Numerical Analysis of Ordinary Differential Equations

Guido Kanschat

June 19, 2018

Preface

These notes are a short presentation of the material presented in my lecture. They follow the notes by Rannacher (Numerik 1 in German) as well as the books by Hairer, Nørsett, and Wanner [HNW93] and Hairer and Wanner [HW10]. Furthermore, I used the book by Deuflhard and Hohmann [DB08]. Historical remarks are in part taken from the article by Butcher [But96].

I am always thankful for hints and errata. But please verify that you have the latest version, which is available on github.

My thanks go to Dörte Jando, Markus Schubert, Lukas Schubotz, and David Stronczek for their help with writing and editing these notes.

Index for shortcuts

IVP	Initial value problem, s. definition **??** on page **??**
BDF	Backward differencing formula, s. example 5.0.4 on page 79
ODE	Ordinary differential equation
DIRK	Diagonal implicit Runge-Kutta method
ERK	Explicit Runge-Kutta method
IRK	Implicit Runge-Kutta method
LMM	Linear multistep method, s. Definition 5.1.1 on page 79
VIE	Volterra integral equation, s. Remark **??** on page ??

Index for symbols

$\mathbb{C}$	The set of complex numbers
e_i	The unit vector of $\mathbb{C}^d$ in direction d
$\Re$	Real part of a complex number
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^d$	The d-dimensional vectorspace of the real d-tuple
u	The exact solution of an ODE or IVP
u_k	The exact solution at time step t_k
y_k	The discrete solution at time step t_k
$\langle x, y \rangle$	The Euclidean scalar product in the space $\mathbb{R}^d$ or $\mathbb{C}^d$
$\lvert x \rvert$	The absolute value of a real number, the modulus of a complex number, or the Euclidean norm in $\mathbb{R}^d$ or $\mathbb{C}^d$, depending on its argument
$\lVert u \rVert$	A norm in a vector space (with exception of the special cases covered by $\lvert x \rvert$)

Contents

Chapter 1

Initial Value Problems and their Properties

1.1 Modeling with ordinary differential equations

Example 1.1.1 (Exponential growth)**.** Bacteria are living on a substrate with ample nutrients. Each bacteria splits into two after a certain time Δt. The time span for splitting is fixed and independent of the individuum. Then, given the amount u_0 of bacteria at time t_0, the amount at $t_1 = t_0 + \Delta t$ is $u_1 = 2u_0$. Generalizing, we obtain

$$u_n = u(t_n) = 2^n u_0, \qquad t_n = t_0 + n\Delta t.$$

After a short time, the number of bacteria will be huge, such that counting is not a good idea anymore. Also, the cell division does not run on a very sharp clock, such that after some time, divisions will not only take place at the discrete times $t_0 + n\Delta t$, but at any time between these as well. Therefore, we apply the continuum hypothesis, that is, u is not a discrete quantity anymore, but a continuous one that can take any real value. In order to accommodate for the continuum in time, we make a change of variables:

$$u(t) = 2^{\frac{t-t_0}{\Delta t}} u_0.$$

Here, we have already written down the solution of the problem, which is hard to generalize. The original description of the problem involved the change of u from one point in time to the next. In the continuum description, this becomes the derivative, which we can now compute from our last formula:

$$\tfrac{d}{dt}u(t) = \frac{\ln 2}{\Delta t} 2^{\frac{t-t_0}{\Delta t}} u_0 = \frac{\ln 2}{\Delta t} u(t).$$

We see that the derivative of u at a certain time depends on u itself at the same time and a constant factor, which we call the growth rate α. Thus, we have arrived at our first differential equation

$$u'(t) = \alpha u(t). \tag{1.1}$$

What we have seen as well is, that we had to start with some bacteria to get the process going. Indeed, any function of the form

$$u(t) = ce^{\alpha t}$$

is a solution to equation (1.1). It is the initial value u_0, which anchors the solution and makes it unique.

Example 1.1.2 (Predator-prey systems)**.** We add a second species to our bacteria example. Let's say, we replace the bacteria by sardines living in a nutrient rich sea, and we add tuna eating sardines. The amount of sardines eaten depends on the likelyhood that a sardine and a tuna are in the same place, and on the hunting efficiency β of the tuna. Thus, equation (1.1) is augmented by a negative change in population depending on the product of sardines u and tuna v:

$$u' = \alpha u - \beta uv.$$

In addition, we need an equation for the amount of tuna. In this simple model, we will make two assumptions: first, tuna die of natural causes at a death rate of γ. Second, tuna procreate if there is enough food (sardines), and the procreation rate is proportional to the amount of food. Thus, we obtain

$$v' = \delta uv - \gamma v.$$

Again, we will need initial populations at some point in time to compute ahead from there.

Remark 1.1.3. The Lotka-Volterra-equations have periodic solutions. Even though none of these exist in closed form the sulotions can be simulated: Lotka and Volterra became interested in this system as they had found that the amount of predatory fish caught had increased during World War I. During the war years there was a strong decrease of fishing effort. In conclusion, they thought, there had to be more prey fish.

A (far too rarely) applied consequence is that in order to diminish the amount of e.g. foxes one should hunt rabbits as foxes feed on rabbits.

Example 1.1.4 (Graviational two-body systems)**.** According to Newton's law of universal gravitation, two bodies of masses m_1 and m_2 attract each other with a force

$$\mathfrak{F}_1 = G\frac{m_1 m_2}{r^3}\mathfrak{r}_1,$$

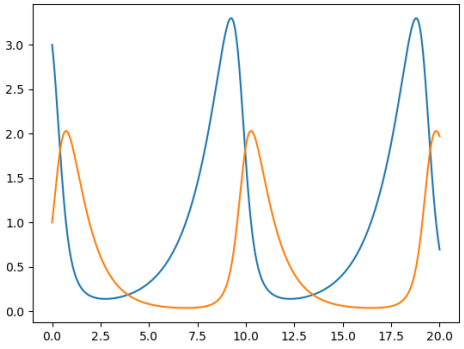


Figure 1.1: Plot of a solution to the Lotka-Volterra equation with parameters $\alpha = \frac{2}{3}$, $\beta = \frac{4}{3}$, $\delta = \gamma = 1$ and initial values $u(0) = 3$, $v(0) = 1$. Solved with a Runge-Kutta method of order five and step size $h = 10^{-5}$

where $\mathfrak{F}_1$ is the force vector acting on m_1 and $\mathfrak{r}_1$ is the vector pointing from m_1 to m_2 and $r = |\mathfrak{r}_1| = |\mathfrak{r}_2|$.

Newton's second law of motion on the other hand relates forces and acceleration:

$$\mathfrak{F} = m\mathfrak{x}'',$$

where $\mathfrak{x}$ is the position of a body in space.

Combining these, we obtain equations for the positions of the two bodies:

$$\mathfrak{x}_i'' = G\frac{m_{3-i}}{r^3}(\mathfrak{x}_i - \mathfrak{x}_{3-i}), \qquad i = 1, 2.$$

This is a system of 6 independent variables. Nevertheless, it can be reduced to three by using that the center of mass moves inertially. Then, the distance vector is the only variable to be computed for:

$$\mathfrak{r}'' = -G\frac{m}{r^3}\mathfrak{r}.$$

Intuitively, that we need an initial position and an initial velocity for the two bodies. Later on, we will see that this can actually be justified mathematically.

Example 1.1.5 (Celestial mechanics)**.** Now we extend the two-body system to a many-body system. Again, we subtract the center of mass, such that we obtain n sets of 3 equations for an $n+1$-body system. Since forces simply add up, this system becomes

$$\mathfrak{x}_i = -G\sum_{j\neq i}\frac{m_j}{r_{ij}^3}\mathfrak{r}_{ij}. \tag{1.2}$$

Here, $\mathfrak{r}_{ij} = \mathfrak{r}_j - \mathfrak{r}_i$ and $r_{ij} = |\mathfrak{r}_{ij}|$. Initial data for the solar system can be obtained from

`https://ssd.jpl.nasa.gov/?horizons`

1.2 Introduction to initial value problems

1.2.1 Definition (Ordinary differential equations): An **ordinary differential equation** (ODE) is an equation for a function $u(t)$, defined on an interval $I \subset \mathbb{R}$ and with values in the real or complex numbers or in the space $\mathbb{R}^d$ ($\mathbb{C}^d$), of the form

$$F\big(t, u(t), u'(t), u''(t), \ldots, u^{(n)}(t)\big) = 0. \tag{1.3}$$

Here $F(\ldots)$ denotes an arbitrary function of its arguments. The **order** n of a differential equation is the highest derivative which occurs. If the dimension d of the value range of u is higher than one, we talk about systems of differential equations.

Remark 1.2.2. A differential equation, which is not ordinary, is called partial. These are equations or systems of equations, which involve partial derivatives with respect to several independent variables. While the functions in an ordinary differential equation may be dependent on additional parameters, derivatives are only taken with respect to one variable, typically, but not exclusively, this variable is time. Due to the fact that this manuscript just deals with ordinary differential equations, the adjective will be omitted in the following.

1.2.3 Definition: An **explicit differential equation** of first order is a equation of the form

$$\begin{aligned} u'(t) &= f(t, u(t)) \\ \text{or shorter:} \qquad u' &= f(t, u). \end{aligned} \tag{1.4}$$

A differential equation of order n is called explicit, if it is of the form

$$u^{(n)}(t) = F\left(t, u(t), u'(t), \ldots, u^{(n-1)}(t)\right)$$

1.2.4 Lemma: Every differential equation of higher order can be written as a system of first-order differential equations. If the equation is explicit, then the system is explicit.

Proof. By the introduction of additional variables $u_0(t) = u(t)$, $u_1(t) = u'(t)$ to $u_{n-1}(t) = u^{(n-1)}(t)$, each differential equation of order n can be transformed into a system of n differential equations of first order. This system has the form

$$\begin{pmatrix} u_0'(t) - u_1(t) \\ u_1'(t) - u_2(t) \\ \vdots \\ u_{n-2}'(t) - u_{n-1}(t) \\ F\big(t, u_0(t), u_1(t), \dots, u_{n-1}(t), u_{n-1}'(t)\big) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}. \tag{1.5}$$

In the case of an explicit equation, the system has the form

$$\begin{pmatrix} u_0'(t) \\ u_1'(t) \\ \vdots \\ u_{n-2}'(t) \\ u_{n-1}'(t) \end{pmatrix} = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_{n-1}(t) \\ F\big(t, u_0(t), u_1(t), \dots, u_{n-1}(t)\big) \end{pmatrix}. \tag{1.6}$$

□

Example 1.2.5. The differential equation

$$u'' + \omega^2 u = f(t) \tag{1.7}$$

can be transformed into the system

$$\begin{aligned} u_1' - u_2 &= 0, \\ u_2' + \omega^2 u_1 &= f(t). \end{aligned} \tag{1.8}$$

The transformation is not uniquely determined. In this example, a more symmetric system can be obtained:

$$\begin{aligned} u_1' - \omega u_2 &= 0, \\ u_2' + \omega u_1 &= f(t). \end{aligned} \tag{1.9}$$

From a numerical perspective, system 1.9 should be chosen over refeq:awa:18 to avoid loss of significance or overflow, i.e. if $|\omega| \ll 1$ or $|\omega| \gg 1$.

1.2.6 Definition: A differential equation of the form (1.4) is called **autonomous**, if the right hand side f is not explicitly dependent on t, i.e.

$$u' = F(u). \tag{1.10}$$

Each differential equation can be transformed into an autonomous differential equation. This is called **autonomization**.

$$U = \begin{pmatrix} u \\ t \end{pmatrix}, \qquad F(U) = \begin{pmatrix} f(t,u) \\ 1 \end{pmatrix}, \qquad U' = F(U)$$

A method which provides the same solution for the autonomous differential equation as for the original IVP, is called **invariant under autonomization**.

Differential equations usually provide sets of solutions from which we have to choose a solution. An important selection criteria is setting an initial value which leads to a well-posed problem (see below).

1.2.7 Definition: Given a point $(t_0, u_0) \in \mathbb{R} \times \mathbb{R}^d$. Furthermore, let the function $f(t,u)$ with values in $\mathbb{R}^d$ be defined in a neighborhood $I \times U \subset \mathbb{R} \times \mathbb{R}^d$ of the initial value. Then an **initial value problem** (IVP) is defined as follows: find a function $u(t)$, such that

$$u'(t) = f\big(t, u(t)\big) \tag{1.11a}$$
$$u(t_0) = u_0 \tag{1.11b}$$

1.2.8 Definition: We call a continuously differentiable function $u(t)$ with $u(t_0) = 0$ a **local solution** of the IVP (1.11), if there exists a neighborhood J of the point in time t_0 in which u and $f(t, u(t))$ are defined and if the equation (1.11a) holds for all $t \in J$.

Remark 1.2.9. We introduced the IVP deliberately in a "local" form because the local solution term is the most useful one for our purpose. Due to the fact that the neighborhood J in the definition above can be arbitrarily small, we will have to deal with the extension to larger intervals below.

Remark 1.2.10. Through the substitution of $t \mapsto \tau$ with $\tau = t - t_0$ it is possible to transform every IVP at the point t_0 to a IVP in point 0. We will make use of this fact and soon always assume $t_0 = 0$.

1.2.11 Lemma: Under the assumption that the right hand side f is continuous in both arguments, the function $u(t)$ is a solution of the initial value problem (1.11) if and only if it is a solution of the **Volterra integral equation** (VIE)

$$u(t) = u_0 + \int_{t_0}^{t} f\big(s, u(s)\big)\,\mathrm{d}s. \tag{1.12}$$

The formulation as integral equation allows on the other hand a more general solution term, because the problem is already well-posed for functions $f(t,u)$, which are just integrable with respect to t. In that case the solution u would be just absolutely continuous and not continuously differentiable.

Remark 1.2.12. Both the theoretical analysis of the IVP and the numerical methods (with exception of the BDF methods) in this lecture notes, solve actually never the IVP (1.11) but always the associated integral equation (1.12).

1.2.13 Theorem (Peano's existence theorem): Let the function $f(t,u)$ be continuous on the closed set

$$\overline{D} = \big\{(t,u) \in \mathbb{R} \times \mathbb{R}^d \;\big|\; |t-t_0| \le \alpha,\ |u-u_0| \le \beta\big\},$$

where $\alpha, \beta > 0$. Then there exists a solution $u(t) \in C^1(I)$ on the interval $I = [t_0 - T, t_0 + T]$ with

$$T = \min\left(\alpha, \frac{\beta}{M}\right),\ M = \max_{(t,u)\in\overline{D}} |f(t,u)|.$$

The proof of this theorem is of little consequence for the remainder of these notes. For its verification, we refer to textbooks on the theory of ordinary differential equations.

Remark 1.2.14. The Peano existence theorem does not make any statements about the uniqueness of a solution and also just guarantees local existence. The second limitation is addressed by the following theorem. The first will be postponed to section 1.4.

1.2.15 Theorem (Peano's continuation theorem): Let the assumptions of Theorem 1.2.13 hold. Then, the solution can be extended to an interval $I_m = [t_-, t_+]$ such that the points $\big(t_-, u(t_-)\big)$ and $\big(t_+, u(t_+)\big)$ are on the boundary of $\overline{D}$. Neither the values of t, nor of $u(t)$ need to be bounded as long as f remains bounded.

Example 1.2.16. The IVP

$$u' = 2\sqrt{|u|}, \qquad u(0) = 0,$$

has solutions $u(t) = t^2$ and $u(t) = 0$.

Example 1.2.17. The functions $1/(t - t_0)$ are solutions to the IVP

$$u' = -u^2, \qquad u(t_0) = 1.$$

1.3 Linear differential equations and Grönwall's inequality

1.3.1. The examination of linear differential equation turns out to be particularly simple. On the other hand, results obtained here will provide us with important statements for general non-linear IVP. Therefore we pay particular attention to the linear case.

1.3.2 Definition: An IVP according to definition 1.2.7 is called **linear** if the right hand side f is an affine function of u. Thus, we can write it in the form

$$u'(t) = A(t)u(t) + b(t) \qquad \forall t \in \mathbb{R} \tag{1.13a}$$
$$u(t_0) = u_0 \tag{1.13b}$$

with a continuous matrix function $A : \mathbb{R} \to \mathbb{C}^{d\times d}$. If in addition $b(t) \equiv 0$, we call it **homogeneous**.

factor.tex factor.tex

1.3.3 Definition: Let the matrix function $A : I \to \mathbb{C}^{d\times d}$ be continuous. Then the function defined by

$$M(t) = \exp\left(-\int_{t_0}^{t} A(s)\,\mathrm{d}s\right) \tag{1.14}$$

is called **integrating factor** of the equation (1.13a).

factor.tex

Corollary 1.3.4. *The integrating factor $M(t)$ has the properties*

$$M(t_0) = \mathbb{I} \tag{1.15}$$
$$M'(t) = -M(t)A(t). \tag{1.16}$$

1.3.5 Lemma: A solution of the IVP (1.13) is given through the representation

$$u(t) = M(t)^{-1}\left(u_0 + \int_{t_0}^{t} M(s)b(s)\,\mathrm{d}s\right) \tag{1.17}$$

with the integrating factor $M(t)$ of the equation (1.14). This solution exists for all $t \in \mathbb{R}$.

Proof. We consider the auxiliary function $w(t) = M(t)u(t)$ with the integrating factor $M(t)$ of the equation (1.14). Using the chain rule, there holds

$$w'(t) = M(t)u'(t) + M'(t)u(t) = M(t)u'(t) - M(t)A(t)u(t). \tag{1.18}$$

Comparing this to the differential equation (1.13a), we see that w solves

$$w'(t) = M(t)b(t).$$

This can be integrated directly to obtain

$$w(t) = u_0 + \int_{t_0}^{t} M(t)b(t),$$

where we use that $w(t_0) = u_0$. According to lemma A.1.3, about the matrix exponential, $M(t)$ is invertible for all t. With the definition of $w(t)$ we are therefore able to solve for $u(t)$, which results in the equation (1.17). The global solvability follows from the fact that the solution is defined for arbitrary $t \in \mathbb{R}$. □

Example 1.3.6. The equation in example 1.2.5 is linear and can be written in the form of (1.13) with

$$A(t) = A = \begin{pmatrix} 0 & \omega \\ -\omega & 0 \end{pmatrix}$$
$$b(t) = f(t).$$

Let now $f(t) \equiv 0$. The Jordan canonical form of A is

$$A = C^{-1}\begin{pmatrix} \omega i & \\ & -\omega i \end{pmatrix} C$$

with a suitable transformation matrix C. The integrating factor is

$$M(t) = e^{At} = C^{-1}\begin{pmatrix} e^{\omega i} & \\ & e^{-\omega i} \end{pmatrix} C = \begin{pmatrix} \cos\omega t & \sin\omega t \\ -\sin\omega t & \cos\omega t \end{pmatrix}.$$

Thus, given an initial value $(u_0, v_0)^T$, the solution is

$$u(t) = \begin{pmatrix} \cos \omega t & \sin \omega t \\ -\sin \omega t & \cos \omega t \end{pmatrix} \begin{pmatrix} u_0 \\ v_0 \end{pmatrix}.$$

The missing details in this argument and the case for an inhomogenety $f(t) = \cos \alpha t$ are left as an exercise.

Remark 1.3.7. If the function $b(t)$ in (1.13a) is only integrable, the function $u(t)$ defined in (1.17) is absolutely continuous and thus differentiable almost everywhere. The chain rule (1.18) is applicable in all points of differentiability and $w(t)$ solves the Volterra integral equation corresponding to (1.13). Thus, the representation formula (1.17) holds generally for solutions of linear Volterra integral equations.

1.3.8 Lemma (Grönwall): Let be $w(t)$, $a(t)$ and $b(t)$ be nonnegative, integrable functions, such that $a(t)w(t)$ is integrable. Furthermore, let $b(t)$ be monotonically nondecreasing and let $w(t)$ satisfy the integral inequality

$$w(t) \leq b(t) + \int_{t_0}^{t} a(s)w(s)\,\mathrm{d}s, \qquad t \geq t_0. \tag{1.19}$$

Then, for almost all $t \geq t_0$ there holds:

$$w(t) \leq b(t) \exp\left(\int_{t_0}^{t} a(s)\,\mathrm{d}s\right). \tag{1.20}$$

Proof. Using the integrating factor

$$m(t) = \exp\left(-\int_{t_0}^{t} a(s)\,\mathrm{d}s\right), \quad \frac{1}{m(t)} = \exp\left(\int_{t_0}^{t} a(s)\,\mathrm{d}s\right),$$

we introduce the auxiliary function

$$v(t) = m(t) \int_{t_0}^{t} a(s)w(s)\,\mathrm{d}s,$$

This function is absolutely continuous and almost everywhere

$$v'(t) = m(t)a(t)\left[w(t) - \int_{t_0}^{t} a(s)w(s)\,\mathrm{d}s\right].$$

By assumption (1.19), the bracket on the right is bounded by $b(t)$. Thus,

$$v'(t) \leq m(t)a(t)b(t)$$

and since $v(t_0) = 0$ by its definition,

$$v(t) \leq \int_{t_0}^{t} m(s)a(s)b(s)\,\mathrm{d}s.$$

From the definition of $v(t)$, we obtain

$$\int_{t_0}^{t} a(s)w(s)\,\mathrm{d}s = \frac{1}{m(t)}v(t) \leq \frac{1}{m(t)}\int_{t_0}^{t} m(s)a(s)b(s)\,\mathrm{d}s$$

Finally, since $b(t)$ is nondecreasing we obtain almost everywhere

$$\begin{aligned}\int_{t_0}^{t} a(s)w(s)\,\mathrm{d}s &\leq \frac{b(t)}{m(t)}\int_{t_0}^{t} a(s)\exp\left(-\int_{t_0}^{s} a(r)\,\mathrm{d}r\right)\mathrm{d}s \\ &= \frac{b(t)}{m(t)}\left[-\exp\left(-\int_{t_0}^{s} a(r)\,\mathrm{d}r\right)\right]_{t_0}^{t} \\ &= \frac{b(t)}{m(t)}\big(m(t_0) - m(t)\big) = \frac{b(t)}{m(t)} - b(t)\end{aligned}$$

Now, entering into the integral inequality (1.19), we obtain

$$w(t) \leq b(t) + \int_{t_0}^{t} a(s)w(s)\,\mathrm{d}s = \frac{b(t)}{m(t)},$$

which proves the lemma. □

Remark 1.3.9. On the form of the requirements (1.19) as well as the estimation (1.20), we can see that Grönwall's inequality is basically based on the construction of a majorant for $w(t)$, which satisfies a linear IVP.

1.3.10 Corollary: Let the functions $u(t)$ and $v(t)$ be two solutions of the linear differential equation (1.13a). If both functions coincide in a point t_0 then they are identical.

Proof. The difference $w(t) = v(t) - u(t)$ solves the integral equation

$$w(t) = \int_{t_0}^{t} A(s)w(s)\,\mathrm{d}s.$$

Hence $|w(t)|$ satisfies the integral inequality

$$|w(t)| \leq \int_{t_0}^{t} |A(s)||w(s)|\,\mathrm{d}s,$$

from which we conclude with Grönwall's inequality (1.20) for $b(t) = 0$, that $|w(t)| = 0$ for all t and therefore $u(t) = v(t)$. □

Corollary 1.3.11. *The representation formula* (1.17) *in Lemma 1.3.5 defines the unique solution to the IVP* (1.13)*. In particular, solutions of linear IVP are always defined on the whole real axis.*

Example 1.3.12. Let $A \in \mathbb{C}^{d\times d}$ be diagonalizable with possibly repeated eigenvalues $\lambda_1, \dots, \lambda_d$ and corresponding eigenvectors $\psi^{(i)}$. Let Ψ be the matrix of column vectors $\psi^{(i)}$. Then, the solution of the IVP

$$\begin{aligned} u' &= Au, \\ u(0) &= u_0, \end{aligned}$$

is given by the formula

$$u(t) = e^{At}u_0 = \Psi \exp \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{pmatrix} \Psi^{-1}u_0.$$

This is due to the fact, that $M(t) = e^{-At}$ and $e^{-\Psi A\psi^{-1}t} = \Psi e^{-At}\Psi^{-1}$.

1.3.13 Lemma: The solutions of the homogeneous, linear differential equation

$$u'(t) = A(t)u(t) \tag{1.21}$$

with $u : \mathbb{R} \to \mathbb{R}^d$, define a vector space of dimension d. Let $\{\psi^{(i)}\}_{i=1,\dots,d}$ be a basis of $\mathbb{R}^d$. Then the solutions $\varphi^{(i)}(t)$ of the equation (1.21) with initial values $\varphi^{(i)}(0) = \psi^{(i)}$ form a basis of the solution space. The vectors $\{\varphi^{(i)}(t)\}$ are linear independent for all $t \in \mathbb{R}$.

Proof. At first we observe that for two solutions $u(t)$ and $v(t)$ of the equation (1.21), their sum and their scalar multiples are solutions too, due to linearity of the derivative and the right hand side. Therefore the vector space structure is proven.

Let now $\varphi^{(i)}(t)$ be solutions of the IVP with linear independent initial values $\{\psi^{(i)}\}$. As a consequence the functions are linear independent as well.

Assume that $w(t)$ is a solution of the equation (1.21), which cannot be written as a linear combination of $\psi^{(i)}$. Then $w(0)$ is not a linear combination of the vectors $\psi^{(i)}$: else let's say $w(0) = \sum \alpha_i\psi^{(i)}$, then $w(t) = \sum \alpha_i\varphi^{(i)}(t)$ would be a linear combination because of uniqueness proven in corollary 1.3.10. Since $\{\psi^{(i)}\}$ according to the assumtions is a basis of $\mathbb{R}^d$, such a $w(0)$ cannot exist. Hence it is shown that $\varphi^{(i)}(t)$ is a basis of the solution space of dimension d.

It remains to show that the $\varphi^{(i)}(t)$ are linearly independent for all $t \in \mathbb{R}$. To this end, assume that the set $\varphi^{(i)}(t)$ is linearly dependent for a value t_1. Then the following holds true without loss of generality

$$\varphi^{(d)}(t_1) = \sum_{i=1}^{d-1} \alpha_i \varphi^{(i)}(t_1) =: w(t).$$

Again according to corollary 1.3.10 we have $\varphi^{(d)} \equiv w$, moreover $\varphi^{(d)}(0) = w(0)$ which again is a contradiction to the assumption $\psi^{(d)}$ is a linear combination of the other initial values. □

1.3.14 Definition: A basis $\{\varphi^{(1)}, \ldots, \varphi^{(d)}\}$ of the solution space of the linear differential equation (1.21), in particular the basis with initial values $\varphi^{(i)}(0) = e_i$, is called **fundamental system** of solutions. The matrix function

$$Y(t) = \left(\varphi^{(1)}(t) \ldots \varphi^{(d)}(t)\right) \tag{1.22}$$

with column vectors $\varphi^{(i)}(t)$ is called **fundamental matrix**.

1.3.15 Corollary: The fundamental matrix is regular for all $t \in \mathbb{R}$ and solves the IVP

$$\begin{aligned} Y'(t) &= A(t)Y(t) \\ Y(0) &= \mathbb{I}. \end{aligned}$$

Proof. The initial value is part of the definition. On the other hand, splitting the the matrix valued IVP into its column vectors, we obtain the original IVP defining the solution space. Regularity follows from linear independence of solutions for any t. □

1.4 Well-posedness of the IVP

1.4.1 Definition: A mathematical problem is called **well-posed** if the following **Hadamard conditions** are satisfied:

1. A solution exists.
2. The solution is unique.
3. The solution is continuously dependent on the data.

The third condition in this form is purely qualitative. Typically, in order to characterize problems with good approximation properties, we will require Lipschitz continuity, which has a more quantitative character.

Example 1.4.2. The IVP

$$u' = \sqrt[3]{u}, \qquad u(0) = 0,$$

has solutions of the form

$$u(t) = \begin{cases} 0 \\ c\left(\frac{2}{3}t\right)^{3/2}. \end{cases}$$

Thus, the solution is not unique and therefore, the IVP is not well-posed. Let now the initial value be nonzero, but slightly positive. Then, a small perturbation, which changes its sign, will have dramatic effect on the solution.

1.4.3 Definition: The function $f(t, y)$ satisfies on its domain $D = I \times \Omega \subset \mathbb{R} \times \mathbb{R}^d$ an uniformly continuous **Lipschitz condition** if it is Lipschitz continuous with regard to y, i.e., it exists a positive constant L, such that

$$\forall t \in I;\, x, y \in \Omega \;:\; |f(t, x) - f(t, y)| \leq L|x - y| \tag{1.23}$$

It satisfies a local Lipschitz condition if the same holds true for all compact subsets of D.

Example 1.4.4. Let $f(t, u) \in C^1(\mathbb{R} \times \mathbb{R}^d)$ and let all partial derivatives with respect to components of u be bounded by

$$\max_{\substack{t\in\mathbb{R} \\ u\in\mathbb{R}^d \\ 1\leq i,j\leq d}} |\frac{\partial}{\partial u_i} f_j(t, u)| \leq K.$$

Then, f satisfies the Lipschitz condition (1.23) with $L = K$. Indeed, by using Taylor expansion, we see that

$$\begin{aligned} f_j(t,u) - f_j(t,v) &= \int_0^1 \frac{d}{ds} f_j\big(t, u + s(v-u)\big)\,\mathrm{d}s \\ &= \int_0^1 \sum_{i=1}^{d} (u_i - v_i)\partial_i f_j\big(t, u + s(v-u)\big)\,\mathrm{d}s. \end{aligned}$$

It is an easy conclusion that

$$|f(t,u) - f(t,v)| \le K|u - v|.$$

1.4.5 Theorem (Stability): Let $f(t,u)$ and $g(t,u)$ be two continuous functions on a cylinder $D = I \times \Omega$ where the interval I contains t_0 and Ω is a convex set in $\mathbb{R}^d$. Furthermore, let f admit a Lipschitz condition with constant L on D. Let u and v be solutions to the IVP

$$u' = f(t,u) \quad \forall t \in I, \qquad u(t_0) = u_0, \tag{1.24}$$
$$v' = g(t,v) \quad \forall t \in I, \qquad v(t_0) = v_0. \tag{1.25}$$

Then, there holds

$$|u(t) - v(t)| \le e^{L|t-t_0|}\left[|u_0 - v_0| + \int_{t_0}^{t} \max_{x\in\Omega} |f(s,x) - g(s,x)|\,\mathrm{d}s\right]. \tag{1.26}$$

Proof. Both $u(t)$ and $v(t)$ solve their respective Volterra integral equations. Taking the difference, we obtain

$$\begin{aligned} u(t) - v(t) &= u_0 - v_0 + \int_{t_0}^{t} \big[f(s,u(s)) - g(s,v(s))\big]\,\mathrm{d}s \\ &= u_0 - v_0 + \int_{t_0}^{t} \big[f(s,u(t)) - f(s,v(s))\big]\,\mathrm{d}s + \int_{t_0}^{t} \big[f(s,v(t)) - g(s,v(s))\big]\,\mathrm{d}s. \end{aligned}$$

Thus, its norm admits the integral inequality

$$\begin{aligned} |u(t) - v(t)| &\le |u_0 - v_0| + \int_{t_0}^{t} |f(s,u(t)) - f(s,v(s))|\,\mathrm{d}s + \int_{t_0}^{t} |f(s,v(t)) - g(s,v(s))|\,\mathrm{d}s \\ &\le \underbrace{|u_0 - v_0| + \int_{t_0}^{t} \max_{x\in\Omega} |f(s,x) - g(s,x)|\,\mathrm{d}s}_{b(t)} + \int_{t_0}^{t} L|u(s) - v(s)|\,\mathrm{d}s. \end{aligned}$$

This inequality is in the form of the assumption in Grönwall's lemma, and its application yields the stability result. □

1.4.6 Theorem (Picard-Lindelöf): Let $f(t, y)$ be continuous on a cylinder

$$D = \{(t, y) \in \mathbb{R} \times \mathbb{R}^d |\ |t - t_0| \leq a, |y - u_0| \leq b\}.$$

Let f be bounded such that there is a constant $M = \max_D |f|$ and satisfy the Lipschitz condition (1.23) with constant L on D. Then the IVP

$$\begin{aligned} u' &= f(t, u) \\ u(t_0) &= u_0 \end{aligned}$$

is uniquely solvable on the interval $I = [t_0 - T, t_0 + T]$ where $T = \min\{a, \frac{b}{M}\}$.

Proof. First, we assume for simplicity $t_0 = 0$ or we transform the problem accordingly. Abbreviate $I = [-T, T]$ and

$$\Omega = \big\{x \in \mathbb{R}^d \big| |x - u_0| \leq b\big\}.$$

We introduce the operator $F(u)$ which is defined through the Volterra integral equation (1.12) as

$$F(u)(t) = u_0 + \int_0^t f(s, u(s))\, \mathrm{d}s. \tag{1.27}$$

Obviously u is a solution of the Volterra integral equation (1.12) if and only if u is a fixed point of F i.e., $u = Fu$. We can obtain such a fixed-point by the iteration $u^{(k+1)} = F(u^{(k)})$ with some initial guess $u^{(0)} : I \to \Omega$. From the boundedness of f, we obtain for $t - t_0 \leq T$

$$|u^{(k+1)}(t) - u_0| = |\int_{t_0}^t f(s, u^{(k)}(s))\, \mathrm{d}s| \leq \int_{t_0}^t |f(s, u^{(k)}(s))|\, \mathrm{d}s \leq TM \leq b.$$

Thus, from $u^{(0)} : I \to \Omega$ follows $u^{(k)} : I \to \Omega$ for all k and the iteration is well-defined.

We now show that F is a contraction under the assumtions of the theorem. We follow the technique in [Heu86, §117] and choose on the space $\mathcal{C}(I)$, which is the space of the continuous functions on I, the norm

$$\|u\|_e := \max_{t \in I}\ e^{-2Lt} |u(t)|.$$

With estimating the difference of operator F applied to two functions:

$$\begin{aligned}
|F(u)(t) - F(v)(t)| &= \left| u_0 - u_0 + \int_0^t (f(s,u(s)) - f(s,v(s)))\,\mathrm{d}s \right| \\
&\leq \int_0^t \left| f\big(s,u(s)\big) - f\big((s,v(s)\big) \right| \mathrm{d}s \\
&\leq \int_0^t L|u(s) - v(s)| \underbrace{e^{-2Ls}e^{2Ls}}_{=1} \,\mathrm{d}s \\
&\leq L\|u - v\|_e \int_0^t e^{2Ls}\,\mathrm{d}s \\
&= L\|u - v\|_e \frac{e^{2Lt} - 1}{2L} \\
&\leq \frac{1}{2} e^{2Lt} \|u - v\|_e .
\end{aligned}$$

It follows

$$e^{-2Lt}|F(u)(t) - F(v)(t)| \leq \frac{1}{2}\|u - v\|_e,$$

for all t and we observe:

$$|F(u)(t) - F(v)(t)|_e \leq \frac{1}{2}\|u - v\|_e.$$

Thus, we have shown that F is a contraction on the space of the continuous functions with the norm $\|.\|_e$. Therefore, we can apply the Banach fixed-point theorem, concluding that F has exactly one fixed-point. This proves the theorem. □

Remark 1.4.7. The norm $\|u\|_e$ had been chosen with regard to Grönwall's inequality, which was not used in the proof explicitly. It is equivalent to the norm $\|u\|_\infty$ because e^{-2Lt} is strictly positiv and bounded. On the other hand one could have performed the proof with some more calculations with respect to the ordinary Tchebychev distance (maximum norm) $\|u\|_\infty$.

Remark 1.4.8. Currently our solution is restricted to $I = [t_0 - T, t_0 + T]$. Since T is chosen in such a way in equation 1.4.6 that the graph of u does not leave the domain, this extension always ends on the boundary of D. One can now extend the solution by solving the next IVP $\left\{ \begin{array}{c} u' = f(t,u) \\ u(t_1) = u_1 \end{array} \right\}$ on the interval I_1. This way one obtains a solution on $I \cup I_1 \cup I_2 \cup ....$

Corollary 1.4.9. *Let the function $f(t,u)$ admit the Lipschitz condition on $\mathbb{R}\times\mathbb{C}^d$. Then, the IVP has a unique solution on the whole real axis.*

Proof. The boundedness was used in order to guarantee that $u(t)\in\Omega$ for any t. This is not necessary anymore, if $\Omega=\mathbb{C}^d$. Thus, the limitation of the interval I becomes unnecessary as well. Finally, the fixed point argument does not depend on boundedness of the set. □

Chapter 2

Explicit One-Step Methods and Convergence

2.1 Introduction

Example 2.1.1 (Euler's method)**.** We begin this section with the method which serves as prototype for a whole class of schemes which solves an IVP or rather the Volterra integral equation numerically. Here, as always for problems with infinite dimensional solution spaces, numerical solution refers to finding an approximation by applying a discretization method, and studying the error of this method.

Consider the following problem: given an IVP of the form (1.11), calculate the value $u(T)$ at a later point in time T.

To this end, we note first of all that for an IVP at the initial point 0, not only the function value $u(0) = u_0$ is known, but also the derivative $u'(0) = f(0, u_0)$. Thus we are capable to replace the solution $u(t)$ in blue by a straight line $y(t)$ in red, which we can see on the left of Figure 2.1. The figure suggests that in general the accuracy of this method may not be very good. The first improvement is that we do not draw the line through the whole interval from 0 to T. Instead, we insert intermediate points and apply the method to each subinterval, where we use the result of a previous interval as the initial point for the next subinterval. As a result one obtains a chain of straight lines and the so-called **Euler method**.

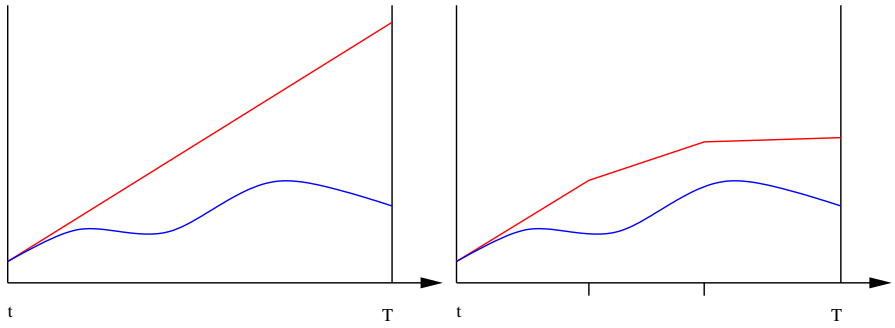
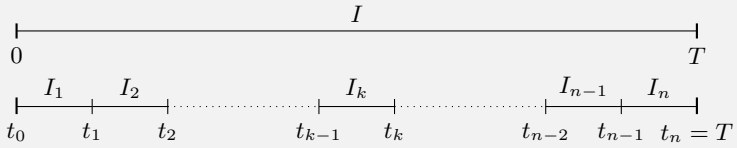


Figure 2.1: Derivation of the Euler method. Left: replacement of the solution of the IVP by a line with slope and initial point given by the IVP. Right: Euler method with three subintervals.

2.1.2 Definition: On a time interval $I = [0, T]$, we define a partitioning in n subintervals, also known as **time steps**. Here we choose the following notation:



The time steps $I_k = [t_{k-1}, t_k]$ have the step size $h_k = t_k - t_{k-1}$. A partitioning in n time steps implies $t_n = T$. The term k-th time step is used for both the interval I_k and for the point in time t_k, but it should always be clear through context which one is meant.
Very often, we will consider evenly spaced time steps, in which case we denote the step size by h and $h_k = h$ for all k.

Definition 2.1.3. In the following chapters we will regularly compare the solution of an IVP with the results of discretization methods. Therefore, we introduce the following convention for notations and symbols.

The solution of the IVP is called the **exact** or **continuous solution**. The term "continuous" indicates here the solution of the non-discretized problem. Its symbol is in general u and we set as abbreviation

$$u_k = u(t_k).$$

If u is vector-valued we also use the alternative superscript $u^{(k)}$ and $u_i^{(k)}$ for a entry of the vector $u(t_k)$.

In general we write the **discrete solution** with the symbol y. We write y_k or $y^{(k)}$ for the value of the discrete solution at the point in time t_k. In contrast to

the continuous solution, y only defined at discrete time steps, unless for special methods discussed later.

2.1.4 Definition (Explicit one-step method): An **explicit one-step method** is a method which, given u_0 at $t_0 = 0$ computes a sequence of approximations $y_1 \ldots, y_n$ to the solution of an IVP in the time steps $t_1, \ldots, t_n$ using an update formula of the form[a]

$$y_k = y_{k-1} + h_k F_{h_k}(t_{k-1}, y_{k-1}). \tag{2.1}$$

The function $F()_{h_k}$ is called **increment function**. We will often omit the index h_k on $F_{h_k}()$ because it is clear that the method is always applied to time intervals.
The method is called **one-step method** because the value y_k explicitly depends only of the values y_{k-1} and $f(t_{k-1}, y_{k-1})$, not on previous values.

[a] The adjective 'explicit' is here in contrast to 'implicit' one-step methods, where the increment function depends on y_k and equation (2.1) must be solved for y_k.

Remark 2.1.5. For one-step methods every step is *per definitionem* similar. Therefore, it is sufficient to consider the first step only. Hence, we will define and analyze methods by stating the dependence of y_1 on y_0 which then can be transferred to the general step from y_{n-1} to y_n. The general one-step method above then reduces to

$$y_1 = y_0 + hF(t_0, y_0).$$

This implies that the values y_k with $k \geq 2$ are computed through formula (2.1) with the respective h_k and the same increment function.

2.1.6 Example: Given the IVP

$$u' = u, \qquad u(0) = 1,$$

the solution is $u(t) = e^t$. The Euler method reads

$$y_1 = y_0 + hy_0.$$

The results for $h = 1$ and $h = 1/2$ are:

exact		$h = 1$			$h = 1/2$		
$t = 0$	1	y_0	1		y_0	1	
$t = 1$	2.71828	y_1	2	0.718	y_2	2.25	0.468
$t = 2$	7.38906	y_2	4	3.389	y_4	5.0625	2.236
$t = k$	2.71828^k	y_k	2^k		y_{2k}	2.25^k	

We note that the error is growing in time. The approximation of the solution can be improved by shrinking h from 1 to $1/2$. The goal of error analysis will be establishing these dependencies.

2.2 Error analysis

Remark 2.2.1. In Figure 2.1, we observe that the error consists of two parts at a given time t_{k+1}. First, an error on the interval I_k due to replacing the differential equation by the discrete method. Second, we have to add the error which results from the fact that our initial value y_k is already not exact due to previous errors. This situation is displayed in Figure 2.2. After one time step, a local error has appeared. In the second time step, we already start with an erroneous initial value. Therefore, we split the error into the local error and an accumulated error. The local error compares continuous and discrete solutions on a single interval with the same initial value. In the analysis, we will have the options of using the exact (right figure) or the approximated initial value (left figure).

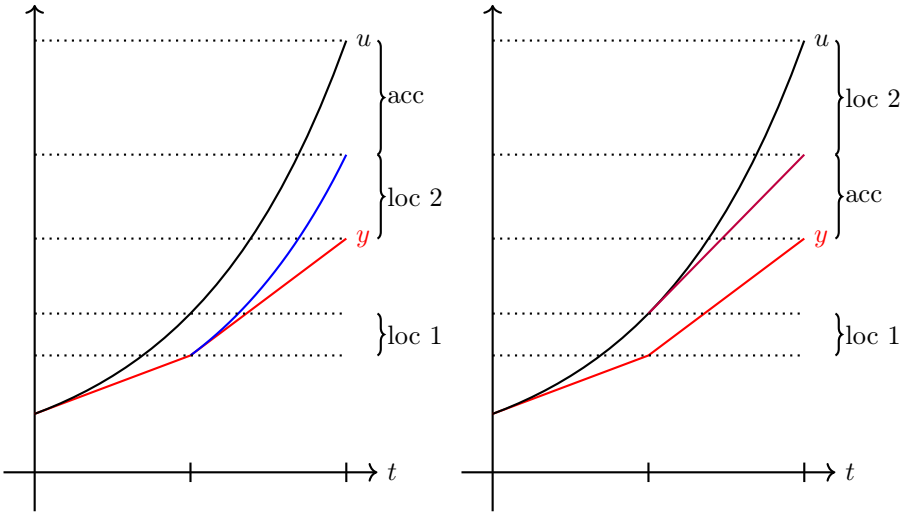


Figure 2.2: Local and accumulated errors. Exact solution in black, the Euler method in red. On the left, in blue the exact solution of an IVP on the second interval with initial value y_1. On the right, in purple the second step of the Euler method, but with exact initial value u_1.

2.2.2 Definition: Let u be a solution of the differential equation $u' = f(t,u)$ on the interval $I_n = [t_{n-1}, t_n]$. Then, the **local error** of a discrete method F is the difference between the solution u_n of the differential equation at t_n and the result of one time step (2.1) with this method with exact initial value:

$$\eta_n = \eta_n(u) = u_n - \big[u_{n-1} + h_n F_{h_n}(t_{n-1}, u_{n-1})\big]. \tag{2.2}$$

The **truncation error** is the quotient of the local error and h_n:

$$\tau_n = \tau_n(u) = \frac{u_n - u_{n-1}}{h_n} - F_{h_n}(t_{n-1}, u_{n-1}). \tag{2.3}$$

The one-step method $F_h(t,y)$ is **consistent** of **order** p with the ODE, if there is a constant c independent of h such that for $h \to 0$:

$$\max_n |\tau_n| \le c h^p \tag{2.4}$$

Example 2.2.3 (Euler method)**.** To find out the order of consistency of the Euler method, we consider the Taylor expansion of the solution at the point t_{n-1}:

$$u(t_n) = u(t_{n-1}) + h_n u'(t_{n-1}) + \frac{1}{2} h_n^2 u''(\zeta)$$

As a result the truncation error reduces to:

$$\begin{aligned}\tau_n &= \frac{u_n - u_{n-1}}{h_n} - F(h; t_{n-1}, u(t_{n-1})) \\ &= \frac{u_{n-1} + h_n f(t_{n-1}, u_{n-1}) + \frac{1}{2}h_n^2 u''(\zeta) - u_{n-1}}{h_n} - f(t_{n-1}; u_{n-1}) \\ &= \frac{1}{2}h_n u''(\zeta)\end{aligned}$$

Under the assumption that $f \in C^1$ on a compact set around the graph of u, this term is bounded, yielding.

$$|\tau_n| \leq \frac{h_n}{2} \max_{\zeta \in I_n} |u''(\zeta)| = \frac{h_n}{2} \max_{\zeta \in I_n} \left|\partial_x f(\zeta, u(\zeta)) + \partial_u f(\zeta, u(\zeta)) f(\zeta, u(\zeta))\right|$$

Here, we enter the assumption that f is sufficiently smooth to conclude that the Euler method is consistent of order 1.

2.2.4 Lemma (Discrete Grönwall inequality): Let (w_n), (a_n) and (b_n) be non-negative sequences of real numbers. Let b_n be monotonically nondecreasing. Then, if

$$w_0 \leq b_0 \quad \text{and} \quad \forall n \geq 1 : \; w_n \leq \sum_{k=0}^{n-1} a_k w_k + b_n, \tag{2.5}$$

there holds

$$w_n \leq \exp\left(\sum_{k=1}^{n-1} a_k\right) b_n. \tag{2.6}$$

Proof. Define the functions $w(t)$, $a(t)$, and $b(t)$ such that for $k \geq 1$ and $t \in [k-1, k)$ there holds

$$w(t) = w(t_{k-1}), \quad a(t) = b(t_{k-1}), \quad b(t) = b(t_{k-1}).$$

These functions are bounded and piecewise continuous on any finite interval. Thus, they are integrable on $[0, n]$. Therefore, the continuous Grönwall inequality of Lemma 1.3.8 applies and proves the result. □

2.2.5 Theorem (Discrete stability): If $F(t, y)$ is Lipschitz continuous in y for any $t = t_k$, $k < n$, with constant L_h, then the one-step method is **discretely stable**, i. e. for arbitrary sequences (y_n) and (z_n), there holds: if $\eta_k(y)$ and $\eta_k(z)$ are both bounded independent of the sequences (y_n) and (z_n), then

$$|y_n - z_n| \leq e^{L_h(t_n - t_0)} \left(|y_0 - z_0| + \sum_{k=1}^{n} |\eta_k(y) - \eta_k(z)| \right)$$

Proof. Subtracting the equations

$$\begin{aligned} \eta_k(y) &= y_k - y_{k-1} - F_{h_k}(t_{k-1}, y_{k-1}), \\ \eta_k(y) &= z_k - z_{k-1} - F_{h_k}(t_{k-1}, z_{k-1}), \end{aligned}$$

we obtain

$$\begin{aligned} y_k - z_k = y_{k-1} - z_{k-1} + \eta_k(y) - \eta_k(z) \\ + h_k\big(F_{h_k}(t_{k-1}, y_{k-1}) - F_{h_k}(t_{k-1}, z_{k-1})\big). \end{aligned}$$

Recursive application yields

$$|y_n - z_n| \leq |y_0 - z_0| + \sum_{k=1}^{n} |\eta_k(y) - \eta_k(z)| + \sum_{k=1}^{n} L_h h_k |y_k - z_k|.$$

The estimate now follows from the discrete Grönwall inequality in Lemma 2.2.4. □

2.2.6 Corollary (One-step methods with finite precision): Let the one-step method F be run on a computer, yielding a sequence (z_n), such that each time step is executed in finite precision arithmetic. Let (y_n) be the mathematically correct solution of the one-step method. Then, the difference equation (2.1) is fulfilled only up to machine accuracy ε_m:

$$\begin{aligned} y_0 - z_0 &\approx \varepsilon_m \\ |\eta_k(y) - \eta_k(z)| = |\eta_k(z)| &\approx \varepsilon_m |z_k|. \end{aligned}$$

Then, the error between the true solution of the one-step method (y_n) and the computed solution is bounded by

$$|y_n - z_n| \leq e^{L_h(t_n - t_0)} n \varepsilon_m \max_k |z_k|.$$

2.2.7 Theorem (Convergence of one-step methods): Let the one-step method $F(.,.)$ be consistent of order p and discretely stable, that is, $F(.,.)$ is Lipschitz continuous in its second argument. Let $f(t,u) \in C^p$. Furthermore, let be $y_0 = u_0$. Let $h = \max h_n$ and let there be a positive number γ such that $\min h_n = \gamma h$. Then, the method converges with order p and there holds for

$$|u_n - y_n| \leq ce^{L_h(t_n - t_0)} h^p, \tag{2.7}$$

where the constant c is independent of h.

Proof. Again we use the discrete stability theorem: with the definition of the order of the method, we obtain

$$|\eta_k(u) - \eta_k(y)| = |\eta_k(u)| \leq ch^{p+1}, \tag{2.8}$$

where c depends on the derivatives of u (and thus of f), but not on $u_k - y_k$. On the other hand, we have

$$n \leq \frac{t_n - t_0}{\min h_n} \leq \frac{t_n - t_0}{\gamma h}.$$

Thus, we obtain by summing up (2.8) over all n

$$|u_n - y_n| \leq e^{L_h(t_n - t_0)} \sum_{k=1}^{n} h_k^{p+1} \leq ce^{L_h(t_n - t_0)} h^p.$$

□

Corollary 2.2.8. *The Euler method converges of first order.*

2.3 Runge-Kutta methods

2.3.1. We are searching for methods which approximate the solution to an IVP numerically. In fact we are not solving the IVP, but the Volterra integral equation (1.12). Hence we can consider solving differential equations as a quadrature problem; with the difficulty that the function, which we integrate, is not known. This consideration leads to a class of methods for IVP, the Runge-Kutta methods.

2.3.2 Definition: An **explicit Runge-Kutta method (ERK)** is a one-step method with the representation

$$g_i = y_0 + h \sum_{j=1}^{i-1} a_{ij} k_j \qquad i = 1, \ldots, s \tag{2.9a}$$

$$k_i = f\left(hc_i, g_i\right) \qquad i = 1, \ldots, s \tag{2.9b}$$

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i k_i \tag{2.9c}$$

In this method the values hc_i are the quadrature points on the interval $[0, h]$. The values k_i are approximations to function values of the integrand in these points and the values g_i constitute approximations to the solution $u(hc_i)$ in the quadrature points. This method uses s intermediate values and is thus called an s-stage method.

Remark 2.3.3. Pursuant to remark 2.1.5 we present the formula for the calculation of y_1 from y_0 on the interval from $t_0 = 0$ to $t_1 = h$. The formula for a later time step k is obtained by replacing y_0 and $t_0 = 0$ by y_k and t_k, respectively to obtain y_{k+1}.

Remark 2.3.4. The intermediate values g_i will not be saved separately in typical implementations, because it is possible to execute the method with the values k_i alone. Nevertheless, the values g_i are useful for highlighting the structure of the method.

2.3.5 Definition (Butcher tableau): It is customary to write Runge-Kutta methods in the form of a **Butcher tableau**, containing only the coefficients of equation (2.9) in the following matrix form:

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array} \tag{2.10}$$

Remark 2.3.6. The first row of the tableau is to read in such a manner, that $g_1 = y_0$ and k_1 is computed directly by $f(t_0, y_0)$. The coefficients a_{1j} and c_0 do not appear in formulas (2.9a) and (2.9b) (or are considered zero).

The further rows indicate the rules for the computation of the further values k_i in each case according to the formulas (2.9a) and (2.9b). The the method

is explicit since the computation of k_i only involves coefficients with index less than i.

The last row below the line is then the short form of formula (2.9c) and lists quadrature weights.

We see, that the coefficients a_{ij} form the strict lower triangle of a square $s \times s$-matrix A. Therefore, in order to simplify the summation bounds, we implicitly complete this matrix with values $a_{ij} = 0$ for $j \geq i$.This way, the sum in (2.9a) can be taken from 1 to s, independent of i. We will also associate to an s-stage method the vector $b = (b_1, \ldots, b_s)^T$.

Example 2.3.7. The Euler method Euler method has the Butcher tableau:

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

That leads to the already known formula:

$$y_1 = y_0 + hf(t_0, y_0)$$

The values $b_1 = 1$ and $c_1 = 0$ indicate that this is a quadrature rule with a single point at the left end of the interval. Such a rule is exact for constant polynomials and thus of order 1.

2.3.8 Example (Two-stage methods): The **modified Euler method** is a variation of the Euler method of the following form:

$$\begin{aligned} k_1 &= f(t_0, y_0) \\ k_2 &= f(t_0 + \frac{1}{2}h, y_0 + h\frac{1}{2}k_1) \\ y_1 &= y_0 + hk_2 \end{aligned} \qquad \begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array}$$

The so-called **Heun method** of order 2 is characterized through the equation

$$\begin{aligned} k_1 &= f(t_0, y_0) \\ k_2 &= f(t_0 + h, y_0 + hk_1) \\ y_1 &= y_0 + h(\frac{1}{2}k_1 + \frac{1}{2}k_2) \end{aligned} \qquad \begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Remark 2.3.9. The modified Euler method uses an approximation to the value of $f(h/2, u(h/2))$ in its quadrature, corresponding to the midpoint quadrature rule. The Heun method is constructed analogous to the trapezoidal rule. Both quadrature rules are of second order, and so are these one-step methods. Both methods were discussed by Runge in his article of 1895 [Run95].

2.3.10 Lemma: The Heun method and the modified Euler method are consistent of second order[a].

[a]Here and in the following proofs of consistency order, we will always assume that all necessary derivatives of f exist and are bounded. We say "f is sufficiently smooth".

Proof. The proof uses Taylor expansion of the continuous solution u and the discrete solution y around t_0 with respect to h. First, abbreviating $f_t = \partial_t f(t_0, u_0)$ and $f_u = \partial_u f(t_0, u_0)$ and so forth[1] and replacing $u'(t_0) = f(t_0, u_0) = f$:

$$u_1 = u(t_0 + h) = u_0 + hf(t_0, u_0) + \frac{h^2}{2}\big(f_t + f_u f\big) + \frac{h^3}{6}\big(f_{tt} + 2f_{tu}f + f_{uu}f^2 + f_u f_t + f_u^2 f\big) + \dots . \quad (2.11)$$

For the discrete solution of the modified Euler step on the other hand, there holds

$$\begin{aligned} y_1 &= u_0 + hf\left(t_0 + \frac{h}{2}, u_0 + \frac{h}{2}f(t_0, u_0)\right) \\ &= u_0 + hf(t_0, u_0) + \frac{h^2}{2}\big(f_t + f_u f\big) \\ &\quad + \frac{h^3}{8}\big(f_{tt} + 2f_{tu}f + f_{uu}f^2 + f_u f_t + f_u^2 f\big) + \dots . \end{aligned}$$

Thus, $|u_1 - y_1| = \mathcal{O}(h^3)$ and the method is of second order. The proof for the Heun method is left as an exercise. □

2.3.11 Example: The three stage Runge-Kutta method is

$$\begin{aligned} k_1 &= f(t_0, y_0) \\ k_2 &= f(t_0 + \frac{1}{2}h, y_0 + \frac{1}{2}hk_1) \\ k_3 &= f(t_0 + h, y_0 - hk_1 + 2hk_2) \\ y_{n+1} &= y_0 + h(\frac{1}{6}k_1 + \frac{4}{6}k_2 + \frac{1}{6}k_3) \end{aligned}$$

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & -1 & 2 & \\ \hline & \frac{1}{6} & \frac{4}{6} & \frac{1}{6} \end{array}$$

This method is obviously based on the Simpson rule.

Remark 2.3.12. Computations become tedious very fast, in part due to the sum of partial derivatives of $f(t, u)$. This can be simplified by considering

[1]Note that f_u, f_{uu} and so on are tensors of increasing rank.

Runge-Kutta methods for the autonomized ODE (see Definition 1.2.6)

$$\begin{pmatrix} u' \\ t' \end{pmatrix} = \begin{pmatrix} f(t,u) \\ 1 \end{pmatrix}.$$

Then,the Runge-Kutta method (2.9) simplifies to

$$\begin{aligned} g_i &= y_0 + \sum_{j=1}^{i-1} a_{ij} h f(g_j), \quad i = 1, \dots, s \\ y_1 &= y_0 + \sum_{j=1}^{s} b_j h f(g_j). \end{aligned} \tag{2.12}$$

2.3.13 Lemma: An ERK is invariant under autonomization (or in short, autonomizable), if and only if

$$c_i = \sum_{j=1}^{i-1} a_{ij}, \quad i = 1, \dots, s. \tag{2.13}$$

Proof. Observing the last component of the vector u in the previous remark and the method applied to it yields the condition. □

2.3.14 Lemma: An autonomizable ERK with s stages is consistent of third order, if and only if the following conditions are met:

$$b_1 + \cdots + b_s = 1, \tag{2.14a}$$
$$b_1 c_1 + \cdots + b_s c_s = 1/2, \tag{2.14b}$$
$$b_1 c_1^2 + \cdots + b_s c_s^2 = 1/3, \tag{2.14c}$$
$$\sum\nolimits_{i,j} b_i a_{ij} c_j = 1/6. \tag{2.14d}$$

It is consistent of fourth order, if and only if additionally

$$b_1 c_1^3 + \cdots + b_s c_s^3 = 1/4, \tag{2.14e}$$
$$\sum\nolimits_{i,j} b_i a_{ij} c_j^2 = 1/12, \tag{2.14f}$$
$$\sum\nolimits_{i,j,k} b_i a_{ij} a_{jk} c_k = 1/24, \tag{2.14g}$$
$$\sum\nolimits_{i,j} b_i c_i a_{ij} c_j = 1/8. \tag{2.14h}$$

Remark 2.3.15. We can rephrase these conditions, such that an ERK is of order k if the quadrature with support points c_i and corresponding weights b_i is exact for polynomials of degree $k-1$:

$$\sum_{i=1}^{s} b_i p(c_i) = \int_0^1 p(t)\,\mathrm{d}t, \qquad \forall p \in \mathbb{P}_{k-1}.$$

Furthermore, for $k \geq 3$

$$\sum_{ij} b_i a_{ij} p(c_j) = \int_0^1 \int_0^t p(s)\,\mathrm{d}s\,\mathrm{d}t, \qquad \forall p \in \mathbb{P}_{k-2}.$$

Additionally, for $k \geq 4$

$$\sum_{ijk} b_i a_{ij} a_{jk} p(c_k) = \int_0^1 \int_0^t \int_0^s p(r)\,\mathrm{d}r\,\mathrm{d}s\,\mathrm{d}t, \quad \forall p \in \mathbb{P}_{k-3},$$

$$\sum_{ij} b_i p(c_i) a_{ij} q(c_j) = \int_0^1 p(t) \int_0^t q(s)\,\mathrm{d}s\,\mathrm{d}t \qquad \forall p \in \mathbb{P}_{k_1}, q \in \mathbb{P}_{k_2}, k_1 + k_2 = k-2.$$

2.3.16 Lemma: The Taylor expansion of a single component of $u_1 = u(h)$ with respect to h is

$$\begin{aligned}
(u_1)_n &= (u_0)_n \\
&\quad + h f_n \\
&\quad + \frac{h^2}{2} \sum_{\lambda} \partial_\lambda f_n f_\lambda \\
&\quad + \frac{h^3}{6} \sum_{\lambda,\mu} \left[\partial_{\lambda\mu} f_n f_\lambda f_\mu + \partial_\lambda f_n \partial_\mu f_\lambda f_\mu\right] \\
&\quad + \frac{h^4}{24} \sum_{\lambda,\mu,\nu} \big[\partial_{\lambda\mu\nu} f_n f_\lambda f_\mu f_\nu + 3\partial_{\lambda\mu} f_n \partial_\nu f_\lambda f_\mu f_\nu \\
&\qquad\qquad + \partial_\lambda f_n \partial_{\mu\nu} f_\lambda f_\mu f_\nu + \partial_\lambda f_n \partial_\mu f_\lambda \partial_\nu f_\mu f_\nu\big] \\
&\quad + \dots
\end{aligned} \tag{2.15}$$

where we have omitted the arguments $f = f(u(t_0))$ and all sums are taken from 1 to d.

Proof. Taking derivatives of u and replacing every occurrence of u' by $f(u)$. For

scalar valued functions, we clarify this at the example

$$\begin{aligned}
u'(t) &= f(u(t)) \\
u''(t) &= \big(u'(t)\big)' = f(u(t))' = f'(u(t))u'(t) \\
&= f'(u(t))f(u(t)) \\
u^{(3)} &= \big(u''(t)\big)' = \big(f'(u(t))f(u(t))\big)' \\
&= f''(u(t))u'(t)f(u(t)) + f'(u(t))f'(u(t))u'(t) \\
&= f''(u(t))f(u(t))^2 + f'(u(t))^2 f(u(t)).
\end{aligned}$$

After the concept is clear, we have to keep track of the vector indices and compute with brute force. It may be worth noting, that in the 4th order term, we used the fact that we can swap summation indices and get

$$\sum_{\lambda,\mu,\nu} \partial_{\lambda\mu} f_n \partial_\nu f_\lambda f_\mu f_\nu = \sum_{\lambda,\mu,\nu} \partial_{\lambda\mu} f_n \partial_\nu f_\mu f_\lambda f_\nu = \sum_{\lambda,\mu,\nu} \partial_{\lambda\nu} f_n \partial_\mu f_\lambda f_\mu f_\nu.$$

□

2.3.17 Lemma: The Taylor expansion of y_1 with respect to h is

$$\begin{aligned}
&(y_1)_n = (u_0)_n \\
&+ h \sum_{j=1}^{s} b_i f_n \\
&+ \frac{h^2}{2} \sum_{\lambda} \left[2 \sum_i b_i c_i \partial_\lambda f_n f_\lambda \right] \\
&+ \frac{h^3}{6} \sum_{\lambda,\mu} \left[3 \sum_i b_i c_i^2 \partial_{\lambda\mu} f_n f_\lambda f_\mu + 6 \sum_{i,j} b_i a_{ij} c_j \partial_\lambda f_n \partial_\mu f_\lambda f_\mu \right] \\
&+ \frac{h^4}{24} \sum_{\lambda,\mu,\nu} \left[6 \sum_i b_i c_i^3 \partial_{\lambda\mu\nu} f_n f_\lambda f_\mu f_\nu + 3 \sum_{i,j} b_i c_i a_{ij} c_j \partial_{\lambda\mu} f_n \partial_m f_\lambda f_\mu f_\nu \right. \\
&\qquad \left. + 2 \sum_{i,j} b_i a_{ij} c_j^2 \partial_\lambda f_n \partial_{\mu\nu} f_\lambda f_\mu f_\nu + \sum_{i,j,k} b_i a_{ij} a_{jk} c_k \partial_\lambda f_n \partial_\mu f_\lambda \partial_\nu f_\mu f_\nu \right].
\end{aligned} \tag{2.16}$$

Proof. We begin with the observation that for an arbitrary function φ holds

$$\frac{d^q}{dh^q}\big(h\varphi(h)\big)\bigg|_{h=0} = \left[h \frac{d^q}{dh^q}\varphi(h) + q h' \frac{d^{q-1}}{dh^{q-1}}\varphi + \binom{q}{2} h'' \dots \right]_{h=0} = q \frac{d^{q-1}}{dh^{q-1}}\varphi.$$

Next, we use (2.9c) to obtain

$$\begin{aligned} y(h) &= u_0, \\ y^{(q)}(h)\big|_{h=0} &= q \sum_{j=1}^{s} b_j \frac{d^{q-1}}{dh^{q-1}} f(g_j)\bigg|_{h=0}. \end{aligned}$$

We observe $g_i(0) = u_0$. Further, from (2.9a), we obtain

$$\begin{aligned} g_i(h)\big|_{h=0} &= u_0, \\ g_{i;n}^{(q)}(h)\big|_{h=0} &= q \sum_{j=1}^{i-1} a_{ij} \frac{d^{q-1}}{dh^{q-1}} f_n(g_j)\bigg|_{h=0}. \end{aligned}$$

Here, $g_{i;n}$ refers to the component k of vector g_i. Finally, we need

$$\begin{aligned} \frac{d}{dh} f_n(g_i(h))\big|_{h=0} &= \sum_{\lambda} \partial_\lambda f_n g'_{i;\lambda} \\ \frac{d^2}{dh^2} f_n(g_i(h))\big|_{h=0} &= \sum_{\lambda,\mu} \partial_{\lambda\mu} f_n g'_{i;\lambda} g'_{i;\mu} + \sum_{k} \partial_\lambda f_n g''_{i;\lambda}. \end{aligned}$$

Summarizing, we obtain

$$\begin{aligned} y'_n &= \sum_{j=1}^{s} b_j f_n(g_j) \\ y''_n &= 2 \sum_{j=1}^{s} b_j \frac{d}{dh} f_n(g_j) = \sum_{j=1}^{s} \sum_{k=1}^{j-1} b_j a_{jk} \sum_{\lambda} \partial_\lambda f_n f_\lambda \\ y'''_n &= 3 \sum_{j=1}^{s} b_j \frac{d^2}{dh^2} f_n(g_j) \\ &= 3 \sum_{j=1}^{s} b_j \left[\sum_{\lambda,\mu} \partial_{\lambda\mu} f_n g'_{j;\lambda} g'_{j;\mu} + \sum_{\lambda} \partial_\lambda f_n g''_{j;\lambda} \right] \\ &= 3 \sum_{j=1}^{s} b_j \left[\sum_{\lambda,\mu} \partial_{\lambda\mu} f_n \sum_{k=1}^{j-1} a_{jk} f_\lambda \sum_{k=1}^{j-1} a_{jk} f_\mu + \sum_{\lambda} \partial_\lambda f_n 2 \sum_{k=1}^{j-1} a_{jk} \sum_{l=1}^{k-1} a_{kl} \sum_{\mu} \partial_\mu f_\lambda f_\mu \right] \end{aligned}$$

□

Proof of Lemma 2.3.14. The proof utilizes Taylor expansion of u_1 and y_1 provided in Lemmas 2.3.16 and 2.3.17, respectively. Once we have computed these expansions, we compare coefficients in front of equal derivatives in order to get the result. □

Remark 2.3.18. Butcher introduced a graph theoretical method for order conditions based on trees. While this simplifies the process of deriving these conditions for higher order methods considerably, it is beyond the scope of this course.

2.3.19 Example (The classical Runge-Kutta method of 4th order):

$$
\begin{aligned}
k_1 &= f(t_n, y_n) \\
k_2 &= f(t_n + \frac{1}{2}h_n, y_n + \frac{1}{2}h_n k_1) \\
k_3 &= f(t_n + \frac{1}{2}h_n, y_n + \frac{1}{2}h_n k_2) \\
k_4 &= f(t_n + h_n, y_n + h_n k_3) \\
y_{n+1} &= y_n + h_n(\frac{1}{6}k_1 + \frac{2}{6}k_2 + \frac{2}{6}k_3 + \frac{1}{6}k_4)
\end{aligned}
$$

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

This formula is based on the Simpson rule as well, but it uses two approximations for the value in the center point. It is of fourth order.

Remark 2.3.20 (Order conditions and quadrature)**.** The order conditions derived by excessive Taylor expansion have a very natural interpretation through the analysis of quadrature formulas for the Volterra integral equation, where (hc_i) are the quadrature points and the other values are quadrature weights. First, we observe that

$$\sum_i b_i f(g_i) \quad \text{approximates} \quad \frac{1}{h}\int_0^1 f(u(hs)\,\mathrm{d}s.$$

In this view, conditions (2.14a)–(2.14c) and (2.14e) state that the formula $\sum_i b_i p(c_i)$ is an exact integral for polynomials of degree up to 3. In a previous semester, we have made use of this property to prove that the formula is of 4th order.

Equally we deduce from formula (2.9a) for g_i that

$$\sum_j a_{ij} f(g_j) \quad \text{approximates} \quad \frac{1}{h}\int_0^{c_i} f(u(hs)\,\mathrm{d}s.$$

The condition (2.13) that the method be autonomizable states nothing but that this be exact for constant functions. For higher order, the accuracy of the value of g_i only implicitly enters the accuracy of the Runge-Kutta method by integrating this value again. Thus, we actually look at approximations of integrals of the form

$$\int_0^1 \varphi(s) \int_0^s \psi(r)\,\mathrm{d}r\,\mathrm{d}s.$$

Condition (2.14d) for 3rd order states, that this condition must be true for linear polynomials $\psi(r)$ and constant $\varphi(s)$, thus, after the interior integration again a polynomial of second order. Equally, conditions (2.14h) and (2.14f) state this for linear polynomials $\psi(r)$ with linear $\varphi(s)$ and for quadratic polynomials $\psi(r)$ with constant $\varphi(s)$, respectively. Finally, condition (2.14g) states that the quadrature has to be exact for any linear polynomial $\varphi(\tau)$ in

$$\int_0^1 \int_0^s \int_0^r \varphi(\tau)\,\mathrm{d}\tau\,\mathrm{d}r\,\mathrm{d}s.$$

Remark 2.3.21 (Butcher barriers)**.** The maximal order of an explicit Runge-Kutta method is limited through the number of stages, or vice versa, a minimum number of stages is required for a certain order. The **Butcher barriers** state that in order to achieve order p one requires s stages, where p and s relate as follows:

p	1	2	3	4	5	6	7	8	9	10
# cond.	1	2	4	8	17	37	85	200	486	1205
s	p	p	p	p	p+1	p+1	p+2	p+3	?	17?

These order bounds refer to systems of differential equations. For a simple equation they may be better. For instance, there exists a five-stage method which solves the one dimensional IVP with order 5.

For $p = 10$ there is only known a method with $r = 17$ until now. It is possible that there exists a method that needs less stages, because currently no proof for a minimal number of stages is available.

2.3.22 Lemma: Let $f(t, u)$ admit the uniform Lipschitz condition. Then, every autonomizable ERK which is consistent of order one admits a uniform Lipschitz condition.

Proof. We observe that the increment function is

$$F(0, y) = \sum_{j=1}^{s} b_j f(hc_i, g_i(y)), \tag{2.17}$$

with $g_i(y)$ defined recursively by

$$g_i(y) = y + h \sum_{j=1}^{i-1} a_{ij} f(hc_j, g_j(y)).$$

Let L be the Lipschitz constant of f. Let $d_i = |g_i(x) - g_i(y)|/|x-y|$. We have

$$\begin{aligned}
d_1 &= 1 \\
d_2 &= |x - y + ha_{21}\Big(f\big(hc_1, g_1(x)\big) - f\big(c_1, g_1(x)\big)\Big)|/|x-y| \\
&\le (1 + ha_{21}L) = (1 + hc_1L) \\
d_3 &\le \Big(1 + hL\big(a_{31} + a_{32}(1 + ha_{21}L)\big)\Big) \\
&\le \big(1 + hLc_2(1 + hc_1L)\big) \\
d_4 &\le \Big(1 + hLc_3\big(1 + hLc_2(1 + hLc_1)\big)\Big) \\
d_s &\le \Big(1 + hLc_s\big(1 + \ldots(1 + hLc_1)\ldots\big)\Big).
\end{aligned}$$

Since $c_i \le 1$, the factor is bounded by $d_s \le (1 + hL)^{s-1}$. Moreover, if $hL \le 1$, we realize that

$$d_s = \Big(1 + hL\big(1 + \ldots(1 + hL)\ldots\big)\Big) \le s.$$

Finally, we enter this result into (2.17) to obtain

$$\begin{aligned}
|F(0,x) - F(0,y)| &\le \sum_{j=1}^{s} b_j L d_j |x-y| \\
&\le d_s L |x-y|.
\end{aligned}$$

Thus, the increment function F admits a Lipschitz condition with constant $L_h = L(1 + hL)^{s-1}$ for general step size h and $L_h = sL$ for $h \le 1/L$. □

2.4 Estimates of the local error and time step control

2.4.1. In the preceding paragraphs, we have used a crude a priori estimate of the local error based on high order derivatives of the right hand side $f(t,u)$. In the case of a complex nonlinear system, such an estimate is bound to be inefficient, since it involves global bounds on the derivatives. Obviously, the local error cannot be computed exactly either, because that would require or imply the knowledge of the exact solution.

In this section, we discuss two methods which allow an estimate of the truncation error from computed solutions. These estimates are local in nature and therefore usually much sharper. Thus, they can be used to control the step size, which in turn gives good control over the balance of accuracy and effort. Nevertheless, it should be pointed out that in these estimates there is an implicit assumption

that the true solution u is sufficiently regular and the step size is sufficiently small, such that the local error already follows the theoretically predicted order.

Given an estimate for the local error, we can devise an algorithm step size control, which controls the local error and thus in a certain way the global error.

Algorithm 2.4.2 (Adaptive step size control)**.** Let there be an estimate for the local error based on $|y_1 - \hat{y}_1|$. Then, the following algorithm can be used to guarantee that the local error of a one-step method remains below a threshold ε in every time step:

1. Given y_{k-1}, compute y_k and $\hat{y}_k$ with time step h_k.
2. Compute
$$h_{\text{opt}} = h \left(\frac{\varepsilon}{y_k - \hat{y}_k} \right)^{\frac{1}{p+1}} . \tag{2.18}$$
3. If $h_{\text{opt}} < h_k$ the time step is rejected: let $h_k = h_{\text{opt}}$ and recompute y_k and $\hat{y}_k$.
4. If the time step was accepted, let $h_{k+1} = h_{\text{opt}}$.
   (a) If $t_k + h_{k+1} > t_n$, let $h_{k+1} = t_n - t_k$.

   Increase k by one and proceed with the first step.

Remark 2.4.3. It might happen, that the value t_k is just below t_n with a difference close to machine accuracy. As a result, the next time step with $h_{k+1} \approx \varepsilon_m$ would suffer from round-off errors. Therefore, it is advisable to avoid this situation by expanding the last time step, if $t_n - t_k \leq c h_{k+1}$ where c is a moderate constant of size around 1.1.

Remark 2.4.4. This algorithm controls and equilibrates the local error. Nevertheless, the global estimate still retains the exponential term. The error estimation techniques in this section are thus not optimal controlling the global error, which involves considerably more effort and will be discussed in a later course.

2.4.1 Extrapolation methods

2.4.5. Here, we estimate the local error by a method called Richardson extrapolation. It is based on computing two approximations with the same method, but different step size, say an approximation y_2 with two steps of size h and an approximation $\hat{y}_2$ with one step of size $2h$.

2.4.6 Theorem: Let $y_2 = y(t_2)$ be the result of a Runge-Kutta method of order p after 2 steps with step size h and let $\hat{y}_2 = \hat{y}(t_2)$ be the result after one step of step size $2h$. Then, the error admits the estimate

$$u_2 - y_2 = \frac{y_2 - \hat{y}_2}{2^p - 1} + O(h^{p+2}) \tag{2.19}$$

Moreover, we obtain an approximation

$$\tilde{y_2} = y_2 + \frac{y_2 - \hat{y}_2}{2^p - 1}, \tag{2.20}$$

such that

$$u(t_2) - \tilde{y_2} = O(h^{p+2}). \tag{2.21}$$

Proof. For the proof we need a refined version of the local error estimates as well as the global error estimate in Theorem (2.2.7) which can be obtained by adding one more step of Taylor expansion. Then, we get for the local error of a method of order p estimates of the form

$$e_1 = u_1 - y_1 = Ch^{p+1} + \mathcal{O}(h^{p+2}), \tag{2.22}$$

with a constant (vector) C with not necessarily positive entries. In the same way, we refine the estimate for error propagation from basic Lipschitz continuity to

$$e_{2;\text{acc}} = \left(\mathbb{I} + h\frac{\partial f}{\partial y} + \mathcal{O}(h^2)\right)(u_1 - y_1). \tag{2.23}$$

The local error on the second interval is of the same structure as (2.22), but on the interval starting at t_1 with initial value $y_1 = y_0 + \mathcal{O}(h)$.

Thus, we obtain for the error after two steps of size h:

$$\begin{aligned} u_2 - y_2 = \underbrace{\big(\mathbb{I} + \mathcal{O}(h)\big)Ch^{p+1}}_{\text{local 2}} + \big(C + \mathcal{O}(h)\big)h^{p+1} + \mathcal{O}(h^{p+2}) \\ = 2Ch^{p+2} + \mathcal{O}(h^{p+2}). \end{aligned} \tag{2.24}$$

We compare this to a single step for $\hat{y}$ with

$$u_2 - \hat{y}_2 = C(2h)^{p+1} + \mathcal{O}(h^{p+2}). \tag{2.25}$$

Subtracting equations (2.24) and (2.25), we obtain

$$y_2 - \hat{y}_2 = \big(2 - 2^{p+1}\big)Ch^{p+1} + \mathcal{O}(h^{p+2}),$$

such that

$$Ch^{p+1} = \frac{y_2 - \hat{y}_2}{2^{p+1} - 2} + \mathcal{O}(h^{p+2}). \tag{2.26}$$

We enter this result into (2.24) to conclude

$$u_2 - y_2 = \frac{y_2 - \hat{y}_2}{2^p - 1} + \mathcal{O}(h^{p+2}). \tag{2.27}$$

Adding y_2 on both sides, we see that

$$\tilde{y}_2 = y_2 + \frac{y_2 - \hat{y}_2}{2^p - 1} \tag{2.28}$$

approximates u_2 of order $\mathcal{O}(h^{p+2})$ and thus one order better than y_2. □

Remark 2.4.7. Formula (2.19) can be evaluated after computation of y_2 and $\hat{y}_2$ in order to obtain an estimate for the local error of y_2. This estimate can be used to control the step size control according to the algorithm above. We do not have an estimate for the error of $\tilde{y}$. Nevertheless, we expect its values to be more accurate, such that we should use $\tilde{y}$ as approximation and initial value for the next time step.

2.4.2 Embedded Runge-Kutta methods

Instead of estimating the local error by doubling the step size, embedded Runge-Kutta methods use two methods of different order to achieve the same effect. The key to efficiency is here, that the computed stages g_i are the same for both methods, and only the quadrature weights b_i differ.

Definition 2.4.8 (Embedded Runge-Kutta methods)**.** An embedded s-stage Runge-Kutta method with orders of consistence p and $\hat{p}$ computes two solutions y and $\hat{y}$ with the same function evaluations. For this purpose we will first compute contributions g_i and k_i for $i = 1, \ldots, s$ as in the normal Runge-Kutta method of stage s. The function values at the end of the time step result as follows

$$\begin{aligned} y_1 &= y_0 + h \sum b_i k_i \\ \hat{y}_1 &= y_0 + h \sum \hat{b}_i k_i. \end{aligned} \tag{2.29}$$

The methods for y and $\hat{y}$ are consistent of order p and $\hat{p}$, respectively. We let $\hat{p} < p$, for example $\hat{p} = p - 1$.

This can be achieved by e.g. using the same mathod twice and omitting $\hat{b}_j$ for one $j \in \{1, \ldots, s\}$.

2.4.9 Definition: The Butcher tableau for the embedded method has the form:

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \\ \hline & \hat{b}_1 & \hat{b}_2 & \cdots & \hat{b}_{s-1} & \hat{b}_s \end{array}$$

Remark 2.4.10. For higher order methods or functions $f(t, u)$ with complicated evaluation, most of the work lies in computation of the stages. Thus, the additional quadrature for the computation of $\hat{y}$ is almost for free. Nevertheless, due to the different orders of approximation, y is much more accurate and we obtain

$$u_1 - \hat{y}_1 = y_1 - \hat{y}_1 + \mathcal{O}(h^p). \tag{2.30}$$

Thus, $y_1 - \hat{y}_1$ is a good estimate for the local error of $\hat{y}_1$. This is the error which is used in step size control below. Similar to Richardson extrapolation above, we use the more accurate value y_1 for further computation, even if we do not have a computable estimate for its local error.

2.4.11 Definition (Dormand-Prince 45): The embedded Runge-Kutta method of orders 4 for $\hat{y}$ and 5 for y due to Dormand and Prince has the Butcher tableau

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

Remark 2.4.12. The Dormand-Prince method of orders 4 and 5 has become a standard tool for the integration of IVP. It is the backbone of `ode45` in Matlab.

2.5 Continuous Runge-Kutta methods

2.5.1. The Runge-Kutta methods discussed so far compute highly accurate approximations to the solution $u(t)$ in the discrete points $(t_k)_{k=1,\dots,n}$. Such approximations are useful, if only the value at the interval end $t_n = T$ is needed, or if the time steps are sufficiently small in order to generate a plot of the solution history. There are some problems though, where the values in discrete points are not sufficient:

1. The step size control managed to use very large steps from which for instance a plot cannot read easily. Thus, we require the solution in the continuum between two time steps.
2. Accurate approximations inside an interval are required. This may be due to the fact that we want to measure the length of a period of a periodic solution, or that the equation contains switches which change discretely when the solution attains certain values.

In all of these cases, we need an interpolation formula for these intermediate values. Unfortunately, as the example of the classical Runge-Kutta method shows, the values g_i have questionable value in this business. Therefore, in order to be better than linear interpolation between y_{k-1} and y_k, we have to consider formulas, which provide the information for accurate interpolation with low additional cost.

Definition 2.5.2. A continuous Runge-Kutta method is a method of the same type as in definition 2.3.2, for which the coefficients b_i are replaced by continuous functions $b_i(\vartheta)$ on the interval $[0, 1]$. For this reason the equation (2.9c) is augmented by

$$y(t_0 + \vartheta h) = y_0 + \sum_{i=1}^{s^*} b_i(\vartheta) k_i. \tag{2.31}$$

Here the stage number s^* may be higher than s. Then additional intermediate values k_i have to be generated.

Remark 2.5.3. The local error $u(t_0 + \vartheta h) - y(t_0 + \vartheta h)$ is of order p^* if the derivative $\partial_\vartheta^k y$ approximates $\partial_\vartheta^k u$ with an error of order h^{p^*-k+1}. Note that the first derivatives involve the derivatives of g_i with respect to h.

For a later time, we observe that the error of the initial value of an interval is only $\mathcal{O}(h^p)$. Thus, an optimal continuous formula balancing the global error of the original method with the local error of the continuous method should be of order $p^* = p - 1$.

Remark 2.5.4. If $s^* > s$ choose $k_{s+1} = k_1 = f(t_n, y_1)$ of the next time step.

2.5.5 Example: For the classical Runge-Kutta method with 4 stages a continuous interpolation with $s^* = s$ is defined by the coefficients

$$b_1(\vartheta) = \vartheta - \frac{3}{2}\vartheta^2 + \frac{2}{3}\vartheta^3$$
$$b_2(\vartheta) = b_3(\vartheta) = \vartheta^2 - \frac{2}{3}\vartheta^3$$
$$b_4(\vartheta) = -\frac{1}{2}\vartheta^2 + \frac{2}{3}\vartheta^3.$$

The error of this method is $y(\vartheta) - u(\vartheta) = O(h^3)$.

2.5.6 Example: A continuous interpolation for the Dormand-Prince method of orders 4/5 is given by

$$b_1(\vartheta) = \vartheta^2(3-2\vartheta)b_1 + \vartheta(\vartheta-1)^2 - 5\tfrac{2558722523-31403016\vartheta}{11282082432}\vartheta^2(\vartheta-1)^2$$
$$b_2(\vartheta) = 0$$
$$b_3(\vartheta) = \vartheta^2(3-2\vartheta)b_3 + 100\vartheta^2(\vartheta-1)^2\tfrac{882725551-15701508\vartheta}{327004410799}$$
$$b_4(\vartheta) = \vartheta^2(3-2\vartheta)b_4 + 25\vartheta^2(\vartheta-1)^2\tfrac{443332067-31403016\vartheta}{1880347072}$$
$$b_5(\vartheta) = \vartheta^2(3-2\vartheta)b_5 + 32805\vartheta^2(\vartheta-1)^2\tfrac{23143187-3489224\vartheta}{199316789632}$$
$$b_6(\vartheta) = \vartheta^2(3-2\vartheta)b_6 + 55\vartheta^2(\vartheta-1)^2\tfrac{29972135-7076736\vartheta}{822651844}$$

(no warranty)

Remark 2.5.7. Collocation methods will provide a natural way to obtain continuous method in the next chapter.

Chapter 3

Implicit One-Step Methods and Long-Term Stability

3.0.1. In the first chapter, we studied methods for the solution of IVP and the analysis of their convergence with shrinking step size h. We could gain a priori error estimates from consistency and stability for sufficient small h.

All of these error estimates are based on Grönwall's inequality. Therefore, they contain a term of the form e^{Lt} which increases fast with increasing length of the time interval $[t_0, T]$. Thus, the analysis is unsuitable for the study of long-term integration, since the exponential term will eventually outweigh any term of the form h^p.

On the other hand, for instance our solar system has been moving on stable orbits for several billion years and we do not observe an exponential increase of velocities. Thus, there are in fact applications for which the simulation of long time periods is worthwhile and where exponential growth of the discrete solution would be extremely disturbing.

This chapter first studies conditions on differential equations with bounded long term solutions, and then discusses numerical methods mimicking this behavior.

3.1 Monotonic initial value problem

Example 3.1.1. We consider for $\lambda \in \mathbb{C}$ the linear initial value problem

$$\begin{aligned} u' &= \lambda u \\ u(0) &= 1. \end{aligned} \tag{3.1}$$

Splitting $\lambda = \Re(\lambda) + i\Im(\lambda)$ into its real and imaginary part, the (complex valued) solution to this problem is

$$u(t) = e^{\lambda t} = e^{\Re(\lambda)t}\big(\cos(\Im(\lambda)t) + i\sin(\Im(\lambda)t)\big).$$

The behavior of $u(t)$ for $t \to \infty$ is determined by the real part of λ:

$$\begin{aligned} \Re(\lambda) < 0 : &\qquad u(t) \to 0 \\ \Re(\lambda) = 0 : &\qquad |u(t)| = 1 \\ \Re(\lambda) > 0 : &\qquad u(t) \to \infty \end{aligned} \tag{3.2}$$

Moreover, the solution is bounded for λ with non-positive real part for all points in time t.

Remark 3.1.2. Since we deal in the following again and again with eigenvalues of real-valued matrices, we will always consider complex valued IVP hereafter, due to the well known fact, that these eigenvalues can be complex.

Remark 3.1.3. Due to Grönwall's inequality and the stability theorem 1.4.5, the solution to the IVP above admits the estimate $|u(t)| \leq e^{|\lambda|t}|u(0)|$. This is seen easily by applying the comparison function $v(t) \equiv 0$. As soon as $\lambda \neq 0$ has a non-positive real part, this estimate is still correct but very pessimistic and therefore useless for large t. Since problems with bounded long-term behavior are quite important in applications, we will have to introduce an improved notation of stability.

3.1.4 Definition: The function $f(t, y)$ satisfies on its domain $D \subset \mathbb{R} \times \mathbb{C}^d$ a **one-sided Lipschitz condition** if the inequality

$$\Re\langle f(t, y) - f(t, x), y - x\rangle \leq \nu |y - x|^2 \tag{3.3}$$

holds with a constant ν for all $(t, x), (t, y) \in D$. Moreover such a function is called **monotonic** if $\nu = 0$, thus

$$\Re\langle f(t, y) - f(t, x), y - x\rangle \leq 0. \tag{3.4}$$

An ODE $u' = f(u)$ is called monotonic if its right hand side F is monotonic.

Remark 3.1.5. The term monotonic from the previous definition is consistent with the term *monotonically decreasing*, which we know from real-valued functions. We can see this by observing for $y > x$

$$\big(f(t, y) - f(t, x)\big)(y - x) \leq 0 \quad \Leftrightarrow \quad f(t, y) - f(t, x) < 0.$$

3.1.6 Theorem: Let $u(t)$ and $v(t)$ be two solutions of the equation

$$u' = f(t,u), \qquad v' = f(t,v),$$

with initial values $u(t_0) = u_0$ and $v(t_0) = v_0$, respectively. Let the function f be continuous and let the one-sided Lipschitz condition (3.3) hold. Then we have for $t > t_0$:

$$|v(t) - u(t)| \leq e^{\nu(t-t_0)} |v(t_0) - u(t_0)|. \tag{3.5}$$

Proof. We consider the auxiliary function $m(t) = |v(t)-u(t)|^2$ and its derivative

$$\begin{aligned} m'(t) &= 2\Re\langle v'(t) - u'(t), v(t) - u(t)\rangle \\ &= 2\Re\big\langle f\big(t, v(t)\big) - f\big(t, u(t)\big), v(t) - u(t)\big\rangle \\ &\leq 2\nu |v(t) - u(t)|^2 \\ &= 2\nu m(t). \end{aligned}$$

According to Grönwall's inequality (lemma 1.3.8 on page 12) we obtain for $t > t_0$:

$$m(t) \leq m(t_0) e^{2\nu(t-t_0)}.$$

Taking the square root yields the stability estimate (3.5). □

Remark 3.1.7. Analog to example 3.1.1 on page 45 we obtain from the stability estimate, that for the difference of two solutions $u(t)$ and $v(t)$ of the differential equation $u' = f(t,u)$ we obtain in the limit $t \to \infty$:

$$\begin{aligned} \nu < 0 &: \qquad |v(t) - u(t)| \to 0 \\ \nu = 0 &: \qquad |v(t) - u(t)| \leq |v(t_0) - u(t_0)| \end{aligned} \tag{3.6}$$

3.1.8 Lemma: The linear differential equation $u' = Au$ with $u(t) \in \mathbb{C}^d$ and a diagonalizable matrix function $A(t) \in \mathbb{C}^{d\times d}$ admits the one-sided Lipschitz condition (3.3) with the constant

$$\nu = \max_{\substack{i=1,\ldots,d \\ t\in\mathbb{R}}} \Re(\lambda_i).$$

Accordingly, this ODE is monotonic if and only if for all eigenvalues λ_i of $A(t)$ there holds

$$\Re(\lambda_i) \leq 0. \tag{3.7}$$

This is the vector-valued form of example 3.1.1.

Proof. For the right hand side of the equation we have

$$\Re\langle A(t)y - A(t)x, y - x\rangle \leq \Re\frac{\langle A(t)y - A(t)x, y - x\rangle}{|y - x|}|y - x| \leq \max_{i=1,\ldots,d} \Re(\lambda_i)|y - x|.$$

Hence, we obtain already $\nu \leq \max_{i=1,\ldots,d} \Re(\lambda_i)$. If we now insert for $x - y$ an eigenvector of eigenvalue λ for which the maximum is accepted, then we obtain the equality and therefore $\nu = \max_{i=1,\ldots,d} \Re(\lambda_i)$. □

3.1.1 Stiff initial value problems

Example 3.1.9. We consider the IVP

$$u' = \begin{pmatrix} -1 & 0 \\ 0 & -100 \end{pmatrix} u, \qquad u(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{3.8}$$

This has the solution

$$u(t) = \begin{pmatrix} e^{-t} \\ e^{-100t} \end{pmatrix}.$$

We see, that the solution has a component which decreases slowly in time with e^{-t} and a second one, which decreases fast with e^{-100t}. If we apply the Euler method with step size h to this equation, then we obtain the method step

$$y^{(n)} = y^{(n-1)} + h \begin{pmatrix} -1 & 0 \\ 0 & -100 \end{pmatrix} y^{(n-1)} = \begin{pmatrix} 1 - h & 0 \\ 0 & 1 - 100h \end{pmatrix} y^{(n-1)}$$

with the solution

$$y^{(n)} = \begin{pmatrix} (1 - h)^n \\ (1 - 100h)^n \end{pmatrix}$$

If we are interested in the second solution component, the one which decreases fast, we choose h to be small, say $h < 1/100$. Thus, for $n \to \infty$ we have $y_n \to 0$, slowly in the first component, fast in the second one, just like the solution $u(t)$ of the continuous solution. (Recall that for fixed chosen step size h the limits $t \to \infty$ and $n \to \infty$ are equal.)

If we are just interested in the first, the slow component, at a time where it has changed significantly. then a considerably larger step size is appropriate, say $h = 1/10$. For this step size the first solution component is still converging to zero with $y_1^{(n)} = 0.9^n$. For the second one we have however $|y_2^{(n)}| = |(-9)^n| \to \infty$. Therefore the approximate solution for this step size diverges for $n \to \infty$, very much in contrast to the behavior of the exact solution $u(t) \to 0$ for $t \to \infty$.

Remark 3.1.10. Of course, it would have been possible to ignore the second component in the previous example. But this is not a simple task in general, due to the fact that most solution components are coupled through the equation. In such cases the step size of the Euler method must be chosen to accommodate the "fast components". This can lead to significant computational overhead. Therefore, we define in the following characteristic properties of such problems and develop to that specially adapted solution methods.

3.1.11 Definition: An initial value problem is called **stiff**, if it has the following characteristic properties:

1. The right hand side of the ODE is monotonic, or at least admits a one-sided Lipschitz condition with a small parameter ν.
2. The time scales on which the different solution components are evolving differ a lot, or in mathematical terms, the Lipschitz constant L of the right hand side is greater than ν by orders of magnitude.
3. The time scales which are of interest for the application are much longer than the fastest time scales of the equation. Again in the language of our parameters: there is a constant c of moderate size, such that

$$e^{LT} \leq ce^{\nu T}. \tag{3.9}$$

Remark 3.1.12. Even though we used the term definition, the notion of "stiffness of an IVP" has something vague or even inaccurate about it. In fact that is due to the very nature of the problems and cannot be fixed. Instead we are forced to sharpen our understanding by means of a few examples.

Remark 3.1.13. The third condition in the definition of stiffness is rather rare to find in the literature, but it is in general implicitly assumed by the discussion for time step methods for stiff IVP. It is important though to realize, that the methods of the previous chapter do not cause problems computing a good resolution for the fastest time scales. In this case, e^{Lt} will be not too much greater than $e^{\nu t}$.

Example 3.1.14. First of all we will have a look at equation (3.8) in example 3.1.9. The first condition of the stiffness definition is fulfilled. The decrease to $1/e$ happens at $t = 1$ and at $t = 1/100$ for the first and second component, respectively. Thus, the second condition holds as well.

According to the discussion of example 3.1.9, the third condition depends on the purpose of the computation. If we want to compute the solution at time $t = 1/100$, we would not denote the problem as stiff. As one is interested on the

solution at time $t = 1$, on which the second component with e^{-100} is already below typical machine accuracy, the problem is stiff indeed. Here we have seen that Euler's method requires disproportionately small time steps.

Remark 3.1.15. The definition of stiffness and the discussion of the examples reveal that numerical methods are needed, which are not just convergent for time steps $h \to 0$ but also for fixed step size h, even in the presence of time scales clearly below h. In this case, methods still have to produce solutions with correct limit behavior for $t \to \infty$.

Example 3.1.16. The **implicit Euler method** is defined by the one-step formula

$$y_1 = y_0 + hf(t_1, y_1) \quad \Leftrightarrow \quad y_1 - hf(t_1, y_1) = y_0. \tag{3.10}$$

Applied to our example (3.8), we observe

$$y^{(n)} = \begin{pmatrix} 1+h & 0 \\ 0 & 1+100h \end{pmatrix}^{-1} y^{(n-1)}.$$

This yields the solution

$$y^{(n)} = \begin{pmatrix} \frac{1}{(1+h)^n} \\ \frac{1}{(1+100h)^n}, \end{pmatrix}$$

which converges to zero for $n \to \infty$, independent of h. Thus, although the implicit Euler method requires in general the solution of a nonlinear system in each step, it allows for much larger time steps than the explicit Euler method,when applied to a stiff problem.

3.2 A- and B-stability

3.2.1. In this section, we will investigate desirable properties of one-step methods for stiff IVP (3.11). We will first study linear problems of the form

$$u' = Au \qquad u(t_0) = u_0. \tag{3.11}$$

and the related notion of A-stability in detail From the conditions for stiffness we derive the following problem characteristics:

1. All eigenvalues of the matrix A lie in the left half-plane of the complex plane. With (3.2) all solutions are bounded for $t \to \infty$.
2. There are eigenvalues close to zero and eigenvalues with a large negative real part.

3. We are interested in time spans which make it necessary, that the product $h\lambda$ of a time step and an arbitrary eigenvalue, is allowed to be large.

For this case we now want to derive criteria for the boundedness of the discrete solution for $t \to \infty$. The important part is not to derive an estimate holding for $h \to 0$, but one that holds for any value of $h\lambda$ in the left half-plane of the complex numbers.

3.2.2 Definition: The **stability function** $R(z) = R(h\lambda)$ is the function generated by applying the one-step method

$$y_1 = y_0 + hF_h(t_0, y_0)$$

to the linear test problem $u'(t) = \lambda u(t)$. Therefore,

$$y_1 = R(h\lambda)u_0, \tag{3.12}$$

and

$$y^{(n)} = R(h\lambda)^n u_0. \tag{3.13}$$

The **stability region** of a one-step method is the set

$$S = \big\{z \in \mathbb{C} \big| |R(z)| \leq 1\big\}. \tag{3.14}$$

Example 3.2.3. The stability function of the explicit Euler method is derived as follows:

$$\begin{aligned} y_1 &= y_0 + h\lambda y_0 = (1 + h\lambda)y_0 \\ \Rightarrow R(h\lambda) &= 1 + h\lambda \\ R(z) &= 1 + z \end{aligned} \tag{3.15}$$

The stability region for the explicit Euler is a circle with radius 1 around the point (-1,0) in the complex plane (see Figure 3.1)

Example 3.2.4. The stability function of the implicit Euler method is derived as follows:

$$\begin{aligned} y_1 &= y_0 + hf(t_1, y_1) \\ y_1 &= y_0 + h\lambda y_1 \\ (1 - h\lambda)y_1 &= y_0 \\ \Rightarrow R(h\lambda) &= \frac{1}{1 - h\lambda} \\ R(z) &= \frac{1}{1 - z} \end{aligned} \tag{3.16}$$

The stability region for the implicit Euler is the complement of a circle with radius 1 around the point (1,0) in the complex plane (see Figure 3.1).

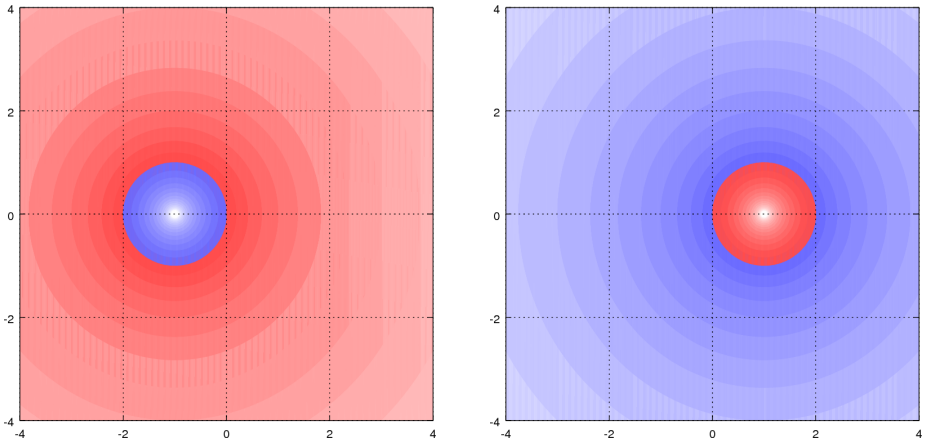


Figure 3.1: Stability regions of the explicit and implicit Euler methods (blue stable, red unstable)

3.2.5 Definition (A-stability): A method is called **A-stable**, if its stability region contains the left half-plane of $\mathbb{C}$, hence

$$\{z \in \mathbb{C} | \Re(z) \leq 0\} \subset S \tag{3.17}$$

3.2.6 Theorem: Let $\left\{y^{(k)}\right\}$ be the sequence generated by an A-stable one-step method of step size h for the linear, autonomous IVP

$$u' = Au, \qquad u(t_0) = u_0$$

with a diagonalizable matrix A an initial value $y^{(0)} = u_0$. If additionally all eigenvalues of A have a non-positive real part, then the sequence members $y^{(k)}$ are uniformly bounded for all h.

Proof. Let $\{v_\ell\}_{\ell=1,\dots,d}$ be a basis of $\mathbb{C}^d$ consisting of eigenvectors of A. Such a basis exists since A is diagonalizable. Let $y_0 = \sum_{\ell=1}^{d} \alpha^\ell v_\ell$ be the representation of y_0 in this basis. Furthermore, we introduce the representations $g_i = \sum_{\ell=1}^{d} \gamma_i^\ell v_\ell$. Then, we see that equations of the form

$$g_i = y_0 + h \sum_{j=1}^{s} a_{ij} A g_i$$

decouple into

$$\gamma_i^\ell = \alpha^\ell + h \sum_{j=1}^{s} a_{ij} \lambda_\ell \gamma_j^\ell.$$

Similarly, if $y_1 = \sum_{\ell=1}^{d} \eta^\ell v_\ell$ we have the separation

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i g_i \quad \longrightarrow \quad \eta^\ell = \alpha^\ell + h \sum_{i=1}^{s} b_i \gamma_i^\ell.$$

Thus, instead of a vector valued problem, the method solves d decoupled scalar problems, inside and across time steps. But for each of the scalar problems, the definition of A-stability implies boundedness of the solution, if $\Re(\lambda_\ell) \leq 0$. □

3.2.7 Theorem: No explicit Runge-Kutta method is A-stable.

Proof. We show that for such methods $R(z)$ is a polynomial. Then, the theorem follows immediately, it is known for polynomials, that the absolute value of its values goes to infinity, if the absolute value of the argument goes to infinity.

From the equation (2.9b) follows $k_i = \lambda g_i$. If we insert that into the equation (2.9a), we obtain

$$g_i = y_0 + h \sum_{j=1}^{i-1} a_{ij} k_j = y_0 + h\lambda \sum_{j=1}^{i-1} a_{ij} g_j.$$

With $g_1 = y_0$ and $z = h\lambda$ one has

$$g_2 = y_0 + a_{21} z y_0 = (1 + a_{21} z) y_0$$
$$g_3 = y_0 + a_{32} z g_1 = y_0 + a_{32} z (1 + a_{21} z) y_0 = (1 + a_{32} z (1 + a_{21} z)) y_0.$$

Therefore one shows easily per induction that k_j results as multiplication of a polynomial of order $j - 1$ with y_0. With formula (2.9c) we have that $R(z)$ is a polynomial of order $s - 1$. □

Remark 3.2.8. The notion of A-stability is only applicable to linear problems with diagonalizable matrices. Now we are considering its extension to nonlinear problems with monotonic right hand sides.

3.2.9 Definition: A one-step method is called **B-stable**, if for monotonic initial value problems $u' = f(u)$ with arbitrary initial value y_0 and z_0 there holds:

$$|y_1 - z_1| \leq |y_0 - z_0| \tag{3.18}$$

independent of the time step size h.

3.2.10 Theorem: Let be $\left\{y^{(k)}\right\}$ the sequence generated by a B-stable method for the IVP

$$u' = f(u), \qquad u(t_0) = u_0$$

with initial values $y^{(0)} = u_0$. If the right hand side f is monotonic, then the values $y^{(k)}$ of the sequence are uniformly bounded for for $k \to \infty$ independent of the time step size h.

Proof. The theorem follows immediately by iterating over the definition of B-stability. □

3.2.11 Corollary: A B-stable method is A-stable.

Proof. Apply the method to the linear differential model equation, which is monotonic for $\Re(\lambda) \leq 0$. Now, the definition of B-stability implies $|R(z)| \leq 1$, and thus, the method is A-stable. □

3.2.1 L-stability

An undesirable feature of complex differentiable functions in the context of stability of Runge-Kutta methods is the fact, that the limit $\lim_{z\to\infty} R(z)$ is well-defined on the Riemann sphere, independent of the path chosen to approach this limit in the complex plane. Thus, for any real number x, we have

$$\lim_{x\to\infty} R(x) = \lim_{x\to\infty} R(ix). \tag{3.19}$$

Thus, a method, which has exactly the left half-plane of $\mathbb{C}$ as its stability domain, seemingly a desirable property, has the undesirable property that components in eigenspaces corresponding to very large negative eigenvalues, and thus decaying very fast in the continuous problem, are decaying very slowly if such a method is applied.

This gave raise to the following notion of L-stability. We nevertheless point out, that the L-stable methods are not always to be considered better than A-stable ones, like it is not always necessary to require A-stability. Judgment must be applied according to the problem being solved.

3.2.12 Definition: An A-stable one-step method is called **L-stable**, if for its stability function there holds

$$\lim_{z\to\infty} |R(z)| = 0. \tag{3.20}$$

Some authors refer to L-stable methods as **strongly A-stable**.

3.3 General Runge-Kutta methods

3.3.1. According to theorem 3.2.7, ERK cannot be A- or B-stable. Thus, they are not suitable for long term integration of stiff IVP. The goal of this chapter is the study of methods not suffering from this limitation. The cure will be implicit methods, where stages may not only depend on known values from the past, but also on the value to be computed.

We point out at the beginning of this chapter, that the main drawback of these methods is the fact that they require the solution of a typically nonlinear system of equations and thus involve much higher computational effort. Therefore, careful judgment should always be applied to determine whether a problem is really stiff or an implicit method is needed for other reasons.

3.3.2 Definition: A **Runge-Kutta method** is a one-step method of the form

$$g_i = y_0 + h\sum_{j=1}^{s} a_{ij}k_j \qquad i = 1,\dots,s \tag{3.21a}$$

$$k_i = f(t_0 + hc_i, g_i) \qquad i = 1,\dots,s \tag{3.21b}$$

$$y_1 = y_0 + h\sum_{i=1}^{s} b_i k_i \tag{3.21c}$$

The method is called

ERK if $j \geq i \Rightarrow a_{ij} = 0$ ("explicit")

DIRK if $j > i \Rightarrow a_{ij} = 0$ ("diagonal implicit")

SDIRK if DIRK and $\forall i, j : a_{ii} = a_{jj}$ ("singly diagonal implicit")

IRK "implicit" in all other cases.

Example 3.3.3 (Two-stage SDIRK)**.** Both SDIRK methods in table 3.1 are of order three

$$\begin{array}{c|cc} \frac{1}{2}-\frac{\sqrt{3}}{6} & \frac{1}{2}-\frac{\sqrt{3}}{6} & 0 \\ \frac{1}{2}+\frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{3} & \frac{1}{2}-\frac{\sqrt{3}}{6} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \begin{array}{c|cc} \frac{1}{2}+\frac{\sqrt{3}}{6} & \frac{1}{2}+\frac{\sqrt{3}}{6} & 0 \\ \frac{1}{2}-\frac{\sqrt{3}}{6} & -\frac{\sqrt{3}}{3} & \frac{1}{2}+\frac{\sqrt{3}}{6} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \tag{3.22}$$

Table 3.1: Two-stage SDIRK method of order 3

3.3.4 Lemma: The stability function of an s-stage Runge-Kutta method with coefficients

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1s} \\ \vdots & & \vdots \\ a_{s1} & \cdots & a_{ss} \end{pmatrix}, b = \begin{pmatrix} b_1 \\ \vdots \\ b_s \end{pmatrix},$$

is given by the two expressions

$$R(z) = 1 + zb^T\big(I - zA\big)^{-1}\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \frac{\det\left(I - zA + z\begin{pmatrix} b_1 & \cdots & b_s \\ \vdots & & \vdots \\ b_1 & \cdots & b_s \end{pmatrix}\right)}{\det\big(I - zA\big)} \tag{3.23}$$

Proof. Entering $f(u) = \lambda u$ into the definition of the stages g_i, we obtain the linear system

$$g_i = y_0 + h\sum_{j=1}^{s} a_{ij}\lambda g_j, \quad i = 1, \ldots, s.$$

In matrix notation with $z = h\lambda$, we obtain $(I - zA)g = (y_0, \ldots, y_0)^T$, where g is the vector $(g_1, \ldots, g_s)^T$. Equally, we obtain

$$\begin{aligned} R(z)y_0 = y_1 = y_0 + h\sum_{i=1}^{s} b_i\lambda g_i &= y_0 + zb^T g \\ &= y_0 + zb^T(I - zA)^{-1}\begin{pmatrix} y_0 \\ \vdots \\ y_0 \end{pmatrix} \\ &= \left(1 + zb^T(I - zA)^{-1}\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}\right) y_0. \end{aligned}$$

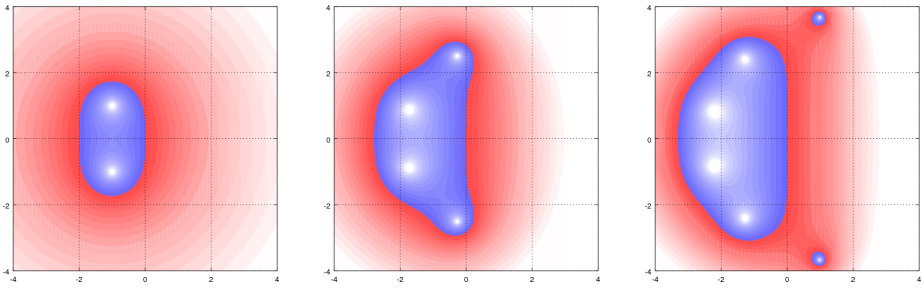


Figure 3.2: Stability regions of the modified Euler method, the classical Runge-Kutta method of order 4 and the Dormand/Prince method of order 5 (blue stable, red unstable)

In order to prove the second representation, we write the whole Runge-Kutta method as a single system of equations of dimension $s + 1$:

$$\begin{pmatrix} I - zA & 0 \\ -zb^T & 1 \end{pmatrix} \begin{pmatrix} g \\ y_1 \end{pmatrix} = y_0 \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}.$$

Applying Cramer's rule yields the result. □

3.3.5 Example: Stability functions of the modified Euler method, of the classical Runge-Kutta method of order 4 and of the Dormand-Prince method of order 5 are

$$\begin{aligned} R_2(z) &= 1 + z + \tfrac{z^2}{2} \\ R_4(z) &= 1 + z + \tfrac{z^2}{2} + \tfrac{z^3}{6} + \tfrac{z^4}{24} \\ R_5(z) &= 1 + z + \tfrac{z^2}{2} + \tfrac{z^3}{6} + \tfrac{z^4}{24} + \tfrac{z^5}{120} + \tfrac{z^6}{600} \end{aligned}$$

respectively. Their stability regions are shown in Figure 3.2.

3.3.6 Definition: The ϑ-scheme is the one-step method, defined for $\vartheta \in [0,1]$ by

$$y_1 = y_0 + h\big((1-\vartheta)f(y_0) + \vartheta f(y_1)\big). \tag{3.24}$$

It is an RKM with the Butcher Tableau

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1-\vartheta & \vartheta \\ \hline & 1-\vartheta & \vartheta \end{array}. \tag{3.25}$$

Three special cases are distinguished:

$\vartheta = 0$	explicit Euler method
$\vartheta = 1$	implicit Euler method
$\vartheta = 1/2$	Crank-Nicolson method

Furthermore, we define the variable ϑ-scheme where ϑ is of the form

$$\vartheta = \frac{1}{2} + \gamma h.$$

3.3.7 Theorem: The ϑ-scheme is A-stable for $\vartheta \geq 1/2$. Furthermore, if there exists a constant c such that $\vartheta - 1/2 \leq ch$, the method is consistent of second order.

Proof. Left as a homework question. Additionally, we show stability regions for different ϑ in Figure □

3.3.1 Existence and uniqueness of discrete solutions

While it was clear that the steps of an explicit Runge-Kutta method can always be executed, implicit methods require the solution of a possibly nonlinear system of equations. The solvability of such a system is not always clear. We will investigate several cases here: First, Lemma 3.3.8 based on a Lipschitz condition on the right hand side. Since this result suffers from a severe step size constraint, we add Lemma 3.3.9 for DIRK methods based on right hand sides with a one-sided Lipschitz condition. Finally, we present Theorem 3.3.10 for general Runge-Kutta methods with one-sided Lipschitz condition.

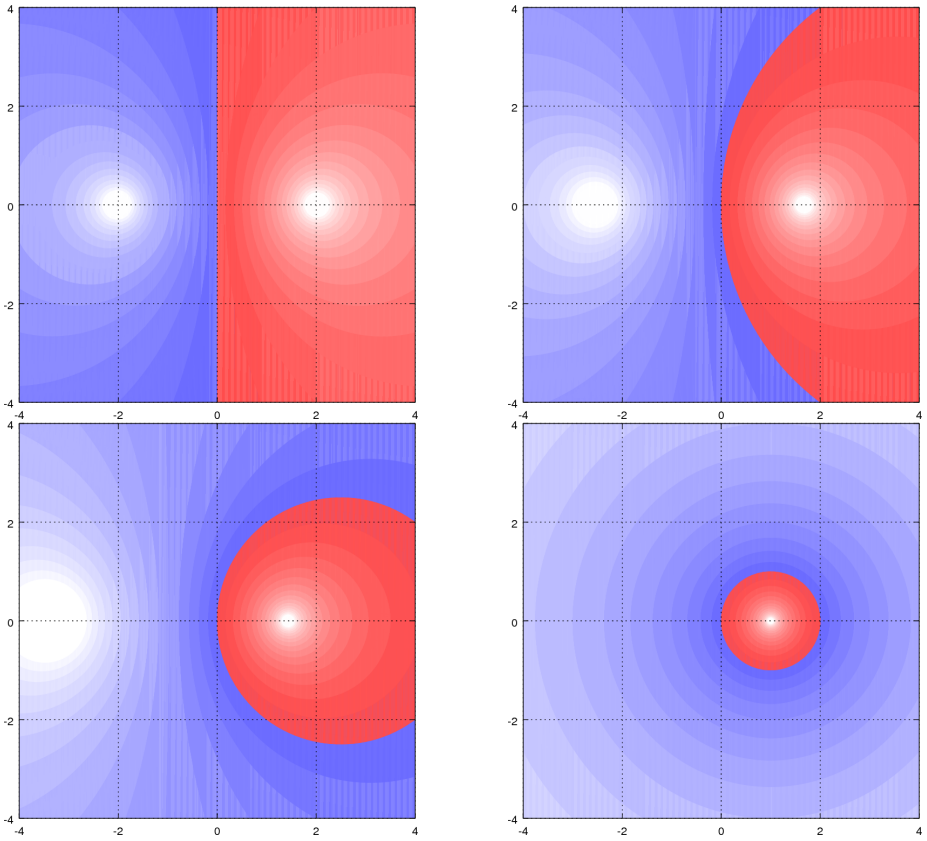


Figure 3.3: Stability regions of the ϑ-scheme with $\vartheta = 0.5$ (Crank-Nicolson), $\vartheta = 0.6$, $\vartheta = 0.7$, and $\vartheta = 1$ (implicit Euler).

3.3.8 Lemma: Let $f : \mathbb{R} \times \mathbb{C}^d \to \mathbb{C}^d$ be continuous and satisfy the Lipschitz condition with constant L. If

$$hL < \frac{1}{\max\limits_{i=1,\ldots,s} \sum\limits_{j=1}^{s} |a_{ij}|}, \tag{3.26}$$

then, for any y_0 the Runge-Kutta method (3.21) has a unique solution y_1.

Proof. We prove existence and uniqueness by a fixed-point argument. To this end, define the vectors $k^{(m)} = (k_1^{(m)}, \ldots, k_s^{(m)})^T \in \mathbb{R}^{sd}$ for $m = 1, \ldots$ and the iteration $k^{(m)} = F(k^{(m-1)}$ by

$$k_i^{(m)} = F_i(k^{(m-1)}) = f\left(t_0 + c_i h, y_0 + h \sum_{j=1}^{s} a_{ij} k_j^{(m-1)}\right), \quad i = 1, \ldots, s.$$

Clearly the vector $k = (k_1, \ldots, k_s)^T \in \mathbb{R}^{sd}$ of the Runge-Kutta method is a fixed-point of this iteration. Using on $\mathbb{R}^{sd}$ the norm $\|k\| = \max_{i=1,\ldots,s}|k_i|$, where $|.|$ is the regular Euclidean norm on $\mathbb{R}^d$, we obtain the estimate

$$\|F(k_1) - F(k_2)\| \leq \left(hL \max_{i=1,\ldots,s} \sum_{j=1}^{s} |a_{ij}|\right) \|k_1 - k_2\|.$$

Under assumption (3.26), the term in parentheses is strictly less than unity and thus, the mapping F is a contraction. The Banach fixed-point theorem yields the unique existence. □

3.3.9 Lemma: Let $f : \mathbb{R} \times \mathbb{C}^d \to \mathbb{C}^d$ be continuous and satisfy the one-sided Lipschitz condition with constant ν. If for $i = 1, \ldots, s$

$$h\nu < \frac{1}{a_{ii}} \tag{3.27}$$

then, for any y_0 the DIRK method (3.21) has a unique solution y_1.

Proof. The proof simplifies compared to the general case of an IRK, since each stage depends explicitly on the previous stages and implicitly only on itself. Thus, we can write

$$g_i = y_0 + v_i + h a_{ii} f(g_i), \quad \text{with} \quad v_i = h \sum_{j=1}^{i-1} a_{ij} f(g_j). \tag{3.28}$$

For linear IVP with diagonalizable matrix A, we have

$$(I - ha_{ii}A)\, g_i = y_0 + v_i,$$

and assumption (3.27) implies that all eigenvalues of $(I - ha_{ii}A)$ are positive, thus, the inverse exists and we obtain a unique solution.

In the nonlinear case, we use a homotopy argument. To this end, we introduce the parameter $\tau \in [0, 1]$ and set up the family of equations

$$g(\tau) = y_0 + \tau v_i + ha_{ii} f(g(\tau)) + (\tau - 1)ha_{ii} f(y_0).$$

For $\tau = 0$ this equation has the solution $g(0) = y_0$, and for $\tau = 1$ the solution $g(1) = g_i$. By showing, that $\frac{d}{d\tau} g$ is bounded, we conclude that a solution exists, since

$$g(1) = g(0) + \int_0^1 g'(s)\, \mathrm{d}s \tag{3.29}$$

There holds

$$g'(\tau) = v_i + ha_{ii}\partial_y f g'(\tau) + ha_{ii} f(y_0).$$

Thus

$$\begin{aligned} |g'|^2 &= \langle g', v_i + ha_{ii} f(y_0)\rangle + ha_{ii}\langle g', \partial_y f g'\rangle \\ &\leq |g'||v_i + ha_{ii} f(y_0)| + ha_{ii}\nu |g'|^2. \end{aligned}$$

Here, we used that by the mean value theorem, there holds

$$\langle \partial_y f u, u\rangle \leq \nu |u|^2, \quad \forall u \in \mathbb{C}^d.$$

We continue by stating that by assumption $1 - ha_{ii}\nu > 0$ and thus

$$|g'| \leq \frac{|v_i + ha_{ii} f(y_0)|}{1 - ha_{ii}\nu}.$$

Thus, we have proved existence of the stages g_i. On the other handy_1 is just a fixed linear combination of these values, such that it exists as well. Uniqueness follows immediately from A- or B-stability of the method. □

3.3.10 Theorem: Let be f continuously differentiable and let it satisfy the one-sided Lipschitz condition with constant ν. If the Runge-Kutta matrix A is invertible and if there is a vector $(d_1, \ldots, d_s)$ with positive entries, such that

$$h\nu < \frac{\langle x, A^{-1}x\rangle}{\sum_{i=1}^{s} d_i x_i^2}, \quad \forall x \in \mathbb{R}^s, \tag{3.30}$$

then the nonlinear system 3.21a has a solution $(g_1, ..., g_s)$.

Proof. We omit the proof here and refer to [HW10, Theorem IV.14.2] □

3.3.11 Definition (Simplifying order conditions): Define the conditions

$$B(\xi): \quad \sum_{i=1}^{s} b_i c_i^{q-1} = \frac{1}{q} \qquad q = 1, \dots, \xi \tag{3.31a}$$

$$C(\xi): \quad \sum_{j=1}^{s} a_{ij} c_j^{q-1} = \frac{c_i^q}{q} \qquad \begin{matrix} q = 1, \dots, \xi \\ i = 1, \dots, s \end{matrix} \tag{3.31b}$$

$$D(\xi): \quad \sum_{i=1}^{s} b_i a_{ij} c_i^{q-1} = \frac{b_j}{q}(1 - c_j^q) \qquad \begin{matrix} q = 1, \dots, \xi \\ j = 1, \dots, s \end{matrix} \tag{3.31c}$$

3.3.12 Theorem: If for a Runge-Kutta method condition $B(p)$ from (3.31a), condition $C(\xi)$ from (3.31b), and condition $D(\eta)$ from (3.31c) are satisfied with $\xi \geq p/2 - 1$ and $\eta \geq p - \xi - 1$, then the method is of order p.

Proof. For the proof, we refer to [HNW93, Ch. II, Theorem 7.4]. Here, we only observe, that

$$\int_0^1 t^{q-1} \, \mathrm{d}t = \frac{1}{q}, \qquad \int_0^{c_i} t^{q-1} \, \mathrm{d}t = \frac{c_i^q}{q}.$$

If we now insert the function x at the places c_i into the quadrature formula with the quadrature weights b_i, then we obtain (3.31a). Similarly we get (3.31b), if we insert the value t^q/q at the places c_i from the quadrature formula with weights a_{ij} for $j = 1, \dots, s$. In both cases we carry this out for all momomials until the desired degree is reached. Due to linearity of the formulas the exactness holds for all polynomials until this degree. □

3.4 Methods based on quadrature and B-stability

3.4.1 Gauss-, Radau-, and Lobatto-quadrature

3.4.1. In this subsection, we review some of the basic facts of quadrature formulas based on orthogonal polynomials.

3.4.2 Definition: Let $L_n(t)$ be the Legendre polynomial of degree n on $[0,1]$, up to scaling,

$$L_n(t) = \frac{d^n}{dt^n}(t^2-1)^n.$$

A quadrature formula, which uses the n roots of L_n as its quadrature points and the integrals of the Lagrange interpolating polynomials as its weights is called **Gauß quadrature**, more precisely, Gauß-Legendre quadrature.

3.4.3 Definition: The **Radau quadrature** formulas use one end point of the interval $[0,1]$ and the roots of orthogonal polynomials of degree $n-1$ as their abscissas. We distinguish left and right Radau quadrature formulas, depending on which end is included. **Lobatto quadrature** formulas use both end points and the roots of a polynomial of degree $n-2$. The polynomials are

$$\text{Radau left} \qquad p_n(t) = \frac{d^{n-1}}{dt^{n-1}}\big(t^n(t-1)^{n-1}\big), \tag{3.32}$$

$$\text{Radau right} \qquad p_n(t) = \frac{d^{n-1}}{dt^{n-1}}\big(t^{n-1}(t-1)^{n}\big), \tag{3.33}$$

$$\text{Lobatto} \qquad p_n(t) = \frac{d^{n-2}}{dt^{n-2}}\big(t^{n-1}(t-1)^{n-1}\big). \tag{3.34}$$

3.4.4 Lemma: A Gauß quadrature formula with n points is exact for polynomials of degree $2n-1$. A Radau quadrature formula with n points is exact for polynomials of degree $2n-2$. A Lobatto quadrature formula with n points is exact for polynomials of degree $2n-3$. The quadrature weights of these formulas are positive.

3.4.2 Collocation methods

3.4.5. An alternative to solving IVP in individual points in time, is to develop methods, which first approximate the solution function through a simpler function. For example this could be a polynomial.

Polynomials are especially suitable for the computation with computers. They are not suited though for high-order interpolation of large intervals. Therefore, we apply them not on the entire interval but rather on subintervals. The subintervals correspond to the time steps, which we used until now.

3.4.6 Definition: An s-stage **collocation method** with support points $c_1, \ldots, c_s$ defines a **collocation polynomial** $y(t) \in \mathbb{P}_s$ through

$$y(t_0) = y_0 \tag{3.35a}$$
$$y'(t_0 + c_i h) = f\big(t_0 + c_i h, y(t_0 + c_i h)\big) \quad i = 1, \ldots, s. \tag{3.35b}$$

We define the value at the end of the time step as

$$y_1 = y(t_0 + h). \tag{3.35c}$$

3.4.7 Lemma: A s-stage collocation method with the points c_1 to c_s defines a Runge-Kutta method of definition 3.3.2 with the coefficients c_i and

$$a_{ij} = \int_0^{c_i} L_j(t)\,\mathrm{d}t, \qquad b_i = \int_0^1 L_j(t)\,\mathrm{d}t. \tag{3.36}$$

Here is $L_j(t)$ Lagrange's interpolation polynomial to point c_j and to the point set $\{c_1, \ldots, c_s\}$:

$$L_j(t) = \prod_{\substack{k=1 \\ k \neq j}}^{s} \frac{t - c_k}{c_j - c_k}.$$

Proof. The polynomial $y'(t)$ is of degree $s - 1$ and therefore uniquely defined through s interpolation conditions in equation (3.35b). We set $y'(x_0 + c_i h) = f\big(t_0 + c_i h, y(t_0 + c_i h)\big) = k_i$, such that we have

$$y'(x_0 + th) = \sum_{j=1}^{s} k_j \cdot L_j(t) \tag{3.37}$$

with the Lagrange interpolation polynomial $L_j(t)$. By integration we obtain:

$$g_i = y(x_0 + c_i h) = y_0 + h \int_0^{c_i} y'(x_0 + th)\,\mathrm{d}t = y_0 + h \sum_{j=1}^{s} k_j \int_0^{c_i} L_j(t)\,\mathrm{d}t, \tag{3.38}$$

which defines the coefficients a_{ij} by comparison with (3.21a). If we integrate to one instead of until c_i, then we obtain the coefficients b_j by comparison with (3.21c). □

3.4.8 Lemma: An implicit s-stage Runge-Kutta method of order s or higher, with pairwise different support points c_i is a collocation method if and only if simplifying condition $C(s)$ in (3.31b) is satisfied. In other words, an s-stage method is a collocation method as soon as all the "quadrature formulas" involved are of order at least s.

Proof. Condition $C(s)$ from (3.31b) results in s^2 interpolation conditions for s^2 coefficients a_{ij}. Therefore these coefficients are defined uniquely. On the other hand (3.31b) yields for $q < s$:

$$\sum_{j=1}^{s} a_{ij} c_j^q = \frac{c_i^{q+1}}{q+1} = \int_0^{c_i} t^q \, \mathrm{d}t.$$

As a consequence of linearity we have

$$\sum_{j=1}^{s} a_{ij} p(c_i) = \int_0^{c_i} p(t) \, \mathrm{d}t, \qquad \forall p \in \mathcal{P}_{s-1}.$$

Applying this to Lagrange interpolation polynomials $L_j(t)$, we obtain the coefficients of equation (3.36), which were in turn computed from the collocation polynomial. □

3.4.9 Theorem: Consider a collocation method with s pairwise different support points c_i and define

$$\pi(t) = \prod_{i=1}^{s} (t - c_i). \tag{3.39}$$

If $\pi(t)$ is orthogonal on $[0, 1]$ to all polynomials of degree $r - 1$ for $r \leq s$, then the collocation method (3.35) is of order $p = s + r$.

Proof. The condition on π implies that the quadrature rule is exact for polynomials of degree $s + r - 1$, thus $B(s + r)$ holds. We have already shown in the proof of Lemma 3.4.8, that $C(s)$ holds. Therefore, it remains to show $D(r)$.

First, we observe that first by $C(s)$ and then by $B(s + r)$ for any $p < s$ and $q \leq r$ there holds

$$\sum_{i=1}^{s} \sum_{j=1}^{s} b_i c_i^{q-1} a_{ij} c_j^{p-1} = \frac{1}{p} \sum_{i=1}^{s} b_i c_i^{p+q-1} = \frac{1}{p(p+q)}.$$

Furthermore, since $B(s+r)$ we have for the same p and q:

$$\frac{1}{q}\sum_{j=1}^{s}\left(b_j c_j^{p-1} - b_j c_j^{p+q-1}\right) = \frac{1}{q}\left(\frac{1}{p} - \frac{1}{p+q}\right) = \frac{1}{p(p+q)}.$$

Subtracting these two and integrating the last result yields

$$0 = \frac{1}{p(p+q)} - \frac{1}{p(p+q)} = \sum_j c_j^{p-1} \underbrace{\left(\sum_i b_i c_i^{q-1} a_{ij} - \frac{1}{q} b_j \left(1 - c_j^q\right)\right)}_{:=\xi_i}.$$

This holds for $p = 1, \ldots, s-1$ and thus amounts to a homogeneous, linear system in the variables ξ_i. Thus, $\xi_i = 0$ and the theorem holds. □ Oops!

Corollary 3.4.10. *An s-stage collocation method is at least of order s and at most of order $2s$.*

Proof. The polynomial $\pi(t)$ in (3.39) is of degree s. As a result it can be orthogonal on all polynomials of degree $s-1$ in the best case. Otherwise it would be orthogonal to itself. The transformed Legendre polynomial of degree s on the interval $[0,1]$ satisfies this condition and theorem 3.4.9 holds true with $r = s$. On the other hand $\pi(t)$ is not orthogonal on the constants. In this case the theorem holds true with $r = 0$. □

3.4.11 Theorem: The collocation polynomial $y(t)$, defined through an s-stage collocation method of the form (3.35), defines a continuous Runge-Kutta method of order s. This means for the difference of the exact solution $u(t)$ of the initial value problem and the collocation polynomial $y(t)$ we get the estimate

$$|u(t) - y(t)| \leq Ch^{s+1}. \tag{3.40}$$

Additionally we obtain for the derivatives of order $k \leq s$ the estimate

$$|u^{(k)}(t) - y^{(k)}(t)| \leq Ch^{s+1-k}. \tag{3.41}$$

Proof. y' is the interpolation polynomial of degree $s-1$ in the interpolation points $c_1, \ldots, c_s$. There holds

$$\max_{t\in[t_0,t_1]} |y'(t) - u'(t)| \leq c \max_{t\in[t_0,t_1]} |u^{(s+1)}(t)| \cdot h^s.$$

We now write

$$y(t) - u(t) = \int_0^t y'(\tau) - u'(\tau)\mathrm{d}\tau \leq \int_0^t h^s \cdot c \max_{t\in[t_0,t_1]} |u^{(s+1)}(t)|\mathrm{d}\tau = ch^s t \leq ch^{h+1}.$$

Since by taking the derivative one loses one order, we obtain

$$\max_{t\in[t_0,t_1]} |y^{(k)}(t) - u^{(k)}(t)| \leq ch^{s-k+1} \max_{t\in[t_0,t_1]} |u^{(s+1)}(t)|.$$

Defining $C = c \max_{t\in[t_0,t_1]} |u^{(s+1)}(t)|$ yields the desired result. □

3.4.12 Definition: An s-stage **Gauß-Collocation method** is a collocation method, where the collocation points are the set of s Gauß points in the interval $[0, 1]$, namely the roots of the Legendre polynomial of degree s.

3.4.13 Example (2- and 3-stage Gauss collocation methods):

$\frac{3-\sqrt{3}}{6}$	$\frac{1}{4}$	$\frac{1}{4} - \frac{\sqrt{3}}{6}$
$\frac{3+\sqrt{3}}{6}$	$\frac{1}{4} + \frac{\sqrt{3}}{6}$	$\frac{1}{4}$
	$\frac{1}{2}$	$\frac{1}{2}$

$\frac{5-\sqrt{15}}{10}$	$\frac{5}{36}$	$\frac{2}{9} - \frac{\sqrt{15}}{15}$	$\frac{5}{36} - \frac{\sqrt{15}}{30}$
$\frac{1}{2}$	$\frac{5}{36} + \frac{\sqrt{15}}{24}$	$\frac{2}{9}$	$\frac{5}{36} - \frac{\sqrt{15}}{24}$
$\frac{5+\sqrt{15}}{10}$	$\frac{5}{36} + \frac{\sqrt{15}}{30}$	$\frac{2}{9} + \frac{\sqrt{15}}{15}$	$\frac{5}{36}$
	$\frac{5}{18}$	$\frac{4}{9}$	$\frac{5}{18}$

see [HNW93, Tables 7.3, 7.4]

3.4.14 Theorem: The s-stage Gauß-collocation method is consistent of order $2s$ and thus of optimal order.
The s-stage Radau- and Lobatto-collocation methods are of orders $2s-1$ and $2s-2$, respectively.

Proof. Gauß quadrature is exact for polynomials of degree $2s-1$ and we have that π in Theorem 3.4.9 is of order s. Therefore, the same theorem concludes that the method is of order $2s$. The same proof applies to Radau- and Lobatto-quadrature rules with their reduced orders. □

3.4.15 Theorem: Collocation methods with Gauß-, right Radau- and Lobatto-quadrature are B-stable. The stability region of Gauß-collocation is exactly the left half-plane of $\mathbb{C}$.

Proof. We only prove the theorem for Gauß-collocation, where the proof is simple and instructive. The proof for Radau- and Lobatto-collocation can be found in [HW10].

Let be $y(t)$ and $z(t)$ the collocation polynomials according to (3.35) with respect to initial values y_0 or z_0. Analogous to the proof of theorem 3.1.6 we introduce the auxiliary function $m(t) = |z(t)-y(t)|^2$. In the collocation points $\xi_i = t_0+c_ih$, there holds

$$\begin{aligned} m'(\xi_i) &= 2\Re\,\langle z'(t) - y'(t), z(t) - y(t)\rangle \\ &= 2\Re\,\langle f(\xi_i, z(\xi_i)) - f(\xi_i, y(\xi_i)), z(t) - y(t)\rangle \le 0. \end{aligned} \tag{3.42}$$

Since Gauß quadrature is exact for polynomials of degree $2s-1$, we have:

$$\begin{aligned} |z_1 - y_1|^2 = m(t_0+h) &= m(t_0) + \int_{t_0}^{t_0+h} m'(t)\,\mathrm{d}t \\ &= m_0 + h\sum_{i=1}^{s} b_i m'(\xi_i) \le m(t_0) = |z_0 - y_0|^2. \end{aligned}$$

Applying this argument to $f(t,u) = \lambda u$, we see A, we see from (3.42) that

$$m'(t) = 2\Re(\lambda)|z(t) - y(t)|^2,$$

which proves the statement about the stability domain. □

3.4.16 Example (2- and 3-stage right Radau collocation methods):

$$\begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array} \qquad \begin{array}{c|ccc} \frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ 1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array}$$

Remark 3.4.17. The Radau-collocation methods with right end point of the interval $[0,1]$ included in the quadrature set are L-stable. The stability regions of the first three are shown in Figure 3.4.

Observe that the stability domains are growing with order of the method. Also, observe that the computation of y_1 coincides with that of g_s, such that we can save a few operations.

3.5 Considerations on implementation

3.5.1. Implicit Runge-Kutta methods require the solution of a nonlinear system of size $s \cdot d$, where s is the number of stages and d the dimension of the system of ODE. DIRK methods are simpler and only require the solution of a system of dimension d. Thus, we should prefer this class of methods, weren't it for

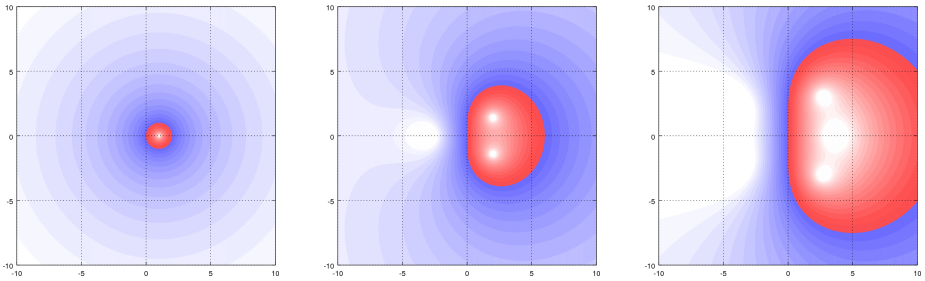


Figure 3.4: Stability domains of right Radau-collocation methods with one (implicit Euler), two, and three collocation points (left to right). Note the different scaling of coordinate axes in comparison with previous figures.

3.5.2 Theorem: A B-stable DIRK method has at most order 4

Proof. See [HW10, Theorem 13.13]. □

Remark 3.5.3. In each step of an IRK, we have to solve a (non-)linear system for the quantities g_i. First, we note that in order to reduce round-off errors, it is advantageous to solve for $z_i = g_i - y_0$, since, especially for small time steps, z_i is expected to be much smaller than g_i. Thus, we have to solve the system

$$z_i = h \sum_{j=1}^{s} a_{ij} f(t_0 + c_j h, y_0 + z_j), \quad i = 1, \dots, s. \tag{3.43}$$

Using the Runge-Kutta matrix A, we rewrite this as

$$\begin{pmatrix} z_1 \\ \vdots \\ z_s \end{pmatrix} = A \begin{pmatrix} hf(t_0 + c_1 h, y_0 + z_1) \\ \vdots \\ hf(t_0 + c_s h, y_0 + z_s) \end{pmatrix}. \tag{3.44}$$

The latter can be used to avoid additional function evaluations by computing

$$y_1 = y_0 + b^T A^{-1} z, \tag{3.45}$$

which again is numerically much more stable than evaluating f with a possibly large Lipschitz constant.

Chapter 4

Newton and quasi-Newton methods

4.1 Basics of nonlinear iterations

4.1.1. The efficient solution of nonlinear problems is an important ingredient to implicit timestepping schemes as well as shooting methods. Without attempting completeness, we present some important facts about iterative methods for this problem. We introduce the two generic schemes, Newton and gradient methods, discuss their respective pros and cons and combine their features in order to obtain better methods.

4.1.2 Definition: We consider two formulations of nonlinear root finding problems

$$f(x) = 0, \qquad f : \mathbb{R}^d \to \mathbb{R}^d \tag{4.1}$$

and

$$x = \operatorname{argmin} F(y) \qquad F : \mathbb{R}^d \to \mathbb{R}. \tag{4.2}$$

These two problems are equivalent by either choosing for instance

$$f(x) = \nabla F(x) \qquad \text{or} \qquad F(x) = |f(x)|.$$

4.1.3 Definition: An iteration

$$x^{(k+1)} = G\left(x^{(k)}\right)$$

is said to be **convergent of order** p if there holds for $p \geq 1$:

$$\|x^{(k+1)} - x^*\| \leq c\|x^{(k)} - x^*\|^p,$$

and if for $p = 1$ there holds $c < 1$. For $p > 1$, such a method converges only locally, namely if $\|x^{(0)} - x^*\|$ is sufficiently small, for instance

$$\|x^{(0)} - x^*\|^{p-1} < \frac{1}{c}.$$

4.1.4 Definition: The **Newton method** for finding the root of the nonlinear equation $f(x) = 0$ reads: given an initial value $x^{(0)}$, compute iterates $x^{(k)}$, $k = 1, 2, \ldots$ by the rule

$$\begin{aligned} J &= \nabla f\left(x^{(k)}\right), \\ Jd^{(k)} &= f(x^{(k)}), \\ x^{(k+1)} &= x^{(k)} - d^{(k)}. \end{aligned} \tag{4.3}$$

We denote by the term **quasi-Newton method** any modification of this scheme employing an approximation $\widetilde{J}$ of the Jacobian J.

4.1.5 Theorem (Newton-Kantorovich): Let $f : \mathbb{R}^d \to \mathbb{R}^d$ be differentiable with

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \qquad x, y \in \mathbb{R}^d, \tag{4.4}$$

$$\left|0\left(\nabla f\left(x^{(0)}\right)\right)^{-1}\right| 0 \leq M. \tag{4.5}$$

If

$$\beta_0 := LM\left|0 f\left(x^{(0)}\right)\right| 0 \leq \frac{1}{2}, \tag{4.6}$$

the Newton method converges to a root of $f(x)$.For $\beta_0 < 1/2$, this convergence is quadratic.

Remark 4.1.6. Instead of proving the Newton-Kantorovich theorem, we discuss its main assumptions and features. First, we note that it does not require that the initial value be close to a root, or even assumes the existence of a root. The theorem is actually an existence proof.

The Lipschitz condition on ∇f can be seen as the deviation of f from being linear. Indeed, if f were linear, then $L = 0$ and provided $M \neq 0$ the method converges in a single step for any initial value.

The larger the constant M, the smaller wone of the eigenvalues of the Jacobian. Therefore, the function becomes flat in that direction and the root finding problem becomes unstable.

If we have convergence due to $\beta_0 \leq 1/2$ (the proof shows contraction) there holds $\beta_1 := LM \left|0f\left(x^{(1)}\right)\right| 0 < 1/2$ and we have quadratic convergence from the second step on.

4.1.7 Definition: The **gradient method** for finding minimizers of a nonlinear functional $F(x)$ reads: given an initial value $x^{(0)}$, compute iterates $x^{(k)}$, $k = 1, 2, \ldots$ by the rule

$$\begin{aligned} d^{(k)} &= -\nabla F(x^{(k)}), \\ \alpha_k &= \operatorname*{argmin}_{\alpha>0} F\left(x^{(k)} + \alpha d^{(k)}\right) \\ x^{(k+1)} &= x^{(k)} + \alpha_k d^{(k)}. \end{aligned} \tag{4.7}$$

The minimization process used to compute α_k, also called **line search**, is one-dimensional and therefore simple. It may be replaced by a heuristic choice of α_k.

4.1.8 Theorem: Let $F(x) : \mathbb{R}^d \to \mathbb{R}$ be continuously differentiable and let $x^{(0)} \in \mathbb{R}^d$ be chosen such that the set

$$K = \left\{x \in \mathbb{R}^d \middle| F(x) \leq F(x^{(0)})\right\}$$

is compact. Then, each sequence defined by the gradient method has at least one accumulation point and each accumulation point is a stationary point of $F(x)$.

4.2 Globalization

4.2.1. The convergence of the Newton method is only local, and it is the faster, the closer to the solution we start. Thus, finding good initial guesses is an important task. A reasonable initial guess in a one-step method seems to be y_0, but on closer inspection, this is true only if the time step is small. Therefore, the convergence requirements of Newton's method would insert a new time

step restriction, which we want to avoid in the context of implicit methods. Therefore, and for other cases like the shooting methods of chapter 6, we present methods which extend the domain of convergence.

As a rule, Newton'a method should never be implemented without some globalization strategy!

4.2.2 Definition: The **Newton method with line search** for finding the root of the nonlinear equation $f(x) = 0$ reads: given an initial value $x^{(0)}$, compute iterates $x^{(k)}$, $k = 1, 2, \ldots$ by the rule

$$\begin{aligned} J &= \nabla f\left(x^{(k)}\right), \\ Jd^{(k)} &= f(x^{(k)}), \\ \alpha_k &= \operatorname{argmin} f(x^{(k)} - \alpha d^{(k)}) \\ x^{(k+1)} &= x^{(k)} - \alpha_k d^{(k)}. \end{aligned} \tag{4.8}$$

4.2.3 Definition: The **Newton method with step size control** for finding the root of the nonlinear equation $f(x) = 0$ reads: given an initial value $x^{(0)}$, compute iterates $x^{(k)}$, $k = 1, 2, \ldots$ by the rule

$$\begin{aligned} J &= \nabla f\left(x^{(k)}\right), \\ Jd^{(k)} &= f(x^{(k)}), \\ x^{(k+1)} &= x^{(k)} - 2^{-j} d^{(k)}. \end{aligned} \tag{4.9}$$

Here, j is the smallest integer number, such that

$$f(x^{(k)} - 2^{-j} d^{(k)}) < f(x^{(k)}), \tag{4.10}$$

for practical purposes.

Remark 4.2.4. The step size control algorithm can be implemented with very low overhead. In fact, in each Newton step we only have to compute the norm of the residual, which is typically needed for the stopping criterion anyway. Additional work is only needed if the residual grows. But this is the case, when the original method was likely to fail.

The convergence proof does not guarantee that the values of j remain bounded. Practically, this is irrelevant, since typically the step size control only triggers within the first few steps, then the quadratic convergence of the Newton method starts.

4.2.5 Definition: For a given vector $v \in \mathbb{R}^d$ and $\gamma > 0$ we define the spherical disc

$$\mathcal{S}_\gamma(v) = \Big\{ s \in \mathbb{R}^d \Big| |s| = 1 \ \wedge \ v \cdot s \geq \gamma |v| \Big\}. \tag{4.11}$$

A **descent method** is an iterative method for finding minimizers of the functional $F(x)$ that computes iterate $x^{(k+1)}$ from iterate $x^{(k)}$ by the following steps:

1. Choose a search direction:

$$s \in \mathcal{S}_\gamma\big(\nabla F(x^{(k)})\big),$$

   and a positive parameter μ.

2. Update:

$$x^{(k+1)} = x^{(k)} - \mu s.$$

Remark 4.2.6. Obviously, the gradient method is a descent method, where the direction s is chosen parallel to $\nabla F(x^{(k)})$ and μ is chosen in an optimal way. It is also called the method of **steepest descent**.

4.2.7 Lemma: The Newton method applied to the function $f(x)$ is a descent method applied to the functional $F(x) = |f(x)|^2$. The same holds for the Newton method with line search or step size control.

Proof. By the product rule, there holds

$$\nabla F(x) = 2 f^T(x) \nabla f(x).$$

The search direction of the Newton method is

$$s = -\frac{d^{(k)}}{|d^{(k)}|} = \frac{\big(\nabla f(x^{(k)})\big)^{-1} f(x^{(k)})}{|\ldots|}$$

Thus, omitting the arguments $x^{(k)}$, we obtain

$$\frac{\nabla F\, s}{\|\nabla F\|} = \frac{f^T \nabla f(x) \big(\nabla f\big)^{-1} f}{\|\big(\nabla f\big)^{-1} f\| \, \|f^T \nabla f(x)\|} \geq \frac{|f|^2}{\|\big(\nabla f\big)^{-1}\| \, \|f\|^2 \|\nabla f(x)\|} = \frac{1}{\mathrm{cond}_2(\nabla f(x))},$$

where we used the operator norm $\|.\|$ of matrices with respect tot the Euclidean norm of $\mathbb{R}^d$. $\mathrm{cond}_2(A)$ is the spectral condition of A, namely

$$\mathrm{cond}_2(A) = \|A\| \, \|A^{-1}\|.$$

With 4.11 we conclude that $s \in \mathcal{S}_\gamma(\nabla F)$ for any γ with

$$\gamma \leq \frac{1}{\mathrm{cond}_2(\nabla f(x))}.$$

The different variants of the Newton method are only distinguished by a different choice of the scaling parameter μ. □

4.2.8 Lemma: Let $F : \mathbb{R}^d \to \mathbb{R}$ be continuously differentiable. For a given point x, assume $\nabla F = \nabla F(x) \neq 0$. Then, there is a constant $\lambda > 0$ such that for any $s \in \mathcal{S}_\gamma(\nabla F(x))$ and any $0 \leq \mu \leq \lambda$ there holds

$$F(x - \mu s) \leq F(x) - \frac{\gamma\mu}{2}|\nabla F(x)|. \tag{4.12}$$

In particular, a positive scaling factor μ for the descent method can always be found.

Proof. First, define

$$U_1(x) = \big\{y \in \mathbb{R}^d \big| |\nabla F(y) - \nabla F(x)| \leq \tfrac{\gamma}{2}|\nabla F(x)|\big\}.$$

Since ∇F is continuous and $\nabla F(x) \neq 0$, this set is a nonempty neighborhood of x. Choose now λ such that

$$B_\lambda(x) \subseteq U_1(x),$$

Hence, for any $\mu \in (0, \lambda)$ and $s \in \mathcal{S}_\gamma(\nabla F(x))$, there holds by the mean value theorem with $0 < \vartheta < 1$

$$\begin{aligned} F(x) - F(x - \mu s) &= \mu \nabla F(x - \vartheta\mu s)s \\ &= \mu\Big(\big(\nabla F(x - \vartheta\mu s) - \nabla F(x)\big) + \nabla F(x)\Big). \end{aligned}$$

Using the definitions of $U_1(x)$ and $U_2(x)$, we obtain

$$\begin{aligned} F(x) - F(x - \mu s) &\geq -\tfrac{\gamma\mu}{2}|\nabla F(x)| + \mu DF(x)s \\ &\geq -\tfrac{\gamma\mu}{2}|\nabla F(x)| + \mu\gamma|\nabla F(x)| \\ &= \tfrac{\gamma\mu}{2}|\nabla F(x)|. \end{aligned}$$

□

4.3 Practical considerations

4.3.1. Quadratic convergence is an asymptotic statement, which for any practical purpose can be replaced by "fast" convergence. Most of the effort spent in a single Newton step consists of setting up the Jacobian J and solving the linear system in the second line of (4.3). Therefore, we will consider techniques here, which avoid some of this work. We will have to consider two cases

1. Small systems with $d \lesssim 1000$. For such systems, a direct method like LU- or QR-decomposition is advisable in order to solve the linear system. To this end, we compute the whole Jacobian and compute its decomposition, an effort of order d^3 operations. Comparing to d^2 operations for applying the inverse and order d for all other tasks, this must be avoided as much as possible.

2. Large systems, where the Jacobian is typically sparse (most of its entries are zero). For such a system, the effort of order d^2 for a full matrix vector multiplication is already not affordable. Therefore, the linear problem is solved by an iterative method and we will not have to compute the Jacobian at all.

Remark 4.3.2. In order to save numerical effort constructing and inverting Jacobians, the following strategies have been successful.

- Fix a threshold $0 < \eta < 1$ which will be used as a bound for error reduction. In each Newton step, first compute the update vector $\widehat{d}$ using the Jacobian $\widehat{J}$ of the previous step. This yields the modified method

$$\begin{aligned} & & J_k &= J_{k-1} \\ & & \widehat{x} &= x^{(k)} - J_k^{-1} f(x^{(k)}) \\ \text{If } |f(\widehat{x})| \le \eta |f(x^{(k)})| & & x^{(k+1)} &= \widehat{x} \\ \text{Else } J_k = \left(\nabla f(x^{(k)})\right)^{-1} & & x^{(k+1)} &= x^{(k)} - J_k^{-1} f(x^{(k)}). \end{aligned} \tag{4.13}$$

  Thus, an old Jacobian and its inverse are used until convergence rates deteriorate. This method is again a quasi-Newton method which will not converge quadratically. However, we can obtain linear convergence at any rate η.

- If Newton's method is used within a time stepping scheme, the Jacobian of the last Newton step in the previous time step is often a good approximation for the Jacobian of the first Newton step in the new time step. This holds in particular for small time steps and constant extrapolation. Therefore, the previous method should also be extended over the bounds of time steps.

- An improvement of the method above can be achieved by so called rank-1 updates. Given $x^{(k)}$ and $x^{(k-1)}$, compute

$$\begin{aligned} p &= x^{(k)} - x^{(k-1)} \\ q &= f(x^{(k)}) - f(x^{(k-1)}) \\ J_k &= J_{k-1} + \frac{1}{|p|^2}\left(q - J_{k-1}p\right)p^T \end{aligned} \tag{4.14}$$

  The fact that the rank of $J_k - J_{k-1}$ is at most one can be used to obtain a decomposition of J_k in a cheap way from one for J_{k-1}.

Remark 4.3.3. For problems leading to large, sparse Jacobians, typically space discretizations of partial differential equations, computing inverses of LU-decompositions is infeasible. These matrices typically only feature a few nonzero elements per row, while the inverse and the LU-decomposition is fully populated, thus increasing the amount of memory from d to d^2.

Linear systems like this are often solved by iterative methods, leading for instance to so called Newton-Krylov methods. Iterative methods approximate the solution of a linear system

$$Jd = f$$

only using multiplications of a vector with the matrix J. On the other hand, for any vector $v \in \mathbb{R}^d$, the term Jv denotes the directional derivative of f in direction J. Thus, it can be approximated easily by

$$Jv \approx \frac{f\left(x^{(k)} + \varepsilon v\right) - f\left(x^{(k)}\right)}{\varepsilon}.$$

The term $f\left(x^{(k)}\right)$ must be calculated anyway as it is the current Newton residual. Thus, each step of the iterative linear solver requires one evaluation of the nonlinear function, and no derivatives are computed.

The efficiency of such a method depends on the number of linear iteration steps which is determined by two factors: the gain in accuracy and the contraction speed. It turns out that typically gaining two digits in accuracy is sufficient to ensure fast convergence of the Newton iteration. The contraction number is a more difficult issue and typically requires preconditioning, which is problem-dependent and as such must be discussed when needed.

Chapter 5

Linear Multistep Methods

5.0.1. In the previous methods we obtained the value after the next time step always by using *one* initial value at the beginning of the current time interval, possibly with the help of intermediate steps. These methods often are accused to have a higher computation time than methods which use several previous points, the argument being that function values at these points have been computed already. Such methods using values of several time steps in the past are called multistep methods. They are constructed such that using more steps yields a method of higher order.

We will begin this chapter by introducing some of the formulas. Afterwards, we will study their stability and convergence properties.

Example 5.0.2 (Adams-Moulton formulas)**.** Basically, there are two construction principles for the multistep methods: Quadrature and numerical differentiation. We postpone the latter to example 5.0.4 and deal with the former for now. As first example we choose the class of Adams-Moulton methods for which the integral from point t_{k-1} to point t_k is approximated by a quadrature of the points t_{k-s} to t_k, hence

$$y_k = y_{k-1} + \sum_{r=0}^{s} f_{k-r} \int_{t_{k-1}}^{t_k} L_r(t)\,\mathrm{d}t, \tag{5.1}$$

where f_j denotes the function value $f(t_j, y_j)$ and $L_r(t)$ the Lagrange interpolation polynomial to point t_r with respect to the points $t_{k-s}, \ldots, t_k$. This is shown in Figure 5.1. Since the integral involves the point being computed itself,

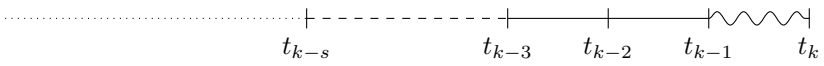


Figure 5.1: The quadrature of Adams-Moulton formulas: the integration interval is marked by the wavy line in the end. The support points of the quadrature are stated under the line.

Figure 5.2: The quadrature of Adams-Bashforth formulas: the integration interval is marked by the wavy line in the end. The support points of the quadrature are stated under the line.

these methods are implicit. The first of these are

$$\begin{aligned}
y_k &= y_{k-1} + hf_k \\
y_k &= y_{k-1} + \frac{1}{2}h\big(f_k + f_{k-1}\big) \\
y_k &= y_{k-1} + \frac{1}{12}h\big(5f_k + 8f_{k-1} - f_{k-2}\big) \\
y_k &= y_{k-1} + \frac{1}{24}h\big(9f_k + 19f_{k-1} - 5f_{k-2} + f_{k-3}\big)
\end{aligned}$$

Example 5.0.3 (Adams-Bashforth formulas)**.** With the same principle we obtain explicit methods by omitting the point in time t_k in the definition of the interpolation polynomial. See Figure 5.2. This yields quadrature formulas of the form

$$y_k = y_{k-1} + \sum_{r=1}^{s} f_{k-r} \int_{t_{k-1}}^{t_k} L_r(t)\,\mathrm{d}t. \tag{5.2}$$

Again, we list the first few:

$$\begin{aligned}
y_k &= y_{k-1} + hf_{k-1} \\
y_k &= y_{k-1} + \frac{1}{2}h\big(3f_{k-1} - 1f_{k-2}\big) \\
y_k &= y_{k-1} + \frac{1}{12}h\big(23f_{k-1} - 16f_{k-2} + 5f_{k-3}\big) \\
y_k &= y_{k-1} + \frac{1}{24}h\big(55f_{k-1} - 59f_{k-2} + 37f_{k-3} - 9f_{k-4}\big)
\end{aligned}$$

Example 5.0.4. Backward differencing formulas (BDF) are as well based on Lagrange interpolation at the points t_{k-s} to t_k. In contrast to Adams formulas

they do not use quadrature for the right hand side, but rather the derivative of the interpolation polynomial in the point t_k. Using Lagrange interpolation polynomials $L_i(t)$, we let

$$y(t) = \sum_{r=0}^{s} y_{n-r} L_{n-r}(t),$$

where y_n is still to determine. Now we assume that y solves the ODE in point t_n, hence

$$y'(t_n) = f(t_n, y_n) = \sum_{r=0}^{s} y_{n-r} L'_{n-r}(t_n).$$

This yields the following schemes:

$$\begin{aligned} y_k - y_{k-1} &= h f_k \\ y_k - \tfrac{4}{3} y_{k-1} + \tfrac{1}{3} y_{k-2} &= \tfrac{2}{3} h f_k \\ y_k - \tfrac{18}{11} y_{k-1} + \tfrac{9}{11} y_{k-2} - \tfrac{2}{11} y_{k-3} &= \tfrac{6}{11} h f_k \\ y_k - \tfrac{48}{25} y_{k-1} + \tfrac{36}{25} y_{k-2} - \tfrac{16}{25} y_{k-3} + \tfrac{3}{25} y_{k-4} &= \tfrac{12}{25} h f_k \end{aligned}$$

Remark 5.0.5. We know from introduction to numerical analysis, that the numerical differentiation and the extrapolation, the evaluation of interpolation polynomials outside of the interval which is spanned through the interpolation points, are not stable. Therefore, we expect stability problems for the Adams-Bashforth and BDF methods. Moreover we remember that Lagrange interpolation with equidistant support points is unstable for a high polynomials. Therefore, we also expect that all methods above perform well only with moderate order.

5.1 Definition and consistency of LMM

5.1.1 Definition: A **linear multistep method** (LMM) with s steps is a method of form

$$\sum_{r=0}^{s} \alpha_{s-r} y_{k-r} = h \sum_{r=0}^{s} \beta_{s-r} f_{k-r}, \tag{5.3}$$

where $f_k = f(t_k, y_k)$ and $t_k = t_0 + hk$. There are explicit ($\beta_s = 0$) and implicit ($\beta_s \neq 0$) methods. For these methods, we define the first and second **generating polynomials**

$$\varrho(x) = \sum_{r=0}^{s} \alpha_{s-r} x^{s-r} = \sum_{r=0}^{s} \alpha_r x^r \qquad \sigma(x) = \sum_{r=0}^{s} \beta_r x^r. \tag{5.4}$$

Remark 5.1.2. The LMM was defined for constant step size h. In principle it is possible to implement the method with a variable step size but we restrict ourselves to the constant case. Notes to the step size control can be found later on in this chapter.

Remark 5.1.3. One-step methods were always denoted by describing how to compute y_1 from y_0. Here, the notation becomes more complicated, but sometimes we consider only y_s computed from $y_0, \ldots, y_{s-1}$ implying the same rues for y_k computed from $y_{k-s}, \ldots, y_{k-1}$.

5.1.4 Definition: We express the LMM with the linear **difference operator**

$$(L_h u)(t_k) = \sum_{r=0}^{s} \Big(\alpha_{s-r} u(t_{k-r}) - h\beta_{R-r} f\big(t_{k-r}, u(t_{k-r})\big) \Big) \tag{5.5}$$

and for a continuous function u we define the **truncation error**

$$\tau_h(t_k) = \tfrac{1}{h} L_h u(t_k). \tag{5.6}$$

The **local error** of a linear multistep method is defined by

$$y(t_s) - y_s$$

where $u(t)$ denotes the exact solution of $u' = f(t,u), u(t_0) = u_0$ and y_s the numerical solution by using the exact initial values $y_i = u(t_i)$ for $i = 0, 1, ..., s-1$.

Lemma 5.1.5. *Consider the differential equation*

$$y' = f(t,y) \qquad y(t_0) = y_0$$

where f is given continuously differentiable and $y(t)$ is the exact solution. For the local error we obtain

$$y(t_k) - y_k = \left(\alpha_0 \mathbb{I} - h\beta_0 \frac{\partial f}{\partial y}(t_k, \eta) \right)^{-1} (L_h u)(t_k). \tag{5.7}$$

Here η is a value between $y(t_k)$ and y_k if f is a scalar function. If f is multidimensional, the matrix $\frac{\partial f}{\partial y}(t_k, \eta)$ is the Jacobi matrix, which rows are evaluated at possible places between $y(t_k)$ and y_k.

Proof. Considering the local error we can assume exact initial values and therefore we can transform 5.3 to:

$$\alpha_s y_k + \sum_{r=1}^{s} \alpha_{s-r} y(t_{k-r}) = h \left(\beta_s f_k + \sum_{r=1}^{s} \beta_{s-r} f_{k-r} \right)$$

We transform further:

$$\sum_{r=0}^{s} (\alpha_r y(t_{k-r}) - h\beta_r f(t_{k-r}, y(t_{k-r}))) \\ - \alpha_0 y(t_k) + h\beta_0 f(t_k, y(t_k)) + \alpha_0 y_k - h\beta_0 f(t_k, y_k) = 0.$$

We now insert 5.5 which results in

$$(L_h u)(t_k) = \alpha_0 \left(y(t_k) - y_k\right) - h\beta_0 \left(f(t_k, y(t_k)) - f(t_k, y_k)\right)$$
$$(y(t_k) - y_k) \left(\alpha_0 \mathbb{I} - h\beta_0 \frac{f(t_k, y(t_k)) - f(t_k, y_k)}{y(t_k) - y_k} \right)$$

By application of the mean value theorem and subsequent transformation we obtain the statement of the theorem. □

5.1.6 Definition: An LMM is consistent of order p, if for all sufficient regular functions u and all relevant k there holds

$$\tau_h(t_k) = \mathcal{O}(h^p), \tag{5.8}$$

or equivalently, that the local error is $\mathcal{O}(h^{p+1})$.

5.1.7 Lemma: An LMM is consistent of order p if and only if for all polynomials φ_q of degree $q \leq p$ and $f(t, q(t)) = q'(t)$ there holds:

$$L_h \varphi_q = 0. \tag{5.9}$$

Proof. We start with the Taylor expansion of a solution u of the ODE and the corresponding right hand side f for t_k, where we insert, unlike usual, $f = u'$:

$$u(t) = \sum_{i=0}^{p} \frac{u^{(i)}(t_k)}{i!}(t - t_k)^i + \frac{u^{(p+1)}(\xi)}{(p+1)!}(t - t_k)^{p+1} =: \varphi(t) + r_u(t)$$
$$f\big(t, u(t)\big) = \sum_{i=1}^{p} \frac{u^{(i)}(t_k)}{(i-1)!}(t - t_k)^{i-1} + \frac{u^{(p+1)}(\xi)}{p!}(t - t_k)^p =: \varphi'(t) + r_f(t),$$

with the Taylor polynomial $\varphi(t)$ of degree p and remainder $r_u(t)$ and $r_f(t)$. Out of this we calculate:

$$\begin{aligned} L_h u(t_k) = &\sum_{r=0}^{s} \alpha_{s-r}\varphi(t_{k-r}) - h\sum_{r=0}^{s}\beta_{s-r}\varphi'(t_{k-r}) \\ &+ \sum_{r=0}^{s}\alpha_{s-r}r_u(t_{k-r}) - h\sum_{r=0}^{s}\beta_{s-r}r_f(t_{k-r}). \end{aligned}$$

Since $t_{k-r} - t_k = rh$, the first row equals a polynomial $\psi(h)$ in h of degree p. For the second row we insert the reminder estimate $r_u(t) = \mathcal{O}((t-t_k)^{p+1}) = hr_f(t)$ and get:

$$L_h u(t_k) = L_h\varphi(t_k) + \mathcal{O}(h^{p+1}) = \psi(h) + \mathcal{O}(h^{p+1}). \tag{5.10}$$

According to the definition of the truncation error, this term has to be of order $p+1$, such that the method is of order p. However it is ψ of degree p. This can only hold true if $L_h\varphi = \psi \equiv 0$. On the other hand $\tau_h(t_k)$ automatically is of order p. Since u is the solution of an arbitrary right hand side, this condition has to be satisfied for all kind of Taylor polynomials φ of degree p. □

5.1.8 Theorem: A LMM with constant step size is consistent of order p if and only if

$$\begin{aligned} \sum_{r=0}^{s}\alpha_r &= 0, \\ \sum_{r=0}^{s}\big(\alpha_r r^q - q\beta_r r^{q-1}\big) &= 0, \qquad q = 1,\dots,p \end{aligned} \tag{5.11}$$

Proof. According to lemma 5.1.7 it is sufficient to show that (5.11) is equivalent to $L_h\varphi_q = 0$ for polynomials of degree $q \le p$. Due to linearity of the method it however is sufficient to show this for a basis of the polynomial space of degree p. For that we choose the monomial basis of the form

$$\pi_q(t) = \left(\frac{t - t_{k-s}}{h}\right)^q, \qquad q = 0,\dots,p.$$

For those it holds: $\pi_q(t_{k-r}) = (s-r)^q$. Now we see that the first condition is $L_h\pi_0 = 0$ (here is $\pi_0' \equiv 0$) and the second condition is $L_h\pi_q = 0$. □

Remark 5.1.9. As shown in a homework problem, a consistent LMM is not necessary convergent. To understand this behavior and develop criteria for convergence we need to diverge into the theory of difference equations.

5.2 Properties of difference equations

5.2.1. The stability of LMM can be understood by employing the fairly old theory of difference equations. In order to keep the presentation simple in this section, we use a different notation for numbering indices in the equations. Nevertheless, the coefficients of the characteristic polynomial are the same as for LMM.

5.2.2 Definition: An equation of the form

$$\sum_{r=0}^{s} \alpha_r y_{n+r} = 0 \tag{5.12}$$

is called a homogeneous **difference equation**. A sequence $\{y_n\}_{n=0,\dots,\infty}$ is solution of the difference equation, if the equation holds true for all $n \geq s$. The values y_n may be from any of the spaces $\mathbb{R}$, $\mathbb{C}$, $\mathbb{R}^d$ or $\mathbb{C}^d$. The **generating polynomial** of this difference equation is

$$\chi(x) = \sum_{r=0}^{s} \alpha_r x^r. \tag{5.13}$$

5.2.3 Lemma: The solutions of the equation (5.12) with $y_n \in \mathbb{R}$ or $y_n \in \mathbb{C}$ form a vector space of dimension s.

Proof. Since the equation (5.12) is linear and homogeneous, it is obvious that if two sequences of solutions $\{y^{(1)}\}$ and $\{y^{(2)}\}$ satisfy the equation, sums of multiples of them satisfy it too.

As soon as the initial values y_0 to y_{s-1} are chosen, all other sequence members are uniquely defined. Moreover it holds

$$y_0 = y_1 = \cdots = y_{s-1} = 0 \quad \Longrightarrow \quad y_n = 0,\ n \geq 0.$$

Therefore it is sufficient to consider the first s values. If they are linear independent, then the overall sequences are and vice versa. Thus, the initial values form a s dimensional vector space. □

5.2.4 Lemma: For each root ξ of the generating polynomial $\chi(x)$ the sequence $y_n = \xi^n$ is a solution of the difference equation (5.12).

Proof. Inserting of the solution $y_n = \xi^n$ into the difference equation results in

$$\sum_{r=0}^{s} \alpha_r \xi^{n+r} = \xi^n \sum_{r=0}^{s} \alpha_r \xi^r = \xi^n \chi(\xi) = 0.$$

□

5.2.5 Theorem: Let be $\{\xi_i\}_{i=1,\dots,\iota}$ the roots of the generating polynomial χ with multiplicity ν_i. Then, the sequences of the form

$$y_n^{(i,k)} = n^{k-1}\xi_i^n \quad i = 1, \dots, \iota; \quad k = 1, \dots, \nu_i \tag{5.14}$$

form a basis of the solution space of the difference equation (5.12).

Proof. First we observe that the sum of the multiplicities of the roots results in the degree of the polynomial:

$$s = \sum_{i=1}^{\iota} \nu_i.$$

Moreover we know because of Lemma 5.2.3, that s is the dimension of the solution space. We show that the sequences $\{y_n^{(i,k)}\}$ are linear independent. This is clear for sequences of different index i. It is also clear for different roots, because for $n \to \infty$ the exponential function nullifies the influence of the polynomials.

It remains to show that the sequences $\{y_n^{(i,k)}\}$ in fact are solutions of the difference equations. For $k = 0$ we have proven this already in lemma 5.2.4. We proof the fact here for $k = 2$ and for a double zero ξ_i; the principle for higher order roots should be clear then. Equation (5.12) applied to the sequence $\{n\xi_i^n\}$ results in

$$\begin{aligned}\sum_{r=0}^{s} \alpha_r (n+r)\xi_i^{n+r} &= n\xi_i^n \sum_{r=0}^{s} \alpha_r \xi_i^r + \xi_i^{n+1} \sum_{r=1}^{s} \alpha_r r \xi_i^{r-1} \\ &= n\xi_i^n \varrho(\xi_i) + \xi_i^{n+1} \varrho'(\xi_i) = 0.\end{aligned}$$

Here the term with α_0 vanishes, because it is multiplied with $r = 0$. $\varrho(\xi_i) = \varrho'(\xi_i) = 0$ because ξ_i is a multiple root. □

5.2.6 Corollary (Root test): All solutions $\{y_n\}$ of the difference equation (5.12) are bounded for $n \to \infty$ if and only if it holds:

- all roots of the generating polynomial $\chi(x)$ lie in the closed unit circle $\left\{z \in \mathbb{C} \middle| \, |z| \leq 1\right\}$ and
- all roots on the boundary of the unit circle are simple.

Proof. According to theorem 5.2.5 we can write all solutions as linear combinations of the sequences $y^{(i,k)}$ in equation (5.14). Therefore,

1. all solutions to $|\xi_i| < 1$ for $n \to \infty$ converge to zero
2. all solutions to $|\xi_i| > 1$ for $n \to \infty$ converge to infinity
3. all solutions to $|\xi_i| = 1$ for $n \to \infty$ stay bounded if and only if ξ_i is simple.

This proves the statement of the theorem. □

5.3 Stability and convergence

Remark 5.3.1. In contrast to one-step methods the convergence of multistep methods follows not directly from the consistency of the method, if the right hand side of the differential equation satisfies the Lipschitz condition (1.23). Analog to the A-stability we will discuss this by means of a simple model problem and we will deduce stability conditions.

Remark 5.3.2. In the following we investigate the solution to a fixed point in time t with a shrinking step size h. Therefore we choose n steps of step size $h = t/n$ and let n go towards infinity.

5.3.3 Definition: An LMM is **stable** if, applied to the trivial ODE

$$u' = 0 \tag{5.15}$$

with arbitrary initial values y_0 to y_{s-1}, it generates solutions y_k which stay bounded at each point in time $t > 0$, if the step size h converges to zero. This property is also called **zero stable** or **D-stable**.

5.3.4 Theorem: A LMM is stable if and only if all roots of the first generating polynomial $\varrho(x)$ of equation (5.4) lie in the unit circle of the complex plane and all roots on the boundary of the unit circle are simple.

Proof. The application of the LMM to the equation (5.15) results in the difference equation

$$\sum_{r=0}^{s} \alpha_{s-r} y_{n-r} = 0.$$

Now we have to proof that the solutions for fixed $t = hn$ stay bounded if $h \to 0$. But we also see that the upper equation does not contain h. Therefore we have to examine, if the solutions y_n stay bounded for $n \to \infty$. By resorting the summation we obtain a difference equation of the form (5.12). Due to corollary 5.2.6 it follows the statement of the theorem. □

5.3.5 Corollary: Adams-Bashforth and Adams-Moulton methods are stable.

Proof. For all of these methods the first generating polynomial is $\varrho(x) = x^s - x^{s-1}$. It has the simple root $\xi_1 = 1$ and the $s-1$-fold root 0. □

5.3.6 Theorem: The BDF methods are stable for $s \leq 6$ and not stable for $s \geq 7$.

5.3.7 Definition: An LMM is convergent of order p, if for any IVP with sufficiently smooth right hand side f there exists a positive constant h_0 such that for $h \leq h_0$ there holds

$$|u(t_n) - y(t_n)| \leq ch^p, \tag{5.16}$$

whenever the initial values satisfy

$$|u(t_i) - y(t_i)| \leq c_0 h^p. \tag{5.17}$$

Here, u is the continuous solution of the IVP and y is the solution generated by the LMM.

5.3.8 Lemma: Every multistep method can be recast as a one-step method

$$Y_k = (A \otimes I)Y_{k-1} + hF_h(t_{k-1}, Y_{k-1}) \tag{5.18}$$

where with $\alpha'_r = \alpha_r/\alpha_s$

$$Y_k = \begin{pmatrix} y_k \\ \vdots \\ y_{k-s+1} \end{pmatrix}, \quad A = \begin{pmatrix} -\alpha'_{s-1} & -\alpha'_{s-2} & \cdots & -\alpha'_0 \\ 1 & 0 & \cdots & 0 \\ & \ddots & \cdots & 0 \\ & & 1 & 0 \end{pmatrix}, \tag{5.19}$$

and $F_h(t_k, Y_k) = (e_1 \otimes I)\psi_h(t_{k-1}, Y_{k-1})$ with $\beta'_r = \beta_r/\alpha_s$ and ψ_h defined implicitly by

$$\begin{aligned} \psi_h(t_{k-1}, Y_{k-1}) = &\sum_{r=1}^{s} \beta_{s-r} f(t_{k-r}, y_{k-r}) \\ &+ \beta'_s f\left(t_k, h\psi_h(t_{k-1}, Y_{k-1}) - \sum_{r=1}^{s} \alpha'_{s-r} y_{k-r}\right). \end{aligned} \tag{5.20}$$

Proof. From the general form of LMM we obtain

$$\frac{1}{\alpha_s}\sum_{r=0}^{s} \alpha_{s-r} y_{k-r} = \frac{h}{\alpha_s}\sum_{r=0}^{s-1} \beta_{s-r} f_{k-r} + \beta_s f_k.$$

We rewrite this to

$$y_k = -\sum_{r=1}^{s} \alpha'_{s-r} y_{k-r} + h\psi_h(t_{k-1}, Y_{k-1}),$$

where we implicitly enter this formula as value for y_k in the computation of f_k. It remains to realize that this is the first set of d equations in (5.18), and that the remaining ones are just shifting y_i to y_{i+1}. □

5.3.9 Lemma: Let $u(t)$ be the exact solution of the IVP. For $k = s, \ldots,$ we define the vector $\widehat{Y}_k$ as the solution of a single step

$$\widehat{Y}_k = (A \otimes I)U_{k-1} + hF_h(t_{k-1}U_{k-1}),$$

with correct initial values $U_{k-1} = (u_{k-1}, u_{k-2}, \ldots, u_{k-s})^T$.
If the multistep method is consistent of order p, and f is sufficiently smooth, then there exist constants $h_0 > 0$ and M such that for $h \leq h_0$ there holds

$$\|Y_k - \widehat{Y}_h\| \leq Mh^{p+1}. \tag{5.21}$$

Proof. The first component of $Y_k - \widehat{Y}_k$ is the local error of step k, which is of order h^{p+1} by the assumption. The other components vanish by the definition of the method. □

5.3.10 Lemma: Assume that an LMM is stable. Then, there exists a vector norm on $\mathbb{C}^{sd}$ such that the operator norm of the matrix A satisfies

$$\|A \otimes I\| \leq 1. \tag{5.22}$$

Proof. We notice that $\widehat{\varrho}(x) = \sum \alpha'_{s-r} x^r$ is the characteristic polynomial of the matrix A and thus its eigenvalues are the roots of $\widehat{\varrho}(x)$, which has the same roots as the generating polynomial $\varrho(x)$. By the root test, we know that simple roots, which correspond to irreducible blocks of dimension one have maximal modulus one. Furthermore, every Jordan block of dimension greater than one corresponds to a multiple root, which by assumption has modulus strictly less than one. It is easy to see that such a block admits a modified canonical form

$$J_i = \begin{pmatrix} \lambda_i & 1-|\lambda_i| & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1-|\lambda_i| \\ & & & \lambda_i \end{pmatrix}.$$

Thus, the canonical form $J = T^{-1}AT$ has norm $\|J\|_\infty \leq 1$. If we define the norm

$$\|x\| = \|(T^{-1} \otimes I)x\|_\infty,$$

we obtain the result by

$$\begin{aligned} \|(A \otimes I)x\| = \|(T^{-1} \otimes I)(A \otimes I)x\|_\infty = \|(J \otimes I)(T^{-1} \otimes I)x\|_\infty \\ \leq \|(T^{-1} \otimes I)x\|_\infty = \|x\|. \end{aligned}$$

□

5.3.11 Theorem: If a linear multi-step method is stable and consistent of order p, then it is convergent of order p.

Proof. We reduce the proof to convergence of a one-step method with

$$Y_k = G(Y_{k-1}) = (A \otimes I)Y_{k-1} + hF_h(t_{k-1}, Y_{k-1}). \tag{5.23}$$

Let Y_{k-1} and Z_{k-1} be two initial values for the interval I_k. By the previous lemma, we have in the norm defined there, for sufficiently small h, and assuming a Lipschitz constant L_h for F_h :

$$\|G(Y_{k-1}) - G(Z_{k-1})\| \leq (1 + hL_h)\|Y_{k-1} - Z_{k-1}\|. \tag{5.24}$$

Thus, the local error $\eta_k = U_k - \widehat{Y}_k$ at step k, which by Lemma 5.3.9 is bounded by Mh^{p+1}, accumulates until step n at most to $h^{p+1}(1_h L_h)^{n-k}$.

We have:

$$\begin{aligned}
\|U_1 - Y_1\| &\leq (1 + hL_h)\|U_0 - y_0\| + Mh^{p+1} \\
\|U_2 - Y_2\| &\leq (1 + hL_h)^2\|U_0 - y_0\| + Mh^{p+1}\big(1 + (1 + hL_h)\big) \\
\|U_3 - Y_3\| &\leq (1 + hL_h)^3\|U_0 - y_0\| + Mh^{p+1}\Big(\big(1 + (1 + hL_h) + (1 + hL_h)^2\big)\Big) \\
\|U_n - Y_n\| &\leq e^{nhL_h}\|U_0 - Y_0\| + \frac{Mh^p}{L_h}\big(e^{nhL_h} - 1\big).
\end{aligned}$$

□

5.3.1 Starting procedures

5.3.12. In contrast to one-step methods, where the numerical solution is obtained solely from the differential equation and the initial value, multistep methods require more than one start value. An LMM with s steps requires s known start values $y_{k-s}, \ldots, y_{k-1}$. Mostly, they are not provided by the IVP itself. Thus, general LMM decompose into two parts:

- a *starting phase* where the start values are computed in a suitable way and
- a *run phase* where the LMM is executed.

It is crucial that the method of the starting phase provides a suitable order corresponding to the LMM of the run phase, recall Definition 5.3.7. Moreover, it should have analog properties to the LMM, like explicit/implicit or applicability to stiff problems. Possible choices for the starting phase include multistep methods with variable order and one-step methods.

Example 5.3.13 (Self starter)**.** A 2-step BDF method requires y_0 and y_1 to be known. y_0 is given by the initial value while y_1 is unknown so far. To guarantee that the method has order 2, y_1 needs to be locally of order 2 at least

$$|u(t_1) - y_1| \leq c_0 h^2. \tag{5.25}$$

This is ensured, for example, by one step of the 1-step BDF method.

However, starting an LMM with $s > 2$ steps by a first-order method and then successively increasing the order until s is reached does not provide the desired global order. That is due to the fact that the first step limits the overall convergence order to 2, compare (5.25). Nevertheless, self starters are often used in practice.

Example 5.3.14 (Runge-Kutta starter)**.** One can use Runge-Kutta methods to start LMM. Since only a fixed number of starting steps are performed, the local order of the Runge-Kutta approximation is crucial. For an implicit LMM with convergence order p and stepsize h one could use an RK method with consistency order $p - 1$ with the same stepsize h.

Consider a 3-step BDF method. Thus, beside y_0, we need start values y_1, y_2 with errors less than $c_0 h^3$. They can be computed by RK methods of consistency order 2, for example by two steps of the 1-stage Gauß collocation method with step size h since it has consistency order $2s = 2$, see theorem 3.4.14.

Example 5.3.15 (Continuous Runge-Kutta starter)**.** Another option is to use continuous Runge-Kutta methods and to evaluate the continuous approximation to obtain the required starting values.

In constrast to Example 5.3.14 one could also use the continuous polynomial approximation of Gauß collocation to start a 3-step BDF method. One step with step size $2h$ of a 2-stage Gauß collocation method would give a polynomial of degree 2 which is then evaluated in $t_1 = t_0 + h$ and $t_2 = t_1 + h$ to obtain y_1, y_2. According to Theorem 3.4.11 y_1, y_2 have the appropriate order.

Remark 5.3.16. In practice not the order of a procedure is crucial but rather the fact that the errors of all approximations (the start values and all approximations of the run phase) are bounded by the user-given tolerance, compare Section 2.4. Thus, the step sizes of all steps are controlled using local error estimates. Hence, self starting procedures usually start with very small step sizes and increase them successively. Due to their higher orders RK starters usually are allowed to use moderate step sizes in the beginning. Generally, LMM are applied with variable step sizes and orders in practice (see e.g. Exercise 7.2).

5.4 LMM and stiff problems

5.4.1 Definition (A-stability of LMM): The linear model difference equation

$$\sum_{r=0}^{s} (\alpha_{s-r} - z\beta_{s-r}) y_{n-r}. \tag{5.26}$$

is obtained by applying an LMM to the model equation $u' = \lambda u$ and inserting $z = h\lambda$.
The **stability region** of an LMM is the set of points $z \in \mathbb{C}$, for which all solution sequences $\{y_n\}$ of the equation (5.26) stay bounded for $n \to \infty$.
An LMM is called **A-stable**, if the stability region contains the left half-plane of $\mathbb{C}$.

5.4.2 Definition: The stability polynomial of an LMM is obtained by inserting $y_n = x^n$ into the linear model difference equation to obtain

$$r_z(x) = \sum_{r=0}^{s} (\alpha_{s-r} - z\beta_{s-r}) x^{s-r}. \tag{5.27}$$

Remark 5.4.3. Instead of the simple amplification function $r(z)$ of the one-step methods, we get here a function of two variables. The point z for which we want to show stability and the artificial variable x from the analysis of the method.

5.4.4 Lemma: Let $\{\xi_1(z), \ldots, \xi_s(z)\}$ be the set of roots of the stability polynomial $r_z(x)$ as functions of z. A point $z \in \mathbb{C}$ is in the stability region of a LMM, if these roots satisfy the root test in corollary 5.2.6.

Proof. The proof is analog to theorem 5.3.4. □

5.4.5 Theorem (2nd Dahlquist barrier): There is no A-stable LMM of order $p > 2$. Among the A-stable LMM of order 2, the trapezoidal rule (Crank-Nicolson) has the smallest error constant.

k	1	2	3	4	5	6
α	90°	90°	86.03°	73.35°	51.84°	17.84°
D	0	0	0.083	0.667	2.327	6.075

Table 5.1: Values for A(α)- and stiff stability for BDF methods of order k.

5.4.1 Relaxed A-stability

5.4.6. Motivated by the fact that there are no higher order A-stable LMM and by highly dissipative problems, people have introduced relaxed concepts of A-stability.

5.4.7 Definition: A set is called **A(α)-stable**, if its stability region contains the sector

$$\left\{ z \in \mathbb{C} \;\middle|\; \Re z < 0 \;\wedge\; \left|\frac{\Im z}{\Re z}\right| \leq \tan\alpha \right\}.$$

It is called **A(0)-stable**, if the negative real axis is contained in the stability region.
It is called **stiffly stable**, if it contains the set $\{\Re(z) < -D\}$.

Remark 5.4.8. The introduction of the A(0)-stability is motivated by linear systems of the form $u' = -Au$ with symmetric, positive definite matrix A. In fact one requires there only stability on the real axis because all eigenvalues are real. Thus, any positive angle α is sufficient.

Similarly A(α)-stable LMM are suitable for linear problems in which high frequently vibration ($\Im\lambda$ large) decay fast ($-\Re\lambda$ large).

In all cases one observes corresponding properties of the Jacobian matrix $\partial_u f$ for the application of nonlinear problems.

Example 5.4.9. The stability regions of the stable BDF methods are in Figure 5.3. The corresponding values for A(α)-stability and stiff stability are in Table 5.1.

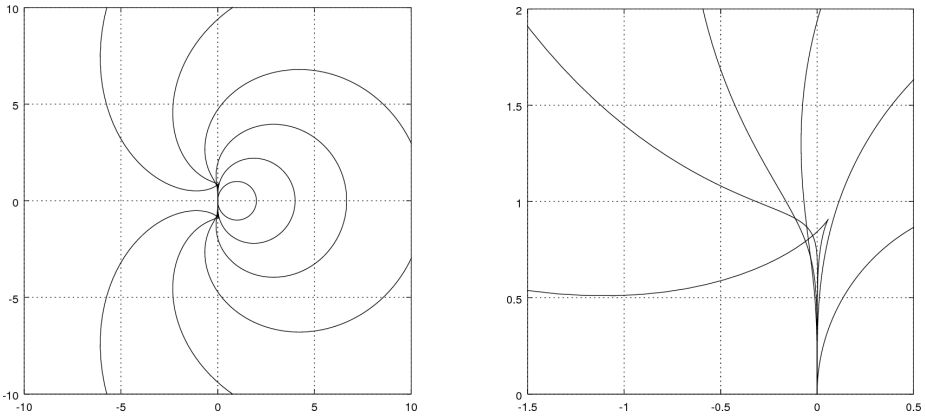


Figure 5.3: Boundaries of stability regions of BDF1 to BDF6. Unstable region right of the origin. Zoom on the right

5.5 Predictor-corrector schemes

5.5.1 Definition (Predictor-corrector methods): Assume a pair of time stepping schemes, one explicit, one implicit,

$$\hat{y}_k = \hat{F}_p(y_{k-1})$$
$$y_k = F_c(y_{k-1}, y_k),$$

we can use $\hat{y}_k$ as initial value for the Newton iteration for y_k. In an extreme case, we let

$$y_k = F_c(y_{k-1}, \hat{y}_k),$$

without any further iteration.

Remark 5.5.2. Predictor-corrector methods were developed strongly around Adams-Moulton and Adams-Bashforth methods, since the implicit ones have much smaller error constants. Given that these methods offer no considerable advantages compared to Runge-Kutta methods, but stability properties and implementation are weak points, We omit their discussion.

A simple predictor for BDF methods can be obtained, since they are based on an interpolating polynomial. Thus, we simply extrapolate this polynomial to the next point in time.

Example 5.5.3. While the predictor-corrector idea sounds reasonable, we have to be careful with stiff problems, the original reason for using implicit methods.

Take again our favorite IVP

$$u' = \lambda u, \qquad u(0) = 1.$$

We apply the BDF(1) scheme, namely the implicit Euler method, with step size 1. According to its stability function (3.16), we obtain

$$y_1 = \frac{1}{1-\lambda}.$$

Accordingly, the interpolating polynomial is

$$y(t) = (1-t) + \frac{1}{1-\lambda}t = 1 + \frac{\lambda}{1-\lambda}t.$$

For the mildly stiff problem $\lambda = -3$, we obtain

$$y_1 = 0.25, \qquad y_2 = 0.0625, \qquad \hat{y}_2 = y(2) = -0.5.$$

Thus, the extrapolated value is already a much worse initial value for a Newton iteration than using the value from the previous time step.

While this example was particularly chosen to exhibit such failure, it does show that extrapolation of stiff problems has its pitfalls. Here, we end up with a time step restriction which is comparable to the stability condition of the explicit method.

Chapter 6

Boundary Value Problems

6.1 Introduction

6.1.1. This chapter deals with problems of a fundamentally different type than the problems we examined in chapter 1 and which we solved with previous numerical methods, namely boundary value problems. Here, we have prescribed values at the beginning and at the end of an interval of interest.

The representation we use here is based primarily on [DB08] and [Ran17].

6.1.2 Definition: A **boundary value problem** (BVP) is a differential equation problem of the form: Find $u : [a, b] \to \mathbb{R}^d$, such that

$$u'(t) = f\big(t, u(t)\big) \qquad t \in (a, b) \tag{6.1a}$$

$$r\big(u(a), u(b)\big) = 0. \tag{6.1b}$$

6.1.3 Definition: A BVP (6.1) is called linear, if the right hand side f as well as the boundary conditions are linear in u. It has the form: find $u : [a, b] \to \mathbb{R}^d$, such that

$$u'(t) = A(t)u(t) + b(t) \qquad \forall t \in (a, b) \tag{6.2a}$$

$$B_a u(a) + B_b u(b) = g. \tag{6.2b}$$

Remark 6.1.4. Since boundary values are imposed at two different points in time, the concept of local solutions from definition 1.2.8 is not applicable. Thus, tricks as going forward from interval to interval, which is for instance done with

Euler's method in the proof of Péano's theorem, are here not applicable. For this reason nothing can be concluded with the local properties of the right hand side f at the points a and b. In fact, it is just possible in a few special cases to conclude that a solution exists.

6.2 Derivatives of the solutions of IVP with respect to data

6.2.1 Derivatives with respect to the initial values

6.2.1. In order to understand BVP, we have to introduce the notion of the derivative of the solution to an IVP with respect to its initial values. Our interest lies in the change of u if the initial value is changed. We will thus denote in these cases $u = u(t; v)$, where t is the usual "time" variable and v the initial value, which now is a variable as well. Thus, $u(t; v)$ is the solution to the IVP

$$\begin{aligned} u'(t;v) = \tfrac{\partial}{\partial t}u(t;v) = f\big(t, u(t;v)\big) \\ u(t_0;v) = v. \end{aligned} \tag{6.3}$$

The purpose of this section is the study of the derivative

$$\tfrac{\partial}{\partial v}u(t;v),$$

which is fundamentally different from $\partial/\partial t u(t;v)$. It can be obtained by solving the variational equation of the original IVP, defined as follows.

6.2.2 Definition: Let $F(v)$ be a function defined on some function space. Then, the Gâteaux derivative of F at a point v in direction w with v and w in this function space is defined as

$$\frac{\partial}{\partial w}F(v) = \lim_{\varepsilon \to 0} \frac{F(v + \varepsilon w) - F(v)}{\varepsilon}, \tag{6.4}$$

if this limit exists.
Here, we have used the notation for directional derivatives in $\mathbb{R}^n$, since this is indeed the character of the Gâteaux derivative.

6.2.3 Definition: The **variational equation** to the first order system of ODE

$$u' = f,$$

of dimension d is the linear matrix-valued system of ODE

$$Y' = \nabla_u f\big(t, u(t)\big)Y \tag{6.5a}$$

for $d \times d$ matrices Y. Here is u a solution of the equation (1.4) and

$$\nabla_u f(t,u) = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_d} \\ \vdots & & \vdots \\ \frac{\partial f_d}{\partial u_1} & \cdots & \frac{\partial f_d}{\partial u_d} \end{pmatrix}$$

is the matrix of the derivatives of f with respect to the components of u. The **fundamental matrix** $Y(t;t_0)$ is solution of the IVP to this equation with

$$Y(t_0) = \mathbb{I}. \tag{6.5b}$$

Remark 6.2.4. The fundamental matrix Y can also be read column by column. Then each column is a vector-valued function $\varphi^{(i)}(t)$ and solves the IVP

$$\begin{aligned} \tfrac{d}{dt}\varphi^{(i)}(t) &= \nabla_u f\big(t, u(t)\big)\varphi^{(i)}(t), \\ \varphi^{(i)}(t_0) &= e_i. \end{aligned}$$

Remark 6.2.5. The definition of the fundamental matrix here is consistent with the one in Definition 1.3.14 for linear equations. Namely, for $f(u) = Au$, we have $\nabla_u f(u) = A$.

6.2.6 Lemma: For fundamental matrices there hold the relations

$$Y(t;s) = Y(s;t)^{-1} \tag{6.6}$$

$$Y(t;r) = Y(t;s)Y(s;r), \tag{6.7}$$

where r, s, t are arbitrary real numbers, such that the solution u of the original IVP exists on the maximal interval spanned by these numbers.

Proof. In order to prove the first equation, denote by $V(\tau;s)$ the solution to the IVP

$$V'(\tau;s) = \nabla_u f(\tau, u(\tau))V(\tau;s), \qquad V(s;s) = Y(s;t).$$

Because of uniqueness, we must have $V(\tau;s) = Y(\tau;t)$ for any τ between s and t, in particular for $r = t$, such that $V(t;s) = Y(t;t) = \mathbb{I}$. On the other hand, by linearity, we have $V(\tau;s) = Y(\tau;s)Y(s;t)$, and thus the equation is proven by

$$\mathbb{I} = V(t;s) = Y(t;s)Y(s;t).$$

Now, assume without loss of generality that s is between r and t. Indeed, if for instance t is between r and s, multiply equation (6.7) from the left by $Y(s;t)$ and prove the equation for

$$Y(s;t)Y(t;r) = Y(s;t)Y(t;s)U(s;r) = Y(s;r).$$

Take the auxiliary function $V(\tau;s)$ as defined above. By uniqueness, it is equal to $Y(\tau;t)$ for all τ. But, on the other hand, we have by linearity $V(\tau;s) = Y(\tau;s)Y(s;r)$, in particular for $\tau = t$.

The statement follows from the definition as a solution of an IVP and the fact that solutions of linear IVP are linear combinable. □

6.2.7 Theorem: Let be $f(t,u)$ continuous in t and continuously differentiable in u. Then, the solution $u(t;v)$ of the IVP (6.3) depends differentiably on the initial value v and the derivative is given by

$$\tfrac{\partial}{\partial v}u(t;v) = Y(t;t_0), \tag{6.8}$$

where $Y(t;t_0)$ is the fundamental matrix with respect to the initial time t_0.

Proof. We write the IVP in its full dependence on v as

$$\begin{aligned}\frac{\partial u(t;v)}{\partial t} &= f\big(t,u(t;v)\big)\\ u(t_0;v) &= v.\end{aligned}$$

From the second equation, we immediately obtain

$$\frac{\partial u(t_0;v)}{\partial v} = \mathbb{I}.$$

Assuming differentiability of f with respect to u, the first equation yields

$$\frac{\partial}{\partial v}\frac{\partial u(t;v)}{\partial t} = \frac{\partial f\big(t,u(t;v)\big)}{\partial v} = \nabla_u f(t,u(t;v))\frac{\partial u(t;v)}{\partial v}.$$

Thus, $\frac{\partial}{\partial v}u$ solves the IVP (6.3)

$$\left(\frac{\partial}{\partial v}u\right)' = \nabla_u f(t,u(t;v))\frac{\partial u(t;v)}{\partial v}.$$

$\frac{\partial}{\partial v}u$ solves the same IVP as the fundamental matrix and thus, they coincide. □

6.2.2 Derivatives with respect to the right hand side function

6.2.8. We close this section by studying the differential dependence of the solution $u(t)$ of an ODE at time t on the function $f(t,u)$, that is, the derivative of a value with respect to a function. In order to keep things simple, we reduce this question to a regular derivative of a function with respect to a real variable by using the Gâteaux derivative. Back to differential equations, our task is now to compute the derivative of $u(t)$ with respect to changes in f, denoted as

$$\frac{\partial}{\partial g}u(t) = \lim_{\varepsilon\to 0}\frac{u_\varepsilon - u}{\varepsilon} = \left.\frac{d}{d\varepsilon}u_\varepsilon(t)\right|_{\varepsilon=0}, \tag{6.9}$$

where u and u_ε respectively solve the IVPs

$$\begin{aligned} u' &= f(t,u) & u(t_0) &= u_0 \\ u'_\varepsilon &= f(t,u_\varepsilon) + \varepsilon g(t,u_\varepsilon) & u_\varepsilon(t_0) &= u_0. \end{aligned}$$

For this derivative, we have the following theorem.

6.2.9 Theorem: Let $f(t,u)$ and $g(t,u)$ be continuous in their first and continuously differentiable in their second argument. Let u be the solution of the IVP $u' = f(u)$ with $u(t_0) = u_0$. Then, the Gâteaux derivative of u in f with respect to a perturbation g exists and there holds

$$\frac{\partial}{\partial g}u(t) = \int_{t_0}^{t} Y(t;s)g\big(s,u(s)\big)\,\mathrm{d}s. \tag{6.10}$$

Proof. We set out by devising a differential equation for the Gâteaux derivative $\mathcal{U}(t) := \frac{d}{d\varepsilon}u_\varepsilon(t)\big|_{\varepsilon=0}$. The differential equation for u yields

$$\begin{aligned} \mathcal{U}'(t) &= \left(\left.\frac{d}{d\varepsilon}u_\varepsilon(t)\right|_{\varepsilon=0}\right)' \\ &= \left.\frac{d}{d\varepsilon}u'_\varepsilon(t)\right|_{\varepsilon=0} \\ &= \left.\frac{d}{d\varepsilon}\Big(f\big(t,u_\varepsilon(t)\big) + \varepsilon g\big(t,u_\varepsilon(t)\big)\Big)\right|_{\varepsilon=0} \\ &= \left.\nabla_u f(t,u_\varepsilon)\frac{d}{d\varepsilon}u_\varepsilon(t) + \varepsilon\nabla_u g\big(u_\varepsilon(t)\big)\frac{d}{d\varepsilon}u_\varepsilon(t) + g\big(t,u_\varepsilon(t)\big)\right|_{\varepsilon=0} \\ &= \nabla_u f(t,u_\varepsilon)\mathcal{U}(t) + g\big(t,u_\varepsilon(t)\big)\big|_{\varepsilon=0} \end{aligned}$$

Furthermore, we have

$$\mathcal{U}(t_0) = \frac{d}{d\varepsilon} u_\varepsilon(t_0) = 0.$$

According to Lemma 1.3.5, the solution of this initial value problem can be represented with the integrating factor $M(t)$ as

$$\mathcal{U}(t) = M^{-1}(t) \int_{t_0}^{t} M(s) g\big(s, u(s)\big) \, \mathrm{d}s.$$

Noticing that $M(\tau)^{-1} = Y(\tau; t_0)$, we obtain

$$u(t) = \int_{t_0}^{t} Y(t; s) g\big(s, u(s)\big) \, \mathrm{d}s$$

□

6.3 Theory of boundary value problems

Remark 6.3.1. The very general boundary condition (BC) (6.1b) usually has more simple forms. Often it is a **linear** linear boundary condition which we can note in the following form

$$B_a u(a) + B_b u(b) = g \tag{6.11}$$

with $d \times d$ matrices B_a and B_b as well as a vector $g \in \mathbb{R}^d$. Another very common case is the one of **separated** boundary conditions, which has the form

$$r_a\big(u(a)\big) = 0, \qquad r_b\big(u(b)\big) = 0, \tag{6.12}$$

or

$$B_a u(a) = g_a, \qquad B_b u(b) = g_b. \tag{6.13}$$

Example 6.3.2. Take the second order differential equation

$$u''(t) = -2, \quad \forall t \in (0, 1),$$

with boundary conditions

$$u(0) = u(1) = 0.$$

We can deduce from the differential equation that the solution is a parabola open to the bottom, and we verify easily that

$$u(t) = t(1 - t)$$

is a solution.

Example 6.3.3. Take the first order differential equation

$$u'(t) = u(t), \quad \forall t \in (0,1),$$

with boundary values

$$u(0) = u(1) = 1.$$

From the theory of the first chapter, we know that the initial value problem with only the initial value at zero has the unique solution $u(t) = e^t$. Thus, this BVP does not have a solution.

If we changed the condition at the right end of the interval to $u(1) = e$, the problem is solvable, albeit we have to admit that this solution seems somewhat accidental.

Remark 6.3.4. As the examples show, a satisfying theory for the existence of solutions will be difficult to obtain for any boundary values. For instance, in the linear case it is obvious that neither B_a, nor B_b may have full rank, because that would imply the unique existence either of the IVP with $B_a u(a) = g_a$ or $B_b u(b) = g_b$, and thus no freedom to match the condition at the other end.

As a consequence we now turn our attention to a "restricted" solution theory and wonder: assume a solution of the problem exists. Which further conditions are necessary to obtain well-posedness of the problem in terms of Hadamard (definition 1.4.1 on page 16).

The key is the following definition which grants us the possibility of the approximation of a solution, at least after a quantification of the neighborhood.

6.3.5 Definition: A solution $u(t)$ of the BVP (6.1) is called **locally unique** or **isolated**, if there is no second solution $v(t)$ of the BVP, which is arbitrary close to $u(t)$. In mathematical language: there exists an $\varepsilon > 0$, such that for any two solutions of the BVP there holds

$$\max_{t\in[a,b]} |u(t) - v(t)| < \varepsilon \qquad \Rightarrow \qquad u(t) = v(t) \quad \forall t \in [a,b].$$

6.3.6 Lemma: Let be $f(t,u)$ continuous in t and continuously differentiable in u. Let additionally $r(x,y)$ be continuously differentiable and set

$$B_a = \left.\frac{\partial r(x,y)}{\partial x}\right|_{x=u(a),y=u(b)}, \qquad B_b = \left.\frac{\partial r(x,y)}{\partial y}\right|_{x=u(a),y=u(b)}. \quad (6.14)$$

Let $u(t)$ be a continuously differentiable solution of the BVP (6.1). Then the derivative of the boundary condition $r(u(a),u(b))$ with respect to the function value $u(t)$ inside the interval $[a,b]$ is

$$E(t) := \frac{\partial r\big(u(a),u(b)\big)}{\partial u(t)} = B_a Y(a;t) + B_b Y(b;t), \quad (6.15)$$

where $Y(t;t_0)$ is the fundamental matrix.

Remark 6.3.7. The definition of the matrices B_a and B_b above is consistent with the usage of the matrix B_a and B_b in equation (6.11).

Proof. We consider the auxiliary function $w_t(\tau;v)$ as solution of the IVP with initial value v in t:

$$\frac{\partial}{\partial\tau} w_t(\tau;v) = f\big(\tau, w_t(\tau;v)\big), \qquad w_t(t;v) = v.$$

Choosing $v = u(t)$, we have by uniqueness $w_t(a;v) = u(a)$ and $u(b) = w_t(b;v)$. Our task of computing the derivative with respect to $u(t)$ has thus become computing the derivative with respect to v. The derivative of the boundary condition can therefore be written as

$$\begin{aligned}\frac{\partial r\big(u(a),u(b)\big)}{\partial u(t)} &= \frac{\partial r\big(w_t(a;v),w_t(b;v)\big)}{\partial v}\\ &= B_a\frac{\partial w_t(a;v)}{\partial v} + B_b\frac{\partial w_t(b;v)}{\partial v}\\ &= B_a Y(a;t) + B_b Y(b;t),\end{aligned}$$

where the last equality is due to Theorem 6.2.7. □

Remark 6.3.8. In order to study local uniqueness of solutions and well-posedness of BVP, we have to change our view on boundary conditions and consider them as functions of solutions to the differential equation. Thus, we will consider the function

$$\varrho(v) := r\big(v(a),v(b)\big),$$

mapping solutions of the differential equation to their boundary values. ϱ is a continuous function, and, as the following theorem shows, even differentiable.

6.3.9 Theorem: Let the assumptions of Lemma 6.3.6 hold. If the matrix $E(t)$ is regular for at least a single value $t \in [a, b]$, then it is regular for all $t \in [a, b]$ and the solution $u(t)$ is locally unique.

Proof. First assume that $E(t)$ is regular for some t. Then we have for $\tau \neq t$:

$$\begin{aligned} E(\tau) &= B_a Y(a;\tau) + B_b Y(b;\tau) \\ &= B_a Y(a;t)Y(t;\tau) + B_b Y(b;t)Y(t;\tau) = E(t)Y(t;\tau), \end{aligned}$$

where the first factor is regular by assumption, the second one as the fundamental matrix of a linear ODE.

Now, we consider the matrix $E(t) = E_u(t)$ as a function of $u(t)$, which is continuous by assumption. Therefore, if it is regular in $u(t)$, it is regular in a neighborhood of $u(t)$ of some positive diameter ε. Let now v be a second solution of the differential equation with $|u(t) - v(t)| < \varepsilon$. Then,

$$\varrho(u) - \varrho(v) = E_\varphi(t)\big(u(t) - v(t)\big),$$

where $\varphi(t)$ is between $u(t)$ and $v(t)$ and thus $E_\varphi(t)$ is regular. If both functions solve the BVP, then the left hand side is zero, and thus $v(t) = u(t)$. □

Now that we established the local uniqueness of the solution, it yet remains to show the continuous dependency of data (stability). Here we are in particular interested, in analogy to the stability theorem, in the derivative of the solution at time t after perturbations of values on the boundary. For this purpose we have:

6.3.10 Theorem: Let the assumptions of Lemma 6.3.6 hold and let $u(t)$ be the solution of the BVP (6.1) with

$$u(a) = g_a \quad \text{and} \quad u(b) = g_b.$$

Then, there holds

$$\frac{\partial u(t)}{\partial g_a} = E^{-1}(t)B_a, \qquad \frac{\partial u(t)}{\partial g_b} = E^{-1}(t)B_b. \tag{6.16}$$

In particular, the conditioning with respect to changes of size ε in the boundary conditions g_a and g_b is

$$\delta u(t) \leq \varepsilon \max\big\{\|E^{-1}(t)B_a\|, \|E^{-1}(t)B_b\|\big\}. \tag{6.17}$$

Proof. We demonstrate the proof for the derivative with respect to the left boundary value. The second equation can be proven the same way. With the chain rule we obtain

$$\frac{\partial u(t)}{\partial g_a} = \frac{\partial u(t)}{\partial r(g_a, g_b)} \frac{\partial r(g_a, g_b)}{\partial g_a}$$

The second derivative is (6.14) B_a. For the first one we notice that $E(t)$ is the derivative of the inverse mapping of $\varrho(u)$. By application of the implicit function theorem, we obtain the result. □

6.3.11 Theorem: Let $f(t,u)$ and $g(t,u)$ be continuous in t and continuously differentiable in u. Then, the variation of the value $u(t)$ of the solution u of the boundary value problem

$$u' = f(t,u), \qquad r\big(u(a), u(b)\big) = 0,$$

with respect to perturbations $f + g$ is

$$\frac{\partial}{\partial g} u(t) = \int_a^b G(t,s) g\big(s, u(s)\big)\, ds, \tag{6.18}$$

where

$$G(t,s) = \begin{cases} -E(t)^{-1} B_a Y(a;s) & a \le s \le t \\ E(t)^{-1} B_b Y(b;s) & t < s \le b \end{cases}. \tag{6.19}$$

Proof. We begin by estimating the influence of perturbations of the right hand side on the boundary values. Using the corresponding Theorem 6.2.9 for IVP, where we swap the meaning of t and the interval boundaries, we obtain

$$\frac{\partial u(a)}{\partial g} = \int_t^a Y(a;s) g\big(s, u(s)\big)\, \mathrm{d}s,$$
$$\frac{\partial u(b)}{\partial g} = \int_t^b Y(b;s) g\big(s, u(s)\big)\, \mathrm{d}s,$$

By the chain rule and Lemma 6.3.6 and Theorem 6.3.10, we get

$$\frac{\partial r\big(u(a), u(b)\big)}{\partial g} = \frac{\partial r\big(u(a), u(b)\big)}{\partial u(t)} \frac{\partial u(t)}{\partial g} = E(t) \frac{\partial u(t)}{\partial g}.$$

Assembling everything, we obtain

$$\begin{aligned} \frac{\partial u(t)}{\partial g} &= E(t)^{-1} \frac{\partial r\big(u(a), u(b)\big)}{\partial g} = E(t)^{-1} \left(B_a \frac{\partial u(a)}{\partial g} + B_b \frac{\partial u(b)}{\partial g} \right) \\ &= E(t)^{-1} \left(B_a \int_t^a Y(a;s) g\big(s, u(s)\big)\, \mathrm{d}s + B_b \int_t^b Y(b;s) g\big(s, u(s)\big)\, \mathrm{d}s \right). \end{aligned}$$

□

Remark 6.3.12. Theorems 6.3.9, 6.3.10 and 6.3.11 represent the verification of the second and third Hadamard conditions. Thus, even if the existence of a solution for the BVP is not always guaranteed, solutions can be approximated under certain conditions.

The case for linear boundary value problems is much simpler and we have an existence an uniqueness result.

6.3.13 Corollary: The linear BVP (6.2) has an unique solution $u(t)$ for arbitrary data $f(t)$ and g if and only if the $d \times d$ matrix

$$E(a) = B_a + B_b Y(b; a)$$

is regular.

Proof. By Theorem 6.3.6 and linearity of the BVP, we deduce that the mapping ϱ from $u(t)$ to the boundary condition is affine and can be written in the form

$$r\big(u(a), u(b)\big) = E(t)u(t) + b(t),$$

where $b(t)$ is some vector in $\mathbb{R}^d$ or $\mathbb{C}^d$ independent of u. Since everything in this equation exept $u(t)$ is given, the unique solvability is equivalent to the invertibility of $E(t)$, which by Theorem 6.3.9 is equivalent to regularity of $E(a)$. □

6.4 Shooting methods

6.4.1 Single shooting method

Example 6.4.1. We illustrate the shooting method on a simple, scalar example

$$u'' = -g, \quad u(0) = 0, \quad u(1) = 0.$$

What we can solve is the IVP

$$u'' = -g, \quad u(0) = 0, \quad u'(0) = s.$$

The latter has a unique solution u(t;s) for each initial value s. Now it is our task to find a value s^*, such that $u(1; s^*) = 0$. With other words, we search for a root of the function

$$F(s) = u(1; s).$$

This can be done with an arbitrary, convergent iteration method. For an example with the Bisection method. Of course the Newton method would be a better choice but for that we need to calculate the derivatives of F. This can be achieved with theorem 6.2.7 by calculating the fundamental matrix Y.

6.4.2 Definition: The **single shooting method** for the BVP (6.1) reads as follows: find an initial vector $u_0 \in \mathbb{R}^d$, such that the solution $u(t) = u(t; u_0)$ of the IVP

$$\tfrac{\partial}{\partial t} u(t; u_0) = f\big(t, u(t; u_0)\big), \qquad u(a; u_0) = u_0$$

satisfies the boundary conditions $r\big(u(a), u(b)\big) = 0$.

Remark 6.4.3. The task of the shooting method is solved normally with the help of Newton's method, which searches for a root (with respect to v) of the function

$$F(v) = r\big(v, u(b; v)\big) \tag{6.20}$$

For Newton's method we require the partial derivatives

$$\frac{\partial}{\partial v^{(i)}} F(v) = \partial_1 r\big(v, u(b; v)\big) + \partial_2 r\big(v, u(b; v)\big) \frac{\partial}{\partial v^{(i)}} u(b; v),$$

which involves the fundamental matrix $Y(b; a)$ of derivatives of the solution at point b with respect to the initial values (see Lemma 6.3.6).

6.4.4 Algorithm: The single shooting method with standard Newton method consists of the following steps:

1. Start with an initial guess $v^{(0)} \in \mathbb{R}^d$.
2. For $v^{(n)}$ given, solve the IVP and the associated variation equation

$$\begin{aligned} \tfrac{\partial}{\partial t} u^{(n)}(t) &= f\big(t, u^{(n)}(t)\big) & u^{(n)}(a) &= v^{(n)}, \\ \tfrac{\partial}{\partial t} Y^{(n)}(t; a) &= \nabla_u f\big(t, u^{(n)}(t)\big) Y^{(n)}(t; a) & Y^{(n)}(a; a) &= \mathbb{I}. \end{aligned}$$

3. Set

$$v^{(n+1)} = v^{(n)} - \big(B_a + B_b Y^{(n)}(b; a)\big)^{-1} r\big(v^{(n)}, u^{(n)}(b)\big) \tag{6.21}$$

4. Stop the iteration if the value $r\big(v, u(t; u_0^{(n+1)})\big)$ is sufficiently small, otherwise repeat from 2.

Remark 6.4.5. The IVP in this algorithm usually cannot be solved analytically. Thus, we have to use what we learned in the previous chapters to choose a time stepping scheme.

Newton's method is known to converge locally, not globally. Therefore, a good initial value is needed for this algorithm to converge. This is the more true, since the solution of the IVP may grow much faster and may be less stable than that of the BVP (see homework), and is only approximated. There are two ways out of this problem: first, a Newton method should never be implemented without any globalization strategy, which modifies the update to increase the domain of convergence. The second part of the solution consists in choosing a method which is more robust than single shooting in the next section.

Remark 6.4.6. Step 2 of the single shooting algorithm requires solving the variational equation of our original ODE. If f describes a complex nonlinear process, the computation and implementation of its derivative may be a daunting and error prone task. This can be avoided by three different algorithmical tools. For each of them, we describe the main advantages and disadvantages:

Automatic differentiation Software like for instance the module Sacado of the Trilinos package, has the standard rules of differentiation (Leibniz, quotient, chain rules, derivatives of polynomials and standard functions) built in. By implementing $f(u)$ in a conforming way, the software can automatically generate the code for its derivatives. Clearly, the advantage is that there is no approximation involved in those derivatives and thus equation and variational equation are always consistent. On the other hand, derivatives, which are not simplified analytically, can become fairly complex. Here, we rely on a good implementation of the automatic differentiation as well as on the optimizing capabilities of the compiler.

Internal numerical differentiation While using a time stepping scheme for solving the IVP, for example a one-step method, in each time step, we compute

$$y^k = \Phi(y^{k-1})$$

as well as approximations

$$\frac{\partial}{\partial y_i} y^k \approx \frac{\Phi(y^{k-1} + \varepsilon e_i) - \Phi(y^{k-1})}{\varepsilon}$$

with the same integration method Φ and the same step sizes h_k. This approximation can also be replaced by more accurate differentiation formulas. Such an implementation requires d additional evaluations of Φ to compute the full gradient, which is feasible only for moderately sized d. Furthermore, the choice of ε is tricky, since the approximation is inaccurate for large values and unstable for small. Implementations of time

stepping schemes which compute both y and its derivatives are available in the optimization community as well among reseach groups implementing filtering techniques in stochastic methods.

External numerical diferentiation Here we forget about the variational equation altogether and approximate by difference quotients the derivative of $u(b)$ with respect to changes in the initial value directly:

$$\frac{\partial}{\partial v_i} u(b) \approx \frac{u(b; v + \varepsilon e_i) - u(b; v)}{\varepsilon}.$$

In practice, $u(b; v + \varepsilon e_i)$ and $u(b; v)$ are computed numerically by any integration method and in general using different step sizes. The robust choice of ε is more critical here and the search for a general algorithm to address this issue has failed. Therefore, this method is only used rarely nowadays (judgement from [DB08]).

A final remark on computing the Jacobian and its inverse: if an iterative method is used to solve the linear system in each Newton step, the complete matrix is not needed, but only the matrix applied to a direction vector. This is something that can be used to accelerate all methods described above by avoiding the computation of unneeded values.

6.4.2 Multiple shooting method

Example 6.4.7. Take on the interval $[0, 2]$ the (admittedly somewhat artificial) boundary value problem

$$u' = u^2, \qquad u(2) = 1,$$

which has the bounded solution

$$u(t) = \frac{1}{3 - t}.$$

We might be tempted to start our shooting method with $v = 1$ after realizing that $v = 0$ leads nowhere. But then, we get

$$u^{(0)}(t) = \frac{1}{1 - t},$$

which only exists on the interval $[0, 1)$. Clearly, the single shooting method is not suited to solve this otherwise harmless BVP.

This observation leads to the idea of applying the shooting method on smaller subintervals and gluing those together.

6.4.8 Definition: Choose a partitioning of the interval $[a, b]$ such that

$$a = t_0 < t_1 < t_2 < \cdots < t_m = b.$$

On each subinterval $I_k = [t_{k-1}, t_k]$, $k = 1, \ldots, m$ define the IVP

$$u_k' = f(t, u_k), \qquad u_k\big(t_{k-1}\big) = v_k,$$

The **multiple shooting method** consists of finding vectors $v_1, \ldots, v_m$, such that

$$\begin{aligned} v_{k+1} &= u_k\big(t_k\big) && k = 1, \ldots, m-1, \\ r(v_1, u_m(b)) &= 0. \end{aligned}$$

The function

$$u(t) = u_k(t) \qquad \text{for } t \in I_k$$

is continuous, solves the ODE, and obeys the boundary conditions.

Remark 6.4.9. The numbering of intervals, vectors, and time partitioning has been chosen to be consistent with the previous definitions in this class. The governing entity is the subinterval $I_k = [t_{k-1}, t_k]$ with solutions u_k and initial value v_k. As a result, the initial values v_k are imposed at t_{k-1}.

Other authors have used the time subdivisions t_k as governing entity, which leads to a shift of several of the indices. Whenever reading or writing about multiple shooting methods, connections between indices must be considered carefully, since every system will exhibits inconsistencies at some points.

Remark 6.4.10. The multiple shooting method is a typically nonlinear system of equations of dimension $d \cdot m$, where d is the dimension of the ODE and m the number of subintervals. Nevertheless, the formulation and typically the implementation hides a much larger number of unknowns involved in the discretization of the subdomain solves. We will keep ignoring this inner discretization of the intervals.

Remark 6.4.11. In order to keep the implementation and presentation simpler, we introduce an additional shooting vector $v_{m+1} = u_m(t_m)$. As a result, the boundary condition in the shooting method simplifies the last conditions to

$$r\big(v_1, v_{m+1}\big) = 0. \tag{6.22}$$

The advantage becomes obvious, when we compute derivatives for the Newton method. With the new vector, we compute

$$\frac{\partial r\big(v_1, v_{m+1}\big)}{\partial v_1} = B_a, \qquad \frac{\partial r\big(v_1, v_{m+1}\big)}{\partial v_{m+1}} = B_b.$$

Without, we have to compute

$$\frac{\partial r\big(v_1, u_m(b)\big)}{\partial v_m} = B_b Y(t_m; t_{m-1}).$$

6.4.12 Definition: A step of Newton's method for the multiple shooting system consists of the update

$$v^{(n+1)} = v^{(n)} - \nabla F(v^{(n)})^{-1} F(v^{(n)}), \tag{6.23}$$

where $v^{(n)} = [v_1^{(n)}, \ldots, v_{m+1}^{(n)}]^T$,

$$F(v) = \begin{bmatrix} F_1(v_1, v_2) \\ \vdots \\ F_m(v_m, v_{m+1}) \\ F_{m+1}(v_1, v_{m+1}) \end{bmatrix}, \quad \nabla F(v) = \begin{bmatrix} G_1 & -\mathbb{I} & & \\ & \ddots & \ddots & \\ & & G_m & -\mathbb{I} \\ B_a & & & B_b \end{bmatrix}, \tag{6.24}$$

and,

$$\begin{aligned} F_k(v_k, v_{k+1}) &= u_k(t_k) - v_{k+1} && k = 1, \ldots, m, \\ F_{m+1}(v_1, v_{m+1}) &= r\big(v_1, v_{m+1}\big), \\ G_k &= Y(t_k; t_{k-1}) && k = 1, \ldots, m. \end{aligned}$$

Remark 6.4.13. Whenever the shooting vectors $v = [v_1, \ldots v_{m+1}]^T$ solve the multiple shooting equations $F(v) = 0$, the solution $u(t)$ of the multiple shooting method is also a solution of the original BVP. This is a consequence of the continuity enforced by these equations. Therefore, the existence of a solution to the original BVP implies existence of a solution to the multiple shooting problem.

6.4.14. We close this section by discussing two important extensions to the multiple shooting method. First, a system may have boundary values in more than just the two end points of the einterval of computation. In such a case, the additional points are included as shooting nodes, such that the boundary conditions can be applied to the shooting vectors v_k instead of solutions of the initial value problems on subintervals.

The second extension is to problems, where the equation itself has a parameter which we try to determine by the shooting method. It turns out that both extensions fit very well into concept of multiple shooting and change the underlying Newton method only slightly.

6.4.15 Definition: A multi-point boundary value problem has boundary conditions of the form

$$r\big(u(t_0), u(t_{k_1}), u(t_{k_2}), \dots, u(t_{k_\ell})\big) = 0, \tag{6.25}$$

with $a = t_0$, $m = k_\ell$, and $b = t_m$ A multiple shooting method for such a problem can be designed by including all values t_{k_i} into the partitioning of the time interval. The corresponding shooting function and Jacobian are

$$F(v) = \begin{bmatrix} F_1(v_1, v_2) \\ \vdots \\ F_m(v_m, v_{m+1}) \\ F_{m+1}(v_1, \dots, v_{m+1}) \end{bmatrix}, \quad \nabla F(v) = \begin{bmatrix} G_1 & -\mathbb{I} & & & \\ & G_2 & -\mathbb{I} & & \\ & & \ddots & \ddots & \\ & & & G_m & -\mathbb{I} \\ B_a & \cdots & B_{k_i} & \cdots & B_b \end{bmatrix}. \tag{6.26}$$

Here,

$$\begin{aligned} F_k(v_k, v_{k+1}) &= u_k(t_k) - v_{k+1} && k = 1, \dots, m, \\ F_{m+1}(v_1, \dots, v_{m+1}) &= r\big(v_1, \dots, v_{m+1}\big), \\ G_k &= Y(t_k; t_{k-1}) && k = 1, \dots, m. \end{aligned}$$

6.4.16 Definition: Given a vector $p \in \mathbb{R}^q$, a parameter dependent boundary value problem depending on p has the form

$$\begin{aligned} u' &= f(t, u; p), \\ r\big(u(a), u(b); p\big) &= 0. \end{aligned} \tag{6.27}$$

Here, $r(\dots) \in \mathbb{R}^{d+q}$ where d is the dimension of the ODE system.

Remark 6.4.17. Every parameter dependent ODE can be transformed into a regular ODE by introducing the $d+q$-dimensional vector $v = (u, p)$ solving the ODE

$$v' = \begin{pmatrix} f\big(t, u(t)\big) \\ 0 \end{pmatrix}.$$

Solving it in this form is nevertheless inefficient, since it involves carrying a differentia lequation for p through all integrators. Instead, we can modify the shooting method in a way, that we incorporate it directly.

6.4.18 Definition: The Jacobian of the Newton method for parameter dependent BVP is

$$\nabla F(v) = \begin{bmatrix} G_1 & -\mathbb{I} & & & P_1 \\ & \ddots & \ddots & & \\ & & G_m & -\mathbb{I} & P_{m-1} \\ B_a & & & B_b & P_m \end{bmatrix}. \tag{6.28}$$

Here,

$$\begin{aligned} P_k &= \frac{\partial u_k(t_k)}{\partial p} && k = 1, \ldots, m, \\ P_m &= \frac{\partial r(\ldots)}{\partial p}, \\ G_k &= Y(t_k; t_{k-1}; p) && k = 1, \ldots, m. \end{aligned}$$

Chapter 7

Second Order Boundary Value Problems

7.1 2nd order two-point boundary value problems

7.1.1. We have already seen, that boundary value problems have very different stability properties than initial value problems. Here, we will discuss a special class of boundary value problems of the form

$$-u''(x) + \beta(x)u'(x) + \gamma(x)u(x) = f(x), \qquad u(a) = u_a, \quad u(b) = u_b. \tag{7.1}$$

In order to make this problem more amenable to mathematical investigation, we introduce the set

$$\mathcal{B} = \Big\{ u \in C^2(a,b) \cap C[a,b] \;\Big|\; u(a) = u_a \;\wedge\; u(b) = u_b \Big\}.$$

Then, we can see the left hand side of the differential equation as a differential operator applied to u and thus mapping $\mathcal{B}$ to the set of continuous functions. Namely, we define

$$\begin{aligned} L : \mathcal{B} &\to C[a,b] \\ u &\mapsto -u'' + \beta u' + \gamma u. \end{aligned} \tag{7.2}$$

In addition, we would like to simplify our life and get rid of the inhomogeneous boundary values u_a and u_b. To this end, let

$$u_B(x) = u_a \frac{b-x}{b-a} + u_b \frac{x-a}{b-a},$$

and introduce the new function $u_0 = u + u_B$. Then, u_0 solves the boudnary value problem

$$-u_0''(x) + \beta(x)u'(x) + \gamma(x)u(x) = f(x) - \beta(x)\frac{u_b - u_a}{b-a} - \gamma(x)u_B(x),$$
$$u(a) = u(b) = 0.$$

Thus, it is sufficient to consider the boundary value problem

7.1.2 Definition: Given an interval $I = [a, b]$, find a function

$$u \in V = \Big\{ u \in C^2(a,b) \cap C[a,b] \;\Big|\; u(a) = u(b) = 0 \Big\}, \tag{7.3}$$

such that for a differential operator of second order as defined above and a right hand side $f \in C[a, b]$ there holds

$$Lu = f. \tag{7.4}$$

Remark 7.1.3. This definition exhibits a major change in paradigm. Before, we considered a differential equation as an equation which determines the derivative of a function in a point. Now, we are looking at a linear system of equations, albeit one, which is not of finite dimension. This paradigm change will be essential when we consider partial differential equations in future semesters.

On the other hand, the equality in equation (7.4) is understood point-wise, such that in fact nothing but our point of view has changed.

7.1.4. Again, we subdivide the interval $I = [a, b]$ into subintervals, but the subdivision does not involve IVP solvers on subintervals, but much more like in the original subdivision in Definition 2.1.2, the the solution will only be defined at the partitioning points t_k, $k = 0, \ldots, n$.

Thus, like with one-step and multistep methods we will have values $y_0, y_1, \ldots, y_n$, but the sequence has a defined end at y_n due to the right boundary of the interval.

While one-step methods directly discretize the Volterra integral equation in order to compute a solution at every new step, **finite difference methods** discretize the differential equation on the whole interval at once and then solve the resulting discrete (finite-dimensional) system of equations.

We have accompished the first step and decided that instead of function values in every point of the interval I, we only approximate $u(t_k)$ in the points of the partition. What is left is the definition of the discrete operator representing the equation.

7.1.5 Definition (Finite differences): In order to approximate first derivatives of a function u, we introduce the operators

$$\text{Forward difference} \qquad D_h^+ u(x) = \frac{u(x+h) - u(x)}{h}, \tag{7.5}$$

$$\text{Backward difference} \qquad D_h^- u(x) = \frac{u(x) - u(x-h)}{h}, \tag{7.6}$$

$$\text{Central difference} \qquad D_h^c u(x) = \frac{u(x+h) - u(x-h)}{2h}. \tag{7.7}$$

For second derivatives we introduce the

$$\text{3-point stencil} \qquad D_h^2 u(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}. \tag{7.8}$$

Remark 7.1.6. The 3-point stencil is the product of forward and backward difference operators.

$$D_h^2 u(x) = D_h^+ u(x) D_h^- u(x) = D_h^- u(x) D_h^+ u(x).$$

For simplicity, we only present finite differences of uniform subdivisions. Nevertheless, the definition of the operators can be extended easily to h changing between intervals.

7.1.7 Definition: A finite difference operator D_h^α is consistent with the αth derivative of order p, if there holds for any $u \in C^{\alpha+p}$:

$$|u^{(\alpha)} - D_h^\alpha| \leq ch^p. \tag{7.9}$$

7.1.8 Lemma: The difference operators have the following consistency orders

$$|u'(x) - D_h^+ u(x)| \leq ch \tag{7.10}$$

$$|u'(x) - D_h^h u(x)| \leq ch \tag{7.11}$$

$$|u'(x) - D_h^c u(x)| \leq ch^2 \tag{7.12}$$

$$|u''(x) - D_h^2 u(x)| \leq ch^2 \tag{7.13}$$

Proof. We begin to show consistency of the first two operators by Taylor ex-

pansion: for some $\xi \in (x, x+h)$, there holds

$$\begin{aligned} u'(x) - D_h^+ u(x) &= u'(x) - \frac{u(x+h) - u(x)}{h} \\ &= u'(x) - \frac{u(x) + hu'(x) + \frac{h^2}{2}u''(\xi) - u(x)}{h} \\ &= \tfrac{h}{2}u''(\xi). \end{aligned}$$

The same computation can be applied to $D_h^- u(x)$. It is clear that we need an additional symmetry argument for te other two, otherwise their consistency order would be lower. Therefore, we follow the line of argument that we introduced in Lemma 5.1.7, and which here reads: a difference operator D_h^α approximating a derivative of order α is consistent of order p, if and only if it is exact for all polynomials of degree $p + \alpha - 1$. We realize this by computing the Taylor polynomial $p(h)$ of degree $p+\alpha-1$ and the remainder term involving $u^{(p+\alpha)}(\xi)$. Then,

$$D_h^\alpha u(x) = \frac{p(h) + \frac{h^{\alpha+p}}{(\alpha+p)!}u^{(p+\alpha)}(\xi)}{h^\alpha}.$$

Now we employ that the formula is exact for $p(h)$ and thus

$$u^{(\alpha)}(x) - D_h^\alpha u(x) = \frac{h^p}{(\alpha+p)!}u^{(p+\alpha)}(\xi).$$

We now write

$$p(\xi) = a_0 + a_1(\xi - x) + a_2(\xi - x)^2 + a_3(\xi - x)^3 + \cdots$$

The central difference D_h^c is exact for linear polynomials, since ti evaluates to zero for a constant and $D_h^c(\xi - x) = 1$. But additionally, we observe

$$\frac{d}{d\xi}(\xi - x)^2\Big|_{\xi=x} = D_h^c(\xi - x)^2\Big|_{\xi=x} = 0.$$

Thus, the central difference is exact for polynomials of degree 2 and consistent of second order.

For the 3-point stencil, we observe that $D_h^2 u(x) = 0$ for any function u such that $u(x+h) - u(x-h) = u(x)$, in particular any odd polynomial in $\xi - x$. Furthermore,

$$D_h^2(\xi - x)^2\Big|_{\xi=x} = \frac{h^2 - 0 + h^2}{h^2} = 2 = \frac{d^2}{d\xi^2}(\xi - x)^2$$

□

Remark 7.1.9. When applied to the equation $u' = f(t, u)$ the solutions obtained by forward and backward differences correspond to the explicit and implicit Euler methods, respectively.

7.1.10 Definition: The **finite difference method** for the discretization of the boundary value problem $Lu = f$ with homogeneous boundary values on the interval $I = [a, b]$ is obtained by

1. choosing a partition $a = t_0, t_1, \ldots, t_n = b$ with
$$t_{k-1} - t_k = h = (b - a)/n.$$
2. only considering the discrete solution valus y_k, $k = 0, \ldots, n$.
3. replacing all differential operators by finite differences in t_k.

7.1.11 Example: Using the 3-point stencil and central difference, we obtain from the BVP

$$-u''(x) + \beta(x)u'(x) + \gamma(x)u(x) = f(x), \qquad u(a) = u(b) = 0,$$

the discrete system of equations

$$y_0 = 0 \tag{7.14}$$

$$\frac{2y_k - y_{k-1} - y_{k+1}}{h^2} + \beta_k \frac{y_{k+1} - y_{k-1}}{2h} + \gamma_k y_k = f_k \quad k = 1, \ldots, n-1 \tag{7.15}$$

$$y_n = 0, \tag{7.16}$$

or short

$$\widetilde{L}_h y = f \tag{7.17}$$

Remark 7.1.12. Like our view to the continuous boundary value problem has changed, the discrete one is now a fully coupled linear system which has to be solved by methods of linear algebra, not by time stepping anymore. In fact, we have $n+1$ variables $y_0, \ldots, y_n$ and $n+1$ equations, such that here existence and uniqueness of solutions are equivalent.

7.1.13 Example: The linear system obtained in Example 7.1.11 has a tridiagonal matrix L_h and reads

$$\begin{pmatrix} 1 & & & & \\ \mu_1 & \lambda_1 & \nu_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & \lambda_{n-1} & \nu_{n-1} \\ & & & & 1 \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} = \begin{pmatrix} 0 \\ f_1 \\ \vdots \\ f_{n-1} \\ 0 \end{pmatrix}, \tag{7.18}$$

where

$$\lambda_k = \frac{2}{h^2} + \gamma_k \qquad \begin{aligned} \mu_k &= -\frac{1}{h^2} - \frac{\beta_k}{2h} \\ \nu_k &= -\frac{1}{h^2} + \frac{\beta_k}{2h} \end{aligned}$$

Remark 7.1.14. The first and last row of the matrix L_h are redundant, since they simply say $y_0 = y_n = 0$. They can be eliminated, such that we obtain the reduced system

$$L_h y = \begin{pmatrix} \lambda_1 & \nu_1 & & \\ \mu_2 & \lambda_2 & \ddots & \\ & \ddots & \ddots & \nu_{n-2} \\ & & \mu_{n-1} & \lambda_{n-1} \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} f_1 \\ \vdots \\ f_{n-1} \end{pmatrix} = f_h. \tag{7.19}$$

In this form, the operator L_h is consistent with L in the sense that it only describes the differential operator, not the boundary values. Therefore, we will use this form in our further analysis.

When it comes to implementation, both versions have their merits. Obviously, the new operator involves less unknowns. On the other hand, the discretization with boundary unknowns is more straight-forward.

7.2 Existence, stability, and convergence

7.2.1. Since the solution of the discretized boundary value problem is a problem in linear algebra, we have to study properties of the matrix L_h. The shortest and most elegant way to prove stability is through the properties of M-matrices, which we present here very shortly. We are not dwelling on this approach too long, since it is sufficient for stability, but by far not necessary and constrained to low order methods.

The fact that L_h is an M-matrix requires some knowledge of irreducible weakly diagonal dominant matrices, which the author considers as outdated as the whole concept of m-matrices. We will just quote this result without proof.

7.2.2 Definition: An **M-matrix** A is a quadratic $n \times n$-matrix with the following properties:

$$a_{ii} > 0, \quad a_{ij} \leq 0, \qquad i, j = 1, \ldots, n, \quad j \neq i. \tag{7.20}$$

For the entries c_{ij} of A^{-1} there holds

$$C_{ij} \geq 0, \qquad i, j = 1, \ldots, n. \tag{7.21}$$

7.2.3 Lemma: The matrix L_h defined above is an M-matrix provided that

$$\gamma_k \geq 0, \qquad |\beta_k| < \frac{2}{h}. \tag{7.22}$$

Proof. It is clear that these two conditions are sufficient for the first M-matrix property. The proof of positivity of the inverse is based on irreducible diagonal dominance, which is too long and too specialized for these notes. □

Remark 7.2.4. The finite element method provides much more powerful to deduce solvability and stability of the discrete problem.

7.2.5 Lemma: Let A be an M-matrix. If there is a vector w such that for the vector $v = Aw$ there holds

$$v_i \geq 1, \quad i = 1, \ldots, n,$$

then

$$\|A^{-1}\|_\infty \leq \|w\|_\infty. \tag{7.23}$$

Proof. Let $x \in \mathbb{R}^n$ and $y = A^{-1}x$. Then,

$$\begin{aligned} |y_i| &= |\sum c_{ij} x_j| \\ &\leq \sum c_{ij} |x_j| \\ &\leq \|x\|_\infty \sum c_{ij} v_j. \end{aligned}$$

Thus,

$$|y_i| \leq \|x\|_\infty \big(A^{-1}v\big)_i = \|x\|_\infty \big(A^{-1}Aw\big)_i \leq \|x\|_\infty |w_i|.$$

Taking the maximum over all i, we obtain

$$\|A^{-1}\|_\infty = \sup_{u\in\mathbb{R}^n} \frac{\|A^{-1}u\|_\infty}{\|u\|_\infty} \le \|w\|_\infty.$$

□

7.2.6 Theorem: Assume that (7.22) holds. Then, the matrix L_h defined in (7.18) is invertible.
Let there furthermore be a constant $\delta < 2$ such that

$$|\beta_k||b-a| - \gamma_k|b-a|^2 \le \delta. \tag{7.24}$$

Then. the inverse admits the estimate

$$\|L_h^{-1}\|_\infty \le \frac{(b-a)^2}{8-4\delta}. \tag{7.25}$$

Proof. Take the function

$$p(x) = (x-a)(b-x) = -x^2 + (a+b)x - ab,$$

with derivatives $p'(x) = a+b-2x$ and $p''(x) = -2$, and a maximum of $(b-a)^2/4$ at $x = (a+b)/2$. Choose the values $p_k = p(x_k)$. By consistency, we have and $k = 1, \dots, n-1$

$$(L_h p)_k \ge 2 - \beta_k|b-a| + \gamma_k|b-a|^2 \ge 2 - \delta.$$

Thus, the vector with entries $w_k = p_k/(2-\delta)$ can be used to bound the inverse of L_h by Lemma 7.2.5. □

Remark 7.2.7. The assumptions of the previous theorem involve two sets of conditions on the parameters β_k and γ_k. Condition (7.24) is actually a condition on the continuous problem. The condition on γ_k is indeed necessary, as will be seen when we study partial differential equations. The condition on β_k is not necessary in this form, but a better estimate again requires far advanced analysis.

The other set of conditions relates the coefficients to the mesh size. Again, the condition on β_k can be avoided as seen in the next example. The condition on γ_k is already implied by $-\gamma_k \le (b-a)^2$, which is a small restriction compared to (7.24), as soon as the partition has 3 interior points. Thus, it is not crucial.

7.2.8 Example: By changing the discretization of the first order term to an **upwind** finite difference method, we obtain an M-matrix independent of the relation of β_k and h. To this end define

$$\beta(x)D_h^{\uparrow}u(x) = \begin{cases} \beta(x)D_h^{-}u(x) & \text{if } \beta(x) > 0 \\ \beta(x)D_h^{+}u(x) & \text{if } \beta(x) < 0 \end{cases}. \tag{7.26}$$

This changes the matrix L_h to a matrix $L_h^{\uparrow}$ with entries

$$\lambda_k = \frac{2}{h^2} + \frac{|\beta_k|}{h} + \gamma_k \qquad \begin{aligned} \mu_k &= -\frac{1}{h^2} - \frac{\max\{0,\beta_k\}}{h} \\ \mu_k &= -\frac{1}{h^2} + \frac{\min\{0,\beta_k\}}{h} \end{aligned} \tag{7.27}$$

As a consequence, the off-diagonal elements always remain non-positive and the diagonal elements remain positive only subject to a condition on γ_k. Thus, $L_h^{\uparrow}$ is an M-matrix independent of the values β_k. Nevertheless, the consistency order is reduced to one.

7.2.9 Theorem: Let $I = (a, b)$ and V according to Definition 7.1.2. Let $u \in V \cap C^{p+2}(I)$ be the solution of the 2-point boundary value problem $Lu = f$. Let L_h be the matrix of a finite difference approximation $L_h y = f_h$ according to (7.19). Let this method be consistent of order p and stable in the sense that $\|L_h^{-1}\|_\infty$ is bounded independent of h.
Then, the method is consistent of order p and for any right hand side f there is a constant c independent of h such that

$$\max_{0\le k\le n} |u_k - y_k| \le ch^p. \tag{7.28}$$

Proof. We apply the difference operator L_h to u (as the vector of function values in the points t_k) and y to obtain

$$L_h(u - y) = (L_h - L)u + Lu - L_h y = \tau + f - f = \tau,$$

where $\tau = (\tau_1, \ldots, \tau_{n-1})^T$ is the vector, which measures the consistency error $(L_h - L)u$ in each t_k. The entries τ_k are bounded by ch^p by the consistency estimate. For the error, there holds

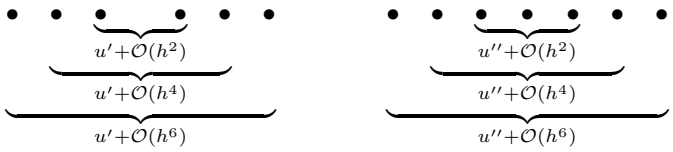
$$u - y = L_h^{-1}L_h(u - y) = L_h^{-1}\tau.$$

Using the stability assumption, we obtain

$$\|u - y\|_\infty \le \|L_h^{-1}\|_\infty \|\tau\|_\infty \le ch^p,$$

where the constant c depends on $\|L_h^{-1}\|_\infty$ and the constant in the consistency estimate, but not on h. □

Remark 7.2.10. Finite differences can be generalized to higher order by extending the stencils by more than one point to the left and right of the current point. Whenever we add two points to the symmetric difference formulas, we can gain two orders of consistency.



Similarly, we can define one-sided difference formulas, which get us close to multistep methods. The matrices generated by these formulas are not M-matrices anymore, although you can show for the 4th order formula for the second derivative that it yields a product of two M-matrices. While this rescues the theory in a particular instance, M-matrices do not provide a theoretical framework for general high order finite differences anymore.

Very much like the starting procedures for high order multistep methods, high order finite differences cause problems at the boundaries. Here, the formulas must be truncated and for instance be replaced by one-sided formulas of equal order.

All these issues motivate the study of different discretization methods in the next course.

7.3 The Laplacian and harmonic functions

7.3.1. The two-point boundary value problem has a natural extension to higher dimensions. There, we deal with partial derivatives $\frac{\partial}{\partial x}$, $\frac{\partial}{\partial y}$, and $\frac{\partial}{\partial z}$. As an outlook towards topics discussed in classes on partial differential equations and their numerical analysis, we close these notes by a short introduction at hand of examples.

7.3.2 Definition: the **Laplacian** in two (three) space dimensions is the sum of the second partial derivatives

$$\Delta u = \frac{\partial^2}{\partial x^2} u + \frac{\partial^2}{\partial y^2} u \left(+ \frac{\partial^2}{\partial z^2} u \right) = \operatorname{div}(\nabla u) \tag{7.29}$$

The **Laplace equation** is the partial differential equation

$$-\Delta u = 0. \tag{7.30}$$

The **Poisson equation** is the partial differential equation

$$-\Delta u = f. \tag{7.31}$$

Solutions to the Laplace equations are called **harmonic functions**.

7.3.1 Properties of harmonic functions

7.3.3 Theorem (Mean-value formula for harmonic functions): Let $u \in C^2(\Omega)$ be a solution to the Laplace equation. Then, u has the mean value property

$$u(\mathbf{x}) = \frac{1}{r^{d-1}\omega(d)} \int_{\partial B_r(\mathbf{x})} u(\mathbf{y})\, ds, \tag{7.32}$$

where $\partial B_r(\mathbf{x}) \subset \Omega$ is the sphere of radius r around $\mathbf{x}$ and $\omega(d)$ is the volume of the unit sphere in $\mathbb{R}^d$.

Proof. First, we rescale the problem to

$$\Phi(r) = \frac{1}{r^{d-1}\omega(d)} \int_{\partial B_r(\mathbf{x})} u(\mathbf{y})\, ds = \frac{1}{\omega(d)} \int_{\partial B_1(0)} u(\mathbf{x} + r\mathbf{z})\, ds.$$

Then, we notice that

$$\begin{aligned}
\Phi'(r) &= \frac{1}{\omega(d)} \int_{\partial B_1(0)} \nabla u(\mathbf{x} + r\mathbf{z}) \cdot \mathbf{z}\, ds_z \\
&= \frac{1}{r^{d-1}\omega(d)} \int_{\partial B_r(\mathbf{x})} \nabla u(\mathbf{y}) \cdot \frac{\mathbf{y} - \mathbf{x}}{r}\, ds_y \\
&= \frac{1}{r^{d-1}\omega(d)} \int_{\partial B_r(\mathbf{x})} \frac{\partial}{\partial \mathbf{n}} u(\mathbf{y})\, ds_y \\
&= \frac{1}{r^{d-1}\omega(d)} \int_{B_r(\mathbf{x})} \Delta u(\mathbf{y})\, d\mathbf{y} = 0.
\end{aligned}$$

Between the last two lines, we used the Gauß theorem for the vector valued function ∇u. Therefore, $\Phi(r)$ is constant. Because of continuity, we have

$$\lim_{r\to 0}\Phi(r) = \lim_{r\to 0}\frac{1}{r^{d-1}\omega(d)}\int_{\partial B_r(\mathbf{x})} u(\mathbf{y})\,ds = u(\mathbf{x}),$$

which proves our theorem. □

7.3.4 Theorem (Maximum principle): Let a function $u \in C^2(\Omega)$ be a solution to the Laplace equation on an open, bounded, connected domain Ω. Then, if there is an interior point $\mathbf{x}_0$ of Ω, such that for a neighborhood $U \subset \Omega$ of $\mathbf{x}_0$ there holds

$$u(\mathbf{x}_0) \geq u(\mathbf{x}) \qquad \forall \mathbf{x} \in U,$$

then the function is constant in Ω.

Proof. Let $\mathbf{x}_0$ be such a maximum and let $r > 0$ such that $B_r(\mathbf{x}_0) \subset \Omega$. Assume that there is a point $\mathbf{x}$ on $\partial B_r(\mathbf{x}_0)$, such that $u(\mathbf{x}) < u(\mathbf{x}_0)$. Then, this holds for points $\mathbf{y}$ in a neighborhood of $\mathbf{x}$. Thus, in order that the mean value property holds, there must be a subset of $\partial B_r(\mathbf{x}_0)$ where $u(\mathbf{y}) > u(\mathbf{x}_0)$, contradicting that $\mathbf{x}_0$ is a maximum. Thus, $u(\mathbf{x}) = u(\mathbf{x}_0)$ for all $\mathbf{x} \in B_r(\mathbf{x}_0)$ for all r such that $B_r(\mathbf{x}_0) \subset \Omega$.

Let now $\mathbf{x} \in \Omega$ be arbitrary. Then, there is a (compact) path from $\mathbf{x}_0$ to $\mathbf{x}$ in Ω. Thus, the path can be covered by a finite set of overlapping balls inside Ω, and the argument above can be used iteratively to conclude $u(\mathbf{x}) = u(\mathbf{x}_0)$. □

Corollary 7.3.5. *Let $u \in C^2(\Omega)$ be a solution to the Laplace equation. Then, its maximum and its minimum lie on the boundary, that is, there are points $\underline{\mathbf{x}}, \overline{\mathbf{x}} \in \partial\Omega$, such that*

$$u(\underline{\mathbf{x}}) \leq u(\mathbf{x}) \leq u(\overline{\mathbf{x}}) \quad \forall \mathbf{x} \in \Omega.$$

Proof. If the maximum of u is attained in an interior point, the maximum principle yields a constant solution and the theorem holds trivially. On the other hand, the maximum principle does not make any prediction on points at the boundary, which therefore can be maxima. The same holds for the minimum, since $-u$ is a solution to the Laplace equation as well. □

Corollary 7.3.6. *Solutions to the Poisson equation with homogeneous boundary conditions are unique.*

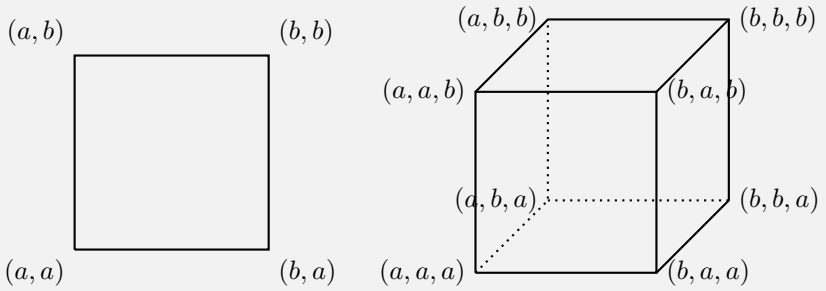
Proof. Assume there are two functions $u, v \in C^2(\Omega)$ with $u = v = 0$ on $\partial\Omega$ such that

$$-\Delta u = -\Delta v = f.$$

Then, $w = u - v$ solves the Laplace equation with $w = 0$ on $\partial\Omega$. Due to the maximum principle, $w \equiv 0$ and $u = v$. □

7.4 Finite differences

7.4.1 Example: The notion of an interval I can be extended to higher dimensions by a square $\Omega = I^2$ or a cube $\Omega = I^3$.



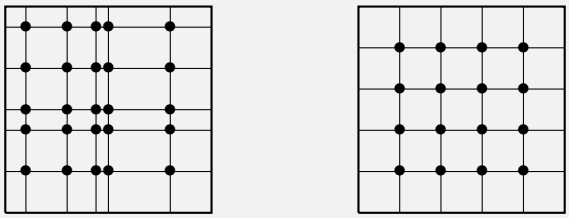
We call such a square or cube **Cartesian**, if its edges and faces are aligned with the coordinate axes.

7.4.2 Example: We consider Dirichlet boundary conditions

$$u(\mathbf{x}) = u_B(\mathbf{x}), \qquad \text{for } \mathbf{x} \in \partial\Omega. \tag{7.33}$$

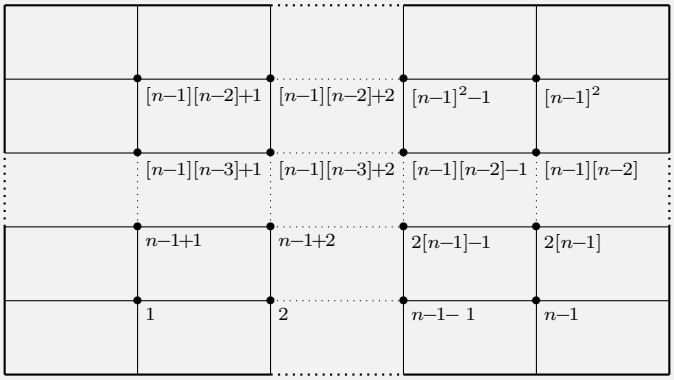
As in the case or two-point boundary value problems, we can reduce our considerations to homogeneous boundary conditions $u_B \equiv 0$ by changing the right hand side in the Poisson equation.

7.4.3 Definition: A **Cartesian grid** on a Cartesian square (cube) consists of the intersection points of lines (planes) parallel to the coordinate axes (planes).

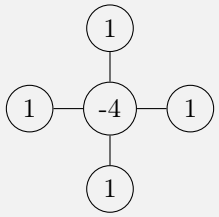


The grid is called uniform, if all lines (planes) are at equal distances.

7.4.4 Definition: The vector of discrete values is defined in points which run in x- and y-direction. In order to obtain a single index for a vector in linear algebra, we use **lexicographic numbering**.



7.4.5 Definition: The **5-point stencil** consists of the sum of a 3-point stencil in x- and a 3-point stencil in y-direction. Its graphical representation is



For a generic row of the linear system, where the associated point is not neighboring the boundary, this leads to

$$\frac{1}{h^2}\left[4y_k - y_{k-1} - y_{k+1} - y_{k-(n-1)} - y_{k+(n-1)}\right] = f_k \tag{7.34}$$

If the point k is next to the boundary, the corresponding entry from the matrix must be omitted.

Remark 7.4.6. From now on, we will call the discrete solution u_h in order to avoid confusion with the coordinate direction y.

7.4.7 Lemma: The matrix obtained for the Laplacian on $\Omega = [0,1]^2$ by the 5-point stencil on a uniform Cartesian mesh of mesh spacing $h = 1/n$ with lexicographic numbering has the structure

$$L_h = \begin{bmatrix} D & -I & & & \\ -I & D & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & D & -I \\ & & & -I & D \end{bmatrix}, \qquad D = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}$$

7.4.8 Theorem: The matrix generated by the 5-point stencil is an M-matrix and the discrete problem

$$L_h u_h = f$$

is stable in the sense that there is a constant c independent of the grid spacing h such that

$$\|L_h^{-1}\| \leq c.$$

Proof. The proof of M-matrix property is identical to the proof for 2-point boundary value problems, which was omitted there. The same way as there, the function

$$w(x, y) = (x - a)(b - x)(y - a)(b - y)$$

can be employed to show boundedness of $\|L_h^{-1}\|$. □

7.4.9 Lemma: The 5-point stencil is consistent of second order.

We summarize:

7.4.10 Theorem: The finite difference methods constructed so far for the Poisson equation is convergent of second order.

Proof. We apply the analysis of Lemma 7.1.8 in x- and y-directions separately, obtaining

$$\left|\frac{\partial^2}{\partial x^2}u(x, y) - \frac{u(x + h, y) - 2u(x, y) + u(x - h, y)}{h^2}\right| \leq ch^2$$
$$\left|\frac{\partial^2}{\partial y^2}u(x, y) - \frac{u(x, y + h) - 2u(x, y) + u(x, y - h)}{h^2}\right| \leq ch^2,$$

an conclude consistency of the sum. □

7.4.11 Theorem: Let y be the solution to the finite difference method for the Laplace equation with the 5-point stencil. Then, the maximum principle holds for y, namely, if there is a point $(\mathbf{x}_k)$ such that $y_k \geq y_j$ for all $j \neq k$ and $y_k \leq y_B$ for any boundary value, then y is constant.

Proof. From equation (7.34), it is clear that a discrete mean value property holds, that is, y_k is the mean value of its four neighbors. Therefore, if $y_k \geq y_j$, for all neighboring indices j of k, we have $y_j = y_k$. We conclude by following a path through the grid points. □

7.5 Evolution equations

After an excursion to second order differential equations depending on a spatial variable, we are now returning to problems depending on time. But this time, on time *and* space. As for the nomenclature, we have encountered ordinary differential equations as equations or systems depending on a time variable only, then partial differential equations (PDE) with several, typically a spatial independent variables. While the problems considered here are covered by the definition of PDE, time and space are fundamentally different as long as we stay away from black holes, such that a distinction is reasonable. Therefore, we introduce the concept of

7.5.1 Definition: An equation of the form

$$\partial_t u(t,x) + F(t,u(t,x)) = 0, \tag{7.35}$$

where $u(t,.)$ is in a function space V on a domain Ω, and F is a differential operator with respect to the spatial variables x only, is an **evolution equation** of first order (in time).
An **initial boundary value problem** (**IBVP**) for this evolution equation completes the differential equation by conditions

$$u(0,x) = u_0 \qquad x \in \Omega \tag{7.36}$$

$$u(t,x) = g \qquad x \in \partial\Omega,\ t > 0. \tag{7.37}$$

7.6 Fundamental solutions

7.6.1 Definition:

$$\Phi(x) = \begin{cases} -\frac{1}{2\pi}\log|x| & d = 2 \\ \frac{1}{d(d-2)\omega_d}\frac{1}{|x|^{d-2}} & d \geq 3, \end{cases} \tag{7.38}$$

for $x \in \mathbb{R}^d$ is the **fundamental solution** to the Laplace equation. Here, ω_d is the volume of the unit ball in $\mathbb{R}^d$.

Appendix A

Appendix

A.1 Properties of matrices

A.1.1 The matrix exponential

Definition A.1.1. The matrix exponential e^A of a matrix $A \in \mathbb{R}^{d\times d}$ is defined by its power series

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}. \tag{A.1}$$

Lemma A.1.2. *The power series* (A.1) *converges for each matrix A.*

Proof. □

Lemma A.1.3 (Properties of the matrix exponential function)**.** *The following relations hold true:*

$$e^0 = \mathbb{I} \tag{A.2}$$

$$e^{\alpha A}e^{\beta A} = e^{(\alpha+\beta)A}, \qquad \forall A \in \mathbb{R}^{d\times d}\ \forall \alpha, \beta \in \mathbb{R}, \tag{A.3}$$

$$e^A e^{-A} = \mathbb{I} \qquad \forall A \in \mathbb{R}^{d\times d}. \tag{A.4}$$

Moreover, e^A is invertible for arbitrary quadratic matrices A.

A.2 The Banach fixed-point theorem

fixed-point theorem.tex fixed-point theorem.tex

A.2.1 Theorem: Let $\Omega \subset \mathbb{R}$ be a closed set and $f\colon \Omega \to \Omega$ a contraction, i.e. there holds $|f(x) - f(y)| \leq \gamma|x - y|$ for a $\gamma \in (0, 1)$.
Then there exists a unique $x^* \in \Omega$ such that $f(x^*) = x^*$.

fixed-point theorem.tex

Proof. Let $x_0 \in \Omega$ and define $f(x_k) = x_{k+1}$. First, we prove existence unsing the cauchy-criterion. Let $k, n \in \mathbb{N}_0$ and consider

$$|x_k - x_{k+m}| = |f(x_{k-1}) - f(x_{k+m-1})| \leq \gamma|x_{k-1} - x_{k+m-1})|.$$

Iteratively, we get

$$|x_k - x_{k+m}| \leq \gamma^k|x_0 - x_m|.$$

We now write $x_0 - x_m = x_0 - x_1 + x_1 - x_2 + \cdots + x_{m-1} - x_m$. The triangle-inequality yields the estimate

$$\begin{aligned}\gamma^k|x_0 - x_m| &\leq \gamma^k|x_0 - x_1| + |x_1 - x_2| + \cdots + |x_{m-1} - x_m| \\ &\leq \gamma^k|x_0 - x_1|(1 + \gamma + \gamma^2 + \cdots + \gamma^m) \\ &\leq \frac{\gamma^k}{1-\gamma}|x_0 - x_1|.\end{aligned}$$

As k gets larger the estimate goes to zero.

Concerning uniqueness, let x^* and y^* be fixpoints.

$$|x^* - y^*| = |f(x*) - f(y^*)| \leq \gamma|x^* - y^*|$$

Since $\gamma \in (0, 1)$ we immediately obtain $|x^* - y^*| = 0$. Using that $|a| = 0$ if and only if $a = 0$ yields $y^* = x^*$. This concludes the proof. □

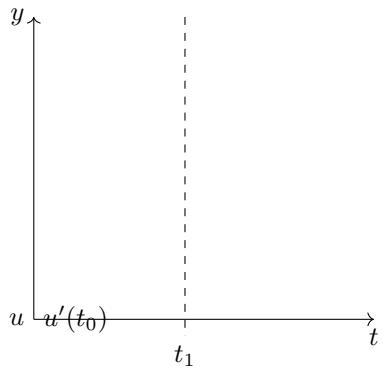
A.3 The implicit and explicit Euler-method

The explicit resp. implicit Euler is given by the one-step method

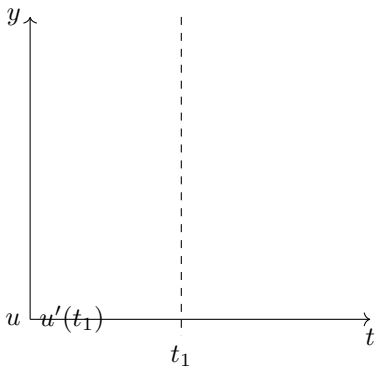
$$y_1 = y_0 + hf(y_0) \qquad \text{resp.} \qquad y_1 = y_0 + hf(y_1)$$

Clearly, the explicit Euler is rather easy to compute since all one needs are f, h and y_0. On the other hand, the implicit Euler is more difficult since for calculating y_1 we need the value of f at y_1.

Consider the following visualizations.



For the explicit Euler we take u_0 and u_0'. y_1, our approximated solution for u_1, is chosen as the intersection point of t_1 and $g(t) = y_0 + t \cdot u'(t_0)$.



For implicit Euler we go backwards. On the t_1-axis we are looking for an the affine function g that fulfills $g(0) = u_0$ and $g'(t_1) = f(t_1)$. Then we set $y_1 = g(t_1)$.

Bibliography

[But96] J. C. Butcher. "A history of Runge-Kutta methods." In: *Appl. Numer. Math.* 20.3 (1996), pp. 247–260. DOI: 10.1016/0168-9274(95)00108-5.

[DB08] P. Deuflhard and F. Bornemann. *Numerische Mathematik 2. Gewöhnliche Differentialgleichungen.* 3. Auflage. de Gruyter, 2008. ISBN: 978-3-11-020356-1.

[Heu86] H. Heuser. *Lehrbuch der Analysis. Teil 2.* 3. Auflage. Teubner, 1986.

[HNW93] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving ordinary differential equations I. Nonstiff problems.* Second edition. Vol. 8. Springer Series in Computational Mathematics. Berlin: Springer, 1993, pp. xvi+528. ISBN: 3-540-56670-8.

[HW10] E. Hairer and G. Wanner. *Solving ordinary differential equations II. Stiff and differential-algebraic problems.* Second edition. Vol. 14. Springer Series in Computational Mathematics. Berlin: Springer-Verlag, 2010, pp. xvi+614. ISBN: 978-3-642-05220-0.

[Ran17] R. Rannacher. *Numerik 1: Numerik gewöhnlicher Differentialgleichungen.* DOI: 10.17885/heiup.258.342. Heidelberg University Publishing, 2017.

[Run95] C. Runge. "Über die numerische Auflösung von Differentialgleichungen." In: *Math. Ann.* 46 (1895), pp. 167–178.

Index