, x), \\
\kappa_2 &= f(t + \frac{\tau}{2}, x + \frac{\tau}{2}\kappa_1), \\
\kappa_3 &= f(t + \frac{\tau}{2}, x + \frac{\tau}{2}\kappa_2), \\
\kappa_4 &= f(t + \tau, x + \tau\,\kappa_3);
\end{aligned}$$

then, we define

$$\psi(t + \tau, t, x) \equiv x + \tau \left(\frac{1}{6}\kappa_1 + \frac{1}{3}\kappa_2 + \frac{1}{3}\kappa_3 + \frac{1}{6}\kappa_4 \right). \tag{2.41}$$

This is an explicit method of order $p = 4$.

In Section 2.4 we introduce the class of one-step methods called the *Runge-Kutta methods (RK)*. This class was inspired by the method in Definition 2.20. In principle, within the class we can derive both *explicit* and *implicit* methods methods of an arbitrary order $p \geq 1$. However, there is a practical restriction as we need to construct Taylor's expansion of the flow of a sufficiently high order, which can be complicated, see Remark 1.23.

The methods which were derived in this section (Euler, Implicit Euler, Runge, Crank-Nicholson, and Heun) belong to the class of Runge-Kutta methods and are of order $p \leq 2$.

2.2 Convergence analysis of one-step methods

We consider the initial value problem (IVP) on the interval $[t_0, T]$, $T < t^+(t_0, x_0)$, and aim to solve the problem numerically by means of a chosen one-step method (2.4). During the coarse of the iterations of the one-step method the local discretisation error, see Definition 2.5, from the previous time steps accumulate. In this section we study the error analysis to estimate the *global error*, which is the accumulated local errors; cf. Theorem 2.23.

Definition 2.21 (Consistency function)**.** Let f be locally Lipschitz continuous on $J \times D$ and ψ be the discrete flow of the vector field f, see Definition 2.1. For each $t \in J, x \in D, \tau \geq 0$ we set

$$\Psi(t, x, \tau) \equiv \frac{\psi(t + \tau, t, x) - x}{\tau} \in \mathbb{R}^n. \tag{2.42}$$

The function Ψ is called the *consistency function* of the discrete flow ψ; i.e., of the relevant one-step method ψ.

There exists a compact subset $K \subset D$ such that

1. $[t_0, T] \times K$ contains the whole trajectory
2. The right-hand side f is Lipschitz continuous on $[t_0, T] \times K$.

Let L be the constant of the Lipschitz continuity of f; then, the domain of the consistency function Ψ contains the compact set $[t_0, T] \times K \times [0, \tau_0]$, provided that $\tau_0 > 0$ is sufficiently small. In this domain, the consistency function Ψ is continuous; i.e.,

$$\Psi \in C([t_0, T] \times K \times [0, \tau_0], \mathbb{R}^n). \tag{2.43}$$

Let *assume* that the consistency function $\Psi = \Psi(t, x, \tau)$ is additionally Lipschitz continuous in the variable x; i.e., there exists $\Lambda \geq 0$ such that

$$\|\Psi(t, x, \tau) - \Psi(t, y, \tau)\| \leq \Lambda \|x - y\| \qquad \forall x, y \in K, t \in [t_0, T], \tau \in [0, \tau_0]. \tag{2.44}$$

We will show later the relationship between the constant Λ and the constant L. Note that the assumption (2.44) is often automatically satisfied.

Definition 2.22. For each partition (2.1) of the interval $[t_0, T]$ we define the norm of the partition:

$$\tau_{\max} = \max_{j=0,\dots,N-1} (t_{j+1} - t_j).$$

Theorem 2.23 (Global error estimate)**.** *Let f be locally Lipschitz continuous on $J \times D$, and denote $u(t) = \phi(t, t_0, x_0)$ the solution of the initial value problem (IVP) on the interval $t \in [t_0, T]$. Consider a one-step method (2.4) of order $p \geq 1$, let ψ be the discrete flow of the vector field f, Ψ be the relevant consistency function, and the condition (2.44) be satisfied. Assume that the local discretisation error can be uniformly estimated: there exists a constants $C > 0$ and a sufficiently small $\tau_1 > 0$ such that*

$$d(t + \tau, t, u(t)) \leq C\tau^{p+1} \qquad \text{for each } \tau \leq \tau_1, t \in [t_0, T]. \tag{2.45}$$

Consider the partition (2.1) of the interval $[t_0, T]$, and let the norm $\tau_{\max}$ of the partition be sufficiently small; then, there exists the approximate solution $\{u_j\}_{j=1}^N$ defined by the recursion

$$u_0 \equiv x_0, \qquad u_{j+1} = \psi(t_{j+1}, t_j, u_j), \qquad j = 0, \dots, N-1. \tag{2.46}$$

Additionally,

$$\|u(t_j) - u_j\| \leq \frac{\mathrm{e}^{\Lambda(t_j - t_0)} - 1}{\Lambda} C\tau_{\max}^p, \qquad j = 0, \dots, N. \tag{2.47}$$

Proof. We need to specify a neighbourhood of the trajectory. Due to the compactness of the set K we can conclude that there exists a constant $\delta_K > 0$ such that

$$\{x \in \mathbb{R}^n : \|x - u(t)\| \leq \delta_K\} \subset K$$

for each $t \in [t_0, T]$. Hence, the system of balls of diameter δ_K, with centre $u(t)$, belong to K. We *assume* that

$$\|u(t_j) - u_j\| \leq \delta_K, \qquad j = 0, \dots, N. \tag{2.48}$$

Therefore, the numerical solution belongs to the specified neighbourhood of the trajectory.

We now analyse the relationship of the exact and approximate solutions at the times t_{j+1} and t_j. We set $\tau_j = t_{j+1} - t_j$; then, by definition, $u(t_{j+1}) = \phi(t_j + \tau_j, t_j, u(t_j))$, and $u_{j+1} = u_j + \tau_j \Psi(t_j, u_j, \tau_j)$; therefore, $u(t_{j+1}) - u_{j+1} = \phi(t_j + \tau_j, t_j, u(t_j)) - u_j - \tau_j \Psi(t_j, u_j, \tau_j)$. By adding and subtracting suitable terms we get that

$$\begin{aligned} u(t_{j+1}) - u_{j+1} = {} & \phi(t_j + \tau_j, t_j, u(t_j)) - u(t_j) - \tau_j \Psi(t_j, u(t_j), \tau_j) \\ & + u(t_j) - u_j + \tau_j \Psi(t_j, u(t_j), \tau_j) - \tau_j \Psi(t_j, u_j, \tau_j)\,. \end{aligned}$$

The triangle inequality yields

$$\begin{aligned} \|u(t_{j+1}) - u_{j+1}\| \leq {} & \|\phi(t_j + \tau_j, t_j, u(t_j)) - u(t_j) - \tau_j \Psi(t_j, u(t_j), \tau_j)\| \\ & + \|u(t_j) - u_j\| + \tau_j \|\Psi(t_j, u(t_j), \tau_j) - \Psi(t_j, u_j, \tau_j)\|\,. \end{aligned}$$

According to the assumption (2.45) we have

$$\|\phi(t_j+\tau_j,t_j,u(t_j))-u(t_j)-\tau_j\Psi(t_j,u(t_j),\tau_j)\| \equiv d(t_i+\tau_j,t_i,u(t_j)) \le K\tau^{p+1},$$

and the assumption (2.44) we estimate

$$\|\Psi(t_j,u(t_j),\tau_j)-\Psi(t_j,u_j,\tau_j)\| \le \Lambda\,\|u(t_j)+u_j\|\,.$$

Therefore,

$$\|u(t_{j+1})-u_{j+1}\| \le (1+\tau_j\Lambda)\,\|u(t_j)+u_j\| + C\tau_j^{p+1}\,, \quad j=0,\dots,N-1. \tag{2.49}$$

Let us denote by $\mathcal{E}_j = \|u(t_j)-u_j\|$ the *global error* at the time t_j, $j=0,\dots,N$.

By induction, we can show that

$$\mathcal{E}_j \le \frac{\mathrm{e}^{\Lambda(t_j-t_0)}-1}{\Lambda} C\tau_{\max}^p, \qquad j=0,\dots,N. \tag{2.50}$$

Base case ($j=0$) . Due to the initial conditions (2.46) and (1.15) $\mathcal{E}_0 = \|u(t_0)-u_0\| = 0$, and we note that from (2.50) that $0 \le \mathcal{E}_0 \le 0$; therefore, the base case is true.

Induction step Assume that the statement (2.50) holds for $j \in \{0,\dots,N-1\}$ and we verify that the statement holds also for $j+1$. From (2.49) we deduce that

$$\begin{aligned}\mathcal{E}_{j+1} &\le (1+\tau_j\Lambda)\mathcal{E}_j + C\tau_j^{p+1}\\ &\le (1+\tau_j\Lambda)\frac{\mathrm{e}^{\Lambda(t_j-t_0)}-1}{\Lambda}C\tau_{\max}^p + C\tau_{\max}^p\tau_j\\ &= \frac{C\tau_{\max}^p}{\Lambda}\left((1+\tau_j\Lambda)\,\mathrm{e}^{\Lambda(t_j-t_0)}-1\right).\end{aligned}$$

Since $1+\tau_j\Lambda \le \mathrm{e}^{\tau_j\Lambda}$,

$$(1+\tau_j\Lambda)\,\mathrm{e}^{\Lambda(t_j-t_0)} \le \mathrm{e}^{\Lambda(t_j+\tau_j-t_0)} = \mathrm{e}^{\Lambda(t_{j+1}-t_0)};$$

therefore, (2.50) also holds for $j+1$.

Let us recall the initial assumption (2.48). We choose $\tau_2 > 0$ such that

$$\frac{C\tau_2^p}{\Lambda}\left(\mathrm{e}^{\Lambda(T-t_0)}-1\right) \le \delta_K, \qquad \tau_2 \le \min(\tau_0,\tau_1).$$

We conclude that for an arbitrary partition (2.1) which satisfies $\tau_{\max} \le \tau_2$, the condition (2.45) holds. Consequently, the numerical solution (2.46) exists and the estimate (2.47) holds. □

We now consider the assumption (2.44) and show that the value of Λ is related to the value of the Lipschitz constant L.

Proposition 2.24. *Let f be locally Lipschitz continuous on $J\times D$, assume that $[t_0,T]\times K \subset J\times D$, where K is a compact set, and let L denote the appropriate constant of the Lipschitz continuity. For the Euler method, Definition 2.3, and the Runge method, Definition 2.10, there exists a sufficiently small $\tau_0 > 0$ such that the consistency function Ψ is Lipschitz continuous in the second variable, cf. (2.44), on $[t_0,T]\times\mathbb{R}^n\times[0,\tau_0]$, with continuity constant*

$$\Lambda \equiv \begin{cases} L & \text{for the Euler method,}\\ L\left(1+\dfrac{\tau_0 L}{2}\right) & \text{for the Runge method.}\end{cases}$$

Proof. In the case of the Runge method, $\Psi(t,x,\tau) \equiv f\left(t+\frac{\tau}{2}, x+\frac{\tau}{2}f(t,x)\right)$; hence,

$$\begin{aligned}\left\|f\left(t+\frac{\tau}{2}, x+\frac{\tau}{2}f(t,x)\right) - f\left(t+\frac{\tau}{2}, y+\frac{\tau}{2}f(t,y)\right)\right\| &\le L\left\|x+\frac{\tau}{2}f(t,x) - y - \frac{\tau}{2}f(t,y)\right\| \\ &\le L\|x-y\| + \frac{\tau L}{2}\|f(t,x)-f(t,y)\| \\ &\le L\left(1+\frac{\tau L}{2}\right)\|x-y\|\end{aligned}$$

for each $x, y \in K$ and each $t \in [t_0, T]$. The restriction on $\tau \le \tau_0$, τ_0 sufficiently small, is required to ensure we remain in the domain D. The proof for Euler method is analogous. □

Proposition 2.25. *Assume that $f \in C([t_0,T]\times\mathbb{R}^n, \mathbb{R}^n)$ is Lipschitz continuous in the second variable on $[t_0,T]\times\mathbb{R}^n$ with constant L. For the Implicit Euler, Definition 2.13, and Implicit Trapezoidal method, Definition 2.17, there exists sufficiently small $\tau_0 > 0$ such that the consistency function Ψ is Lipschitz continuous in the second variable, cf. (2.44), on $[t_0,T]\times\mathbb{R}^n\times[0,\tau_0]$, with continuity constant*

$$\Lambda \equiv \begin{cases} L\left(1-\tau_0 L\right)^{-1} & \text{for the Implicit Euler method,} \\ L\left(1+\frac{\tau_0 L}{2}\right)\left(1-\frac{\tau_0 L}{2}\right)^{-1} & \text{for the Implicit Trapezoidal method.}\end{cases}$$

Proof. In the case of the Implicit Euler method, $\Psi(t,x,\tau) \equiv \kappa_1$, where $\kappa_1 = f(t+\tau, x+\tau\kappa_1)$. The fixed point κ_1 exists for $\tau < 1/L$, see (2.34). Note that κ_1 is a function of t, x and τ; i.e., $\kappa_1 = \kappa_1(t,x,\tau)$.

For each $t \in [t_0,T]$, each pair $x, y \in \mathbb{R}^n$, and each $\tau < 1/L$,

$$\begin{aligned}\|\Psi(t,x,\tau) - \Psi(t,y,\tau)\| &\equiv \|\kappa_1(t,x,\tau) - \kappa_1(t,y,\tau)\| \\ &= \|f(t+\tau, x+\tau\kappa_1(t,x,\tau)) - f(t+\tau, y+\tau\kappa_1(t,y,\tau))\| \\ &\le L\|x-y\| + \tau L\|\kappa_1(t,x,\tau) - \kappa_1(t,y,\tau)\|.\end{aligned}$$

Therefore, $\|\kappa_1(t,x,\tau) - \kappa_1(t,y,\tau)\| \le L(1-\tau L)^{-1}\|x-y\|$. We can choose $\tau_0 = 1/2L$. In the case of Implicit Trapezoidal method we can proceed analogously. □

2.3 Adaptive time-stepping

Recall the definition of the local discretisation error (2.10) and the order of the method (2.11).

Consider the Runge method, Definition 2.10; then, according to Corollary 2.11 it holds that if $f \in C^2(J\times D, \mathbb{R}^n)$ the local discretisation error is of the order two; i.e.,

$$\|\phi(t+\tau,t,x) - \psi(t+\tau,t,x)\| = d(t+\tau,t,x) = \mathcal{O}(\tau^3).$$

If f is sufficiently smooth then there exists a *Taylor expansion* of the local discretisation error: If $f \in C^3(J\times D, \mathbb{R}^n)$ then

$$d(t+\tau,t,x) = K_0\tau^3 + \mathcal{O}(\tau^4),$$

where K_0 is a positive constant. If $f \in C^k(J\times D, \mathbb{R}^n)$, $k \ge 3$, then

$$d(t+\tau,t,x) = K_0\tau^3 + \cdots + K_{k-3}\tau^k + \mathcal{O}(\tau^{k+1}),$$

where $K_0, \ldots, K_{k-3}$ are constants. The constant K_0 is called the *leading term* of the Taylor expansion of the local discretisation error.

We consider two one-step methods, cf. Definition 2.4, which we will call, respectively, the "low order" and of "high order" method:

low order: a one-step method of order p

$$t \in J, x \in D, \tau \geq 0 \quad \longmapsto \quad \psi(t+\tau, t, x) \in \mathbb{R}^n, \tag{2.51}$$

high order: a one-step method of order $p+1$

$$t \in J, x \in D, \tau \geq 0 \quad \longmapsto \quad \bar{\psi}(t+\tau, t, x) \in \mathbb{R}^n. \tag{2.52}$$

We first consider the low order method. If $f \in C^p(J \times D, \mathbb{R}^n)$

$$\|\phi(t+\tau, t, x) - \psi(t+\tau, t, x)\| = \mathcal{O}(\tau^{p+1}).$$

Additionally, if $f \in C^{p+1}(J \times D, \mathbb{R}^n)$ then

$$\|\phi(t+\tau, t, x) - \psi(t+\tau, t, x)\| = K_0 \tau^{p+1} + \mathcal{O}(\tau^{p+2}), \tag{2.53}$$

where K_0 is the leading term of the Taylor expansion of the local discretisation error. Similarly, for the high order method, if $f \in C^{p+1}(J \times D, \mathbb{R}^n)$

$$\left\|\phi(t+\tau, t, x) - \bar{\psi}(t+\tau, t, x)\right\| = \mathcal{O}(\tau^{p+2}). \tag{2.54}$$

Defining

$$\Delta(\tau) = \bar{\psi}(t+\tau, t, x) - \psi(t+\tau, t, x), \tag{2.55}$$

we note that from (2.53) and (2.54) that

$$\|\Delta(\tau)\| = K_0 \tau^{p+1} + \mathcal{O}(\tau^{p+2}). \tag{2.56}$$

It is important to note that $\Delta(\tau) \in \mathbb{R}^n$ is a *computable* quantity. Dropping the terms of the order $\mathcal{O}(\tau^{p+2})$ in (2.54) and (2.57), we claim that

$$\|\phi(t+\tau, t, x) - \psi(t+\tau, t, x)\| = \|\Delta(\tau)\| = K_0 \tau^{p+1}. \tag{2.57}$$

Using this result we can try to compute an *optimal* time step τ, which we denote by τ_{opt}. Given a desired *tolerance* (error) `tol`, we require that

$$\|\phi(t+\tau_{\text{opt}}, t, x) - \psi(t+\tau_{\text{opt}}, t, x)\| = \|\Delta(\tau_{\text{opt}})\| = K_0 \tau_{\text{opt}}^{p+1} = \texttt{tol}. \tag{2.58}$$

Eliminating the constant K_0 we can obtain a formula for the optimal time step

$$\tau_{\text{opt}} = \tau \left(\frac{\texttt{tol}}{\|\Delta(\tau)\|} \right)^{\frac{1}{p+1}}. \tag{2.59}$$

Algorithm 2.1 (Adaptive step size)**.** Consider a low order (2.51) and high order (2.52) one-step method; then, we can define a one-step method with adaptive step size for $(t, x) \in J \times D$, with time step $\tau > 0$ from the previous step, as follows:

$\tau \leftarrow \max(\tau, \texttt{tol})$
$\delta \leftarrow \left\|\bar{\psi}(t+\tau, t, x) - \psi(t+\tau, t, x)\right\|$
while $\delta > \texttt{tol}$ **do**
  $\tau \leftarrow \tau \left(\frac{\texttt{tol}}{\delta}\right)^{1/(p+1)}$
  $\delta \leftarrow \left\|\bar{\psi}(t+\tau, t, x) - \psi(t+\tau, t, x)\right\|$
end while
Accept τ ▷ it now holds that $\left\|\bar{\psi}(t+\tau, t, x) - \psi(t+\tau, t, x)\right\| \leq \texttt{tol}$
$t \leftarrow t + \tau$
$x \leftarrow \bar{\psi}(t+\tau, t, x)$

Remark 2.26. MATLAB (Shampine and Reichelt, 1997) includes solvers **ode23** and **ode45** for the numerical solution of (IVP). Both solvers can be classified as one-step methods with an adaptive step size based on the principles formulated in Algorithm 2.1:

ode23 an explicit method (2.51) of order $p = 2$ and an explicit method (2.52) of order $p+1 = 3$ are used

ode45 an explicit method (2.51) of order $p = 4$ and an explicit method (2.52) of order $p+1 = 5$ are used

There are some rules which allow the optimisation of the coupling of the "low order" and "high order" methods. In Section 2.4.1 we will talk about the so called *embedded formulas*.

Remark 2.27. Algorithm 2.1 depends on a choice of one single parameter `tol`. In the more recent versions of MATLAB, the adaptivity depends on a choice of $n+1$ parameters

$$\texttt{AbsTol} \in \mathbb{R}^n, \qquad \text{and} \qquad \texttt{RelTol} \in \mathbb{R}, \tag{2.60}$$

which are the absolute and the relative tolerances, respectively.

We define $\delta \in \mathbb{R}^n$, with

$$\delta_i = \texttt{AbsTol}_i + \max\left(|x_i|, |\bar{\psi}_i(t+\tau, t, x)|\right) \texttt{RelTol}, \qquad i = 1, \ldots, n,$$

and the error function

$$\texttt{err} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\bar{\psi}_i(t+\tau, t, x) - \psi_i(t+\tau, t, x)}{\delta_i}\right)};$$

then

$$\tau_{\text{opt}} = \tau\left(\frac{1}{\texttt{err}}\right)^{\frac{1}{(p+1)}}. \tag{2.61}$$

By default MATLAB uses

$$\texttt{RelTol} = 10^{-3}, \qquad \texttt{AbsTol}_i = 10^{-6}, \quad i = 1, \ldots, n.$$

2.4 Runge-Kutta methods (RK)

We will now define a class of one-step methods inspired by the Runge-Kutta method, see Definition 2.20 in Section 2.1, called the *Runge-Kutta methods*, often abbreviated as the *RK methods*.

RK methods are defined by a set of constants (data)

$$A = (a_{ij})_{i,j=1}^{s} \in \mathbb{R}^{s\times s}, \qquad b \in \mathbb{R}^{s}, \qquad c \in \mathbb{R}^{s}, \tag{2.62}$$

where the positive integer s is called the *stage* of the method.

Definition 2.28 (Butcher, 1972)**.** Let f be locally Lipschitz continuous on $J \times D$, assume that $(t, x) \in J \times D$, $\tau \geq 0$; then, for the data (2.62) we consider $\kappa_i \in \mathbb{R}^n$, $i = 1, \ldots, s$, to be the solutions of s nonlinear equations

$$\kappa_i = f\left(t + \tau c_i, x + \tau \sum_{j=1}^{s} a_{ij}\kappa_j\right), \qquad i = 1, \ldots, s, \tag{2.63}$$

and we define the discrete flow by

$$\psi(t + \tau, t, x) \equiv x + \tau \sum_{i=1}^{s} b_i \kappa_i \in \mathbb{R}^n. \tag{2.64}$$

Definition 2.29 (Butcher Tableau)**.** The data of the RK methods can be presented in the form of the *Butcher array* (or *Butcher tableau*):

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \ldots & a_{1s} \\ c_2 & a_{21} & a_{22} & \ldots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \ldots & a_{ss} \\ \hline & b_1 & b_2 & \ldots & b_s \end{array} \tag{2.65}$$

We can interpret this tableau as follows:

1. From the i-th row of the Butcher tableau we can construct the i-th equation of the system (2.63), $i = 1, \ldots, s$.

2. The formula for the discrete flow (2.64) is defined by means of the coefficients b_i, $i = 1, \ldots, s$ in the last row of the Butcher tableau.

Consider the matrix A. If the entries $a_{ij} = 0$ for $i \leq j$ then A is a strictly lower triangular matrix; therefore, the underlying RK method is *explicit*, which means that the vectors $\kappa_i \in \mathbb{R}^n$ in (2.64) are defined by *linear* canonical formulas (2.63). Otherwise, we say that the underlying RK method is *implicit*.

All the methods in Section 2.1 were RK methods.

Example 2.2 (Butcher tableaux). The Butcher tableaux of the Euler method (Definition 2.3), the Runge method (Definition 2.10), the Implicit Euler method (Definition 2.13), the Implicit Trapezoidal method or Crank-Nicholson method (Definition 2.17), and the Heun method (Definition 2.19) are given by

0	0
	1

Euler

0	0	0
1/2	1/2	0
	0	1

Runge

1	1
	1

Impl. Euler

0	0	0
1	1/2	1/2
	1/2	1/2

Crank-Nicholson

0	0	0
1	1	0
	1/2	1/2

Heun

For an explicit RK method we may, as a shortcut, skip the entries of A which are zero by definition.

Example 2.3 (Classical Runge-Kutta). We can write the *classical Runge-Kutta* method, cf. Definition 2.20, by the Butcher tableau

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

Given a stage s and a particular Butcher's tableau (2.65) we want to find the order p of the underlying RK method. Consider $\psi(t+\tau,t,x)$ defined by (2.63)–(2.64); then, we need to develop the Taylor expansion of $\psi(t+\tau,t,x)$ at the point $\tau=0$. We proceed analogously as for the expansion of the flow ϕ in Section 1.4, cf. (1.34)–(1.35). We get that

$$\psi(t+\tau,t,x)=\sum_{j=0}^{k}\frac{\tau^j}{j!}\,\frac{\partial^j}{\partial\tau^j}\psi(t+\tau,t,x)\Big|_{\tau=0}+\mathcal{O}(\tau^{k+1}). \tag{2.66}$$

Note that $\psi(t+\tau,t,x)|_{\tau=0}=x$. The expansion is given by linear combinations of the elementary differentials, see Remark 1.23.

As an important special case we consider the expansion (2.66) for the autonomous ODE: Let $f(t,x)\equiv f(x)$, recall (1.17), and define

$$\psi(\tau,x_0)\equiv\psi(\tau,0,x_0). \tag{2.67}$$

As an exercise, we compute all terms of the expansion (2.66) up to the order 3.

Lemma 2.30. *Let $f(t,x)\equiv f(x)$, assume $f\in C^3(D,\mathbb{R}^n)$, and consider the RK method defined by the Butcher tableau (2.29). Choose $x\in D$, $0\le\tau\le\tau_0$, where τ_0 is sufficiently small; then*

$$\psi(\tau,x)=x+\tau\sum_{i=1}^{s}b_i f+\tau^2\sum_{i=1}^{s}b_i\sum_{j=1}^{s}a_{ij}f^{(1)}[f]+\frac{\tau^3}{2}\sum_{i=1}^{s}b_i\left(\sum_{j=1}^{s}a_{ij}\right)\left(\sum_{k=1}^{s}a_{ik}\right)f^{(2)}[f,f]$$
$$+\tau^3\sum_{i=1}^{s}b_i\sum_{j=1}^{s}a_{ij}\sum_{k=1}^{s}a_{jk}f^{(1)}[f^{(1)}[f]]+\mathcal{O}(\tau^4). \tag{2.68}$$

Proof. By definition,

$$\psi(\tau, x) = x + \tau \sum_{i=1}^{s} b_i \kappa_i, \tag{2.69}$$

and since the ODE is autonomous, $\kappa_i = f(x + \tau \sum_{j=1}^{s} a_{ij}\kappa_j)$; hence, we have defined (explicitly or implicitly) a vector function $\kappa_i = \kappa_i(\tau, x)$ of argument τ.

We seek for the expansion (2.66); i.e.,

$$\psi(\tau, x) = x + \tau \frac{\partial}{\partial \tau}\psi(0, x) + \frac{\tau^2}{2}\frac{\partial^2}{\partial \tau^2}\psi(0, x) + \frac{\tau^3}{6}\frac{\partial^3}{\partial \tau^3}\psi(0, x) + \mathcal{O}(\tau^4).$$

By definition (2.69),

$$\begin{aligned}
\frac{\partial}{\partial \tau}\psi(\tau, x) &= \sum_{i=1}^{s} b_i \kappa_i + \tau \sum_{i=1}^{s} b_i \frac{\partial \kappa_i}{\partial \tau}, \\
\frac{\partial^2}{\partial \tau^2}\psi(\tau, x) &= 2\sum_{i=1}^{s} b_i \frac{\partial \kappa_i}{\partial \tau} + \tau \sum_{i=1}^{s} b_i \frac{\partial^2 \kappa_i}{\partial \tau^2}, \\
\frac{\partial^3}{\partial \tau^3}\psi(\tau, x) &= 3\sum_{i=1}^{s} b_i \frac{\partial^2 \kappa_i}{\partial \tau^2} + \tau \sum_{i=1}^{s} b_i \frac{\partial^3 \kappa_i}{\partial \tau^3}.
\end{aligned}$$

Computing the partial derivatives of $\kappa_i = \kappa_i(\tau, x)$ with respect to τ via the chain rule, evaluating at $\tau = 0$, and assuming that $f \in C^3(D, \mathbb{R}^n)$ completes the proof. □

We can give a sufficient condition for a RK method to be of order $p = 3$; initially assuming that the ODE is autonomous.

Lemma 2.31 (Autonomous ODE: RK method of order $p = 3$)**.** *Let $f(t, x) \equiv f(x)$, $f \in C^3(D, \mathbb{R}^n)$, and consider the RK method defined by the Butcher tableau (2.65); then, if*

$$\sum_{i=1}^{s} b_i = 1, \qquad 2\sum_{i,j=1}^{s} b_i a_{ij} = 1, \qquad 3\sum_{i,j,k=1}^{s} b_i a_{ij} a_{ik} = 1, \qquad 6\sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} = 1, \tag{2.70}$$

the RK method is of order $p = 3$ at each point $x \in D$.

Proof. For a given $x \in D$, we estimate the local discretisation error (2.13), and consider the Taylor expansion of the vector field (1.38) and the discrete flow (2.68). Both expansions are linear combinations of four elementary differentials f, $f^{(1)}[f]$, $f^{(2)}[f, f]$, $f^{(1)}[f^{(1)}[f]]$. Comparing the coefficients of the same elementary differentials we deduce that $d(\tau, x) = \mathcal{O}(\tau^4)$. □

Let f be local Lipschitz continuous on $J \times D$. The initial value problem (1.14)–(1.15) can be formulated as the initial value problem (1.30) for autonomous ODE. One step of the RK method applied on the vector field f can be interpreted as one step of the same method (with the same Butcher table) applied on the vector field (1.29). In the notation (2.67),

$$\Psi(\tau, z) \equiv z + \tau \sum_{i=1}^{s} b_i K_i, \qquad K_i = F\left(z + \tau \sum_{j=1}^{s} a_{ij} K_j\right), \qquad i = 1, \dots, s. \tag{2.71}$$

By definition of F,

$$K_i = \begin{bmatrix} 1 \\ f(z + \tau \sum_{j=1}^{s} a_{ij} K_j) \end{bmatrix} = \begin{bmatrix} 1 \\ \kappa_i \end{bmatrix}, \qquad \kappa_i = f\left(t + \tau \sum_{j=1}^{s} a_{ij}, x + \tau \sum_{j=1}^{s} a_{ij} \kappa_j \right).$$

Comparing the resulting κ_i with the formula (2.63) it is clear that one step of the method applied on the field f and one step of the method applied on the autonomous field F will be the same provided that the following condition holds.

Lemma 2.32 (Invariance with respect to "autonomisation"). *The RK method defined by the Butcher tableau* (2.65) *is invariant with respect to the* autonomisation *if and only if*

$$c_i = \sum_{j=1}^{s} a_{ij}, \qquad i = 1, \dots, s. \tag{2.72}$$

Corollary 2.33 (RK method of order $p = 3$). *Assume that $f \in C^3(J \times D, \mathbb{R}^n)$, consider the RK method defined by the Butcher tableau* (2.65) *and let the condition* (2.72) *be satisfied (the autonomisation); then, if*

$$\sum_{i=1}^{s} b_i = 1, \qquad 2\sum_{i=1}^{s} b_i c_i = 1, \qquad 3\sum_{i=1}^{s} b_i c_i^2 = 1, \qquad 6\sum_{i,j=1}^{s} b_i a_{ij} c_j = 1, \tag{2.73}$$

the RK method is of order $p = 3$ at each point $(t, x) \in J \times D$.

Proof. We have shown that RK method can be formulated as (2.71) for the autonomous vector field F provided that (2.72) holds. According to Lemma 2.31, the method is of the order 3, provided that the conditions (2.70) hold; therefore, it is sufficient to check that the conditions (2.70) and (2.73) are equivalent. Due to (2.72),

$$\begin{aligned}
\sum_{i,j=1}^{s} b_i a_{ij} &= \sum_{i=1}^{s} b_i \sum_{j=1}^{s} a_{ij} = \sum_{i=1}^{s} b_i c_i, \\
\sum_{i,j,k=1}^{s} b_i a_{ij} a_{ik} &= \sum_{i=1}^{s} b_i \sum_{j=1}^{s} a_{ij} \sum_{k=1}^{s} a_{ik} = \sum_{i=1}^{s} b_i c_i^2, \\
\sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} &= \sum_{i=1}^{s} b_i \sum_{j=1}^{s} a_{ij} \sum_{k=1}^{s} a_{jk} = \sum_{i=1}^{s} b_i \sum_{j=1}^{s} a_{ij} c_j;
\end{aligned}$$

hence, (2.70) and (2.73) are equivalent. □

We can naturally alter this result for different regularity assumptions on the right-hand side f.

Corollary 2.34 (RK method of order $p = 1$). *Assume that $f \in C^1(J \times D, \mathbb{R}^n)$, consider the RK method defined by the Butcher tableau* (2.65) *and let the condition* (2.72) *be satisfied; then, if*

$$\sum_{i=1}^{s} b_i = 1, \tag{2.74}$$

the RK method is of order $p = 1$ at each point $(t, x) \in J \times D$.

Corollary 2.35 (RK method of order $p = 2$)**.** *Assume that $f \in C^2(J \times D, \mathbb{R}^n)$, consider the RK method defined by the Butcher tableau (2.65) and let the condition (2.72) be satisfied; then, if*

$$\sum_{i=1}^{s} b_i = 1, \qquad 2\sum_{i=1}^{s} b_i c_i = 1, \tag{2.75}$$

the RK method is of order $p = 2$ at each point $(t, x) \in J \times D$.

Corollary 2.36 (RK method of order $p = 4$)**.** *Assume that $f \in C^4(J \times D, \mathbb{R}^n)$, consider the RK method defined by the Butcher tableau (2.65) and let the condition (2.72) be satisfied; then, if*

$$4\sum_{i=1}^{s} b_i c_i^3 = 1, \quad 8\sum_{i,j=1}^{s} b_i a_{ij} c_i c_j = 1, \quad 12\sum_{i,j=1}^{s} b_i a_{ij} c_j^2 = 1, \quad 24\sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} c_k = 1, \tag{2.76}$$

the RK method is of order $p = 4$ at each point $(t, x) \in J \times D$.

2.4.1 Explicit RK methods

We first analyse explicit methods of stage $s \leq 4$, with the aim to be to find the maximal order of the method. It will be shown that for the methods of stage $s \leq 4$ that the maximal order p is equal to the stage s; i.e. $p = s$ for $s \leq 4$ (see Deuflhard and Bornemann, 2012, Theorem 4.24).

Corollaries 2.35, 2.33, and 2.36, yield sufficient conditions for the method to be of order at least $p = 2, 3, 4$, respectively. These conditions represent *nonlinear* constraints on various constants from the Butcher tableau. In the following analysis we deduce *only some* of the possible solutions.

c_1		
c_2	a_{21}	
	b_1	b_2

Explicit RK ($s = 2$)

c_1			
c_2	a_{21}		
c_2	a_{31}	a_{32}	
	b_1	b_2	b_3

Explicit RK ($s = 3$)

c_1				
c_2	a_{21}			
c_2	a_{31}	a_{32}		
c_2	a_{41}	a_{42}	a_{43}	
	b_1	b_2	b_3	b_4

Explicit RK ($s = 4$)

Explicit RK methods ($s = 2$)

We have five unknowns a_{21}, b_1, b_2, c_1, and c_2 to compute, and from Corollary 2.35 we can formulate the following system of four nonlinear equations:

$$\begin{aligned} b_1 + b_2 &= 1 \\ b_1 c_1 + b_2 c_2 &= \frac{1}{2} \\ c_1 &= 0 \\ c_2 &= a_{21} \end{aligned}$$

Hence, we have a system of 5 unknowns with 4 conditions. Let $c_2 \neq 0$; then,

$$c_1 = 0, \qquad b_2 = \frac{1}{2c_2}, \qquad b_1 = 1 - b_2 = 1 - \frac{1}{2c_2}, \qquad a_{21} = c_2. \tag{2.77}$$

With this choice of parameters we get that $p = 2$.

Example 2.4 (Explicit RK methods ($s = 2$)). By setting $c_2 = 1/2$ and $c_2 = 1$ in (2.77) we get the Runge and Heun method, respectively:

0		
1/2	1/2	
	0	1

Runge

0		
1	1	
	1/2	1/2

Heun

Explicit RK methods ($s = 3$)

The Butcher tableau for the explicit RK method with $s = 3$ consists of nine unknown coefficients a_{21}, a_{31}, a_{32}, b_1, b_2, b_3, c_1, c_2, and c_3. From Corollary 2.33 we get:

$$b_1 + b_2 + b_3 = 1 \tag{2.78a}$$

$$b_1 c_1 + b_2 c_2 + b_3 c_3 = \frac{1}{2} \tag{2.78b}$$

$$b_1 c_1^2 + b_2 c_2^2 + b_3 c_3^2 = \frac{1}{3} \tag{2.78c}$$

$$\sum_{i,j=1}^{3} b_i a_{ij} c_j = \sum_{k=1}^{3} c_k \sum_{j=1}^{3} b_j a_{jk} = c_2 b_3 a_{32} = \frac{1}{6} \tag{2.78d}$$

$$c_1 = 0 \tag{2.78e}$$

$$c_2 = a_{21} \tag{2.78f}$$

$$c_3 = a_{31} + a_{32} \tag{2.78g}$$

Hence, we have a system of 9 unknowns with 7 conditions. We can define all the parameters by selecting values for the parameters $c_2 \neq 0$ and $c_3 \neq 0$, such that $c_2 \neq c_3$:

- $c_1 = 0$ by (2.78e).
- We can compute b_2 and b_3 as the solution of the system

  $$\begin{bmatrix} c_2 & c_3 \\ c_2^2 & c_3^2 \end{bmatrix} \begin{bmatrix} b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 1/3 \end{bmatrix},$$

  where the determinant is non-zero by the above requirement on the selection of c_2 and c_3; hence,

  $$b_2 = \frac{2 - 3c_3}{6c_2(c_2 - c_3)}, \qquad b_3 = \frac{2 - 3c_2}{6c_3(c_3 - c_2)}.$$

- Then from (2.78a), (2.78d),(2.78g), and (2.78f) we can simply calculate

  $$b_1 = 1 - b_2 - b_3, \qquad a_{32} = \frac{1}{6b_3c_2}, \qquad a_{31} = c_3 - \frac{1}{6b_3c_2}, \qquad a_{21} = c_2.$$

Referring to Remark 2.27 we shall seek for explicit methods of stage $s = 2$ with order $p = 2$ and explicit methods of stage $s = 3$ with order $p = 3$, aiming to optimise the "low order" and "high order" method. The resulting method will be called of ode23-type.

We are able to construct all explicit methods of stage $s = 2$, order $p = 2$ and all explicit methods of stage $s = 3$, order $p = 3$. We need to consider the *expense* of one step of the ode23-type method. If $s = 2$ then we need to evaluate the right-hand side twice; i.e., evaluate the vector κ_i, $i = 1, 2$, cf. (2.63). Similarly, if $s = 3$ we need to evaluate the right-hand side three times; i.e., evaluate the vector κ_i, $i = 1, 2, 3$, cf. (2.63). Therefore, the expense of one step of the ode23-type method is the five evaluations of the right-hand side. However, it is possible to reduce this expense by selecting κ_i, $i = 1, 2$, related to the "low order" $s = 2$ method and the κ_i, $i = 1, 2$, related to "high order" $s = 3$ method the same; then, there are only three evaluations of the right-hand side.

Example 2.5 (RK3(2) with Heun). Consider the Heun method, shown by the Butcher tableau on the left below. Due to parameter analysis of explicit RK methods of stage $s = 3$, see (2.78), we can construct the following explicit method (in the middle), of order $p = 3$:

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

Heun ($s = 2$) $\qquad$ $s = 3$ $\qquad$ Compact Form

In particular, we choose $c_1 = 0$, $c_2 = 1$ and $c_3 = 1/2$. The key result is that the definition of κ_i, $i = 1, 2$ is carried over from the Heun method, requiring that these are only computed once and then used for both methods when doing adaptive time-stepping, cf. Section 2.3. Additionally, we can use a compact form of the Butcher tableau to denote one step of the adaptive time-stepping method, see above on the right. We say that the "high order" method is *embedded* into the "low order" method.

Example 2.6 (RK3(2) with Runge). As another example of an ode23-type method we can introduce the following embedded formula, using Runge for the "low order" method:

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

Runge ($s = 2$) $\qquad$ $s = 3$ $\qquad$ Compact Form

Definition 2.37 (Embedded RK$p(p-1)$ methods)**.** We can define ode23-type methods via embedded formulas. Instead of ode23-type method we call these methods an *embedded RK3(2) method* or just an *RK3(2) method*. This notation reflects the fact that

- we combine two explicit Runge-Kutta methods in the spirit of Algorithm 2.1,
- RK3(2) is defined via an embedded formula,
- the expenses of RK3(2) are essentially the same as for one step of the RK method of order $p = 3$.

In general, we consider RK$p(p-1)$ methods, where p is a "high order" method and $p-1$ is a "low order" method. There is a whole class of *Embedded Runge-Kutta methods*, see Deuflhard and Bornemann (2012).

Example 2.7 (RK2(1)). In the same spirit we can define embedded RK2(1) methods:

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

Explicit RK methods ($s = 4$)

The Butcher tableau for the explicit RK method with $s = 3$ consists of fourteen unknown coefficients a_{21}, a_{31}, a_{32}, a_{41}, a_{42}, a_{43}, b_1, b_2, b_3, b_4, c_1, c_2, c_3, and c_4. From Definition 2.36 we a system of twelve nonlinear equations, which since $c_1 = 0$, reduces to

$$b_1 + b_2 + b_3 + b_4 = 1 \tag{2.79a}$$

$$b_2c_2 + b_3c_3 + b_4c_4 = \frac{1}{2} \tag{2.79b}$$

$$b_2c_2^2 + b_3c_3^2 + b_4c_4^2 = \frac{1}{3} \tag{2.79c}$$

$$\sum_{i,j=1}^{4} b_i a_{ij} c_j = b_3a_{32}c_2 + b_4(a_{42}c_2 + a_{43}c_3) = \frac{1}{6} \tag{2.79d}$$

$$b_2c_2^3 + b_3c_3^3 + b_4c_4^3 = \frac{1}{4} \tag{2.79e}$$

$$\sum_{i,j=1}^{4} b_i a_{ij} c_i c_j = b_3c_3a_{32}c_2 + b_4c_4(a_{42}c_2 + a_{43}c_3) = \frac{1}{8} \tag{2.79f}$$

$$\sum_{i,j=1}^{4} b_i a_{ij} c_j^2 = b_3a_{32}c_2^2 + b_4(a_{42}c_2^2 + a_{43}c_3^2) = \frac{1}{12} \tag{2.79g}$$

$$\sum_{i,j,k=1}^{4} b_i a_{ij} a_{jk} c_k = b_4a_{43}a_{32}c_2 = \frac{1}{24} \tag{2.79h}$$

$$c_2 = a_{21} \tag{2.79i}$$
$$c_3 = a_{31} + a_{32} \tag{2.79j}$$
$$c_4 = a_{41} + a_{42} + a_{43} \tag{2.79k}$$

Hence, we have a system of 13 unknowns with 11 conditions. Consider the right-hand sides of the equations (2.79a)–(2.79c) and (2.79e); these can be interpreted as integrals $\int_0^1 t^k \,\mathrm{d}t$, $k = 0, 1, 2, 3$:

$$\sum_{i=1}^{4} b_i c_i^k = \int_0^1 t^k \,\mathrm{d}t, \quad k = 0, 1, 2, 3. \tag{2.80}$$

Recall the *Lagrange quadrature formulae*, Remark 2.1, with coefficients $b = (b_1, b_2, b_3, b_4)^\top$ and nodes $c = (c_1, c_2, c_3, c_4)^\top$; i.e., the quadrature

$$\sum_{i=1}^{4} b_i g(c_i) \approx \int_0^1 g(t) \,\mathrm{d}t \tag{2.81}$$

of a sufficiently smooth function $g = g(t)$. According to quadrature theory, see (Quarteroni et al., 2010, p. 372), the quadrature is of order 4 if and only if it is exact for the third order polynomials; i.e. for functions $g(t) = \operatorname{span}\{1, t, t^2, t^3\}$. Therefore, (2.80) is satisfied. In Remark 2.1 we listed two quadratures of order 4: The Simpson rule (2.20) and 3/8-rule (2.21).

Lemma 2.38 (Simpson rule)**.** *If*

$$c = \left(0, \frac{1}{2}, \frac{1}{2}, 1\right)^\top, \qquad b = \left(\frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{1}{6}\right)^\top; \tag{2.82}$$

then, the equations (2.79a)–(2.79c) *and* (2.79e) *of the system* (2.79) *are satisfied.*

Lemma 2.39 ($\frac{3}{8}$-rule)**.** *If*

$$c = \left(0, \frac{1}{3}, \frac{2}{3}, 1\right)^\top, \qquad b = \left(\frac{1}{8}, \frac{3}{8}, \frac{3}{8}, \frac{1}{8}\right)^\top; \tag{2.83}$$

then, the equations (2.79a)–(2.79c) *and* (2.79e) *of the system* (2.79) *are satisfied.*

We can then compute all the coefficients from (2.79) as follows:

- Fix the constants according to either (2.82) or (2.83).
- From equations (2.79d) and (2.79f) we can compute a_{32} and the linear combination $a_{42}c_2 + a_{43}c_4$ as a solution the linear system

$$\begin{bmatrix} b_4 c_2 & b_4 \\ b_4 c_3 c_2 & b_4 c_4 \end{bmatrix} \begin{bmatrix} a_{32} \\ a_{42}c_2 + a_{43}c_4 \end{bmatrix} = \begin{bmatrix} 1/6 \\ 1/8 \end{bmatrix}. \tag{2.84}$$

  This gives us a_{32}.
- From equation (2.79h) we can then compute a_{43}

- Then, we can compute the unknown a_{42} from the second solution component $a_{42}c_2 + a_{43}c_4$ of (2.84).
- The unknowns a_{21}, a_{31}, a_{41} can be computed from (2.79i)–(2.79k).

We note that we have computed all the coefficients without using (2.79g); therefore, we have to check that (2.79g) is linearly dependent. It can be shown that the above solutions a_{ij}, $1 \leq j < i < 4$, satisfy (2.79g).

Example 2.8 (Explicit RK methods ($s = 4$)). By selecting the coefficients b_i and c_i, $i = 1, \ldots, 4$, via either the Simpson rule (2.82) or the $3/8$-rule (2.83) we derive, respectively, the following Butcher tableaux:

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

Classical RK

0				
1/3	1/3			
2/3	−1/3	1		
1	1	−1	1	
	1/8	3/8	3/8	1/8

3/8-rule

If we want to construct an explicit method of order $p \geq 5$ then we face issues. These are related to the fact that an approximation of the vector field $\phi(t + \tau, t, x)$ by means of Taylor expansions "explode", see Remark 1.23.

Theorem 2.40 (Butcher barrier)**.** *Consider an explicit method of stage s and order p; then,*

- *for $p \geq 5$ it is necessary for $s \geq p + 1$,*
- *for $p \geq 7$ it is necessary for $s \geq p + 2$,*
- *for $p \geq 8$ it is necessary for $s \geq p + 3$,*

etc.

Proof. see Hairer (1978). □

Example 2.9 (Butcher method (1963)). One of the first methods which reached the barrier is the following method, which is of stage $s = 6$ and order $p = 5$:

0						
1/4	1/4					
1/4	1/8	1/8				
1/2	0	−1/2	1			
3/4	3/16	0	0	9/16		
1	−3/7	2/7	12/7	−12/7	8/7	
	7/90	0	32/90	12/90	32/90	7/90

(2.85)

The existence of Butcher barriers results in an obvious complication in the construction of embedded methods.

Example 2.10 (RK5(4) — Dormand-Prince (1980)). The Dormand-Prince (1980) method, also called DOPRI5, is an embedded formula of RK5(4), see Definition 2.37, of stage $s = 7$.

$$\begin{array}{c|ccccccc}
0 & 0 & & & & & & \\
1/5 & 1/5 & & & & & & \\
3/10 & 3/40 & 9/40 & & & & & \\
4/5 & 44/45 & -56/15 & 32/9 & & & & \\
8/9 & 19372/6561 & -25360/2187 & 64448/6561 & -212/729 & & & \\
1 & 9017/3168 & -355/33 & 46732/5247 & 49/176 & -5103/18656 & & \\
1 & 35/384 & 0 & 500/1113 & 125/192 & -2187/6784 & 11/84 & \\
\hline
 & 35/384 & 0 & 500/1113 & 125/192 & -2187/6784 & 11/84 & 0 \\
 & 5179/57600 & 0 & 7571/16695 & 393/640 & -92097/339200 & 187/2100 & 1/40
\end{array} \tag{2.86}$$

Note that the MATLAB function `ode45` is related to the formula (2.86).

2.4.2 Implicit RK methods

So far, we have considered only two of implicit RK methods, see Example 2.2, namely the Implicit Euler and Crank-Nicholson methods. We have shown that defining one step of such a method is equivalent to finding the fixed point of a mapping. In case of the implicit Euler method we have shown that this mapping is a *contraction* provided that the step size $\tau > 0$ is sufficiently small, see Remark 2.16. This result will hold for implicit RK methods in general.

Implicit RK methods become important when solving *stiff problems*, see Section 5.3. Those problems require the choice of extremely small step size τ.

We shall analyse the orders of implicit RK methods of stages $s \leq 3$, where we will not always require the maximal order of the resulting method. We use Corollaries 2.35, 2.33, and 2.36 to derive these methods.

$$\begin{array}{c|c}
c_1 & 1_{11} \\
\hline
 & b_1
\end{array}$$

Implicit RK ($s = 1$)

$$\begin{array}{c|cc}
c_1 & a_{11} & a_{12} \\
c_2 & a_{21} & a_{22} \\
\hline
 & b_1 & b_2
\end{array}$$

Implicit RK ($s = 2$)

$$\begin{array}{c|ccc}
c_1 & a_{11} & a_{12} & a_{23} \\
c_2 & a_{21} & a_{22} & a_{23} \\
c_2 & a_{31} & a_{32} & a_{13} \\
\hline
 & b_1 & b_2 & b_3
\end{array}$$

Implicit RK ($s = 3$)

Gauss method

We consider an implicit RK method of stage $s = 1$. The objective is to construct a method with maximal order p; by applying Corollary 2.35 to prove that the method is of order $p = 2$

at least. We can construct a system of nonlinear equations for the unknown coefficients c_1, a_{11}, b_1:

$$\sum_{i=1}^{1} b_i = b_1 = 1$$
$$\sum_{i=1}^{1} b_i c_i = b_1 c_1 = \frac{1}{2}$$
$$\sum_{j=1}^{1} a_{1j} = a_{11} = c_1$$

The system has the unique solution $c_1 = 1/2$, $a_{11} = 1/2$ and $b_1 = 1$. By analysing the conditions from Corollary 2.33 we conclude that the method cannot be of order higher then $p = 2$.

Example 2.11 (Gauss1). An implicit RK method, called *Gauss1*, of stage $s = 1$ of maximal order $p = 2$ is defined by the following Butcher tableau:

$$\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array} \tag{2.87}$$

The reason of the name "*Gauss1*" will become clear later.

We now consider an implicit RK method of order $s = 2$, with maximal order, which we conjecture is $p = 4$. Due to Corollary 2.36 we check defining conditions on the constants

$$b_1 + b_2 = 1 \tag{2.88a}$$
$$b_1 c_1 + b_2 c_2 = \frac{1}{2} \tag{2.88b}$$
$$b_1 c_1^2 + b_2 c_2^2 = \frac{1}{3} \tag{2.88c}$$
$$\sum_{i,j=1}^{2} b_i a_{ij} c_j = b_1(a_{11}c_1 + a_{12}c_2) + b_2(a_{21}c_1 + a_{22}c_2) = \frac{1}{6} \tag{2.88d}$$
$$b_1 c_1^3 + b_2 c_2^3 = \frac{1}{4} \tag{2.88e}$$
$$\sum_{i,j=1}^{2} b_i a_{ij} c_i c_j = b_1 c_1(a_{11}c_1 + a_{12}c_2) + b_2 c_2(a_{21}c_1 + a_{22}c_2) = \frac{1}{8} \tag{2.88f}$$
$$\sum_{i,j=1}^{2} b_i a_{ij} c_j^2 = b_1(a_{11}c_1^2 + a_{12}c_2^2) + b_2(a_{21}c_1^2 + a_{22}c_2^2) = \frac{1}{12} \tag{2.88g}$$
$$\sum_{i,j,k=1}^{2} b_i a_{ij} a_{jk} c_k = (b_1 a_{11} + b_2 a_{21}) \sum_{k=1}^{2} a_{1k} c_k + (b_1 a_{12} + b_2 a_{22}) \sum_{k=1}^{2} a_{2k} c_k = \frac{1}{24} \tag{2.88h}$$
$$c_1 = a_{11} + a_{12} \tag{2.88i}$$
$$c_2 = a_{21} + a_{22} \tag{2.88j}$$

Therefore, we have a system of 8 unknowns and 10 equations; hence, it appears that the problem is overdetermined. The right-hand sides of equations (2.88a)–(2.88c) and (2.88e) can be interpreted as integrals $\int_0^1 t^k \,\mathrm{d}t$, $k = 0, 1, 2, 3$; therefore, these equations can be written as

$$\sum_{i=1}^{2} b_i c_i^k = \int_0^1 t^k \,\mathrm{d}t\,, \quad k = 0, \dots, 3. \tag{2.89}$$

Consider the quadrature formula

$$\sum_{i=1}^{2} b_i g(c_i) \approx \int_0^1 g(t)\,\mathrm{d}t \tag{2.90}$$

with coefficients (weights) $b = (b_1, b_2)^\top$ and nodes $c = (c_1, c_2)^\top$.

Remark 2.41 (Gauss quadrature). From the theory of Gauss quadrature, cf., for example, Süli and Meyers (2003, Example 10.1), it follows that the condition (2.89) holds if

$$c = \left(\frac{1}{2} - \frac{\sqrt{3}}{6}, \frac{1}{2} + \frac{\sqrt{3}}{6}\right)^\top, \qquad b = \left(\frac{1}{2}, \frac{1}{2}\right)^\top. \tag{2.91}$$

In other words, the quadrature is exact for polynomials of degree three (i.e., order of the quadrature is four).

It can be checked that remaining equations (2.88d) and (2.88f)–(2.88j) have the solution

$$a_{11} = \frac{1}{4}, \qquad a_{12} = \frac{1}{4} - \frac{\sqrt{3}}{6}, \qquad a_{21} = \frac{1}{4} + \frac{\sqrt{3}}{6}, \qquad a_{22} = \frac{1}{4}.$$

Example 2.12 (Gauss2). An implicit RK method, called *Gauss2*, of stage $s = 2$ of maximal order $p = 4$ is defined by the following Butcher tableau:

$$\begin{array}{c|cc} 1/2 - \sqrt{3}/6 & 1/4 & 1/4 - \sqrt{3}/6 \\ 1/2 + \sqrt{3}/6 & 1/4 + \sqrt{3}/6 & 1/4 \\ \hline & 1/2 & 1/2 \end{array} \tag{2.92}$$

In general, let us consider the Gauss quadrature formulae

$$\sum_{i=1}^{k} b_i g(c_i) \approx \int_0^1 g(t)\,\mathrm{d}t \tag{2.93}$$

with coefficients (weights) $b = (b_1, \dots, b_k)^\top$ and nodes $c = (c_1, \dots, c_k)^\top$. If we assume $k = 1$ then we get Example 2.11. For $k = 3$, i.e. three quadrature nodes on the interval $[0, 1]$, then

$$c = \left(\frac{1}{2} - \frac{\sqrt{15}}{10}, 1, \frac{1}{2} + \frac{\sqrt{15}}{10}\right)^\top, \qquad b = \left(\frac{5}{18}, \frac{4}{9}, \frac{5}{18}\right)^\top, \tag{2.94}$$

is the solution of the Gauss interpolation problem. We will not give the corresponding Butcher tableau, but it can be found in Hairer et al. (2009, Table 7.4). The method, called *Gauss3*, is of order $p = 6$.

Remark 2.42 (Collocation methods). Analysing *Gauss2*, which is of order $p = 4$, is based on verifying the assumptions of Corollary 2.36; i.e. defining conditions (2.88). For the method *Gauss3* of order $p = 6$ the analysis of the corresponding conditions requires analysing thirty-six conditions; cf., Remark 1.23. As an alternative idea, the initial value problem (IVP) can be approximated by numerical collocation, see Deuflhard and Bornemann (2012, Section 6.3). This technique exploits *Gauss-type* quadratures, which simplifies the formulation of defining conditions, and leads to a class of implicit Butcher methods.

By *Gauss-type* quadrature we mean Gauss quadrature with additional nodes. Here, we consider Gauss-Radau and Gauss-Lobatto quadrature rules.

Radau method

We consider an RK method of stage $s = 2$, constructed to satisfy *as many* defining conditions (2.88) *as possible* while imposing one of two different additional conditions on the nodes c_1 and c_2:

$$\text{Version I}: \qquad c_1 = 0, \qquad 0 < c_2 \leq 1 \tag{2.95}$$

$$\text{Version II}: \qquad 0 \leq c_1 < 1, \qquad c_2 = 1 \tag{2.96}$$

We can satisfy (2.88a)–(2.88c); however, it is impossible to fulfil (2.88e).

Remark 2.43 (Gauss-Radau quadrature). Let us specify

$$\text{Version I}: \qquad c_1 = \left(0, \frac{2}{3}\right)^\top, \qquad b = \left(\frac{1}{4}, \frac{3}{4}\right)^\top, \tag{2.97}$$

$$\text{Version II}: \qquad c_1 = \left(\frac{1}{3}, 1\right)^\top, \qquad b = \left(\frac{3}{4}, \frac{1}{4}\right)^\top, \tag{2.98}$$

The quadrature is exact for second order polynomials; i.e., the quadrature is of order 3.

Example 2.13 (RadauI2 & RadauII2). Implicit RK methods of stage $s = 2$ with nodes specified by (2.97) and (2.97) has maximal order $p = 3$ and are defined by the following Butcher tableaux:

0	1/4	−1/4
2/3	1/4	5/12
	1/4	3/4

RadauI2

1/3	5/12	−1/12
1	3/4	1/4
	3/4	1/4

RadauII2

Assuming $k = 3$, i.e., three quadrature nodes on interval $[0, 1]$, the solution of the Gauss-Radau interpolation problem are two pairs

$$\text{Version I}: \qquad c_1 = \left(0, \frac{6-\sqrt{6}}{10}, \frac{6+\sqrt{6}}{10}\right)^\top, \qquad b = \left(\frac{1}{9}, \frac{16+\sqrt{6}}{36}, \frac{16-\sqrt{6}}{36}\right)^\top,$$

$$\text{Version II}: \qquad c_1 = \left(\frac{4-\sqrt{6}}{10}, \frac{4+\sqrt{6}}{10}, 1\right)^\top, \qquad b = \left(\frac{16-\sqrt{6}}{36}, \frac{16+\sqrt{6}}{36}, \frac{1}{9}\right)^\top,$$

The quadrature is exact for polynomials of order four (the quadrature is of order 5). We do not give the corresponding Butcher tableau, cf. Hairer and Wanner (2010, Table 5.4 & Table 5.6), which we shall call *RadauI3* and *RadauII3*.

Example 2.14 (RadauI1 & RadauII1). Implicit RK methods of stage $s = 1$ and order $p = 1$ are defined by the following Butcher tableaux:

0	1
	1

RadauI1

1	1
	1

RadauII1

Note that *RadauII1* is Implicit Euler.

Lobatto method

We consider implicit RK methods of stage $s = 2$ and $s = 3$, imposing additional conditions on the nodes

$$\begin{aligned} s = 2 &: \quad c_1 = 0, \quad c_2 = 1 \\ s = 3 &: \quad c_1 = 0, \quad 0 < c_2 < 1, \quad c_3 = 1 \end{aligned}$$

Remark 2.44 (Gauss-Lobatto quadrature). Let us specify

$$\begin{aligned} s = 2 &: \quad c = (0, 1)^\top, \quad b = \left(\frac{1}{2}, \frac{1}{2}\right)^\top \\ s = 3 &: \quad c = \left(0, \frac{1}{2}, 1\right)^\top, \quad b = \left(\frac{1}{6}, \frac{2}{3}, \frac{1}{6}\right).^\top \end{aligned}$$

The order of the quadrature is 2 for $s = 2$ and 3 for $s = 3$.

Example 2.15 (Lobatto). Implicit RK methods of stage $s = 2$ and $s = 3$ can be defined, via Gauss-Lobatto quadrature, by the following Butcher tableaux:

0	0	0
1	1/2	1/2
	1/2	1/2

Lobatto2

0	0	0	0
1/2	5/24	1/3	−1/24
1	1/6	2/3	1/6
	1/6	2/3	1/6

Lobatto3

It can be shown that

1. the *Lobatto2* method is of order $p = 2$,
2. the *Lobatto3* method is of order $p = 4$.

Note that the *Lobatto2* method is the Crank-Nicholson method.

CHAPTER 3

Multistep methods

In Chapter 2 we considered *one-step* methods to generate a numerical approximation of the solution of the initial value problem (IVP), which work by computing the numerical solution at the current time step based on the numerical solution at the previous time step; i.e., using *one-step* of the numerical solution. In this chapter, we consider *multistep methods* which use the numerical solution from multiple previous time steps.

3.1 Linear multistep method

The aim is to find a numerical solution of the initial value problem (IVP). We seek a phase curve $u(t) = \phi(t, t_0, x_0)$ on a given *finite* closed interval $t \in [t_0, T]$, and assume that $t_0 < T < t^+(t_0, x_0)$ and that the right-hand side is sufficiently smooth; i.e, $f \in C^k(J \times D, \mathbb{R}^n)$, $k \geq 1$.

We define an *equidistant partition* of the interval $[t_0, T]$ with step size $\tau > 0$ as

$$\{t_j\}_{j=0}^N, \quad t_j = t_{j-1} + \tau, \quad \tau = \frac{T - t_0}{N}. \tag{3.1}$$

Algorithm 3.1 (Linear m-step method)**.** A m-step method, for $m \geq 1$, is defined by a choice of real coefficients

$$\{a_i\}_{i=0}^m, \qquad \{b_i\}_{i=0}^m, \qquad a_m = 1 \tag{3.2}$$

such that $|a_0| + |b_0| \neq 0$. The iterative method is initialised with the first m values of the numerical solution

$$\{u_i\}_{i=0}^{m-1}, \quad u_0 \equiv x_0. \tag{3.3}$$

We then define the m-step recurrence as

$$\begin{aligned} &a_m u_{j+m} + a_{m-1} u_{j+m-1} + \cdots + a_0 u_j \\ &\qquad = \tau \left(b_m f(t_{j+m}, u_{j+m}) + b_{m-1} f(t_{j+m-1}, u_{j+m-1}) + \cdots + b_0 f(t_j, u_j) \right), \end{aligned} \tag{3.4}$$

for $j = 0, \ldots, N - m$.

The initialisation (3.3) can be accomplished by performing a chain of one-step methods

$$u_j = \psi(t_j, t_0, x_0), \quad j = 0, \ldots, m - 1. \tag{3.5}$$

Algorithm 3.1 generates a sequence $\{u_j\}_{j=0}^N$. It is expected that if τ is sufficiently small then the sequence $\{u_j\}_{j=0}^N$ will approximate the sequence $\{u(t_j)\}_{j=0}^N$ of the exact solution evaluated at the points $\{t_j\}_{j=0}^N$.

Linear m-step methods can be distinguished as

- *explicit*, if $b_m = 0$, or
- *implicit*, if $b_m \neq 0$.

Example 3.1 (Multistep explicit method). Let $m = 3$, i.e. we consider a three step method, with coefficients $a_3 = 1$, $a_2 = -1$, $a_1 = a_0 = 0$, $b_3 = 0$, $b_2 = 23/12$, $b_1 = -4/3$, $b_0 = 5/12$. Then,

$$u_{j+3} = u_{j+2} + \tau \left(\frac{23}{12} f(t_{j+2}, u_{j+2}) - \frac{4}{3} f(t_{j+1}, u_{j+1}) + \frac{5}{12} f(t_j, u_j) \right) \tag{3.6}$$

Vector $u_{j+3} \in \mathbb{R}^n$ is *explicitly* defined by the formula (3.6) as a linear combination of the previously computed values $u_{j+2}, u_{j+1}, u_j \in \mathbb{R}^n$, and three right-hand sides $f(t_{j+2}, u_{j+2})$, $f(t_{j+1}, u_{j+1})$, $f(t_j, u_j) \in \mathbb{R}^n$. The evaluation of these right-hand sides represents the substantial cost of the computation. In the next step we compute $u_{j+4} \in \mathbb{R}^n$ as

$$u_{j+4} = u_{j+3} + \tau \left(\frac{23}{12} f(t_{j+3}, u_{j+3}) - \frac{4}{3} f(t_{j+2}, u_{j+2}) + \frac{5}{12} f(t_{j+1}, u_{j+1}) \right).$$

We notice here that we have to evaluate $f(t_{j+3}, u_{j+3})$; however, the values $f(t_{j+2}, u_{j+2})$ and $f(t_{j+1}, u_{j+1})$ were evaluated in the *previous step* (the evaluation of u_{j+3}), and hence can be reused.

Linear multistep method can be interpreted as a *stencil* which is shifted at each step, and at each step we only have to evaluate the right-hand side once.

Example 3.2 (Multistep implicit method). Let $m = 2$, i.e. we consider a two step method, with coefficients $a_2 = 1$, $a_1 = -1$, $a_0 = 0$, $b_2 = 5/12$, $b_1 = 2/3$, $b_0 = -1/12$. Then,

$$u_{j+2} = u_{j+1} + \tau \left(\frac{5}{12} f(t_{j+2}, u_{j+2}) + \frac{2}{3} f(t_{j+1}, u_{j+1}) - \frac{1}{12} f(t_j, u_j) \right). \tag{3.7}$$

The formula (3.7) defines $u_{j+2} \in \mathbb{R}^n$ either as a root of a nonlinear system

$$u_{j+2} - u_{j+1} - \tau \left(\frac{5}{12} f(t_{j+2}, u_{j+2}) + \frac{2}{3} f(t_{j+1}, u_{j+1}) - \frac{1}{12} f(t_j, u_j) \right) = 0 \in \mathbb{R}^n, \tag{3.8}$$

or as a fixed point of the operator

$$u_{j+2} \in \mathbb{R}^n \longmapsto u_{j+1} + \tau \left(\frac{5}{12} f(t_{j+2}, u_{j+2}) + \frac{2}{3} f(t_{j+1}, u_{j+1}) - \frac{1}{12} f(t_j, u_j) \right) \in \mathbb{R}^n. \tag{3.9}$$

In the former case the root u_{j+2} of the system (3.8) could be approximated via the *Newton method* or its variations; while, in the latter case we could consider iterations of the type

$$u_{j+2}^{\text{new}} = u_{j+1} + \tau \left(\frac{5}{12} f(t_{j+2}, u_{j+2}^{\text{old}}) + \frac{2}{3} f(t_{j+1}, u_{j+1}) - \frac{1}{12} f(t_j, u_j) \right) \in \mathbb{R}^n. \tag{3.10}$$

If the step size τ is sufficiently small the iterations converge to the fixed point u_{j+2}.

As a result finding u_{j+2} is not a simple substitution to a formula (as in the case of explicit method), but instead we have to apply a numerical iterative method which is necessarily reflected in the overheads (costs). However, for *stiff problems*, see Section 5.3, it may be necessary to apply implicit methods.

Let us note that the m-step recurrence (3.4) can be formulated as an equation

$$\sum_{i=0}^{m} a_i u_{j+i} - \tau \sum_{i=0}^{m} b_i f(t_{j+i}, u_{j+i}) = 0 \in \mathbb{R}^n, \tag{3.11}$$

where u_{j+i} approximates $u(t_{j+i})$; i.e., $u(t_0 + \tau(j+i))$.

We want to define the local discretisation error. Let $f \in C^1(J \times D, \mathbb{R}^n)$ and recall that for each initial condition the solution is defined as

$$u(t+\tau) = \phi(t+\tau, t, x), \tag{3.12}$$

of the relevant initial problem, see Definition 1.11 of the flow of the vector field. The function $u(t+\tau)$ is continuously differentiable. We define

$$D(t+\tau, t, x) \equiv \sum_{i=0}^{m} a_i u(t+i\tau) - \tau \sum_{i=0}^{m} b_i f(t+i\tau, u(t+i\tau)) \in \mathbb{R}^n;$$

i.e., in the m-step recurrence we set the exact solution (3.12) for $j = 0$ with the initial condition $(t, x = u(t))$. We can estimate the corresponding error by exploiting the differential equation

$$u'(t+i\tau) = f(t+i\tau, u(t+i\tau)), \quad i = 0, \dots, m. \tag{3.13}$$

Definition 3.1 (Local discretisation error, order of the method & consistency)**.** Consider the linear m-step method (3.4) with coefficients (3.2). Assume that $f \in C^1(J \times D, \mathbb{R}^n)$ and for each $(t, x) \in J \times D$ we define the *local discretisation error* as the vector function

$$D(t+\tau, t, x) = \sum_{i=0}^{m} a_i u(t+i\tau) - \tau \sum_{i=0}^{m} b_i u'(t+i\tau). \tag{3.14}$$

If there exists a positive integer $p \geq 1$ such that

$$\|D(t+\tau, t, x)\| = \mathcal{O}(\tau^{p+1}), \qquad \text{for } \tau \to 0, \tag{3.15}$$

we say that the method is of order p at the point (t, x). If the method is of order at least $p = 1$, we say that the method is *consistent*.

Theorem 3.2. *Assume that $f \in C^p(J \times D, \mathbb{R}^n)$, $p \geq 1$, and let*

$$\sum_{i=0}^{m} a_i = 0, \qquad \sum_{i=0}^{m} i^\ell a_i = \ell \sum_{i=0}^{m} i^{\ell-1} b_i, \qquad \ell \in \{1, \dots, p\}, \qquad 0^0 \equiv 1. \tag{3.16}$$

Then, the corresponding linear multistep method is at least of order p at each point $(t, x) \in J \times D$.

Proof. Consider the Taylor expansion

$$u(t+i\tau) = \sum_{\ell=0}^{p} \frac{i^\ell}{\ell!} \tau^\ell u^{(\ell)}(t) + \mathcal{O}(\tau^{p+1})$$

for $i = 0, \dots m$, see (1.34); additionally,

$$\tau u'(t + i\tau) = \sum_{\ell=1}^{p} \frac{i^{\ell-1}}{(\ell-1)!} \tau^{\ell} u^{(\ell)}(t) + \mathcal{O}(\tau^{p+1}).$$

Then, by linear combination of these results we have that

$$a_i u(t + i\tau) - \tau b_i u'(t + i\tau) = a_i u(t) + \sum_{\ell=1}^{p} \frac{1}{\ell!} (i^{\ell} a_i - \ell i^{\ell-1} b_i) \tau^{\ell} u^{(\ell)}(t) + \mathcal{O}(\tau^{p+1}).$$

Summing from $i = 0, \dots, m$ we obtain a result for the local discretisation error (3.14)

$$\sum_{i=0}^{m} a_i u(t + i\tau) - \tau \sum_{i=0}^{m} b_i u'(t + i\tau) = \sum_{\ell=0}^{p} \frac{\tau^{\ell}}{\ell!} C_{\ell} u^{(\ell)}(t) + \mathcal{O}(\tau^{p+1}),$$

where

$$C_0 = \sum_{i=0}^{m} a_i, \qquad C_{\ell} = \sum_{i=0}^{m} (i^{\ell} a_i - \ell i^{\ell-1} b_i), \quad \ell = 1, \dots, p. \tag{3.17}$$

Hence, if $C_{\ell} = 0$ for $\ell = 0, \dots, p$, then the method is at least of order p; this is equivalent to the condition (3.16). □

We can verify that the explicit three step method Example 3.1 is of order $p = 3$ by checking that $C_0 = C_1 = C_2 = C_3 = 0$ and $C_4 \neq 0$. Similarly, we can verify that the two step method Example 3.2 is of order $p = 3$ as well.

Corollary 3.3 (Consistency conditions). *Assume that $f \in C^2(J \times D, \mathbb{R}^n)$; then, the linear m-step method with coefficients (3.2) is consistent if and only if*

$$\sum_{i=0}^{m} a_i = 0, \qquad \sum_{i=0}^{m} i a_i = \sum_{i=0}^{m} b_i. \tag{3.18}$$

Definition 3.4. Consider a linear m-step method with coefficients (3.2), then we can associate the following complex polynomials of variable $z \in \mathbb{C}$:

first characteristic polynomial

$$\rho(z) = \sum_{i=0}^{m} a_i z^i, \tag{3.19}$$

second characteristic polynomial

$$\sigma(z) = \sum_{i=0}^{m} b_i z^i. \tag{3.20}$$

Remark 3.5. The method is consistent in the sense of (3.18) if and only if

$$\rho(1) = 0, \qquad \rho'(1) = \sigma(1). \tag{3.21}$$

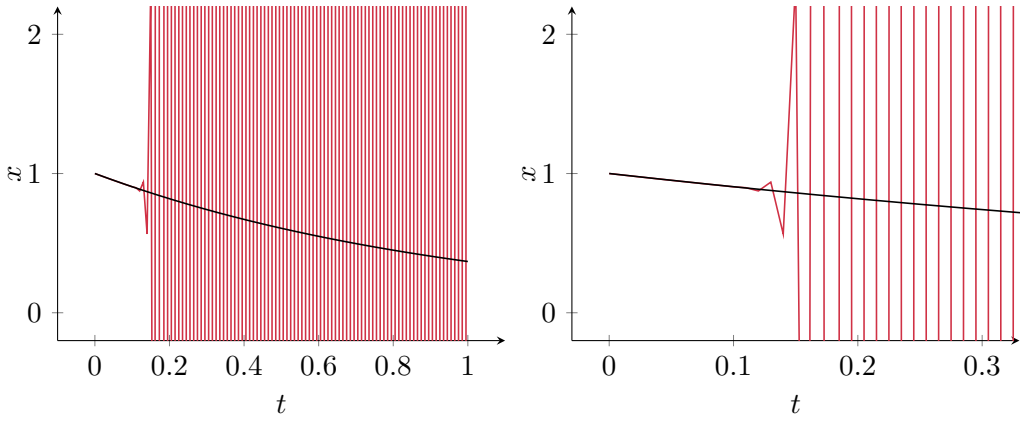


Figure 3.1: Comparison of exact solution and unstable multistep solution of the initial value problem (3.23)

3.2 D-stability & Convergence

We consider an explicit two step method

$$u_{j+2} = -4u_{j+1} + 5u_j + \tau\left(4f(t_{j+1}, u_{j+1}) + 2f(t_j, u_j)\right), \tag{3.22}$$

which can be shown to be of order $p = 3$. Let us solve the following initial value problem

$$x' = -x, \qquad x(0) = 1 \tag{3.23}$$

using this method in the interval $[0, 1]$ with step size $\tau = 0.01$. This problem has a known analytical solution, $\phi(t, 0, 1) = \mathrm{e}^{-t}$. Figure 3.1 shows the comparison of the exact and numerical solution; we observe that dramatic oscillation occur in the numerical solution. We note that if we used the method (3.1) or (3.2) from Section 3.1, which are also of order $p = 3$, with the same discretisation data $\tau = 0.01$, the numerical and exact solutions would appear (almost) identical.

In order to understand the potential issue we consider solving the scalar initial value problem

$$x' = 0, \qquad x(0) = x_0 \tag{3.24}$$

using the method (3.22). The method is initialized via (3.3) by choosing $u_0 = x_0$ and u_1. Since the right hand side of equation (3.24) is zero, the recurrence (3.22) is a linear recurrence driven by the homogeneous linear difference equation

$$u_{j+2} + 4u_{j+1} - 5u_j = 0. \tag{3.25}$$

Using the theory of linear difference equations we can derive the general solution of (3.25). The linear difference equation has the characteristic polynomial $z^2 + 4z - 5$, which is, in fact, the first characteristic polynomial $\rho(z) = z^2 + 4z - 5$ of the method (3.22). This polynomial has the roots $z_1 = 1$ and $z_2 = -5$ and, hence, the general solution $\{u_j\}_{j=0}^{\infty}$ is a linear combination of both fundamental solutions $\left\{1^j\right\}_{j=0}^{\infty}$ and $\left\{(-5)^j\right\}_{j=0}^{\infty}$. Therefore,

$$\{u_j\}_{j=0}^{\infty} = \left\{c_1 1^j + c_2(-5)^j\right\}_{j=0}^{\infty},$$

where c_1 and c_2 are arbitrary constants. These constants can be deduced from the initial values u_0 and u_1 by means of a linear transformation

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 5 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix}.$$

Consider the initial condition $u_0 = x_0 = 1$ and let u_1 be defined via the Runge method with step size τ, cf. (3.5). Evaluating the relevant Butcher tableau it yields $\kappa_1 = f(0, x_0) = 0$, $\kappa_2 = f(0+\tau/2, x_0+\tau\kappa_1/2) = 0$ and $u_1 = u_0+\tau\kappa_2 = 1$. Hence, the numerical solution $\{u_j\}_{j=0}^{\infty} \equiv \{1^j\}_{j=0}^{\infty}$ of the corresponding initial value problem is a constant solution as expected.

We now simulate the influence of the rounding errors on the actual computation. We assume that $u_0 = 1$ and $u_1 = 1 + \tau\varepsilon$, where ε is very small, e.g. it is comparable to the machine precision. The simulated solution is given by the formula

$$\{u_j\}_{j=0}^{\infty} = \frac{1}{6}\left\{6 + \tau\varepsilon(1 - (-5)^j)\right\}_{j=0}^{\infty}, \quad \text{for } \{t_j\}_{j=0}^{\infty} = \{\tau j\}_{j=0}^{\infty}.$$

For a given fixed time $T > 0$, we consider the partition (3.1), where N is a parameter satisfying $N \to \infty$, and set the time step to $\tau = T/N$. We can then compute the numerical solution at time T

$$u_N = \frac{1}{6}(6 + \tau\varepsilon(1 - (-5)^N));$$

hence, $|u_N| \to \infty$ for $N \to \infty$. In contrast, the exact solution at time T is equal to one, $\phi(T, 0, 1) = 1$.

In conclusion, we considered a numerical experiment which illustrated *instability* of a particular multistep method (3.22), cf. Figure 3.1. This is a consistent method of order $p = 3$, but an *instability* of the method appeared when solving the simplest problem (3.24) due to rounding errors while performing the computation. The reason for this instability is due to the fact that one of the roots $z_2 = -5$ of the polynomial $\rho(z) = z^2 + 4z - 5$, lies outside the unit circle. Is it possible to characterise a *stable* multistep method?

Definition 3.6 (D-stability (G. Dahlquist (1956))). A linear m-step method with coefficients (3.2) is *D-stable* provided that each root $z \in \mathbb{C}$ of the first characteristic polynomial $\rho(z) = 0$ satisfies either

- $|z| < 1$, or
- $|z| = 1$ and $\rho'(z) \neq 0$ (i.e., the algebraic multiplicity of the root z is equal to 1).

Both the explicit method (3.1) and the implicit method (3.2) from Section 3.1 are D-stable:

explicit method (3.1)**:** the first characteristic polynomial $\rho(z) = z^2(z-1)$ has roots $\rho(1) = 0$, $\rho'(1) = 1 \neq 0$ and the multiple root $\rho(0) = 0$, $\rho'(0) = 0$.

implicit method (3.2)**:** the first characteristic polynomial $\rho(z) = z(z-1)$ has roots $\rho(1) = 0$, $\rho'(1) = 1 \neq 0$ and $\rho(0) = 0$.

Theorem 3.7 (The global error estimate). *Assume that $f \in C^p(J \times D, \mathbb{R}^n)$, $p \geq 1$ and let $u(t) = \phi(t, t_0, x_0)$ be the solution of the initial value problem* (IVP) *in the interval $t \in [t_0, T]$. We*

consider a D-stable m*-step method* (3.4) *of order* $p \geq 1$ *on the equidistant partition* (3.1)*, with coefficients* (3.2) *and initialisation* (3.3)*, which generates the sequence* $\{u_j\}_{j=0}^N$.

There exists a positive constant $C > 0$ *such that for sufficiently large* N

$$\|u(t_j) - u_j\| \leq C(\varepsilon_0 + \tau^p), \quad j = 0, \dots, N, \qquad \tau = \frac{T - t_0}{N}, \tag{3.26}$$

where

$$\varepsilon_0 \equiv \max_{\ell=0,\dots,m-1} \|u(t_\ell) - u_\ell\|.$$

is the initialisation error for (3.3).

Proof. see Deuflhard and Bornemann (2012, Theorem 7.23). □

In order to initialise Algorithm 3.1, it is suitable to use the one step method (3.5) of order k, where $k \geq p$. In this case $\varepsilon_0 = \mathcal{O}(\tau^k)$ and the initialisation does not effect the order p of the error estimate (3.26).

There are theoretical limits to the maximum achievable order of the D-stable m-step method:

Remark 3.8 (Dahlquist Barrier). Consider the D-stable m-step method of the order $p \geq 1$ on the equidistant partition (3.1); then, it is necessary that

$$p \leq \begin{cases} m+2 & \text{if } m \text{ is even,} \\ m+1 & \text{if } m \text{ is odd,} \\ m & \text{if } {}^{b_m}/_{a_m} \leq 0 \text{ (in particular if the method is explicit i.e., } b_m = 0). \end{cases}$$

Proof. see Hairer et al. (2009, Theorem 3.5). □

Example 3.3 (Implicit 2-step method with maximal order $p = 4$). We will derive a two step method with the highest order. The linear two step method is defined by choosing six coefficients a_2, a_1, a_0, b_2, b_1, b_0 satisfying (3.2). From the proof of Theorem 3.2, namely (3.17),

$$\begin{aligned}
C_0 &= a_0 + a_1 + a_2 \\
C_1 &= a_1 + 2a_2 - b_0 - b_1 - b_2 \\
C_2 &= \frac{1}{2}a_1 + 2a_2 - b_1 - 2b_2 \\
C_3 &= \frac{1}{3}a_1 + \frac{8}{3}a_2 - b_1 - 4b_2 \\
C_4 &= \frac{1}{4}a_1 + 4a_2 - b_1 - 8b_2 \\
C_5 &= \frac{1}{5}a_1 + \frac{32}{5}a_2 - b_1 - 16b_2 \\
&\vdots
\end{aligned}$$

We require that $a_2 = 1$, see (3.2), and if $C_0 = C_1 = C_2 = C_3 = C_4 = 0$ then the method is of the order $p \geq 4$. We can compute the coefficients a_1, a_0, b_2, b_1, and b_0 as a solution of the

system

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & -1 \\ 0 & 1/2 & 0 & -1 & -2 \\ 0 & 1/3 & 0 & -1 & -4 \\ 0 & 1/4 & 0 & -1 & -8 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \\ -2 \\ -8/3 \\ -4 \end{bmatrix}. \tag{3.27}$$

The solution of this system is uniquely determined as

$$a_2 = 1, \quad a_1 = 0, \quad a_0 = -1, \quad b_2 = \frac{1}{3}, \quad b_1 = \frac{4}{3}, \quad b_0 = \frac{1}{3},$$

and additionally we can check that $C_5 \neq 0$ for these coefficients to show that $p = 4$ is the highest order possible. This method is D-stable as the first characteristic polynomial $\rho(z) \equiv a_2 z^2 + a_1 z + a_0 = z^2 - 1$ has the simple roots $\{1, -1\}$ on the unit circle. We conclude that the method

$$u_{j+2} = u_{j+1} + \tau \left(\frac{1}{3} f(t_{j+2}, u_{j+2}) + \frac{4}{3} f(t_{j+1}, u_{j+1}) + \frac{1}{3} f(t_j, u_j) \right) \tag{3.28}$$

defines an implicit ($b_2 \neq 0$) D-stable two step method of maximal order. From Remark 3.8 we see that the method hits the barrier exactly as the method satisfies $p = m + 2$ for even $m = 2$.

Example 3.4 (Explicit 2-step method with maximal order $p = 3$). We are derive an *explicit* two step method with the highest order. The linear two step method is defined by choosing six coefficients a_2, a_1, a_0, b_2, b_1, and b_0 satisfying (3.2), i.e. $a_2 = 1$. We want the method to be explicit, i.e., we require that $b_2 = 0$. We consider the expansion (3.17) subject to the constraint $a_2 = 1$ and $b_2 = 0$. We propose that the maximal order will be $p = 3$; i.e., $C_0 = C_1 = C_2 = C_3 = 0$, and $C_4 \neq 0$. The unknown coefficients a_1, a_0, b_2, b_1 are given as the solution of the system

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 1/2 & 0 & -1 \\ 0 & 1/3 & 0 & -1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \\ -2 \\ -8/3 \end{bmatrix}. \tag{3.29}$$

The solution of this system is uniquely determined as

$$a_2 = 1, \qquad a_1 = 4, \qquad a_0 = -5, \qquad b_2 = 0, \qquad b_1 = 4, \qquad b_0 = 2$$

This is actually the method (3.22), which was shown to not be D-stable. According to Remark 3.8, any explicit D-stable two-step method is of the order $p \leq 2$.

3.3 Construction of multistep methods

The aim is to derive m-step methods (3.4) which approximates the sequence $\{u(t_j)\}_{j=0}^N$ of exact solutions of the initial value problem (IVP) on an equidistant partition (3.1) of the interval $[t_0, T]$.

3.3.1 Adams methods

We start by defining the coefficients (3.2) such that

$$a_m = 1, \qquad a_{m-1} = -1, \qquad a_{m-2} = \cdots = a_0 = 0, \tag{3.30}$$

which satisfies the first condition from Theorem 3.2. Then, for $j = 0, \dots, N - m$, we have that

$$u_{j+m} - u_{j+m-1} = \tau \left(b_m f(t_{j+m}, u_{j+m}) + b_{m-1} f(t_{j+m-1}, u_{j+m-1}) + \cdots + b_0 f(t_j, u_j) \right). \quad (3.31)$$

We then compute the unknown coefficients b_m, ..., b_0 in order to obtain the highest order possible, using the sufficient condition from Theorem 3.2. These yield the so-called *Adams methods*. The first characteristic polynomial for these methods is

$$\rho(z) = z^m - z^{m-1} = (z-1)z^{m-1}, \quad (3.32)$$

which means that the method is is D-stable for all values of m.

We give two examples:

Example 3.5 (Adams method: $m = 2$, implicit). A linear two step method ($m = 2$), satisfying (3.31), of the highest order is given by

$$u_{j+2} = u_{j+1} + \tau \left(\frac{5}{12} f(t_{j+2}, u_{j+2}) + \frac{2}{3} f(t_{j+1}, u_{j+1}) - \frac{1}{12} f(t_j, u_j) \right).$$

In order to derive this method we consider the second condition from Theorem 3.2 with $a_2 = 1$, $a_1 = -1$, and $a_0 = 0$, cf. (3.30), and attempt to derive a method of order $p = 3$; hence, we have that

$$\frac{a_1 + 2^\ell a_2}{\ell} = 0^{\ell-1} b_0 + 1^{\ell-1} b_1 + 2^{\ell-1} b_2, \qquad \ell = 1, \dots, 3.$$

Hence, we can compute b_2, b_1, and b_0 as the unique solution of the system

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_1 + 2a_2 \\ a_1/2 + 2a_2 \\ a_1/3 + 8a_2/3 \end{bmatrix}.$$

Hence,

$$a_2 = 1, \qquad a_1 = -1, \qquad a_0 = 0, \qquad b_2 = \frac{5}{12}, \qquad b_1 = \frac{2}{3}, \qquad b_0 = -\frac{1}{12}.$$

We note that (3.16) does not hold for $\ell = 4$; hence, the highest order of the method is $p = 3$.

Example 3.6 (Adams method: $m = 2$, explicit). An explicit linear two step method ($m = 2$), satisfying (3.31), of the highest order is given by

$$u_{j+2} = u_{j+1} + \tau \left(\frac{3}{2} f(t_{j+1}, u_{j+1}) - \frac{1}{2} f(t_j, u_j) \right).$$

In order to derive this formula we proceed similarly as in Example 3.5. We first require that $a_2 = 1$, $a_1 = -1$, and $a_0 = 0$, cf. (3.30), and additional as we are search for an explicit method we require that $b_2 = 0$. We consider (3.16) for $\ell = 1, 2$, and hence can compute the coefficients b_1 and b_0 as the unique solution of the system

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} a_1 + 2a_2 \\ a_1/2 + 2a_2 \end{bmatrix}.$$

Hence,

$$a_2 = 1, \qquad a_1 = -1, \qquad a_0 = 0, \qquad b_2 = 0, \qquad b_1 = \frac{3}{2}, \qquad b_0 = -\frac{1}{2}$$

gives a method of order $p = 2$, as (3.16) does not hold for $\ell = 2$.

These methods create two classes of methods for different m:

- explicit Adams methods, called *Adams-Bashfort* methods, and
- implicit Adams methods, called *Adams-Moulton* methods.

Example 3.7 (Explicit Adams methods — Adams-Bashfort ($m = 1, 2, 3, 4$)). Adams-Bashfort methods, which are explicit Adams methods, for $m = 1, 2, 3, 4$ are given by the following formulas, respectively:

$$u_{j+1} = u_j + \tau f(t_j, u_j), \tag{ab1}$$

$$u_{j+2} = u_{j+1} + \tau \left(\frac{3}{2} f(t_{j+1}, u_{j+1}) - \frac{1}{2} f(t_j, u_j) \right), \tag{ab2}$$

$$u_{j+3} = u_{j+2} + \tau \left(\frac{23}{12} f(t_{j+2}, u_{j+2}) - \frac{4}{3} f(t_{j+1}, u_{j+1}) + \frac{5}{12} f(t_j, u_j) \right), \tag{ab3}$$

$$u_{j+4} = u_{j+3} + \tau \left(\frac{55}{24} f(t_{j+3}, u_{j+3}) - \frac{59}{24} f(t_{j+2}, u_{j+2}) + \frac{37}{24} f(t_{j+1}, u_{j+1}) - \frac{3}{8} f(t_j, u_j) \right) \tag{ab4}$$

The m-step Adams-Bashfort method is of the order $p = m$. Note that ab1 is the Euler method.

Example 3.8 (Implicit Adams methods — Adams-Moulton ($m = 1, 2, 3, 4$)). Adams-Moulton methods, which are implicit Adams methods, for $m = 1, 2, 3, 4$ are given by the following formulas, respectively:

$$u_{j+1} = u_j + \frac{1}{2} \tau \left(f(t_{j+1}, u_{j+1}) + f(t_j, u_j) \right), \tag{am1}$$

$$u_{j+2} = u_{j+1} + \tau \left(\frac{5}{12} f(t_{j+2}, u_{j+2}) + \frac{2}{3} f(t_{j+1}, u_{j+1}) - \frac{1}{12} f(t_j, u_j) \right), \tag{am2}$$

$$u_{j+3} = u_{j+2} + \tau \left(\frac{3}{8} f(t_{j+3}, u_{j+3}) + \frac{19}{24} f(t_{j+2}, u_{j+2}) - \frac{5}{24} f(t_{j+1}, u_{j+1}) + \frac{1}{24} f(t_j, u_j) \right), \tag{am3}$$

$$u_{j+4} = u_{j+3} + \tau \left(\frac{251}{720} f(t_{j+4}, u_{j+4}) + \frac{646}{720} f(t_{j+3}, u_{j+3}) - \frac{264}{720} f(t_{j+2}, u_{j+2}) + \frac{106}{720} f(t_{j+1}, u_{j+1}) - \frac{19}{720} f(t_j, u_j) \right). \tag{am4}$$

The m-step Adams-Moulton method is of the order $p = m + 1$. Note that am1 is the Crank-Nicholson method.

For the practical implementation of multistep methods, it may be convenient to *shift* the *stencil*; i.e, alter the indexing of $\{u_j\}_{j=0}^N$ and $\{t_j\}_{j=0}^N$. For example, the method (ab4)

$$u_{j+4} = u_{j+3} + \tau \left(\frac{55}{24} f(t_{j+3}, u_{j+3}) - \frac{59}{24} f(t_{j+2}, u_{j+2}) + \frac{37}{24} f(t_{j+1}, u_{j+1}) - \frac{3}{8} f(t_j, u_j) \right)$$

can be formulated as a 4-step recurrence ($m = 4$)

$$u_{j+1} = u_j + \tau \left(\frac{55}{24} f(t_j, u_j) - \frac{59}{24} f(t_{j-1}, u_{j-1}) + \frac{37}{24} f(t_{j-2}, u_{j-2}) - \frac{3}{8} f(t_{j-3}, u_{j-3}) \right),$$

for $j = 3, \dots, N-1$, with the initialisation $u_0 \equiv x_0, u_1, u_2, u_3$. Similarly, the method (am2)

$$u_{j+2} = u_{j+1} + \tau \left(\frac{5}{12} f(t_{j+2}, u_{j+2}) + \frac{2}{3} f(t_{j+1}, u_{j+1}) - \frac{1}{12} f(t_j, u_j) \right)$$

could be equivalently formulated as a two step recurrence ($m = 2$)

$$u_{j+1} = u_j + \tau \left(\frac{5}{12} f(t_{j+1}, u_{j+1}) + \frac{2}{3} f(t_j, u_j) - \frac{1}{12} f(t_{j-1}, u_{j-1}) \right), \tag{3.33}$$

for $j = 1, \dots, N-1$, with the initialisation $u_0 \equiv x_0, u_1$.

More formally, the original m-step recurrence (3.4) can be equivalently formulated as the m-step recurrence

$$\begin{aligned} a_m u_{j+1} + a_{m-1} u_j + \cdots + a_0 u_{j-m+1} \\ = \tau \left(b_m f(t_{j+1}, u_{j+1}) + b_{m-1} f(t_j, u_j) + \cdots + b_0 f(t_{j-m+1}, u_{j-m+1}) \right), \end{aligned} \tag{3.34}$$

for $j = m-1, \dots, N-1$. with the initialisation $u_0 \equiv x_0, u_1, \dots, u_{m-1}$.

Remark 3.9. For example, in Quarteroni et al. (2010), algorithms related to linear multistep methods are reported in a shifted version, such as (3.34). In general, our presentation follows Deuflhard and Bornemann (2012) and thus uses (3.4).

Adams methods were originally derived by numerical integration. We consider the initial value problem (IVP) and by the integral definition of the solution (1.16) we can derive for the equidistant partition (3.1) the identity

$$u(t_{j+1}) = u(t_{j-k}) + \int_{t_{j-k}}^{t_{j+1}} f(s, u(s)) \, \mathrm{d}s, \qquad k = 0, 1, 2, \dots. \tag{3.35}$$

We can approximate $f(s, u(s))$ using Lagrange interpolation of $f(\cdot, u(\cdot))$ at the nodes t_i, $i = j-q, \dots, j+\ell$, $q \in \mathbb{N}_0$, $\ell \in \{0, 1\}$, given by

$$f(s, u(s)) \approx \mathcal{L}_{j-q}(s) f_{j-q} + \cdots + \mathcal{L}_j(s) f_j + \cdots + \mathcal{L}_{j+\ell}(s) f_{j+\ell} \tag{3.36}$$

where

$$f_i = f(t_i, u(t_i)), \qquad i = j-q, \dots, j+\ell,$$

and

$$\mathcal{L}_{j-q+i}(s) = \prod_{\substack{k=0 \\ k \neq i}}^{q+\ell} \frac{s - t_{j-q+k}}{t_{j-q+i} - t_{j-q+k}} \in \mathbb{P}_{q+\ell-1}, \qquad t_{j-k} \leq s \leq t_{j+1}, \quad i = 0, \dots, q+\ell \tag{3.37}$$

are the Lagrange basis functions. Then, we can define the multistep method as

$$u_{j+1} - u_{j-k} = \int_{t_{j-k}}^{t_{j+1}} f(s, u(s)) \, \mathrm{d}s \approx \sum_{i=0}^{q+\ell} f_{j-q+i} \int_{t_{j-k}}^{t_{j+1}} \mathcal{L}_{j-q+i}(s) \, \mathrm{d}s. \tag{3.38}$$

We note that $\ell = 0$ defines an explicit method and $\ell = 1$ defines an implicit method. Letting $q = 1$, $k = 0$ and $\ell = 1$ we get the 3-step recurrence

$$u_{j+1} - u_j = f_{j+1} \int_{t_j}^{t_{j+1}} \mathcal{L}_{j+1}(s) \, \mathrm{d}s + f_j \int_{t_j}^{t_{j+1}} \mathcal{L}_j(s) \, \mathrm{d}s + f_{j-1} \int_{t_j}^{t_{j+1}} \mathcal{L}_{j-1}(s) \, \mathrm{d}s, \tag{3.39}$$

By introducing the substitution $w = {}^{(s-t_{j-q})}/_{\tau}$ into (3.36) we can define

$$L_i(w) = \prod_{\substack{k=0 \\ k\neq i}}^{2} \frac{w-k}{i-k} \in \mathbb{P}_2, \qquad 0 \le w \le 2, \quad w = \frac{1}{\tau}(s - t_{j-q}), \quad i = 0, 1, 2; \tag{3.40}$$

then

$$\begin{aligned}
\int_{t_j}^{t_{j+1}} \mathcal{L}_{j+1}(s)\,\mathrm{d}s &= \tau \int_1^2 L_2(w)\,\mathrm{d}w = \tau\frac{5}{12}, \\
\int_{t_j}^{t_{j+1}} \mathcal{L}_{j}(s)\,\mathrm{d}s &= \tau \int_1^2 L_1(w)\,\mathrm{d}w = \tau\frac{2}{3}, \\
\int_{t_j}^{t_{j+1}} \mathcal{L}_{j-1}(s)\,\mathrm{d}s &= \tau \int_1^2 L_0(w)\,\mathrm{d}w = -\tau\frac{1}{12}.
\end{aligned}$$

These are the coefficients $b_2 = \frac{5}{12}$, $b_1 = \frac{2}{3}$, $b_0 = -\frac{1}{12}$ of the method am2 from Example 3.8 with shifted stencil; cf. (3.33).

3.3.2 Predictor/Corrector methods

In Example 3.2 we found that to evaluate the j-th step of the method it is necessary to solve either a nonlinear problem (3.8) or find a fixed point (3.9). Both procedures only solvable the non-linearity approximatively at an extra cost. In this section, we discuss the third alternative to evaluating the j-th step of the implicit method, called the *Predictor/Corrector* technique. We demonstrate this with an example, using the shifted stencil (3.34) versions of the two-step Adams-Bashfort method (ab2) as a *predictor* and the two-step Adams-Moulton method (am2) as the *corrector*:

$$\begin{aligned}
&\text{ab2 (Predictor):} && u_{j+1} = u_j + \tau\left(\frac{3}{2}f(t_j, u_j) - \frac{1}{2}f(t_{j-1}, u_{j-1})\right), \\
&\text{am2 (Corrector):} && u_{j+1} = u_j + \tau\left(\frac{5}{12}f(t_{j+1}, u_{j+1}) + \frac{2}{3}f(t_j, u_j) - \frac{1}{12}f(t_{j-1}, u_{j-1})\right).
\end{aligned}$$

We then consider three different *predictor/corrector* algorithms:

Algorithm 3.2 ($PECE$)**.** At the time step t_{j+1} the following steps are performed:

Predict compute the *predictor* (ab2):

$$u^P_{j+1} = u_j + \tau\left(\frac{3}{2}f(t_j, u_j) - \frac{1}{2}f(t_{j-1}, u_{j-1})\right),$$

Evaluate evaluate the right-hand side:

$$f^E_{j+1} = f(t_{j+1}, u^P_{j+1}),$$

Correct compute the *corrector* (am2):

$$u^C_{j+1} = u_j + \tau\left(\frac{5}{12}f^E_{j+1} + \frac{2}{3}f(t_j, u_j) - \frac{1}{12}f(t_{j-1}, u_{j-1})\right),$$

Evaluate evaluate the right-hand side:

$$f_{j+1}^E = f(t_{j+1}, u_{j+1}^C).$$

Then, we can define $u_{j+1} = u_{j+1}^C$ and $f(t_{j+1}, u_{j+1}) = f_{j+1}^E$ for the next time step.

In the next algorithm we save one evaluation on the right-hand side:

Algorithm 3.3 (PEC)**.** At the time step t_{j+1} the following steps are performed:

Predict compute the *predictor* (ab2):

$$u_{j+1}^P = u_j + \tau \left(\frac{3}{2} f(t_j, u_j) - \frac{1}{2} f(t_{j-1}, u_{j-1}) \right),$$

Evaluate evaluate the right-hand side:

$$f_{j+1}^E = f(t_{j+1}, u_{j+1}^P),$$

Correct compute the *corrector* (am2):

$$u_{j+1}^C = u_j + \tau \left(\frac{5}{12} f_{j+1}^E + \frac{2}{3} f(t_j, u_j) - \frac{1}{12} f(t_{j-1}, u_{j-1}) \right),$$

Then, we can define $u_{j+2} = u_{j+1}^C$ and $f(t_{j+1}, u_{j+1}) = f_{j+1}^E$ for the next time step.

In the next variant we will iterate the corrector twice:

Algorithm 3.4 ($PECECE = P(EC)^2E$)**.** At the time step t_{j+1} the following steps are performed:

Predict compute the *predictor* (ab2):

$$u_{j+1}^P = u_j + \tau \left(\frac{3}{2} f(t_j, u_j) - \frac{1}{2} f(t_{j-1}, u_{j-1}) \right),$$

Evaluate evaluate the right-hand side:

$$f_{j+1}^E = f(t_{j+1}, u_{j+1}^P),$$

Correct compute the *corrector* (am2):

$$u_{j+1}^C = u_j + \tau \left(\frac{5}{12} f_{j+1}^E + \frac{2}{3} f(t_j, u_j) - \frac{1}{12} f(t_{j-1}, u_{j-1}) \right),$$

Evaluate evaluate the right-hand side:

$$f_{j+1}^E = f(t_{j+1}, u_{j+1}^C).$$

Correct compute the *corrector* (am2):

$$u_{j+1}^C = u_j + \tau \left(\frac{5}{12} f_{j+1}^E + \frac{2}{3} f(t_j, u_j) - \frac{1}{12} f(t_{j-1}, u_{j-1}) \right),$$

Evaluate evaluate the right-hand side:

$$f_{j+1}^E = f(t_{j+1}, u_{j+1}^C).$$

Then, we can define $u_{j+1} = u_{j+1}^C$ and $f(t_{j+1}, u_{j+1}) = f_{j+1}^E$ for the next time step.

These algorithms can be modified in several ways:

- Instead of choosing ab2 (predictor) and am2 (corrector), respectively, we can consider an arbitrary m-step Adams-Bashfort method (predictor) and m-step Adams-Moulton method (corrector), respectively.
- The *evaluate-correct* steps can be repeated multiple times as convenient. Hence, we can consider algorithms $P(EC)^k E$ and $P(EC)^k$, where $k \in \mathbb{N}$ is a positive integer.

Remark 3.10 (Order of Predictor/Corrector methods). Let us consider an arbitrary m-step Adams-Bashfort method and m-step Adams-Moulton method with either the $P(EC)^k E$ or $P(EC)^k$ Predictor/Corrector algorithm, where k is a positive integer. We will assume that N defining the equidistant partition (3.1) is sufficiently large; i.e., the time step τ is sufficiently small. Then, it holds that the Predictor/Corrector method is of the order $p = m + 1$, see Deuflhard and Bornemann (2012, Lemma 7.38). Therefore, the order of the Predictor/Corrector method is equal to the order of the corrector.

The above statement holds asymptotically for sufficiently small τ. There is no theoretical guidance on how to choose k. We can also consider Predictor/Corrector methods combining Adams-Bashfort and Adams-Moulton with different choice of m; e.g.,

$$\begin{aligned} &\text{ab1 (Predictor):} && u_{j+1} = u_j + \tau f(t_j, u_j), \\ &\text{am2 (Corrector):} && u_{j+1} = u_j + \tau \left(\frac{5}{12} f(t_{j+1}, u_{j+1}) + \frac{2}{3} f(t_j, u_j) - \frac{1}{12} f(t_{j-1}, u_{j-1}) \right). \end{aligned}$$

3.3.3 BDF methods

In this section, we define the BDF methods. We first start by defining two-step BDF method:

Example 3.9 (BDF2). We aim to derive a two step method ($m = 2$) of the highest order satisfying the constraint $b_0 = b_1 = 0$. As $a_2 = 1$, then we need to derive the coefficients a_0, a_1, and b_2. From Theorem 3.2 we can show that for $p = 2$ the coefficients can be found as the solution of the linear system

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & -1 \\ 0 & 1/2 & -2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \\ -2 \end{bmatrix}.$$

Hence, the coefficients of the method are

$$a_2 = 1, \qquad a_1 = -\frac{4}{3}, \qquad a_0 = \frac{1}{3}, \qquad b_2 = \frac{2}{3}, \qquad b_1 = 0, \qquad b_0 = 0.$$

It can be shown that the conditions of Theorem 3.2 are not satisfied for $p = 3$. This then gives an implicit method defined by the recurrence relation

$$u_{j+2} = \tfrac{4}{3} u_{j+1} - \tfrac{1}{3} u_j + \tfrac{2}{3} \tau f(t_{j+2}, u_{j+2}).$$

The first characteristic polynomial of this method is $\rho(z) = 1/3 - 4z/3 + z^2$, which has the roots 1 and $\frac{1}{3}$; therefore, the method is D-stable.

We can define a class of linear m-step methods of the highest order satisfying the constraints

$$b_0 = \cdots = b_{m-1} = 0, \tag{3.41}$$

which we call the m-step *BDF methods*.

Example 3.10 (BDF methods ($m = 1, \ldots, 6$)).

$$u_{j+1} - u_j = \tau f(t_{j+1}, u_{j+1}), \tag{BDF1}$$

$$u_{j+2} - \frac{4}{3}u_{j+1} + \frac{1}{3}u_j = \frac{2}{3}\tau f(t_{j+2}, u_{j+2}), \tag{BDF2}$$

$$u_{j+3} - \frac{18}{11}u_{j+2} + \frac{9}{11}u_{j+1} - \frac{2}{11}u_j = \frac{6}{11}\tau f(t_{j+3}, u_{j+3}), \tag{BDF3}$$

$$u_{j+4} - \frac{48}{25}u_{j+3} + \frac{36}{25}u_{j+2} - \frac{16}{25}u_{j+1} + \frac{3}{25}u_j = \frac{12}{25}\tau f(t_{j+4}, u_{j+4}), \tag{BDF4}$$

$$u_{j+5} - \frac{300}{137}u_{j+4} + \frac{300}{137}u_{j+3} - \frac{200}{137}u_{j+2} + \frac{75}{137}u_{j+1} - \frac{12}{137}u_j = \frac{60}{137}\tau f(t_{j+5}, u_{j+5}), \tag{BDF5}$$

$$u_{j+6} - \frac{360}{147}u_{j+5} + \frac{450}{137}u_{j+4} - \frac{400}{147}u_{j+3} + \frac{225}{147}u_{j+2} - \frac{72}{147}u_{j+1} + \frac{10}{147}u_j = \frac{60}{147}\tau f(t_{j+6}, u_{j+6}), \tag{BDF6}$$

The m-step *BDF* methods are of the order $p = m$, and BDF1 is the Implicit Euler method. The methods listed here are *D-stable*; however, m-step BDF methods are *not* D-stable when $m \geq 7$ (Hairer et al., 2009, Theorem 3.4).

We can derive the formula for BDF2 in an alternative way. We consider the Lagrange polynomial with nodes t_j, t_{j+1}, t_{j+2}, which interpolates the exact solution $u(t_j)$, $u(t_{j+1})$, $u(t_{j+2})$. Using the Lagrange basis (3.37), then

$$u(s) \approx \mathcal{L}_{j+2}(s)u(t_{j+2}) + \mathcal{L}_{j+1}(s)u(t_{j+1}) + \mathcal{L}_j(s)u(t_j), \qquad s \in [t_0, T]. \tag{3.42}$$

In this case, the Lagrange basis (3.37) are quadratic functions and, hence, their derivatives $\frac{\mathrm{d}}{\mathrm{d}s}\mathcal{L}_{j+i}(s)$ are linear functions. We differentiate the formula (3.42) with respect to s:

$$\frac{\mathrm{d}}{\mathrm{d}s}u(s) = f(s, u(s)) \approx u(t_{j+2})\frac{\mathrm{d}}{\mathrm{d}s}\mathcal{L}_{j+2}(s) + u(t_{j+1})\frac{\mathrm{d}}{\mathrm{d}s}\mathcal{L}_{j+1}(s) + u(t_j)\frac{\mathrm{d}}{\mathrm{d}s}\mathcal{L}_j(s), \tag{3.43}$$

for $s \in [t_0, T]$. We evaluate (3.43) at the point $s = t_{j+2} = t_j + 2\tau$:

$$f(t_{j+2}, u(t_{j+2})) \approx \frac{3}{2\tau}u(t_{j+2}) - \frac{2}{\tau}u(t_{j+1}) + \frac{1}{2\tau}u(t_j). \tag{3.44}$$

Instead of the exact solutions $u(t_{j+i})$ we consider their approximations u_{j+i}. These approximations are defined by the recurrence

$$f(t_{j+2}, u_{j+2}) = \frac{3}{2\tau}u_{j+2} - \frac{2}{\tau}u_{j+1} + \frac{1}{2\tau}u_j. \tag{3.45}$$

The formulas (3.45) and (BDF2) are equivalent.

It can be shown that BDF methods from the Example 3.10 can be equivalently defined by means of *Backward Differentiation Formulas*; hence, why the methods are called *BDF*. We elaborate on the example of the method BDF2. We have shown that (3.45) and (BDF2) are equivalent; but we can alternatively define the Lagrange interpolation polynomial with nodes t_j, t_{j+1} and t_{j+2} via backward differences, using the *Newton representation* of the Lagrange interpolation polynomial. Therefore,

$$u(s) \approx u(t_{j+2}) + \frac{1}{\tau}(s - t_{j+2})\nabla u(t_{j+2}) + \frac{1}{2\tau^2}(s - t_{j+2})(s - t_{j+1})\nabla^2 u(t_{j+2})\,, \tag{3.46}$$

where ∇ is the operator of the backward difference,

$$\nabla u(t_{j+2}) = u(t_{j+2}) - u(t_{j+1}),$$

and

$$\nabla^2 u(t_{j+2}) = \nabla u(t_{j+2}) - \nabla u(t_{j+1}) = u(t_{j+2}) - 2u(t_{j+1}) - u(t_j).$$

We differentiate the formula (3.46) with respect to s and evaluate it at the point $s = t_{j+2} = t_j + 2\tau$; which yields the same recurrence formula (3.45).

3.3.4 Adaptive time-stepping

We have defined the Adams and BDF methods, and presented them as linear multistep methods of the highest order which satisfy the required constraints. The assumption (3.1) on the equidistant partition is important to this derivation. However, we provided two alternative derivations using Lagrange interpolation polynomials; namely, a *numerical integration* definition (3.38) for *Adams methods* and *numerical differentiation* (3.43) for *BDF methods*. In this respect, the main numerical technique is the construction of Lagrange interpolation polynomial; to this end, the nodal points need not be equidistant.

Adaptive time-stepping from Section 2.3 can also be applied within the framework of multistep methods. The techniques of *adaptive step refinement* are based on *adaptive interpolation*, the main principles can be seen in Deuflhard and Bornemann (2012, Section 7.4 — Adaptive Control of Order and Step Size). The adaptivity for m-step method is substantially more complicated then the adaptivity for one step methods.

MATLAB contains two functions which implement this adaptivity:

`ode113` is based on the $PECE$ implementation, where the predictor and corrector are m-step Adams-Bashfort and m-step Adams-Moulton methods, respectively, where the number of steps m can be changed adaptively in the range $m = 1, \ldots, 13$

`ode15s` is based on the implementation of m-step BDF, where the number of steps m can be changed adaptively in the range $m = 1, \ldots, 5$

CHAPTER 4

Dynamical systems

We consider an autonomous ODE, see Definition 1.5, and the corresponding initial value problem

$$x' = f(x), \qquad x(0) = x_0. \tag{4.1}$$

Let $f \in C^1(D, \mathbb{R}^n)$, where $D \subset \mathbb{R}^n$ is an open set containing x_0 and let ϕ be the corresponding flow of the vector field f, cf. Definition 1.17. The vector function

$$u(t) = \phi(t, x_0) \tag{4.2}$$

solves the initial value problem on the maximal solution interval.

We have interpreted the original initial value problem (IVP) and the autonomous initial value problem (4.1) as a *model of evolution* in a state space; namely in $\mathbb{R}^n$. The models of evolution are called in general *dynamical systems*; see, e.g., Katok and Hasselblatt (1995). We restrict ourself to the dynamical systems systems which are defined (modelled) by the initial problems (IVP) and (4.1), respectively.

The following remark may be skipped.

Remark 4.1. The characteristic feature of the flow ϕ is the following property:

$$\phi(t_1 + t_2, x_0) = \phi(t_2, \phi(t_1, x_0))$$

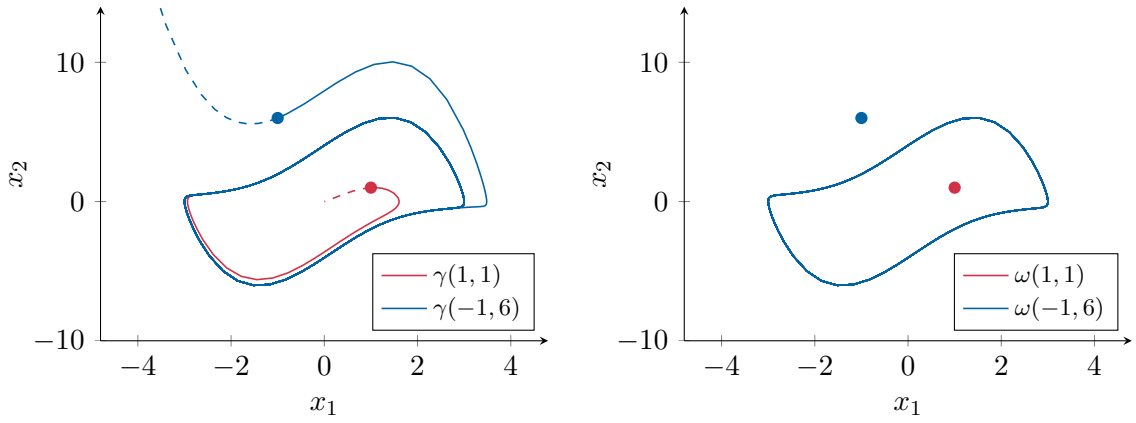
for $t_1 \in \mathbb{R}$ and $t_2 \in \mathbb{R}$. Consider the linear dynamical system $x' = ax$, $x(0) = x_0$. For the corresponding flow it means that $\mathrm{e}^{a(t_1+t_2)}\, x_0 = \mathrm{e}^{at_1}\, \mathrm{e}^{at_2}\, x_0$. We say that the operator ϕ is a representation of the additive group of the state space.

There is another view: Operator ϕ is the representation of a one-parameter group of diffeomorphisms where time t is the parameter. Let Ω be open subdomain $\Omega \subset D$; then, we are interested in changes of Ω in time; namely, the mapping $\Omega \longmapsto \phi(t, \Omega) \equiv \Omega_t$, $t \geq 0$. Denote $\mathrm{meas}(\Omega)$ and $\mathrm{meas}(\Omega_t)$ as the corresponding Lebesgue measures; then, the question is what is the rate $\mathrm{meas}(\Omega)/\mathrm{meas}(\Omega_t)$?

In case of linear dynamical systems the answer was formulated by J. Liouville (1838): Consider $A \in \mathbb{R}^{n\times n}$, $x' = Ax$. We will learn in the sequel, see (4.11), that $\phi(t, x) = \mathrm{e}^{tA}\, x$; then, the rate is given by the formula

$$\frac{\mathrm{meas}(\Omega)}{\mathrm{meas}(\phi(t, \Omega))} = \det \mathrm{e}^{tA} = \mathrm{e}^{t\,\mathrm{tr}(A)},$$

where $\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii}$ is the trace of matrix A.



(a) Orbits

(b) Limit sets

Figure 4.1: Van der Pol oscillator — orbits and limit sets for $a = 1.1$

4.1 Asymptotics of the time evolution

We first introduce several notions from the theory of dynamical systems.

Definition 4.2 (Orbit = Phase curve)**.** Let $x_0 \in D$; then, the set

$$\gamma(x_0) = \bigcup_{t \in (t^-(x_0), t^+(x_0))} \phi(t, x_0)$$

is called the *orbit* of the point x_0.

Definition 1.12 of the phase curve and Definition 4.2 of the orbit are equivalent.

Definition 4.3 (The positive & negative orbit)**.** Let $x_0 \in D$; then, the sets

$$\gamma^+(x_0) = \bigcup_{t \in [0, t^+(x_0))} \phi(t, x_0), \qquad \gamma^-(x_0) = \bigcup_{t \in (t^-(x_0), 0]} \phi(t, x_0)$$

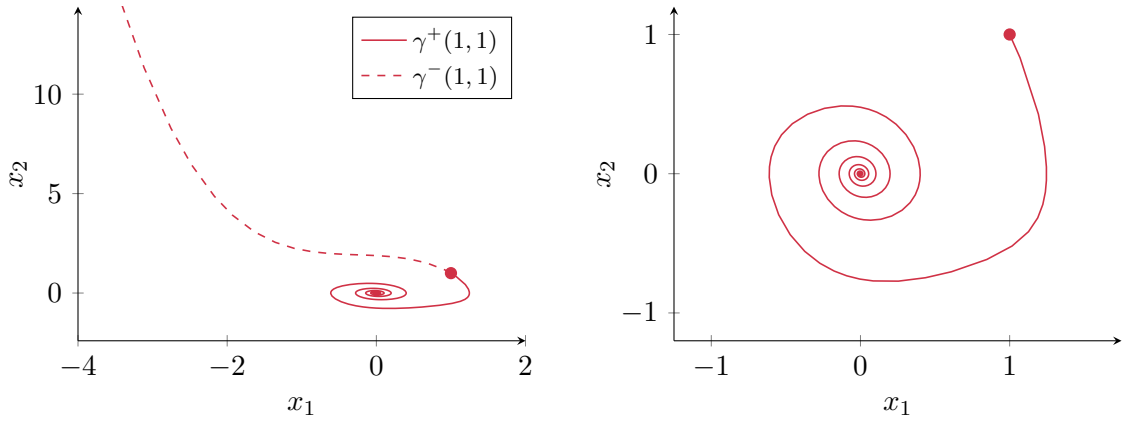
are the *positive* and *negative orbit* of the point x_0, respectively.

Example 4.1 (Van der Pol oscillator). Let us consider the dynamical system

$$\begin{aligned} x_1' &= x_2 \\ x_2' &= -x_1 + 2ax_2 - x_1^2 x_2, \end{aligned}$$

where $a \in \mathbb{R}$ is a parameter.

In Figure 4.1 we analyse the Van der Pol oscillator for the parameter $a = 1.1$. Figure 4.1(a) displays the orbits $\gamma(1,1)$ and $\gamma(-1,6)$, with the positive orbits $\gamma^+(1,1)$ and $\gamma^+(-1,6)$ displayed as a solid line and the negative orbits $\gamma^-(1,1)$ and $\gamma^-(-1,6)$ are plotted with dashed lines. Note that the orbit $\gamma^-(1,1)$ contains the origin $(0,0)$, which is the unstable stationary state (cf. Section 4.2, Definition 4.11).



(a) Positive and negative orbit

(b) Positive orbit

Figure 4.2: Van der Pol oscillator — orbit for $a = -0.1$. Note the ω-limit is a single point: $\omega(1,1) = \{(0,0)\}$

Definition 4.4 (ω-limit set)**.** Let $x_0 \in D$; then the set

$$\omega(x_0) = \bigcap_{\tau \geq 0} \overline{\gamma^+(\phi(\tau, x_0))}$$

is called the *ω-limit set* of the orbit $\gamma^+(x_0)$.

Definition 4.5 (α-limit set)**.** Let $x_0 \in D$; then, the set

$$\alpha(x_0) = \bigcap_{\tau \leq 0} \overline{\gamma^-(\phi(\tau, x_0))}$$

is called the *α-limit set* of the orbit $\gamma^-(x_0)$.

Figure 4.1(b) displays the ω-limit sets of the points $(1,1)$ and $(-1,6)$; i.e., $\omega(1,1)$ and $\omega(-1,6)$. In this particular case, $\omega(1,1) = \omega(-1,6)$. The set $\alpha(1,1)$ reduces to just the single point, $\alpha(1,1) = \{(0,0)\}$ (not shown in Figure 4.1(b)), which is the unstable steady state, cf. Section 4.2.

Remark 4.6. The object $\omega(1,1)$ is well defined mathematically. In Figure 4.1(b), you can just see the *numerical approximation* of $\omega(1,1)$. This particular $\omega(1,1)$ is related to a periodic solution of the initial value problem (4.1), where we had to guess the period; which can be done via a trial and error procedure. Nevertheless, there exists numerical methods and specialised software[1] which rigorously approximate periodic solutions and compute the period.

4.2 The steady state

Figure 4.2 shows an example of the ω-limit set consisting of just a single point. This is the case of the steady state.

[1] Matcont. https://matcont.sourceforge.io/

Definition 4.7 (The steady state)**.** Let $x^* \in D$; then, if $f(x^*) = 0 \in \mathbb{R}^n$ we say that x^* is the *steady state*.

Remark 4.8 (The steady state: synonyms). *The steady state = stationary point = equilibrium = stationary solution*.

Remark 4.9. Let $x^* \in D$ and $f(x^*) = 0$; then, the solution of the initial value problem (4.1) is constant in time:

$$u(t) = \phi(t, x^*) = x^*, \qquad t \in \mathbb{R}.$$

Definition 4.10 (Stability, Asymptotic stability = A-stability)**.** Let $x^* \in D$ and $f(x^*) = 0$; then, we say that the steady state $x^* \in D$ is *stable* provided that for all $\varepsilon > 0$ there exists a $\delta > 0$ such that for all $x \in B_\delta(x^*) \coloneqq \{x \in D : \|x - x^*\| < \delta\}$ it holds that

$$\|\phi(t, x) - x^*\| < \varepsilon \qquad \text{for all } t \geq 0.$$

If, additionally, there exists an $r > 0$ such that for all $x \in B_r(x^*) \coloneqq \{x \in D : \|x - x^*\| < r\}$ it holds that

$$\lim_{t \to +\infty} \phi(t, x) = x^*,$$

then x^* is *asymptotically stable* or, simply, *A-stable*.

Definition 4.11 (Instability)**.** Let $x^* \in D$ and $f(x^*) = 0$. We say that the steady state x^* is *unstable* if it is not stable; i.e., there exists a $\varepsilon > 0$ such that for all $\delta > 0$ it holds that there exists an $x \in B_\delta(x^*) \coloneqq \{x \in D : \|x - x^*\| < \delta\}$ and $t > 0$ such that

$$\|\phi(t, x) - x^*\| \geq \varepsilon.$$

Example 4.2 (Linear dynamical system). We consider the linear dynamical system

$$\begin{aligned} x_1' &= x_2 \\ x_2' &= -x_1. \end{aligned}$$

In matrix notation, we can write this as

$$x' = Ax, \quad \text{where } A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

It can be shown that

$$\phi(t, x) = \mathrm{e}^{tA} x = \begin{bmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{bmatrix} x.$$

Hence the orbit $\gamma(x)$ of a point $x \in \mathbb{R}^2$ is a circle with centre at the origin $(0, 0)$ and radius $\|x\|$. Therefore, $x^* = (0, 0)$ is the steady state, which is not A-stable.

Our aim is to formulate a sufficient condition for the A-stability. To this end we need to recall the following definitions.

Remark 4.12 (Spectrum of a real matrix). Let $A \in \mathbb{R}^{n\times n}$ be a real matrix; then, the set

$$\sigma(A) = \{\lambda \in \mathbb{C} : \det(\lambda I - A) = 0\}$$

is called the *spectrum* of the matrix A. The elements of $\sigma(A)$ are called *eigenvalues* which, in general, we denote them by λ; hence $\lambda \in \sigma(A)$, $\lambda \in \mathbb{C}$. We write $\lambda = \lambda(A)$ to emphasize the fact that λ is an eigenvalue of the particular matrix A. Additionally, complex eigenvalues appear in couples; i.e., if $\lambda \in \sigma(A)$ then $\overline{\lambda} \in \sigma(A)$.

Definition 4.13. Let $A \in \mathbb{R}^{n\times n}$ and $\lambda \in \sigma(A)$; then, $\Re(\lambda)$ denotes the real part of the eigenvalue λ.

Theorem 4.14 (Lyapunov, 1892). *Let $f \in C^1(D, \mathbb{R}^n)$, $x^* \in D$, $f(x^*) = 0$, and*

$$A = \left(\frac{\partial f_i}{\partial x_j}(x^*)\right)_{i,j=1}^{n} \in \mathbb{R}^{n\times n}$$

be the Jacobian of f at the point x^*. If

$$\max_{\lambda\in\sigma(A)} \Re(\lambda) < 0, \tag{4.3}$$

then x^* is A-stable. If there exists a $\lambda \in \sigma(A)$, $\Re(\lambda) > 0$, then x^* is unstable.

Proof. Deuflhard and Bornemann (2012, Theorem 3.30). □

In Example 4.1, there is only one steady state $x^* = (0, 0)$, which is available for any value of the parameter a. The Jacobian at the point $x^* = (0, 0)$ is

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 2a \end{bmatrix}. \tag{4.4}$$

In order to discuss stability, we look for roots of the quadratic equation $\lambda^2 - 2a\lambda + 1 = 0$. For $a = -0.1$, cf. Figure 4.2 the steady state $x^* = (0, 0)$ is A-stable as $\sigma(A) = \{-0.1 \pm i0.995\}$; whereas, for $a = 1.1$, cf. Figure 4.1 the steady state $x^* = (0, 0)$ is unstable as $\sigma(A) = \{1.5583, 0.6417\}$.

We will not give the proof of Theorem 4.14, but just outline the idea which is based on the *principle of linearised stability*. Let the assumptions of Theorem 4.14 be satisfied. We consider the dynamical system and Taylor expansion of the function f at the steady state x^*:

$$x' = f(x) = f(x^*) + A(x - x^*) + g(x - x^*) = A(x - x^*) + g(x - x^*), \tag{4.5}$$

where A is the Jacobian. Note that $f(x^*) = 0$. The vector function $g : \mathbb{R}^n \to \mathbb{R}^n$ describes the higher order terms of the Taylor expansion $g(x - x^*) = o(x - x^*)$.

We consider the linear change of coordinates $x = x^* + y$; then, the transformed dynamical system reads as

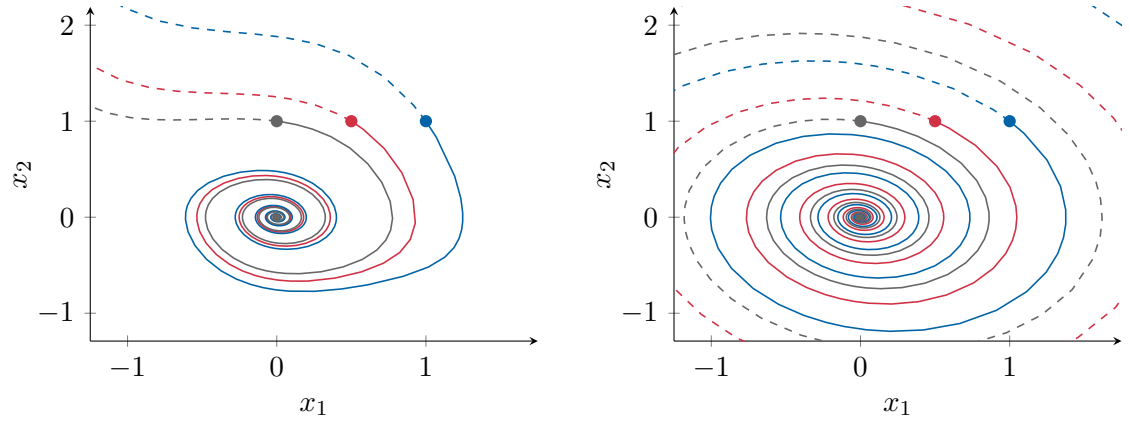
$$y' = f(x^* + y) = Ay + g(y), \tag{4.6}$$

where $y^* = 0 \in \mathbb{R}^n$ is the steady state and $g(y) = o(y)$.

We now consider the dynamical system which stems from (4.6) neglecting the higher order terms $g(y) = o(y)$:

$$z' = Az, \tag{4.7}$$

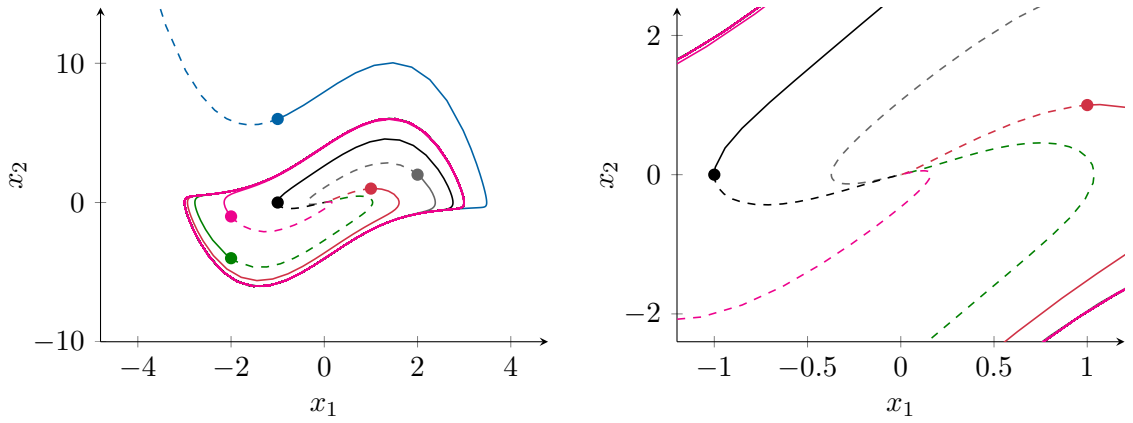
where $z^* = 0 \in \mathbb{R}^n$ is obviously the steady state.



(a) Van der Pol oscillator

(b) Linearised dynamical system

Figure 4.3: Van der Pol oscillator — phase portraits for system and linearised system for $a = -0.1$ around A-stable steady state $x^* = (0, 0)$



(a) Phase portrait

(b) Detail in neighbourhood of steady state

Figure 4.4: Van der Pol oscillator — phase portraits for $a = 1.1$ around unstable steady state $x^* = (0, 0)$

Remark 4.15 (Linearisation). The transition from the dynamical system (4.5) with the steady state x^* to the linear dynamical system (4.7) with the steady state $0 \in \mathbb{R}^n$ is called *linearisation*.

It is intuitive that the solution of the initial value problems (4.1), i.e.

$$x' = A(x - x^*) + g(x - x^*), \qquad x(0) = x_0, \tag{4.8}$$

and the solution of the linearized initial value problem

$$z' = Az, \qquad z(0) = z_0, \tag{4.9}$$

are "similar" provided that $x_0 \approx x^*$ and $z_0 \approx 0 \in \mathbb{R}^n$, respectively.

In Figure 4.3 we compare the phase portraits for the van der Pol oscillator from Example 4.1 (Figure 4.3(a)) to the linearised dynamical system (4.9) with matrix (4.4) (Figure 4.3(b))

for $a = -0.1$. Note, that $\sigma(A) = \{-0.1 \pm i\, 0.995\}$. It is important to realise that the "similarity" of both dynamical systems (4.8) and (4.9) is a local property; namely, we have to restrict to sufficiently small neighbourhoods of the stationary states x^* and $0 \in \mathbb{R}^n$. In Figure 4.4 we consider Example 4.1 for $a = 1.1$. Figure 4.4(a) displays the phase portrait, and demonstrates a limit cycle which is part of the ω-limit set; cf. Figure 4.1(a). Figure 4.4(b) displays the detail of the phase portrait in the neighbourhood of the unstable steady state.

By the end of this section we will have a *mathematical* formulation of the relationship between (4.8) and (4.9); cf. Theorem 4.22. We will first develop some preliminary results. Namely, we give an important formula for solution of the initial value problem (4.9). Let the state variable be denoted by $x \in \mathbb{R}^n$; then, we solve the initial value problem

$$x' = Ax, \qquad x(0) = x_0, \tag{4.10}$$

where $A \in \mathbb{R}^{n\times n}$. We will show that the flow of the vector field

$$t \in \mathbb{R}, x_0 \in \mathbb{R}^n \longmapsto \phi(t, x_0) \in \mathbb{R}^n$$

is defined by the explicit formula

$$\phi(t, x_0) = \mathrm{e}^{tA}\, x_0, \tag{4.11}$$

where $\mathrm{e}^{tA} \in \mathbb{R}^{n\times n}$ is a matrix.

Theorem 4.16 (The matrix exponential)**.** *Let $A \in \mathbb{R}^{n\times n}$, $t \in \mathbb{R}$. The* matrix exponential *is defined by the power series*

$$\mathrm{e}^{tA} = \sum_{k=0}^{+\infty} \frac{(tA)^k}{k!}. \tag{4.12}$$

The series (4.12) converges uniformly *and* absolutely *on each interval $-T \le t \le T$, $T > 0$ and the following holds:*

1. *If $AB = BA$ then $\mathrm{e}^{t(A+B)} = \mathrm{e}^{tA}\,\mathrm{e}^{tB}$.*
2. *Let $A \sim B$ (matrix similarity), i.e., $\exists Z \in \mathbb{R}^n$, $\det Z \neq 0$ such that $ZA = BZ$; then*
$$Z\,\mathrm{e}^{tA} = \mathrm{e}^{tB}\,Z.$$
3. $$\frac{\mathrm{d}}{\mathrm{d}t}\,\mathrm{e}^{tA} = A\,\mathrm{e}^{tA}$$

Proof. Following the *Weierstrass criterion*, the number series

$$\sum_{k=0}^{+\infty} \frac{(T\|A\|)^k}{k!} = \mathrm{e}^{T\|A\|} < +\infty$$

is the *majorant* of the series (4.12) for $-T \le t \le T$. The remaining properties *1-3* follows from the definition of (4.12). □

From Theorem 4.16 it follows that

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathrm{e}^{tA}\,x_0\right) = A\,\mathrm{e}^{tA}\,x_0;$$

hence, $x = \mathrm{e}^{tA}\,x_0$ satisfies the initial value problem (4.10), which confirms the result (4.11).

Remark 4.17 (Complex matrix exponential). If $A \in \mathbb{C}^{n\times n}$ is a complex matrix then we can extend definition (4.12) naturally to the complex field.

Remark 4.18. The exponential $\mathrm{e}^{tA} \in \mathbb{R}^{n\times n}$ can be approximated numerically (Higham, 2008). However, this is quite costly.

Theorem 4.19. *The steady state $x^* = 0 \in \mathbb{R}^n$ of the linear dynamical system (4.10) is asymptotically (A-stable) if and only if*

$$\max_{\lambda\in\sigma(A)} \Re(\lambda) < 0. \tag{4.13}$$

Proof. Deuflhard and Bornemann (2012, Theorem 3.23). □

We already knew that the spectral property (4.13) is a sufficient condition for A-stability, cf. Theorem 4.14. Example 4.2 provides a counterexample that (4.13) is also the necessary condition for A-stability. Nevertheless, the above quoted proof uses tools of linear algebra, namely the transformation of $A \in \mathbb{R}^{n\times n}$ to *Jordan canonical form*.

Corollary 4.20. *Let the spectral property (4.13) be satisfied; then for all $x \in \mathbb{R}^n$*

$$e^{tA}x \longrightarrow 0 \in \mathbb{R}^n. \tag{4.14}$$

Compare Definition 4.10 of a asymptotically stable steady state with the property (4.14). Referring to Definition 4.10, the second property should hold for a chosen positive r; note, that the linearity of the dynamical system (4.10) implies that this property holds for any chosen positive r.

Definition 4.21. Consider a mapping $h : U \subset \mathbb{R}^n \to V \subset \mathbb{R}^n$ where U and V are open subsets. The mapping is called a *homeomorphism* if

a) h is bijection and

b) there exists the continuous inverse $h^{-1} : V \subset \mathbb{R}^n \to U \subset \mathbb{R}^n$.

In other words, the image V is a continuous deformation of the preimage U. We now consider the linearisation again, see Remark 4.15; namely, we consider the relationship of both the original dynamical system (4.8) and the linearized problem(4.9).

Theorem 4.22 (Hartman-Grobman)**.** *There exists a homeomorphism $h : U \subset \mathbb{R}^n \to V \subset \mathbb{R}^n$ with the properties*

- *U and V are open sets containing $0 = z^*$ and x^*, respectively,*
- *$h(0) = x^*$, $h^{-1}(x^*) = z^* = 0$,*
- *$\forall z \in U$*

$$\phi(t, h(z)) = h(\mathrm{e}^{tA}\,z), \qquad 0 \le t < +\infty,$$

- $\forall x \in V$

$$h^{-1}(\phi(t,x)) = \mathrm{e}^{tA}\, h^{-1}(x), \qquad 0 \le t < +\infty.$$

Proof. Katok and Hasselblatt (1995, Theorem 6.3.1) □

The above homeomorphism maps

- the positive orbit of point $z \in U$ on the positive orbit of point $h(z) \in V$,
- the positive orbit of point $x \in V$ on the positive orbit of point $h^{-1}(x) \in U$.

Let us comment on Figure 4.3 in view of Theorem 4.22. There exists a homeomorphism h, i.e. a continuous deformation of coordinates, that transforms the picture Figure 4.3(a) to the picture Figure 4.3(b) and vice versa.

4.3 Discrete-time dynamical systems

Let us consider the Runge-Kutta methods (RK), see Definition 2.28, where

$$t \in J, x \in D, \tau \ge 0 \longmapsto \psi(t+\tau, t, x) \in \mathbb{R}^n$$

is the discrete flow of the vector field. As we are investigating the autonomous ODE we consider the discrete flow of the vector field independent of t; i.e.,

$$x \in D, \tau \ge 0 \longmapsto \psi(\tau, x) \in \mathbb{R}^n \tag{4.15}$$

Specifically, we consider the Runge method as an example. We set $\kappa_1 = f(x)$, $\kappa_2 = f(x + \frac{\tau}{2}\kappa_1)$, and define

$$\psi(\tau, x) \equiv x + \tau\kappa_2 = x + \tau f(x + \frac{\tau}{2} f(x)). \tag{4.16}$$

From the initial condition $x \in \mathbb{R}^n$ at time $0 \in \mathbb{R}$ we move to the new state $\psi(\tau, x) \in \mathbb{R}^n$ at time $\tau \in \mathbb{R}$. Hence,

$$0 \longmapsto \tau, \quad x \longmapsto \psi(x). \tag{4.17}$$

The aim is to model the development of the state variable in discrete time snapshots. We consider the recurrence

$$x \longmapsto \psi(\tau, x) \longmapsto \psi^2(\tau, x) \longmapsto \cdots \longmapsto \psi^j(\tau, x) \longmapsto \ldots, \tag{4.18}$$

where $\psi^j(\tau, x)$ is defined recursively via superposition of a mapping:

$$\psi^j(\tau, x) = \psi(\tau, \psi^{j-1}(\tau, x)), \qquad j \in \mathbb{N}. \tag{4.19}$$

The development of the initial condition $(0 \in \mathbb{R}, x \in \mathbb{R}^n)$ is defined by the *iterations*

$$j \in \mathbb{N}_0 \longmapsto t_j = \tau j, \qquad u_j = \psi^j(\tau, x). \tag{4.20}$$

This defines the sequence of discrete times and states

$$\{t_j\}_{j=0}^{+\infty}, \qquad \{u_j\}_{j=0}^{+\infty}. \tag{4.21}$$

Let us recall the convergence analysis from Theorem 2.23 for the one-step methods and Theorem 3.7 for the multistep methods. We considered a finite interval $[t_0, T]$ and assumed that $T < t^+(t_0, x_0)$. For an equidistant partition it implied that the approximating sequences $\{t_j\}_{j=0}^N$, $\{u_j\}_{j=0}^N$ have the finite length $N = {}^{(T-t_0)}/_{\tau}$.

We aim, for a given step size τ, to investigate the sequences (4.21) up to the "very end", which is ω-limit point $\omega(x)$. More precisely, we compute a numerical approximation of this limit point.

Let us consider the most simple ω-limit point, which is the steady state. We show that the steady state is related to the fixed point of the numerical method.

Proposition 4.23. *Let $x^* \in D$, $\tau > 0$. If $f(x^*) = 0$ then*

$$x^* = \psi(\tau, x^*); \tag{4.22}$$

hence, x^* is a fixed point *of the mapping $x \longmapsto \psi(\tau, x)$.*

Proof. We consider only the method (4.16), i.e. the iterations

$$x \longmapsto \psi(\tau, x) \equiv x + \tau f(x + \frac{\tau}{2} f(x)).$$

If $f(x^*) = 0$ then $\tau f(x^* + \frac{\tau}{2} f(x^*)) = 0$ and hence

$$x^* = x^* + \tau f(x^* + \frac{\tau}{2} f(x^*)).$$

The proof can be generalized to all Runge-Kutta methods (RK), see (2.64) and also to linear m-step methods. □

Let $x^* \in D$ be an A-stable steady state, see Definition 4.10. If the initial condition $x \in D$ is sufficiently close to x^* then $\phi(t, x) \to x^*$. If we consider the orbit $\gamma^+(x)$ of the point x we conclude that $\omega(x) = x^*$. *Each* numerical method for solving the initial value problem (4.1) generates a sequence of iterations $x \longmapsto \psi(\tau, x)$, see (4.20) and (4.21). The sequence (4.21) is a discrete approximation of the orbit $\gamma^+(x)$. According to Proposition 4.23, the above mentioned x^* is the fixed point $x^* = \psi(\tau, x^*)$ of the mapping $x \longmapsto \psi(\tau, x)$. Performance of the appropriate numerical method depends on the step size τ.

We now define the stability and instability of a fixed point $x^* = \psi(\tau, x^*)$. Compare the following Definition 4.24 and Definition 4.25 with Definition 4.10 and Definition 4.11, respectively.

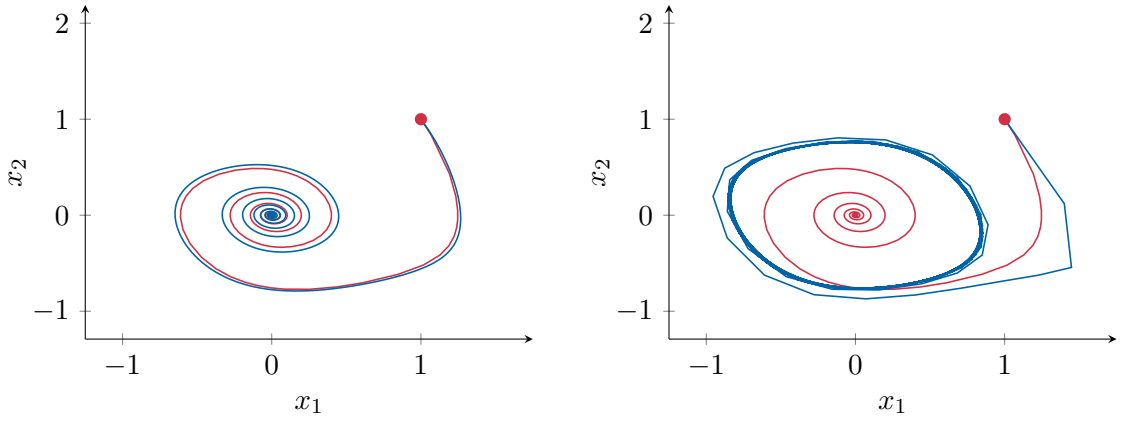
Definition 4.24 (A-stability of a fixed point)**.** Let $x^* = \psi(\tau, x^*) \in D$ be a fixed point of the mapping $x \in D \longmapsto \psi(\tau, x) \in \mathbb{R}^n$ for a given $\tau > 0$. We say that the fixed point $x^* = \psi(\tau, x^*)$ is *stable* provided that for all $\varepsilon > 0$ there exists a $\delta > 0$ such that for all $x \in B_\delta(x^*) := \{x \in D : \|x - x^*\| < \delta\}$ it holds that

$$\|\psi^j(\tau, x) - x^*\| < \varepsilon \qquad \text{for all } j \in \mathbb{N}_0.$$

If, additionally, there exists an $r > 0$ such that for all $x \in B_r(x^*) := \{x \in D : \|x - x^*\| < r\}$ it holds that

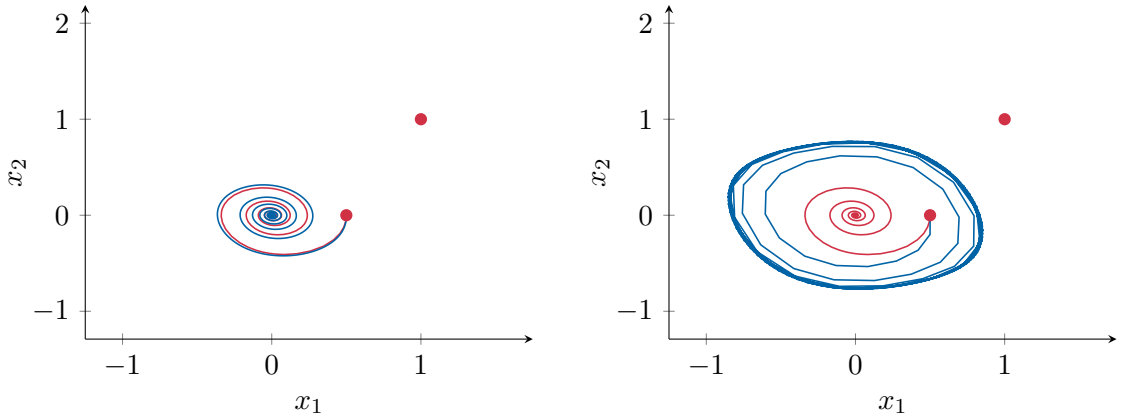
$$\psi^j(\tau, x) \to x^*, \qquad \text{for } j \to +\infty,$$

then x^* is an *A-stable* fixed point.



(a) $\tau = 0.05$ (b) $\tau = 0.4$

Figure 4.5: Van der Pol oscillator, $a = -0.1$ — positive orbit for $(1, 1)$ compared to numerical approximation using Euler



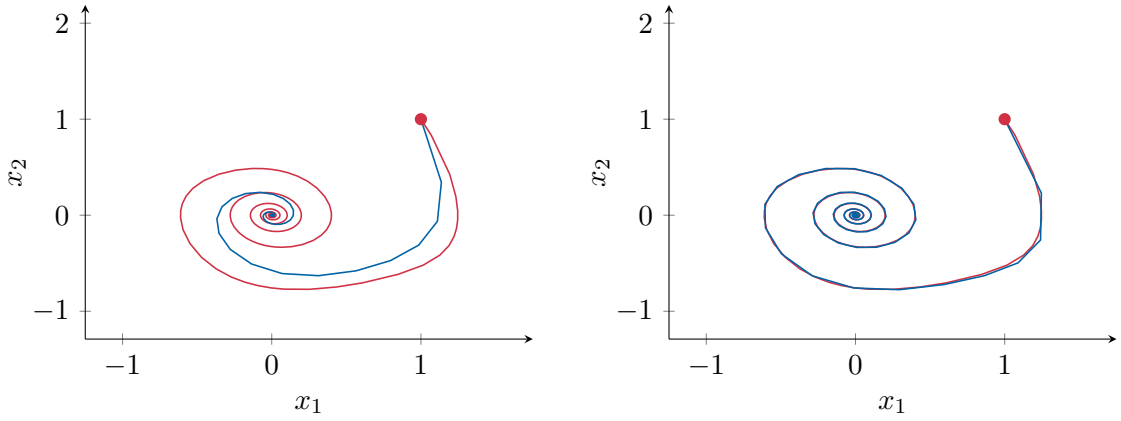
(a) $\tau = 0.05$ (b) $\tau = 0.4$

Figure 4.6: Van der Pol oscillator, $a = -0.1$ — positive orbit for $(0.5, 0)$ compared to numerical approximation using Euler

Definition 4.25 (Instability)**.** Let $x^* = \psi(\tau, x^*) \in D$ be a fixed point of the mapping $x \in D \longmapsto \psi(\tau, x) \in \mathbb{R}^n$ for a given $\tau > 0$. We say that the fixed point $x^* = \psi(\tau, x^*)$ is *unstable* if it is not stable; i.e., there exists a $\varepsilon > 0$ such that for all $\delta > 0$ it holds that there exists an $x \in B_\delta(x^*) := \{x \in D : \|x - x^*\| < \delta\}$ and $j > 0$ such that

$$\|\phi^j(\tau, x) - x^*\| \geq \varepsilon.$$

We illustrate the above notions in Figure 4.5 and Figure 4.6 where we consider the dynamical system from Example 4.1 with $a = -0.1$ which has a A-stable steady state $x^* = 0 \in \mathbb{R}^2$. We consider two orbits, $\gamma^+(1, 1)$ and $\gamma^+(0.5, 0)$ along with their numerical approximation by the Euler method; i.e., by the iterations $x \longmapsto \psi(\tau, x) \equiv x + \tau f(x)$. The approximation depends on step size τ. If τ is comparatively small namely $\tau = 0.05$ then the fixed point



(a) Implicit Euler (b) Crank-Nicholson

Figure 4.7: Van der Pol oscillator, $a = -0.1$ — positive orbit for $(1, 1)$ compared to numerical approximation using implicit one-step methods

$x^* = \psi(\tau, x^*)$ is classified as an *A-stable fixed point* of the iterations $x \longmapsto \psi(\tau, x)$ according to Definition 4.24; cf., the numerical solution in Figure 4.5(a) and Figure 4.6(a). For τ comparatively large, namely $\tau = 0.4$, then the fixed point $x^* = \psi(\tau, x^*)$ of the iterations $x \longmapsto \psi(\tau, x)$ is classified as an *unstable fixed point* according to Definition 4.25; cf., the numerical solution in Figure 4.5(b) and Figure 4.6(b). These iterations are actually attracted to the so called *limit cycle*. This invariant object disappears if the step size is sufficiently small (e.g., $\tau = 0.05$). This analysis is due to experimental observations. The instability of a fixed point is often manifested by *chaotic behaviour* of the iterations.

For the above we stated that $\tau = 0.4$ is comparatively large. If we use an implicit method, as in Figure 4.7, then the step size $\tau = 0.4$ is adequate to declare that the corresponding fixed point is actually an A-stable fixed point.

Remark 4.26. The A-stability and the instability of a fixed point is a *qualitative property*. Figure 4.7 is instructive — we are not primarily interested in the orbit of the point (1,1) and its numerical approximation (i.e., the relationship between the blue and the black lines) in *detail*. Instead we interested in the *asymptotic tendency* and the *affinity*. On the other hand, from Theorem 2.23 and Theorem 3.7 we know that if we restrict ourself to an interval $0 \leq t \leq T$ of a finite length T we can estimate the error of the exact and the numerical solution with respect to the step size $\tau > 0$.

CHAPTER 5

Domain of stability & stiff systems

We consider the the initial value problem (4.1) for an autonomous ODE. Let $x^* \in D$ be a steady state; i.e., $f(x^*) = 0$. Let us denote by

$$A = \left(\frac{\partial f_i}{\partial x_j}(x^*) \right)_{i,j=1}^{n} \in \mathbb{R}^{n \times n} \tag{5.1}$$

the Jacobian at the point x^*. Let us assume that

$$\max_{\lambda \in \sigma(A)} \Re(\lambda) < 0. \tag{5.2}$$

As a consequence of Theorem 4.14, the steady state x^* is A-stable.

We consider the approximation of the initial value problem (4.1) via a chosen discrete dynamical system $x \longmapsto \psi(\tau, x)$. Due to Proposition 4.23, the steady state x^* is a fixed point of the iterations $x^* = \psi(\tau, x^*)$. Is this fixed point A-stable? By Definition 4.24 we know the mapping, and hence stability, depends on step size τ. The question is whether there exists any recommendations for the choice of $\tau > 0$?

In order to simplify the analysis we consider the linearisation, see Remark 4.15. Namely, we consider initial value problem

$$x' = Ax, \qquad x(0) = x_0, \tag{5.3}$$

where A is the Jacobian (5.1). We stress that the assumption (5.2) is assumed to hold, and hence, the origin $0 \in \mathbb{R}^n$ is an A-stable steady state.

Due to Corollary 4.20 it holds that for all $x(0) = x_0 \in \mathbb{R}^n$

$$e^{tA} x_0 \longrightarrow 0 \in \mathbb{R}^n \qquad \text{for } t \to +\infty. \tag{5.4}$$

We will consider a numerical solution of the initial value problem (4.1). Following Proposition 4.23, each numerical solution can be interpreted as a discrete dynamical system; i.e., the iterations $x \longmapsto \psi(\tau, x)$. The steady state $0 \in \mathbb{R}^n$ is interpreted as the fixed point $0 = \psi(\tau, 0)$ of the iterations $x \longmapsto \psi(\tau, x)$. We will check whether for all $x(0) = x_0 \in \mathbb{R}^n$ it holds that

$$\psi^j(\tau, x_0) \longrightarrow 0 \in \mathbb{R}^n \qquad \text{for } j \to +\infty. \tag{5.5}$$

In other words, whether $0 = \psi(\tau, 0)$ is an A-stable fixed point of the iterations $x \longmapsto \psi(\tau, x)$. In general it will be true if the step size $\tau > 0$ will be sufficiently small. We will analyse how large the step size τ should be in order to preserve the A-stability of the fixed point.

This is essentially the meaning of the notion of the *domain of stability*. In Section 5.1 and Section 5.2 we will define the domains of stability for the Runge-Kutta (RK) and linear multistep methods, respectively.

In Section 5.3 we will talk about so called *stiff problems*. The definition is rather vague, but is related to a class of ODE's. Adequate solvers for stiff problems require either extremely small step size τ or the application of implicit methods.

5.1 Domain of stability: one-step method

We consider five one-step methods

0	0		0	0	0		0	0	0		1	1		0	0	0
	1		1/2	1/2	0		1	1	0			1		1	1/2	1/2
				0	1			1/2	1/2						1/2	1/2
Euler			Runge				Heun				Impl. Euler				Crank-Nicholson				

We will solve the linear dynamical system (5.3) where $A \in \mathbb{R}^{n\times n}$. We define one step of the above methods:

Example 5.1 (Euler for linearised ODE). We evaluate the relevant Butcher tableau, see Definition 2.28. We exploit the fact that the right-hand side f does not depend on time; hence, $\kappa_1 = f(x) = Ax$. Then $\psi(\tau, x) = x + \tau Ax = (I + \tau A)x$; therefore,

$$x \longmapsto \psi(\tau, x) \equiv (I + \tau A)x \tag{5.6}$$

is the formula for one iteration of the Euler method.

Example 5.2 (Runge for linearised ODE). $\kappa_1 = f(x) = Ax$, $\kappa_2 = f(x + \frac{\tau}{2}\kappa_1) = A(x + \frac{\tau}{2}Ax)$, and $\psi(\tau, x) = x + \tau(Ax + \frac{\tau}{2}A^2x) = (I + \tau A + \frac{\tau^2}{2}A^2)x$; therefore,

$$x \longmapsto \psi(\tau, x) \equiv \left(I + \tau A + \frac{\tau^2}{2}A^2\right)x. \tag{5.7}$$

Example 5.3 (Heun for linearised ODE). $\kappa_1 = f(x) = Ax$, $\kappa_2 = f(x+\tau\kappa_1) = A(x+\tau Ax)$, and $\psi(\tau, x) = x + \frac{\tau}{2}Ax + \frac{\tau}{2}(A(x + \tau Ax)) = (I + \tau A + \frac{\tau^2}{2}A^2)x$. Therefore,

$$x \longmapsto \psi(\tau, x) \equiv \left(I + \tau A + \frac{\tau^2}{2}A^2\right)x. \tag{5.8}$$

In fact, the formulae (5.7) and (5.8) are the same as the linearisation of an autonomous ODE for both methods are identical.

Let us recall the exponential of a matrix, see (4.11),

$$e^{tA}x = \left(I + \tau A + \frac{\tau^2}{2}A^2 + \ldots + \frac{\tau^j}{j!}A^j + \ldots\right)x.$$

The formulae formulae (5.7) and (5.8) correspond to the second order approximation of the above exponential. By analogy,

Example 5.4 (Classical Runge-Kutta for linearised ODE).

$$x \longmapsto \psi(\tau, x) \equiv \left(I + \tau A + \frac{\tau^2}{2} A^2 + \frac{\tau^3}{6} A^3 + \frac{\tau^4}{24} A^4 \right) x. \tag{5.9}$$

Linearisation of the classical RK and the $^3/_8$-rule, see Example 2.8, are the same.

Next we consider the implicit methods:

Example 5.5 (Implicit Euler for linearised ODE). We evaluate the relevant Butcher tableaux:

$$\kappa_1 = f(x + \tau\kappa_1) = A(x + \tau\kappa_1) \quad \Longrightarrow \quad (I - \tau A)\kappa_1 = Ax \quad \Longrightarrow \quad \kappa_1 = (I - \tau A)^{-1} Ax.$$

Then,

$$\psi(\tau, x) = x + \tau(I - \tau A)^{-1} Ax = (I - \tau A)^{-1} \left((I - \tau A)x + \tau Ax \right) = (I - \tau A)^{-1} x;$$

therefore,

$$x \longmapsto \psi(\tau, x) \equiv (I - \tau A)^{-1} x. \tag{5.10}$$

Example 5.6 (Crank-Nicholson for linearised ODE). Evaluating the relevant Butcher tableaux yields $\kappa_1 = f(x) = Ax$ and

$$\begin{aligned} & \kappa_2 = f\left(x + \tau \left(\frac{1}{2}\kappa_1 + \frac{1}{2}\kappa_2 \right) \right) = A\left(x + \frac{\tau}{2} Ax + \frac{\tau}{2}\kappa_2 \right) \\ \Longrightarrow \qquad & \kappa_2 = \left(I - \frac{\tau}{2} A \right)^{-1} \left(I + \frac{\tau}{2} A \right) Ax. \end{aligned}$$

Then

$$\psi(\tau, x) = x + \frac{\tau}{2} Ax + \frac{\tau}{2} \left(I - \frac{\tau}{2} A \right)^{-1} \left(I + \frac{\tau}{2} A \right) Ax.$$

Taking $(I - \frac{\tau}{2} A)^{-1}$ as a factor, we derive the iteration formula

$$x \longmapsto \psi(\tau, x) \equiv \left(I - \frac{\tau A}{2} \right)^{-1} \left(I + \frac{\tau A}{2} \right) x. \tag{5.11}$$

The matrix $A \in \mathbb{R}^{n \times n}$ of our dynamical system is real; nevertheless, it may have complex eigenvalues. Next, we will define a simple dynamical system which will depend on a complex parameter $\lambda \in \mathbb{C}$; later, the parameter λ will play the role of an eigenvalue $\lambda \in \sigma(A)$.

For a given $\lambda \in \mathbb{C}$, we consider the initial value problem

$$z' = \lambda z, \qquad z(0) = z_0 \in \mathbb{C} \tag{5.12}$$

in the complex plane. The corresponding flow is defined by the complex exponential

$$z(t) = \mathrm{e}^{t\lambda} z_0 \quad \text{for } t \in \mathbb{R}. \tag{5.13}$$

Assume that

$$\Re(\lambda) < 0; \tag{5.14}$$

then, for all $z(0) = z_0 \in \mathbb{C}$

$$e^{t\lambda} z_0 \longrightarrow 0 \in \mathbb{C} \qquad \text{for } t \to +\infty. \tag{5.15}$$

In other words, $0 \in \mathbb{C}$ is an A-stable steady state of the dynamical system (5.12).

We will consider a numerical solution of the initial value problem (5.12), which are iterations $z \longmapsto \psi(\tau, z)$ of a chosen one-step method. For example,

- the Euler method, see (5.6), corresponds to the iterations

$$z \in \mathbb{C} \longmapsto \psi(\tau, z) \equiv (1 + \tau\lambda)z \in \mathbb{C}, \tag{5.16}$$

- the Runge method, see (5.8), is defined by the iterations

$$z \in \mathbb{C} \longmapsto \psi(\tau, z) \equiv \left(1 + \tau\lambda + \frac{\tau^2}{2}\lambda^2\right) z \in \mathbb{C}, \tag{5.17}$$

- the Implicit Euler method, see (5.10), is defined by the iterations

$$z \in \mathbb{C} \longmapsto \psi(\tau, z) \equiv (1 - \tau\lambda)^{-1} z \in \mathbb{C}, \tag{5.18}$$

- and the Crank-Nicholson method, see (5.11), is defined by the iterations

$$z \in \mathbb{C} \longmapsto \psi(\tau, z) \equiv \left(1 - \frac{\tau}{2}\lambda\right)^{-1} \left(1 + \frac{\tau}{2}\lambda\right) z \in \mathbb{C}. \tag{5.19}$$

The steady state $0 \in \mathbb{C}$ is interpreted as the fixed point $0 = \psi(\tau, 0)$ of the iterations $z \longmapsto \psi(\tau, z)$ according to Proposition 4.23; hence, we will check whether for all $z(0) = z_0 \in \mathbb{C}$

$$\psi^j(\tau, z_0) \longrightarrow 0 \in \mathbb{C} \qquad \text{for } j \to +\infty. \tag{5.20}$$

For example, in case of the Runge method, for all $z(0) = z_0 \in \mathbb{C}$

$$\left(1 + \tau\lambda + \frac{\tau^2}{2}\lambda^2\right)^j z_0 \longrightarrow 0 \in \mathbb{C} \qquad \text{for } j \to +\infty,$$

and in case of Crank-Nicholson, for all $z(0) = z_0 \in \mathbb{C}$

$$\left(\frac{1 + \frac{\tau}{2}\lambda}{1 - \frac{\tau}{2}\lambda}\right)^j z_0 \longrightarrow 0 \in \mathbb{C} \qquad \text{for } j \to +\infty,$$

In other words, we will check whether $0 = \psi(\tau, 0)$ is an A-stable fixed point of the iterations $z \longmapsto \psi(\tau, z)$.

We are now approaching the key notion of this section, namely the *domain of stability*, which pertains to each particular numerical method. Let us consider the Euler method, i.e., the iterations (5.6).

Definition 5.1 (Domain of stability for the Euler method)**.** We define the set

$$S = \{\mu \in \mathbb{C} : |1 + \mu| < 1\}. \tag{5.21}$$

The set is called *domain of stability for the Euler method*.

Consider the iterations (5.16) of the Euler method. Then, we require $\tau > 0$ to be chosen such that for all $z(0) = z_0 \in \mathbb{C}$

$$(1 + \tau\lambda)^j z_0 \longrightarrow 0 \in \mathbb{C} \qquad \text{for } j \to +\infty; \tag{5.22}$$

i.e., we require A-stability of the fixed point. This will be satisfied provided that $|1 + \tau\lambda| < 1$; i.e., provided that $\tau\lambda \in S$. The domain S can be explicitly constructed as it is the interior of the circle centred at point $-1 \in \mathbb{C}$ with radius $r = 1$, see Figure 5.1.

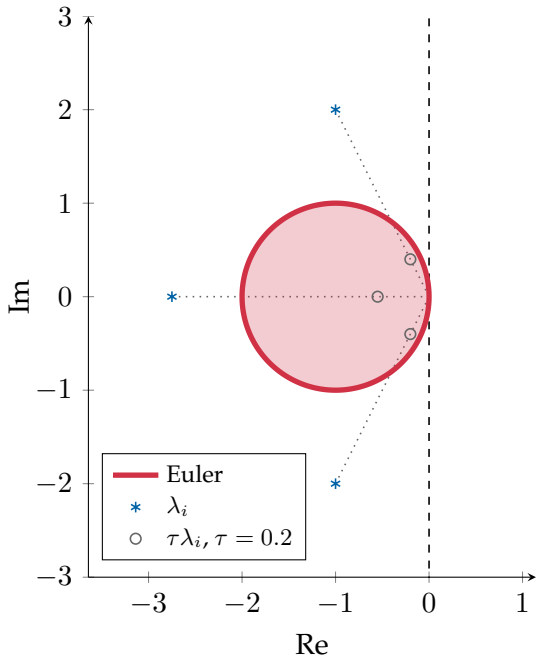


Figure 5.1: Domain of stability for Euler (interior of circle at $-1 \in \mathbb{C}$ with radius $r = 1$) compared to spectrum $\sigma(A) = \{\lambda_1, \lambda_2, \lambda_3\}$, where $\lambda_1 = -2.75$, $\lambda_2 = -1 + 2i$, $\lambda_3 = -1 - 2i$, and spectral transformation $\lambda_i \mapsto \tau\lambda_i$, for $\tau = 0.2$

Example 5.7 (Choice of $\tau > 0$ for Euler). We consider a matrix $A \in \mathbb{R}^{3\times3}$ with spectrum $\sigma(A) = \{\lambda_1, \lambda_2, \lambda_3\}$, where $\lambda_1 = -2.75$, $\lambda_2 = -1 + i2$, and $\lambda_3 = -1 - i2$. The objective is to choose $\tau > 0$ such that $\tau\lambda_i \in S$ for $i = \{1, 2, 3\}$. For example, the choice of $\tau = 0.2$ satisfies this requirement. Figure 5.1 illustrates the spectrum $\sigma(A)$ and the transformed spectral points $\tau\lambda_i$. In summary, if $\tau = 0.2$ then (5.22) holds for $\lambda := \lambda_i$, $i = \{1, 2, 3\}$. The Euler method defines iterations (5.6), $x \in \mathbb{R}^3 \longmapsto \psi(\tau, x) \equiv (I + \tau A)x \in \mathbb{R}^3$. Matrix A can be transformed to the Jordan canonical form as the matrix A is *diagonalisable*; hence, there exists a $Q \in \mathbb{C}^{3\times3}$ such that

$$AQ = QD,$$

where D a diagonal matrix with the diagonal entries λ_1, λ_2 and λ_3. Then, for all $x_0 \in \mathbb{R}^3$

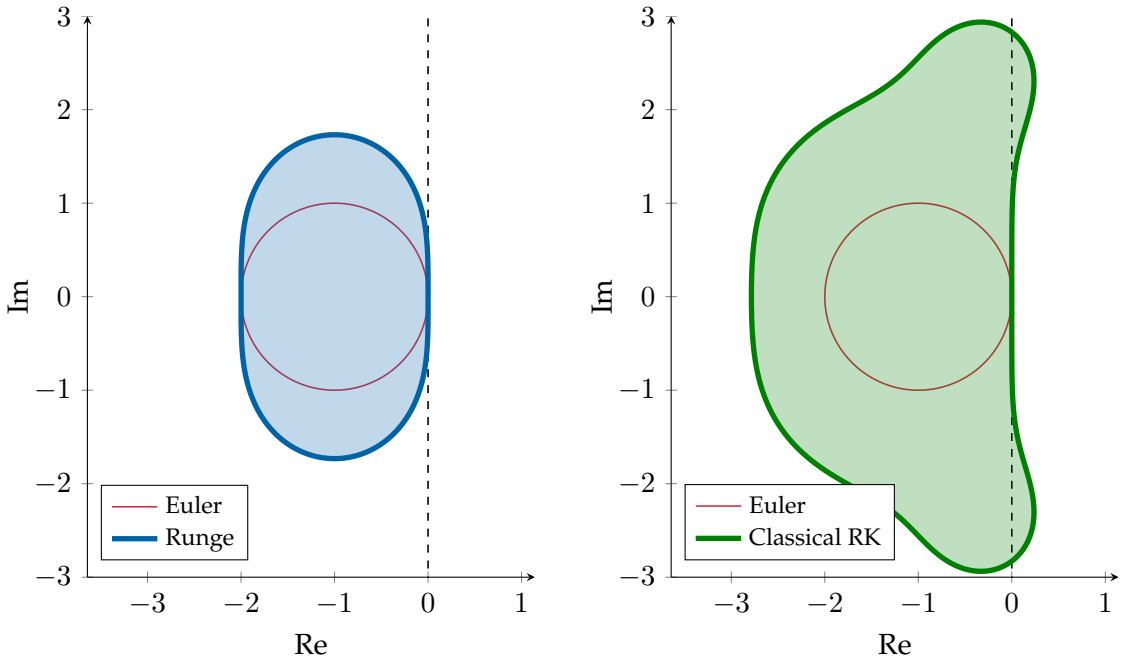
$$(I + \tau A)^j x_0 \longrightarrow 0 \in \mathbb{R}^3 \qquad \text{for } j \to +\infty;$$

i.e., $0 = \psi(\tau, 0)$ is an A-stable fixed point of the iteration $x \longmapsto \psi(\tau, x)$.

Definition 5.2 (Domain of stability for the Runge method)**.** We define the set

$$S = \left\{ \mu \in \mathbb{C} : \left| 1 + \mu + \frac{\mu^2}{2} \right| < 1 \right\}. \tag{5.23}$$

The set is called the *domain of stability for the Runge method*.



(a) Runge

(b) Classical Runge-Kutta

Figure 5.2: Domains of stability for Runge and Classical Runge-Kutta compared to Euler

Consider the iterations (5.17) of the Runge method. We require $\tau > 0$ to be chosen such that for all $z(0) = z_0 \in \mathbb{C}$

$$\left(1 + \tau\lambda + \frac{\tau^2\lambda^2}{2}\right)^j z_0 \longrightarrow 0 \in \mathbb{C} \qquad \text{for } j \to +\infty; \tag{5.24}$$

i.e., we require A-stability of the fixed point. This will be satisfied provided that

$$\left|1 + \tau\lambda + \frac{\tau^2\lambda^2}{2}\right| < 1;$$

i.e., provided that $\tau\lambda \in S$. Domain S can be explicitly constructed as an ellipse, or approximated *numerically* by incremental techniques using the Newton method. Figure 5.2(a) displays the domain of stability for the Runge method as the interior of the blue ellipse compared with the domain of stability for the Euler method (the interior of the red circle).

Definition 5.3 (Domain of stability for the classical RK method)**.** We define the set

$$S = \left\{\mu \in \mathbb{C} : \left|1 + \mu + \frac{\mu^2}{2} + \frac{\mu^3}{6} + \frac{\mu^4}{24}\right| < 1\right\} \tag{5.25}$$

The set is called the *domain of stability for the classical RK method*.

Consider the iterations (5.9) of the classical RK method. We require that $\tau > 0$ to be chosen such that for all $z(0) = z_0 \in \mathbb{C}$

$$\left(1 + \tau\lambda + \frac{\tau^2\lambda^2}{2} + \frac{\tau^3\lambda^3}{6} + \frac{\tau^4\lambda^4}{24}\right)^j z_0 \longrightarrow 0 \in \mathbb{C} \qquad \text{for } j \to +\infty; \tag{5.26}$$

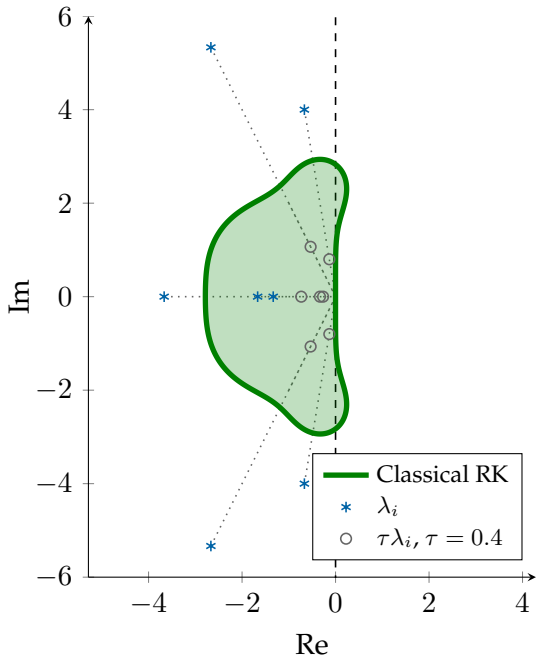


Figure 5.3: Domain of stability for Classical Runge-Kutta compared to spectrum $\sigma(A) = \{\lambda_i\}_{i=1}^7$, and spectral transformation $\lambda_i \mapsto \tau\lambda_i$, for $\tau = 0.4$, for Example 5.8

i.e., we require A-stability of the fixed point. This will be satisfied provided that

$$\left|1 + \tau\lambda + \frac{\tau^2\lambda^2}{2} + \frac{\tau^3\lambda^3}{6} + \frac{\tau^4\lambda^4}{24}\right| < 1;$$

i.e., provided that $\tau\lambda \in S$. Figure 5.2(b) displays the domain of stability for the classical Runge-Kutta method as the interior of the green curve compared with the domain of stability for the Euler method (the interior of the red circle).

Example 5.8 (Choice of $\tau > 0$ for classical Runge-Kutta). We consider a matrix $A \in \mathbb{R}^{7\times 7}$ with spectrum $\sigma(A) = \{\lambda_i\}_{i=1}^7$, where $\lambda_1 = -8/3 + 16i/3$, $\lambda_2 = -8/3 - 16i/3$, $\lambda_3 = -2/3 + 4i$, $\lambda_4 = -2/3 - 4i$, $\lambda_5 = -11/3$, $\lambda_6 = -5/3$, and $\lambda_7 = -4/3$. The objective is to choose $\tau > 0$ such that $\tau\lambda_i \in S$ for $i = 1, \ldots, 7$. For example, the choice of $\tau = 0.4$ satisfies this requirement. Figure 5.3 illustrates the spectrum $\sigma(A)$ and the transformed spectral points $\tau\lambda_i$ compared to the domain of stability. Then, for all $x_0 \in \mathbb{R}^7$

$$\psi^j(\tau, x_0) \longrightarrow 0 \in \mathbb{R}^7 \qquad \text{for } j \to +\infty;$$

i.e., $0 = \psi(\tau, 0)$ is an A-stable fixed point of the iterations

$$x \longmapsto \psi(\tau, x) \equiv \left(I + \tau A + \frac{\tau^2}{2}A^2 + \frac{\tau^3}{6}A^3 + \frac{\tau^4}{24}A^4\right) x.$$

We can simply follow the arguments applied in Example 5.7.

Next we define the domains of stability for two implicit methods.

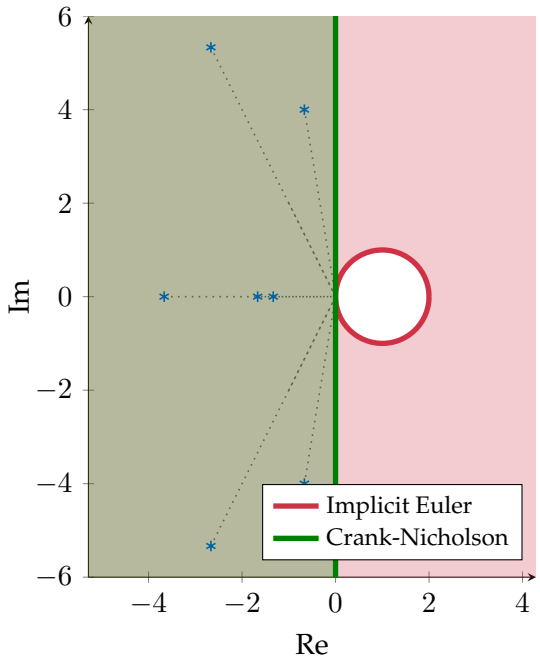


Figure 5.4: Domain of stability for Implicit Euler (*exterior* of circle at $1 \in \mathbb{C}$ with radius $r = 1$) and Crank-Nicholson ($\{\mu \in \mathbb{C} : \Re(\mu) < 0\}$) compared to spectrum $\sigma(A) = \{\lambda_i\}_{i=1}^{7}$ for Example 5.9

Definition 5.4 (Domain of stability for the implicit Euler method)**.** We define the set

$$S = \left\{ \mu \in \mathbb{C} : \frac{1}{|1-\mu|} < 1 \right\}. \tag{5.27}$$

The set is called the *domain of stability for the implicit Euler method*.

Consider the iterations (5.18) of the implicit Euler method. We require $\tau > 0$ to be chosen such that for all $z(0) = z_0 \in \mathbb{C}$

$$\left(\frac{1}{1-\tau\lambda} \right)^j z_0 \longrightarrow 0 \in \mathbb{C} \qquad \text{for } j \to +\infty; \tag{5.28}$$

i.e., we require A-stability of the fixed point. This will be satisfied provided that $|1-\tau\lambda|^{-1} < 1$; i.e., provided that $\tau\lambda \in S$. The domain of stability S is the *exterior* of the circle centred at the point $1 \in \mathbb{C}$ with radius $r = 1$, see Figure 5.4. In summary, the condition (5.28) is satisfied for all $\tau > 0$.

Definition 5.5 (Domain of stability for the Crank-Nicholson method)**.** We define the set

$$S = \left\{ \mu \in \mathbb{C} : \frac{|1+\frac{\mu}{2}|}{|1-\frac{\mu}{2}|} < 1 \right\}. \tag{5.29}$$

The set is called the *domain of stability for the Crank-Nicholson method*.

Consider the iterations (5.19) of the Crank-Nicholson method. We require $\tau > 0$ to be chosen such that for all $z(0) = z_0 \in \mathbb{C}$

$$\left(\frac{1+\frac{\tau\lambda}{2}}{1-\frac{\tau\lambda}{2}}\right)^j z_0 \longrightarrow 0 \in \mathbb{C} \qquad \text{for } j \to +\infty; \tag{5.30}$$

i.e., we require A-stability of the fixed point. This will be satisfied provided that $|1+\frac{\tau\lambda}{2}| < |1-\frac{\tau\lambda}{2}|$; i.e., provided that $\tau\lambda \in S$. The domain of stability S is the left half of the complex plane; i.e, $S = \{\mu \in \mathbb{C} : \Re(\mu) < 0\}$, see Figure 5.4. In summary, the condition (5.28) is satisfied for all $\tau > 0$.

Example 5.9 (Choice of $\tau > 0$ for implicit one-step). We consider a matrix $A \in \mathbb{R}^{7\times 7}$ with the spectrum $\sigma(A) = \{\lambda_i\}_{i=1}^7$, as in Example 5.8. Let us analyse the implicit Euler method and the Crank-Nicholson — the domains of stability of both methods are shown in (5.28) along with the spectrum $\sigma(A)$. Notice that $\tau\lambda_i \in S$, $i = 1, \ldots, 7$, for all $\tau > 0$. In other words, the domain S is invariant with respect to the homotetic transformation for all $\tau > 0$. Hence, for all $\tau > 0$, the fixed point $0 = \psi(\tau, 0)$ is an A-stable fixed point of both the iterations (5.10)

$$x \longmapsto \psi(\tau, x) \equiv (I - \tau A)^{-1} x,$$

and the iterations (5.11)

$$x \longmapsto \psi(\tau, x) \equiv \left(I - \frac{\tau A}{2}\right)^{-1} \left(I + \frac{\tau A}{2}\right) x.$$

We can now define the general definition of the domain of stability for the class of Runge-Kutta (RK) methods. We consider the Butcher tableau of the chosen RK method, see Definition 2.28. We solve the initial value the problem (5.3); i.e., the linearised problem with matrix $A \in \mathbb{R}^{n\times n}$, by means of the chosen RK method. Consequently, we solve (5.12), where $\lambda \in \sigma(A)$.

It can be shown that one iteration of the chosen method has the form

$$z \in \mathbb{C} \longmapsto \psi(\tau, z) \equiv R(\tau\lambda) z \in \mathbb{C}, \tag{5.31}$$

see Deuflhard and Bornemann (2012, Lemma 6.30). We set $\mu \equiv \tau\lambda$ and define the function $\mu \in \mathbb{C} \longmapsto R(\mu) \in \mathbb{C}$, where

1. $R(\mu)$ is a *polynomial*, with $R(0) = 1$, in case of an *explicit method*, and
2. $R(\mu) = {}^{P(\mu)}/_{Q(\mu)}$ is a *rational function*, with $P(0) = Q(0) = 1$, in case of an *implicit method*.

Compare (5.31) with the methods we came across so far:

- the classical Runge-Kutta method,

$$R(\mu) = 1 + \mu + \frac{\mu^2}{2} + \frac{\mu^3}{6} + \frac{\mu^4}{24}$$

- the implicit Euler method,

$$R(\mu) = \frac{1}{1-\mu}$$

- the Crank-Nicholson method,

$$R(\mu) = \frac{1 + \frac{\mu}{2}}{1 - \frac{\mu}{2}}.$$

Definition 5.6 (Domain of stability for RK method)**.** We define the set

$$S = \{\mu \in \mathbb{C} : |R(\mu)| < 1\}. \tag{5.32}$$

The set is called the *domain of stability for RK method*.

Consider the iterations (5.31) of the RK method. We require τ to be chosen such that for all $z(0) = z_0 \in \mathbb{C}$

$$(R(\tau\lambda))^j z_0 \longrightarrow 0 \in \mathbb{C} \qquad \text{for } j \to +\infty; \tag{5.33}$$

i.e., we require A-stability of the fixed point. This will be satisfied provided $|R(\tau\lambda)| < 1$; i.e., provided that $\tau\lambda \in S$.

Definition 5.7 (A-stable RK method)**.** Let us consider a RK method and let S be its domain of stability. Then, we say that the RK method is *A-stable* if the domain of stability contains the left half-plane of the complex plane; i.e.,

$$\{\mu \in \mathbb{C} : \Re(\mu) < 0\} \subset S. \tag{5.34}$$

Hence, the condition (5.33); namely, the A-stability of the fixed point $0 = \psi(\tau, 0)$, is satisfied regardless of the step size $\tau > 0$ for a A-stable RK method.

Consequently, the implicit Euler method and the Crank-Nicholson method are A-stable methods. It can be shown that the implicit RK methods from Section 2.4.2 which are based on *Gauss quadrature* and *Radau quadrature* are A-stable methods, see Deuflhard and Bornemann (2012, Lemma 6.50 & Theorem 6.51).

It can also be shown that there is no *explicit* A-stable method, see Deuflhard and Bornemann (2012, Lemma 6.11). The domain of stability for an explicit RK method is a bounded set.

Theorem 5.8. *Consider the linear dynamical system (5.3) and let the spectral condition (5.2) be satisfied. We consider a particular RK method $x \in \mathbb{R}^n \longmapsto \psi(\tau, x) \in \mathbb{R}^n$ and let S be the corresponding domain of stability. Assume that $\tau > 0$ is chosen such that*

$$\lambda \in \sigma(A) \qquad \Longrightarrow \qquad \tau\lambda \in S;$$

then, for all $x(0) = x_0 \in \mathbb{R}^n$

$$\psi^j(\tau, x) \longrightarrow 0 \in \mathbb{R}^n \qquad \text{for } j \to +\infty; \tag{5.35}$$

i.e., $0 \in \mathbb{R}^n$ is an A-stable fixed point of the iteration $x \in \mathbb{R}^n \longmapsto \psi(\tau, x) \in \mathbb{R}^n$.

We skip the proof, noting that Examples 5.7, 5.8, and 5.9 illustrate the statement of Theorem 5.8.

Theorem 5.9. *Consider the initial value problem (4.1) for an autonomous ODE. Let $x^* \in D$, $f(x^*) = 0$ be the steady state and let the spectral condition (5.2) be satisfied. We consider a particular RK method $x \in \mathbb{R}^n \longmapsto \psi(\tau, x) \in \mathbb{R}^n$, define the domain of stability S by means of the linearisation (5.3) around the steady state x^*, and assume that $\tau > 0$ is chosen such that*

$$\lambda \in \sigma(A) \quad \Longrightarrow \quad \tau\lambda \in S;$$

then, $x^* \in \mathbb{R}^n$ is an A-stable fixed point of the iterations $x \in \mathbb{R}^n \longmapsto \psi(\tau, x) \in \mathbb{R}^n$.

We skip the proof here.

Remark 5.10. We note that the constraint on the step size $\tau > 0$ formulated above is understood locally; e.g., for points x from a sufficiently small neighbourhood of the steady state x^*, see Definition 4.24.

5.2 Domain of stability: multistep method

We consider the linear dynamical system (5.3) with matrix $A \in \mathbb{R}^{n\times n}$. Let the spectral property (4.3) be satisfied. Hence, the origin $0 \in \mathbb{R}^n$ is an A-stable steady state. In order to solve the initial value problem (5.3), we use an m-step method, namely Algorithm 3.1, (3.2)–(3.4). We will analyse asymptotic properties of the numerical solution; i.e., we will generate $\{t_j\}_{j=0}^{+\infty}$ and $\{u_j\}_{j=0}^{+\infty}$.

In Section 3.3 we have shown that the m-step recurrence (3.4) can be formulated as the equation (3.11)

$$\sum_{i=0}^{m} a_i u_{j+i} - \tau \sum_{i=0}^{m} b_i f(t_{j+i}, u_{j+i}) = 0 \in \mathbb{R}^n, \qquad j \in \mathbb{N}_0.$$

Due to linearity, $f(t_{j+i}, u_{j+i}) = Au_{j+i}$; therefore, Algorithm 3.1 represents the m-step recurrence

$$\sum_{i=0}^{m} (a_i I - \tau b_i A)\, u_{j+i} = 0 \in \mathbb{R}^n, \qquad j \in \mathbb{N}_0, \tag{5.36}$$

where $I \in \mathbb{R}^{n\times n}$ is the identity matrix. The algorithm is initialized by the choice of vectors

$$\{u_i\}_{i=0}^{m-1}, \quad u_0 \equiv x_0. \tag{5.37}$$

In order to initialise (5.37), we can exploit an arbitrary one step method.

We formulate the scalar model problem, see (5.12), and let (5.14) be satisfied. Therefore, the exact solution of the model problem is damped exponentially for each initial condition; i.e., (5.15) holds. We require the numerical solution of the model problem behaves similarly.

The numerical solution is a sequence of discrete times $\{t_j\}_{j=0}^{+\infty}$, $t_{j+1} = t_j + \tau$, and discrete states $\{z_j\}_{j=0}^{+\infty}$, $z_j \in \mathbb{C}$, which are defined by the initialisation condition $\{z_i\}_{i=0}^{m-1}$ and the m-step recurrence

$$\sum_{i=0}^{m} (a_i - \tau\lambda b_i)\, z_{j+i} = 0, \qquad j \in \mathbb{N}_0. \tag{5.38}$$

We require that for all $\{z_i\}_{i=0}^{m-1}$ (for all initialisations) it holds

$$z_j \to 0 \quad \text{for } j \to +\infty. \tag{5.39}$$

Let us note that the m-step recurrence (5.38) is a *linear difference equation* (Deuflhard and Bornemann, 2012, Example 3.39). The general solution of this difference equation is determined by the roots of the characteristic polynomial

$$\eta(z) = \sum_{i=0}^{m} (a_i - \tau\lambda b_i)\, z^i, \tag{5.40}$$

which is called *the third characteristic polynomial*. By recalling Definition 3.4,

$$\eta(z) = \rho(z) - \tau\lambda\sigma(z). \tag{5.41}$$

According to Deuflhard and Bornemann (2012, Theorem 3.40) the linear difference equation (5.38) satisfies the conditions (5.39) if and only if for all $z \in \mathbb{C}$

$$\rho(z) - \tau\lambda\sigma(z) = 0 \in \mathbb{C} \qquad \Longrightarrow \qquad |z| < 1. \tag{5.42}$$

Definition 5.11 (Domain of stability)**.** Let us consider a linear m-step method with coefficients (3.2). We call the set

$$S = \{\mu \in \mathbb{C} : \forall z \in \mathbb{C}, \rho(z) - \mu\sigma(z) = 0 \in \mathbb{C} \implies |z| < 1\} \tag{5.43}$$

the *domain of stability for the m-step method*.

If the step size $\tau > 0$ is chosen such that $\tau\lambda = \mu \in S$, then (5.39) holds.

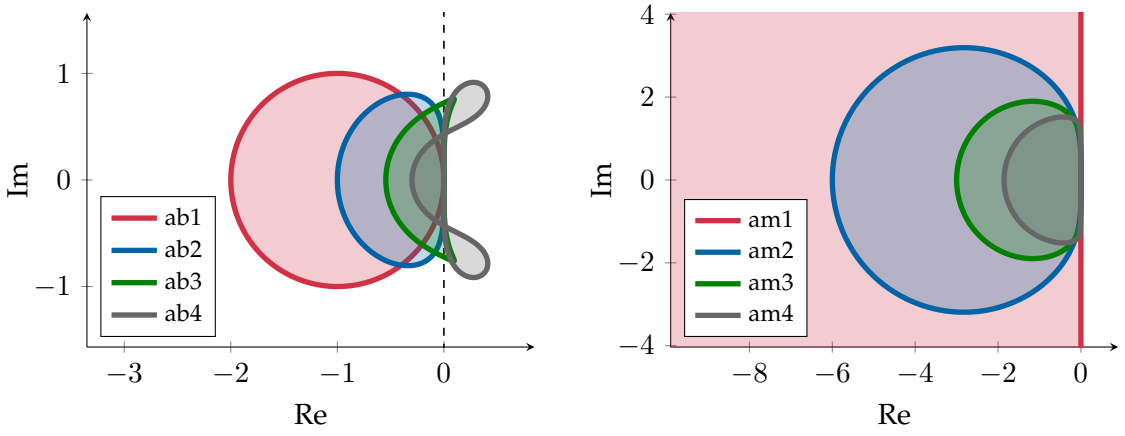
Remark 5.12 (Boundary of stability, Jordan curve). Let S is be an open set; then, we define $\partial S \equiv \overline{S} \setminus S$ to be the *boundary* of S. The boundary ∂S can be parametrised and explicitly constructed

$$\theta \in [0, 2\pi) \to \mathrm{e}^{i\theta} \to \rho(\mathrm{e}^{i\theta}) - \mu\sigma(\mathrm{e}^{i\theta}) = 0 \to \mu = \frac{\rho(\mathrm{e}^{i\theta})}{\sigma(\mathrm{e}^{i\theta})} \in \partial S. \tag{5.44}$$

The mapping (5.44) defines a positively oriented curve in the complex plane. If this mapping is a bijection then we call the curve a *Jordan curve*. In this case, we define the open sets $\mathrm{int}(\partial S)$ and $\mathrm{ext}(\partial S)$ as the interior and the exterior of the curve ∂S, respectively. The Jordan curve is a simple curve (it does not cross itself).

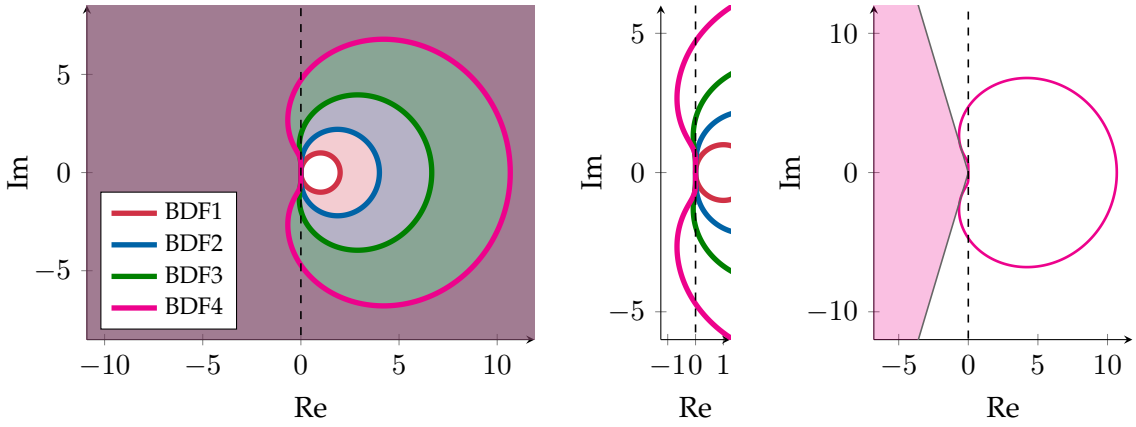
Example 5.10 (Domain of stability for explicit Adams method). We consider the m-step explicit Adams methods (m-step Adams-Bashfort methods) (ab1), (ab2), (ab3), and (ab4); see Example 3.7. The corresponding boundaries of stability are shown in Figure 5.5(a). The first three are Jordan curves and the corresponding domains of stability are $S = \mathrm{int}(\partial S)$ for (ab1), (ab2), and (ab3). The fourth boundary, for (ab4), is not a Jordan curve and the corresponding domain of stability consists of three connected pieces.

Example 5.11 (Domain of stability for implicit Adams method). We consider the m-step implicit Adams methods (m-step Adams-Moulton methods) (am1), (am2), (am3), and (am4); see Example 3.8. The corresponding boundaries of stability are shown in Figure 5.5(b). The method *am1* is the Crank-Nicholson method and, hence, the domain of stability is the left half of the complex plane. The boundaries for(am2), (am3), and (am4) are Jordan curves and, hence, the corresponding domains of stability are $S = \mathrm{int}(\partial S)$.



(a) Explicit Adams (Adams-Bashfort) (b) Implicit Adams (Adams-Moulton)

Figure 5.5: Domains of stability for Adams methods (interior of curves)



(a) Domains of Stability (b) Zoom (c) BDF4 $A(\alpha)$ stability

Figure 5.6: Domains of stability for BDF (exterior of curves)

It is useful to compare Figure 5.5(a) and Figure 5.5(b), namely the sizes of stability domains. The larger the domain of stability, the larger step size can be afforded.

Definition 5.13 (A-stable multistep method)**.** Let us consider a linear m-step method and let S be its domain of stability. We say that this method is *A-stable* if the domain of stability contains the left half of the complex plane; i.e.,

$$\{\mu \in \mathbb{C} : \Re(\mu) < 0\} \subset S. \tag{5.45}$$

The A-stability of the multistep method is a very rare property. It can be proven that the only A-stable methods are (am1), (BDF1), and (BDF2). The first two are one step methods: (am1) is the Crank-Nicholson method and (BDF1) is the Implicit Euler method.

Example 5.12 (Domain of stability for BDF). We consider the BDF methods (BDF1), (BDF2), (BDF3), and (BDF4); see Example 3.10. The corresponding boundaries of stability are shown

in Figure 5.6, and the domains of stability $S = \mathrm{ext}(\partial S)$ are the *exteriors* of the corresponding curves.

BDF1 and BDF2 are A-stable methods. We have already noted that BDF methods are *not* D-stable for $m > 6$. For the methods BDF3–BDF6 we introduce a weaker concept of A-stability. We say that the method is $A(\alpha)$-stable if the domain of stability S contains the sector

$$\{\mu \in \mathbb{C} : |\arg(-\mu)| < \alpha, \quad \mu \neq 0\} \subset S, \tag{5.46}$$

where arg is the principal value of the argument of the complex number μ. Measuring the angles α in degrees then the m-step BDF is $A(\alpha)$-stable for the following angles α:

m	3	4	5	6
α	86.03°	73.35°	51.84°	17.84°

For example, see Figure 5.6(c), where the *infinite* purple arc denotes the sector (5.46) with angle $\alpha = 73.35°$ and the boundary of the domain of stability for (BDF4) is also shown.

Finally, we formulate two assertions which are analogous to Theorem 5.8 and Theorem 5.9.

Theorem 5.14. *We consider the linear dynamical system (5.3) and let the spectral condition (5.2) be satisfied. We consider the m-step method from Algorithm 3.1; i.e., (5.36)–(5.37) and let S be the corresponding domain of stability. Assume that time step $\tau > 0$ is chosen such that*

$$\lambda \in \sigma(A) \quad \Longrightarrow \quad \tau\lambda \in S;$$

then, for each initialisation for all $\{u_i\}_{i=0}^{m-1}$ such that

$$\sum_{i=0}^{m} (a_i I - \tau b_i A)\, u_{j+i} = 0 \in \mathbb{R}^n, \quad j \in \mathbb{N}_0, \tag{5.47}$$

it holds that

$$u_j \to 0 \in \mathbb{R}^n \quad \text{for } j \to +\infty. \tag{5.48}$$

Theorem 5.15. *We consider the initial value problem (4.1) for the autonomous ODE. Let $x^* \in D$, $f(x^*) = 0$, be a steady state and let the spectral condition (5.2) be satisfied. We consider the m-step method from Algorithm 3.1 which generates a sequence of discrete times and states*

$$j \in \mathbb{N}_0 \longmapsto t_j = \tau j, \qquad \{t_j\}_{j=0}^{+\infty}, \qquad \{u_j\}_{j=0}^{+\infty}. \tag{5.49}$$

Let S be the domain of stability of the corresponding method and assume that the time step $\tau > 0$ is chosen such that

$$\lambda \in \sigma(A) \quad \Longrightarrow \quad \tau\lambda \in S.$$

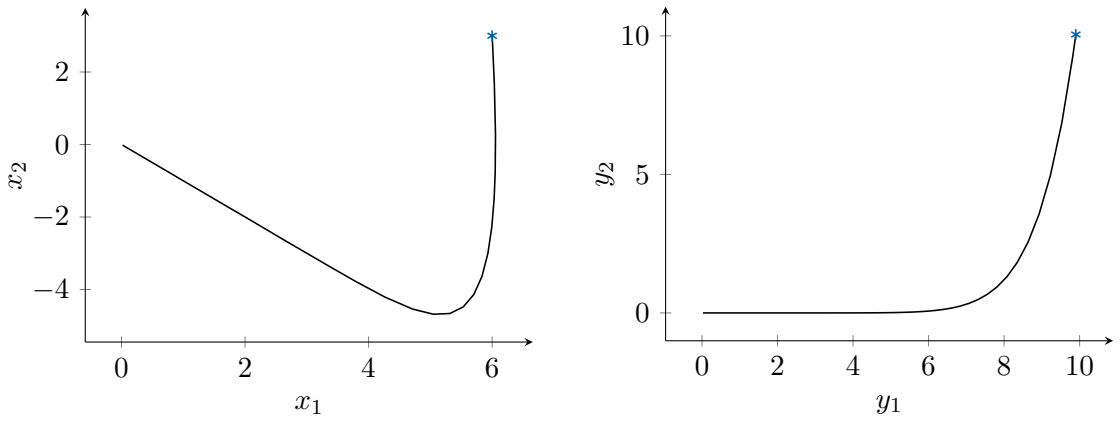
There exists a constant $r > 0$ such that for every initial condition $x_0 \in D$ which satisfies the condition

$$\|x_0 - x^*\| < r;$$

the following holds: for each initialisation $\{u_i\}_{i=0}^{m-1}$, $u_0 \equiv x_0$, which is obtained via the one-step method (3.5) with an initial condition x_0, the method converges to a steady state; i.e.,

$$u_j \to x^* \qquad \text{for } j \to +\infty. \tag{5.50}$$

In summary, the method converges locally.



(a) State variable x

(b) State variable y with $y^0 = V^{-1}x^0$

Figure 5.7: Orbit of Example 5.13 for $x^0 = (6, 3)$

5.3 Stiff problems

We consider a *damped linear oscillator*

$$\begin{bmatrix} x_1' \\ x_2' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -c & -k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \tag{5.51}$$

where $c > 0$ and $k > 0$ are constants representing the damping and stiffness, respectively, of the spring. Let $A \in \mathbb{R}^{2\times 2}$ be the matrix of the system.

Example 5.13 (Damped linear oscillator). We consider (5.51) with $c = 10$ and $k = 11$. Then, $\sigma(A) = \{\lambda_1 = -1, \lambda_2 = -10\}$. Hence, we can transform $A \in \mathbb{R}^{2\times 2}$ to its Jordan normal form J; i.e., there exists a nonsingular matrix $V \in \mathbb{R}^{2\times 2}$ such that

$$J = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}, \qquad AV = VJ, \qquad V = \begin{bmatrix} 0.7071 & -0.0995 \\ -0.7071 & 0.9950 \end{bmatrix}. \tag{5.52}$$

We solve two equivalent problems in the state variables $x \in \mathbb{R}^2$ and $y \in \mathbb{R}^2$:

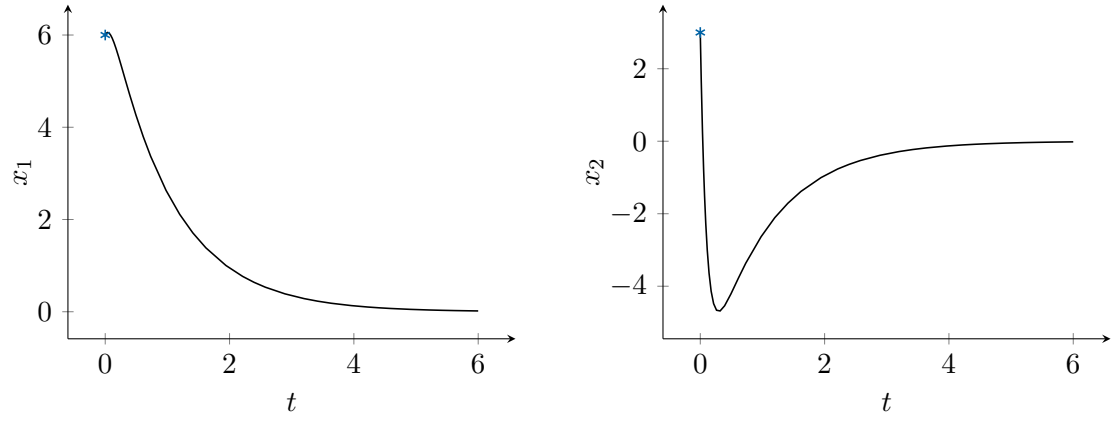
$$\begin{aligned} x'(t) &= Ax(t), & x(0) &= x^0 = Vy^0, \\ y'(t) &= Jy(t), & y(0) &= y^0 = V^{-1}x^0, \end{aligned}$$

where $x(t) = Vy(t)$, $y(t) = V^{-1}x(t)$ for $t \in \mathbb{R}$. Note that due to the form of the matrix J we actually solve two independent problems in the state variable y; namely,

$$\begin{aligned} y_1'(t) &= \lambda_1 y_1(t), & y_1(0) &= y_1^0, \\ y_2'(t) &= \lambda_2 y_2(t), & y_2(0) &= y_2^0. \end{aligned}$$

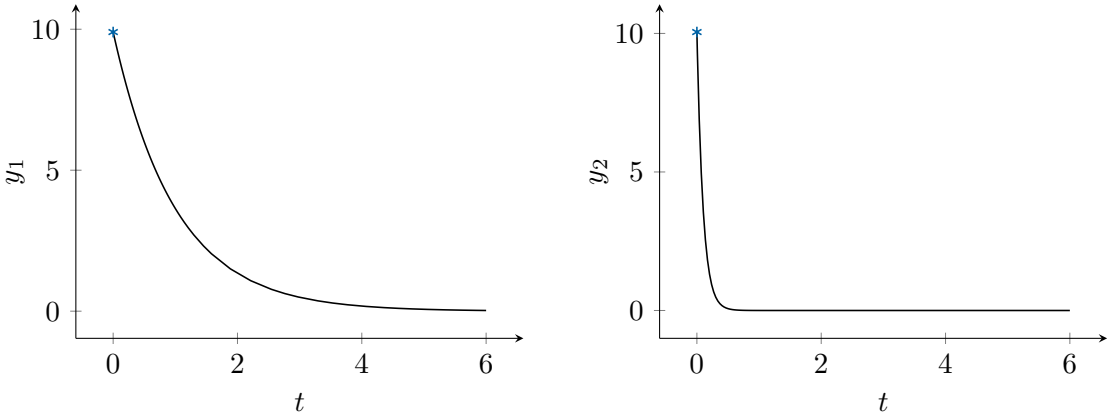
Nevertheless, the time t is synchronized in both systems.

In Figure 5.7 we compare the two orbits of the state variables x and y with initial condition $x^0 = (6, 3)$ on the time interval $t \in [0, T]$, $T = 6$. In Figure 5.8 and Figure 5.9 we depict



(a) x_1 (b) x_2

Figure 5.8: Trajectories of Example 5.13 for state variable x



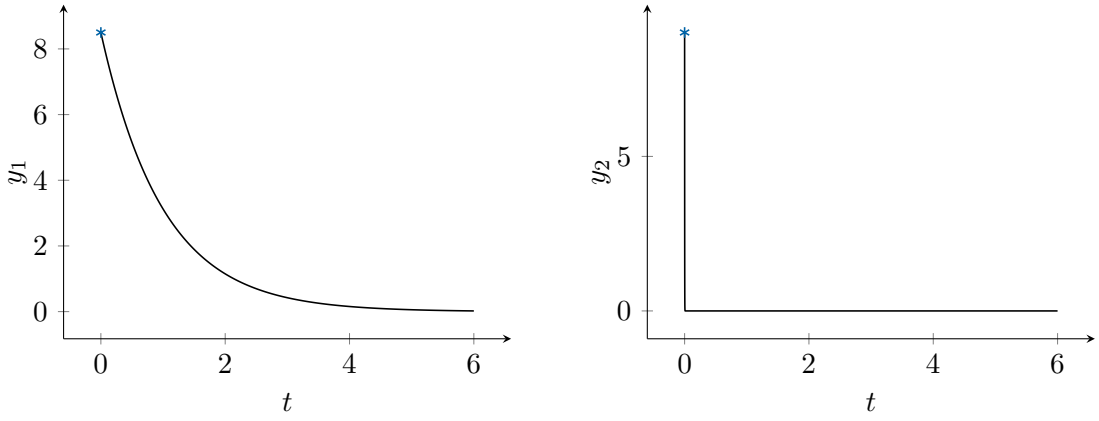
(a) y_1; slow damping $\lambda_1 = -1$ (b) y_2; fast damping $\lambda_2 = -10$

Figure 5.9: Trajectories of Example 5.13 for state variable y

the corresponding trajectories in x and y, respectively, on the same time interval. Introducing y, we notice qualitative differences of the solution components y_1 and y_2 are more visible — in Figure 5.9(a) the state variable y_1 decreases slowly in time; whereas, in Figure 5.9(b) the state variable y_2 decreases much more rapidly. This is related to the fact that $|\lambda_1|$ is comparatively small (the case for Figure 5.9(a)) and $|\lambda_2|$ is comparatively large (the case for Figure 5.9(b)). The length $T > 0$ of the analysed time interval is also important in order to allow enough time for qualitative differences in Figure 5.9 to develop.

Example 5.13 illustrates the phenomenon which is called the *stiffness*. The stiffness characterises the mathematical problem: a dynamical system which is defined by (5.51) is *stiff* for specific values of parameters (c and k in our case). The *stiffness* is also an issue for the numerical solution, as it requires selection of a extremely small step length.

Definition 5.16 (stiffness ratio)**.** Consider the linear dynamical system (5.3) and assume that



(a) y_1; slow damping $\lambda_1 = -1$ (b) y_2; very fast damping $\lambda_2 = -1000$

Figure 5.10: Trajectories of Example 5.14 for state variable y

$A \in \mathbb{R}^{n\times n}$ satisfies the spectral condition (5.2); then, we define the *stiffness ratio* as

$$L = \frac{\max_{i=1,\dots,n} |\Re(\lambda_i)|}{\min_{i=1,\dots,n} |\Re(\lambda_i)|}, \tag{5.53}$$

where $\lambda_i \in \sigma(A)$, $i = 1, \dots, n$. We say that the stiffness ratio is large if $L \gg 1$.

We can generalize the definition of the stiffness ratio by means of the linearisation.

Remark 5.17 (stiffness ratio for nonlinear equations). Let us consider the initial value problem (4.1) for an autonomous ODE, let $x^* \in D$, $f(x^*) = 0$, be the steady state, and the Jacobian $A \in \mathbb{R}^{n\times n}$, see (5.1), satisfy the spectral condition (5.2). Then, the stiffness ratio L is defined by the formula (5.53), where $\lambda_i \in \sigma(A)$, $i = 1, \dots, n$. The analysis of the stiffness ratio L is restricted to a sufficiently small neighbourhood of the steady state.

The problem defined in Example 5.13 is not classified as a stiff problem since $L = 10$ is small.

Example 5.14 (Stiff damped linear oscillator). Consider the damped linear oscillator (5.51) with $c = 1000$ and $k = 1001$; then, $\sigma(A) = \{\lambda_1 = -1\,,\ \lambda_2 = -1000\}$.

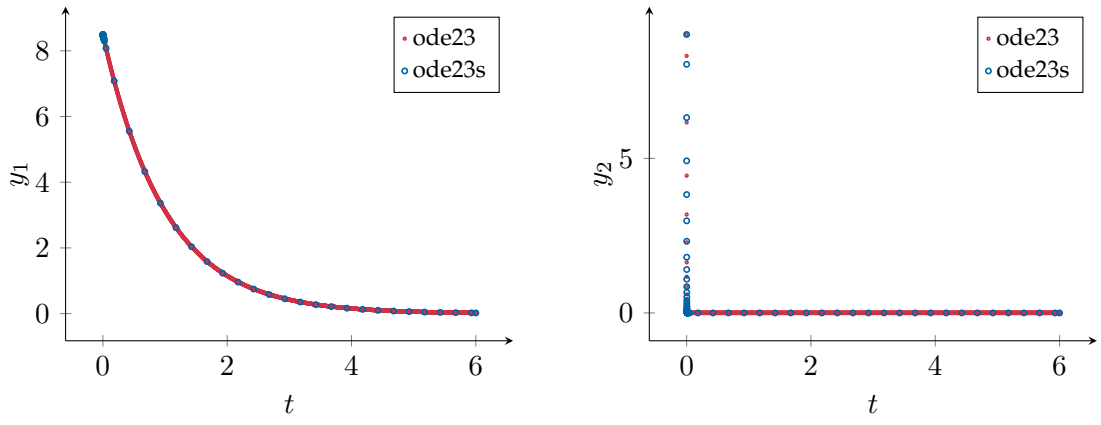
In the analysis we proceed as in Example 5.13. We transform $A \in \mathbb{R}^{2\times 2}$ to Jordan normal form, see (5.52), with eigenvalues $\lambda_1 = -1$, $\lambda_2 = -1000$, and transformation matrix

$$V = \begin{bmatrix} 0.7071 & -0.0010 \\ -0.7071 & 1.0000 \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

In the numerical experiment, which is similar to Example 5.13, we choose the same initial condition $x^0 = (6, 3)$. In Figure 5.10 we investigate the time evolution of the state variable $y \in \mathbb{R}^2$; we note that y_2, see Figure 5.10(b), is damped almost immediately.

Example 5.14 can be classified as a stiff problem as $L = 1000$ is large.

So, what is the issue with the numerical solution of a stiff problem? Consider the problem formulated in Example 5.14 and apply the Euler method with step size $\tau > 0$. Recall the



(a) y_1 (b) y_2

Figure 5.11: Comparison of **ode23** and **ode23s** for Example 5.14

domain of stability (5.21), cf. Example 5.7; then, in the context of Example 5.14 we have to require that

$$\tau\lambda_1 = -\tau \in S, \qquad \tau\lambda_2 = -1000\tau \in S.$$

Hence, we have to choose $\tau < 2/1000 = 0.002$. This restriction of the time step is necessary for the numerical solution to converge to the correct fixed point. However, if we observe the initial stages of the trajectory in Figure 5.10 the solution changes dramatically; then, in order to capture this tendency we need to decrease the step size ten times to $\tau = 0.0002$. Since we are integrating on large interval $0 \leq t \leq 6$, the computational overhead is astronomical.

To solve the problem formulated in Example 5.14, we used **ode23**; i.e., the adaptive step refinement based on explicit methods. After the initial stages when the time step τ was dramatically reduced and settles down to $\tau \approx 10^{-3}$. For a stiff problem it is more reasonable to use implicit methods; namely, A-stable methods. MATLAB provides the function **ode23s** to solve stiff problems with adaptive step refinement. Using this method for numerical computation shows that the time step is $\tau \approx 10^{-1}$ in the stages when the solution does not change much. Figure 5.11 shows the numerical solution of the state variable y for Example 5.14 using both **ode23** and **ode23s**, where each computed point (and hence each time point) is indicated by a circle on the plot. Notice, that for **ode23** the time step size is so small that it is impossible to see the individual points.

Let us cite Hairer and Wanner (2010, pg. 2):

> "Stiff equations are problems for which explicit methods don't work."

The label "Stiff equations" covers "stiff problems" and hence, among other things, the problems with a large stiffness ratio according to Definition 5.16 or Remark 5.17. We also speak of stiff ODEs.

Hairer and Wanner (2010) admit that a rigorous definition of *stiff problems* does not exist; nevertheless, whatever the stiffness might be, the following solution strategy is recommended: If your favourite explicit method fails then try an implicit method. It may work.

The classical examples of stiff problems are the problems which originate by discretization of partial differential equations (PDE).

Example 5.15 (Heat equation). We seek for a function $\boldsymbol{u} = \boldsymbol{u}(x,t)$ on the domain $\Omega = \{(x,t) : 0 \le x \le 1, 0 \le t < +\infty\}$ such that

$$\frac{\partial \boldsymbol{u}}{\partial t} = \frac{\partial^2 \boldsymbol{u}}{\partial x^2} \qquad \text{on } \Omega, \tag{5.54}$$

assuming the homogeneous Dirichlet boundary condition

$$\boldsymbol{u}(0,t) = \boldsymbol{u}(1,t) = 0, \qquad \text{for } t \ge 0,$$

and the initial condition

$$\boldsymbol{u}(x,0) = \boldsymbol{u}^0(x) \qquad \text{for } 0 \le x \le 1,$$

where $\boldsymbol{u}^0 = \boldsymbol{u}^0(x)$ is a given function on the interval $0 \le x \le 1$.

As the discretization technique we use the *method of lines*, also called the *semi-discretisation* in the space variable, by considering a finite difference discretisation in space. To this end, define a mesh on the space domain $0 \le x \le 1$ by a uniform partition into $n+2$ nodes

$$x_j = jh, \qquad h = \frac{1}{n+1}, \qquad j = 0, \dots, n+1,$$

and the vector function $u(t) = (u_0(t), u_1(t), \dots, u_n(t), u_{n+1}(t))^\top$, where each component approximates $\boldsymbol{u}(x_j, t)$, $j = 0, \dots, n+1$. In particular, we approximate the space derivatives via the central difference:

$$\frac{\partial \boldsymbol{u}}{\partial x}(x_j, t) \approx \frac{u_{j-1}(t) - 2u_j(t) + u_{j+1}(t)}{h^2}, \qquad j = 1, \dots, n.$$

Hence, we approximate the heat equation by means of the ODE system

$$\begin{aligned} u_j'(t) &= \frac{u_{j-1}(t) - 2u_j(t) + u_{j+1}(t)}{h^2}, \qquad j = 1, \dots, n, \\ u_0(t) &= u_{n+1}(t) = 0, \end{aligned}$$

with the naturally defined initial condition

$$u_0(0) = u_{n+1}(0) = 0, \qquad u_j(0) = \boldsymbol{u}^0(x_j), \quad j = 1, \dots, n, .$$

Finally, we reduce the vector function to

$$u(t) = (u_1(t), \dots, u_n(t))^\top \in \mathbb{R}^n \tag{5.55}$$

by cutting off the solution components $u_0(t) = u_{n+1}(t) = 0$.

Therefore, we conclude that the heat equation is approximated by the linear initial value problem

$$u' = Au, \qquad u(0) \equiv u^0 \in \mathbb{R}^n, \qquad u_j^0 = \boldsymbol{u}^0(x_j), \quad j = 1, \dots, n\,, \tag{5.56}$$

where $A \in \mathbb{R}^{n \times n}$ is the tri-diagonal matrix

$$A = \frac{1}{h^2} \begin{pmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & 1 & -2 & 1 \\ 0 & \dots & 0 & 1 & -2 \end{pmatrix} \in \mathbb{R}^{n \times n}. \tag{5.57}$$

The eigenvalues $\lambda_j \in \sigma(A)$, $j = 1, \dots, n$, are real and satisfy the spectral condition (5.2). Both the eigenvalues $\lambda_j \in \sigma(A)$ and the corresponding eigenvectors $v_j \in \mathbb{R}^n$, $Av_j = \lambda_j v_j$, are known explicitly. In particular,

$$\lambda_j = -4\frac{\sin(j\pi h/2)}{h^2}, \qquad 0 > \lambda_1 > \cdots > \lambda_j > \cdots > \lambda_n. \tag{5.58}$$

Then, the stiffness ratio is given as

$$L = \frac{\sin(n\pi h/2)}{\sin(\pi h/2)}. \tag{5.59}$$

So, depending on the dimension of the space discretisation we get a different stiffness:

1. if $n = 100$, then $\lambda_{100} \approx -4.0794 \times 10^4$, $\lambda_1 \approx -9.8688$ and, hence, $L \approx 4.1336 \times 10^3$,
2. if $n = 1000$, then $\lambda_{1000} \approx -4.0080 \times 10^6$, $\lambda_1 \approx -9.8696$ and, hence, $L \approx 4.0610 \times 10^5$.

Even in the case $n = 100$, the problem (5.57) is not reasonably solvable by an *explicit* method with a fixed time step τ.

Bibliography

P. Deuflhard and F. Bornemann. *Scientific Computing with Ordinary Differential Equations*. Texts in Applied Mathematics. Springer, New York, 2012. DOI: 10.1007/978-0-387-21582-2.

E. Hairer. A Runge-Kutta Method of Order 10. *IMA Journal of Applied Mathematics*, 21(1): 47–59, 1978. DOI: 10.1093/imamat/21.1.47.

E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer Series in Computational Mathematics. Springer, Berlin, 2nd edition, 2010. DOI: 10.1007/978-0-387-21582-2.

E. Hairer, G. Wanner, and S. P. Nørsett. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer Series in Computational Mathematics. Springer-Verlag, Berlin, 2nd edition, 2009. DOI: 10.1007/978-0-387-21582-2.

N. J. Higham. *Functions of Matrices: Theory and Computation*. SIAM, 2008.

A. Katok and B. Hasselblatt. *Introduction to the Modern Theory of Dynamical Systems*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 1995.

J. Kurzweil. *Obyčejné diferenciální rovnice*. SNTL, Praha, 1973.

J. Kurzweil. *Ordinary Differential Equations*. Studies in Applied Mechanics. Elsevier, Amsterdam, 1986.

A. Quarteroni and F. Saleri. *Scientific Computing with MATLAB*. Texts in Computational Science and Engineering. Springer, Berlin, 2004. DOI: 10.1007/978-3-642-59339-0.

A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Texts in Applied Mathematics. Springer, Berlin, 2nd edition, 2010. DOI: 10.1007/b98885.

L. F. Shampine. *Numerical Solution of Ordinary Differential Equations*. Chapman & Hall/CRC, London, 1994.

L. F. Shampine and M. W. Reichelt. The MATLAB ODE suite. *SIAM Journal on Scientific Computing*, 18(1):1–22, 1997. DOI: 10.1137/S1064827594276424.

E. Süli and D. F. Meyers. *An Introduction to Numerical Analysis*. Cambridge University Press, Cambridge, 2003. DOI: 10.1017/CBO9780511801181.

Index