UNITEXT 145

Taketomo Mitsui · Guang-Da Hu

Numerical Analysis of Ordinary and Delay Differential Equations

Springer

UNITEXT

La Matematica per il 3+2

Volume 145

The **UNITEXT - La Matematica per il 3+2** series is designed for undergraduate and graduate academic courses, and also includes books addressed to PhD students in mathematics, presented at a sufficiently general and advanced level so that the student or scholar interested in a more specific theme would get the necessary background to explore it.

Originally released in Italian, the series now publishes textbooks in English addressed to students in mathematics worldwide.

Some of the most successful books in the series have evolved through several editions, adapting to the evolution of teaching curricula.

Submissions must include at least 3 sample chapters, a table of contents, and a preface outlining the aims and scope of the book, how the book fits in with the current literature, and which courses the book is suitable for.

For any further information, please contact the Editor at Springer: francesca.bonadei@springer.com

THE SERIES IS INDEXED IN SCOPUS

***

UNITEXT is glad to announce a new series of free webinars and interviews handled by the Board members, who rotate in order to interview top experts in their field.

Access this link to subscribe to the events: https://cassyni.com/events/TPQ2UgkCbJvvz5QbkcWXo3

Taketomo Mitsui · Guang-Da Hu

Numerical Analysis of Ordinary and Delay Differential Equations

Springer

Taketomo Mitsui
Nagoya University
Nagoya, Aichi, Japan

Guang-Da Hu
Department of Mathematics
Shanghai University
Shanghai, China

ISSN 2038-5714 ISSN 2532-3318 (electronic)
UNITEXT
ISSN 2038-5722 ISSN 2038-5757 (electronic)
La Matematica per il 3+2
ISBN 978-981-19-9262-9 ISBN 978-981-19-9263-6 (eBook)
https://doi.org/10.1007/978-981-19-9263-6

Mathematics Subject Classification: 65Lxx, 65Qxx

Preface

The aim of the book is to provide a concise textbook for students on an advanced undergraduate or a first-year graduate course from various disciplines, such as applied mathematics, control and engineering, who want to learn a modern standard of numerical methods of ordinary and delay differential equations. Their stability analysis is also explained. Experts in applied mathematics, control and engineering fields can also learn the recent developments in numerical analysis of such differential systems. Major algorithms of numerical solution are clearly described.

Ordinary differential equations (ODEs) have a long history in mathematics and provide a valuable resource for its development. At the same time, they provide a strong mathematical tool to express phenomena of a wide variety in science and engineering. Therefore, graduate students, scientists and engineers are required to have knowledge and experience of ODEs. The idea of mathematical formulation to give a functional relationship of an unknown function and its derivative can be extended furthermore. One powerful way to extend ODEs is to treat equations of functions of several independent variables, called partial differential equations (PDEs). Another way is to incorporate unknown functions with delayed argument, called delay differential equations (DDEs). Mathematical analysis has developed to elucidate properties of differential equations, e.g. existence of solutions of the equations. Hence, in some cases we can obtain a solution of a differential equation by algebraic combinations of known mathematical functions. In many practical cases, however, such a solution method is quite difficult and numerical approximations are called for. Modern computers accelerate the situation and, moreover, launch more possibilities of numerical means.

Henceforth, the knowledge and expertise of numerical solution of differential equations become a requirement in the broad area of science and engineering. One might think that a well-organised software package such as MATLAB can provide the solution. In a sense it is true, but one must be aware that blind employment of a software package does not help the users. An understanding of numerical solution of differential equations is still necessary. The present book is intended to give the principles of numerical solution of ordinary differential equations as well as of delay differential equations. To obtain a broader perspective of numerical analysis in

applied mathematics, a short introduction to polynomial interpolation is also given. In particular, we take note that there are a few concise textbooks of delay differential equations and have tried to give descriptions that are as transparent as possible.

The prerequisite of the book is knowledge of calculus and linear algebra at college level. Each chapter is followed by remarks on further development of the chapter topic and exercises. We hope the readers find the topic interesting and try to obtain further developments by themselves.

Nagoya, Japan — Taketomo Mitsui
Shanghai, China — Guang-Da Hu

Contents

Chapter 1
Introduction

This chapter briefly introduces the concept of an ordinary differential equation through a few examples of it. You will see how an equation is formulated and what its (analytical) solution means. Also the necessity of a numerical solution is explained with examples. The delay differential equation, that is, a differential equation including a delay term with respect to the independent variable, is given through an example, too.

1.1 Mathematical Modelling by Differential Equations

When a function x of independent variable t is subject to another functional relation $F(t, x, x') = 0$ together with its first derivative x', it is said that we are given the differential equation of x. Of course this is a very broad and loose definition of a **differential equation** and later a more strict definition will be introduced. We emphasize, however, that many phenomena can be modelled by a differential equation and its mathematical treatment is significant in real-life applications. Typical origins can be found in physical phenomena. We will explain it briefly.

The first example is classical mechanics, that is, the simple pendulum which consists of a small weight of mass m suspended from a fixed point O by a light (that is, with a negligible mass) rod of length a, and only swings in a vertical plane. We take $\theta(t)$ as the inclination of the rod to the downward vertical with the time t (see Fig. 1.1).

Note that the first derivative $\mathrm{d}\theta/\mathrm{d}t$ represents its angular velocity, while the second derivative $\mathrm{d}^2\theta/\mathrm{d}t^2$ is the angular acceleration. Newton's law of motion states

The original version of this chapter was revised: Figure 1.3 has been corrected. The correction to this chapter is available at https://doi.org/10.1007/978-981-19-9263-6_8

T. Mitsui and G. Hu, *Numerical Analysis of Ordinary and Delay Differential Equations*, UNITEXT 145, https://doi.org/10.1007/978-981-19-9263-6_1

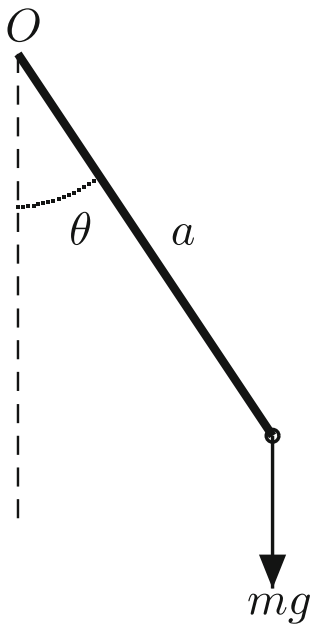


Fig. 1.1 Simple pendulum

acceleration of moving body = external force for the body.

Hence, when no friction exists, the law gives

$$ma^2 \frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} = -mga \sin\theta,$$

where g stands for the gravitational constant. Introduction of $\varphi(t)$ in place of $\mathrm{d}\theta/\mathrm{d}t$ leads to

$$\frac{\mathrm{d}\varphi}{\mathrm{d}t} + \frac{g}{a}\sin\theta = 0, \qquad \frac{\mathrm{d}\theta}{\mathrm{d}t} - \varphi = 0, \tag{1.1}$$

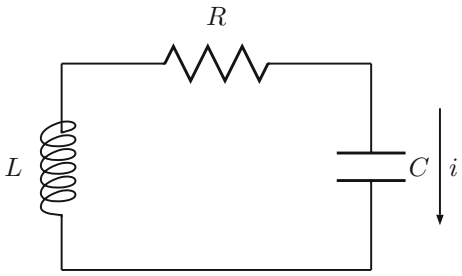
which is a two-dimensional system of ordinary differential equations for $(\theta(t), \varphi(t))$. From the physical point of view, it is interesting to observe that the equation does not depend on the mass m. We impose the condition that at the start $(t = 0)$ $(\theta(0), \varphi(0))$ is equal to $(\theta_0, 0)$ and will analyse the pendulum motion after the start when it is put at the angle θ_0 and gently released. This is called the **initial-value problem** of (1.1).

The second example is the closed LCR circuit in electronics. Suppose that the capacitor with the capacitance C, inductor with the inductance L and resistor with the resistance R are connected in series (see Fig. 1.2).

By the Kirchhoff law the electronic current i in the circuit satisfies

$$L\frac{\mathrm{d}i}{\mathrm{d}t} + Ri + \frac{1}{C}\int i\,\mathrm{d}t = 0.$$

Fig. 1.2 LCR circuit



When we introduce $x = i$, $R/L = 2\alpha$ and $1/(LC) = \omega^2$ ($\omega > 0$), it reduces to

$$\frac{\mathrm{d}^2x}{\mathrm{d}\,t^2} + 2\alpha\frac{\mathrm{d}x}{\mathrm{d}t} + \omega^2 x = 0, \tag{1.2}$$

which can be written in the system form as

$$\frac{\mathrm{d}x}{\mathrm{d}t} - y = 0, \quad \frac{\mathrm{d}y}{\mathrm{d}t} + 2\alpha y + \omega^2 x = 0. \tag{1.2′}$$

Like these, physical laws often have differential equations as their mathematical expression and a similar approach is employed in other disciplines of science and engineering to model phenomena by differential equations. Mathematical problems for differential equation are

- how we can solve it,
- what is the condition which guarantees existence of its solution,
- what is the condition for deriving a unique solution,
- what is the behaviour of the solution when the independent variable t is becoming large

and so on. Each item requires theoretical analysis, but the present volume will be focused on numerical solutions of differential equations. Readers interested in mathematical theory together with mathematical modelling by differential equations can consult [**?**].

1.2 Analytical Versus Numerical Solutions

Prior to detailed descriptions of numerical solutions, we will present background material about the solution of differential equations.

Elementary Solutions
When the angle θ in (1.1) is small in magnitude, we can assume its approximation is given by

$$\frac{\mathrm{d}y}{\mathrm{d}t} + \frac{g}{a}x = 0, \qquad \frac{\mathrm{d}x}{\mathrm{d}t} - y = 0,$$

which leads to

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \frac{g}{a}x = 0 \tag{1.3}$$

with the initial condition $x(0) = \theta_0$ and $\dfrac{\mathrm{d}x}{\mathrm{d}t}(0) = 0$. What is its solution? It is easy to check that $\sin(\omega t)$ and $\cos(\omega t)$ satisfies (1.3) with $\omega = \sqrt{(g/a)}$. The initial condition restricts, however, the solution to $x(t) = \theta_0 \cos(\omega t)$. How about in the case (1.2) (but we restrict ourselves for $\omega > \alpha$)? Due to the similarity between (1.3) and (1.2), we may be inspired to think of a solution in the form of $x(t) = e^{-\alpha t}\cos(\Omega t)$ and to try to substitute it into the equation. A manipulation shows that if $\Omega = \sqrt{\omega^2 - \alpha^2}$, it is a solution. A natural question occurs: Are there any other solutions? The following theorem gives the answer.

First, we introduce:

Definition 1.1 Let $\mathbb{R}^d$ denote the d-dimensional Euclidean space. When we are given the unknown function $\boldsymbol{x} : [t_0, t_F] \longmapsto \mathbb{R}^d$, the known function $\boldsymbol{f} : [t_0, t_F] \times \mathbb{R}^d \longmapsto \mathbb{R}^d$ and the known constant $\boldsymbol{\xi} \in \mathbb{R}^d$, and they satisfy

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{f}(t, \boldsymbol{x}) \ \ (t_0 < t < t_F), \qquad \boldsymbol{x}(t_0) = \boldsymbol{\xi}, \tag{1.4}$$

this is called the initial-value problem of the differential equation with respect to $\boldsymbol{x}$.

Theorem 1.1 *Suppose that the initial-value problem (1.4) is given. We equip $\mathbb{R}^d$ with a certain norm $\|\cdot\|$. Let the function $\boldsymbol{f}$ be continuous on the domain $\mathscr{D} = \{(t, \boldsymbol{x});\ |t - t_0| \le \rho,\ \|\boldsymbol{x} - \boldsymbol{\xi}\| \le R\}$ and satisfy $\|\boldsymbol{f}(t, \boldsymbol{x})\| \le M$. We also assume the following condition (C).*

(C) The function $\boldsymbol{f}(t, \boldsymbol{x})$ satisfies the Lipschitz condition with respect to $\boldsymbol{x}$. That is, there exists a non-negative constant L fulfilling the inequality

$$\|\boldsymbol{f}(t, \boldsymbol{x}) - \boldsymbol{f}(t, \boldsymbol{y})\| \le L\|\boldsymbol{x} - \boldsymbol{y}\| \tag{1.5}$$

on the domain $\mathscr{D}$.

Then the problem (1.4) has a unique solution on the interval $t_0 \le t \le \min(t_0 + \rho, t_0 + R/M)$.

Remark 1.1 We did not give a definition of the norm of $\mathbb{R}^d$ above. Indeed it can be equipped with different kind of norm, whose mathematically rigorous definition can be seen in a standard textbook of linear algebra. However, we stress that every norm of $\mathbb{R}^d$ is topologically equivalent (see Exercise 1.7) and often employ the so-called

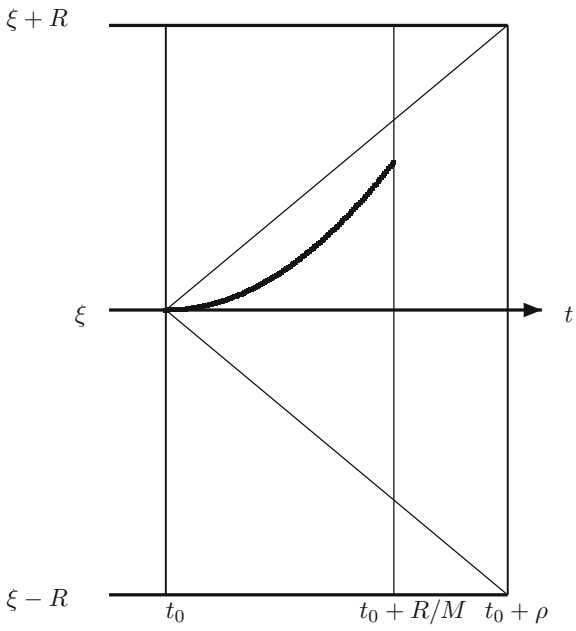


Fig. 1.3 Sketch of Theorem 1.1

2-norm given by $\|\boldsymbol{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_d^2}$ for $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^\mathsf{T}$. This is a special case of the general p-norm defined by $\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{d} |x_i|^p \right)^{1/p}$ $(p \geq 1)$. We also stress that in numerical computations the difference of norms can be meaningful.

We postpone a proof of Theorem 1.1 to Chap. **??** and here try to understand its implications. The theorem is called a **local** unique existence theorem, because it only guarantees the existence at most $t_0 + \rho$ or $t_0 + R/M$, which depends on the initial condition (see Fig. 1.3). However, in the case of (1.3), since the Lipschitz constant L is kept the same beyond the bound, we can shift the same assertion for t greater than the bound and obtain a global existence of the unique solution. Similar things can be said for (1.2). Note that in both cases a constant multiple of each 'solution' satisfies the differential equation again, for the right-hand side of each equation is zero (the case is called homogeneous). The situation becomes different in nonlinear function $\boldsymbol{f}$ with respect to $\boldsymbol{x}$.

Remark 1.2 The formulation $F(t, x, x') = 0$, which is given at the beginning of the chapter, is slightly broader than that of Definition 1.1, for it includes a case that cannot be transformed into (1.4). For example, it includes the differential-algebraic equation in the form $x' = f(t, x)$ and $g(x, x') = 0$. The expression in (1.4) is often called the **normal form** of differential equation and we concentrate ourselves into it.

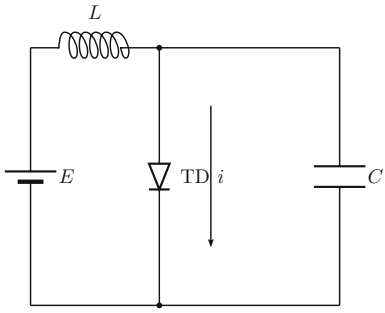


Fig. 1.4 Circuit diagram with a tunnel diode

Nonlinear Case

We will study an electronic circuit similar to Fig. 1.2 but with tunnel-diode TD instead of the resistor and the elements are wired in parallel (see Fig. 1.4).

When the characteristic function $g(v)$ of TD has a cubic nonlinearity with respect to the voltage v biased by E, the circuit equation together with the currency i is given as

$$C\frac{\mathrm{d}v}{\mathrm{d}t} + g(v) + i = 0, \quad L\frac{\mathrm{d}i}{\mathrm{d}t} - v = 0,$$

which reduces to the following equation after normalization:

$$\frac{\mathrm{d}v}{\mathrm{d}t} = \varepsilon v\left(1 - \frac{v^2}{3}\right) - i, \quad \frac{\mathrm{d}i}{\mathrm{d}t} = v$$

or

$$\frac{\mathrm{d}^2v}{\mathrm{d}t^2} + \varepsilon(v^2 - 1)\frac{\mathrm{d}v}{\mathrm{d}t} + v = 0, \tag{1.6}$$

where ε is a positive parameter. This is known as the **van der Pol equation**, which is named after Dutch physicist, Balthasar VAN DER POL.

Thus, an application of Theorem 1.1 is not straightforward for (1.6). Also the elementary method of solution, which is usually given in the undergraduate college class to express the solution by combination of elementary functions such as polynomials, rational, trigonometric, exponential and logarithmic functions, is very hard to handle (1.6) because of its nonlinearity. This is the case for (1.1), too. However, more advanced mathematical analysis can induce the unique existence of the periodic solution of (1.1) and (1.6) [**?**]. This is an interesting result for them, because the equation causes oscillation phenomena without any external forcing. It is called an **autonomous oscillation**.

To get acquainted with the advanced methods, we describe the phase plane analysis by taking Eq. (1.6) as an example. On the xy-plane at the point $(x, y) = (v(t), \mathrm{d}v/\mathrm{d}t\,(t))$ of $v(t)$ satisfying (1.6) with $\varepsilon = 5$ we attach an arrow whose gradient is equal to $\mathrm{d}v/\mathrm{d}t\,(t)$, and repeat the process for many other points on the plane.

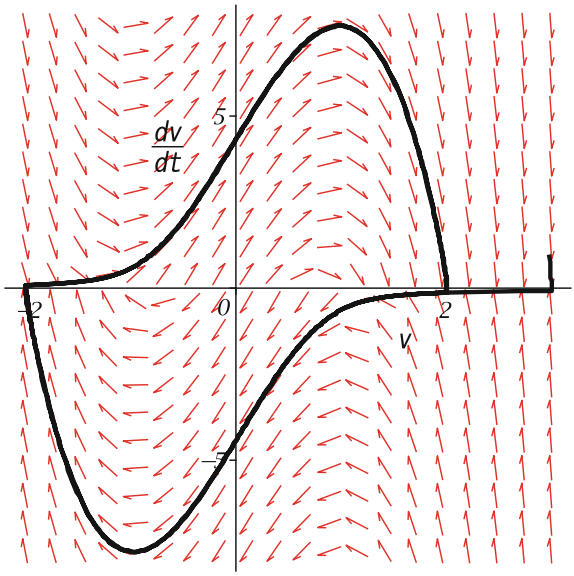


Fig. 1.5 Phase field and phase portrait of van der Pol equation

See Fig. 1.5. The figure consisting of these arrows is called the *gradient field* of the van der Pol equation. Then, taking an initial point, e.g., (3.0, 1.0), on the plane, we obtain a curve, which is depicted as the thick line in the figure, by connecting the arrows one by one. This is called the *phase portrait* of the solution of the van der Pol equation with the initial condition. We can observe that it will wind around a simple closed curve. This suggests the existence of an autonomously oscillating solution of the equation.

The analysis given above is called qualitative for ODEs. On the other hand, we are required to give quantitative information of the solution, too. This can be carried out by numerical solutions which are explained later in the volume.

Furthermore, when the nonlinear element TD has a certain time-delaying effect, the equation becomes

$$\frac{\mathrm{d}v}{\mathrm{d}t}(t) = \varepsilon v(t-\tau)\left(1 - \frac{v^2(t-\tau)}{3}\right) - i(t), \quad \frac{\mathrm{d}i}{\mathrm{d}t}(t) = v(t), \tag{1.7}$$

where the positive τ denotes the time delay. Analytical methods are more difficult than (1.6), while numerical solutions increase their role. For demonstration purposes, we will show results by numerical solutions. A numerical solution by the classical Runge–Kutta method is shown in Fig. 1.6[1] for the problem (1.6) with $\varepsilon = 1$, $v(0) = 2$

[1] Reproduced from '*Introduction to Computational Science with Differential Equations*' (in Japanese) by T. MITSUI, T. KOTO and Y. SAITO, Kyoritsu Shuppan, Tokyo, 2004 (ISBN: 4-320-01753-6).

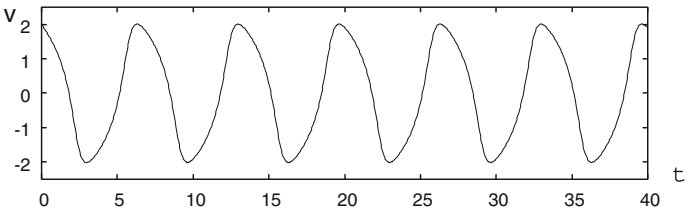


Fig. 1.6 Voltage variation of van der Pol circuit

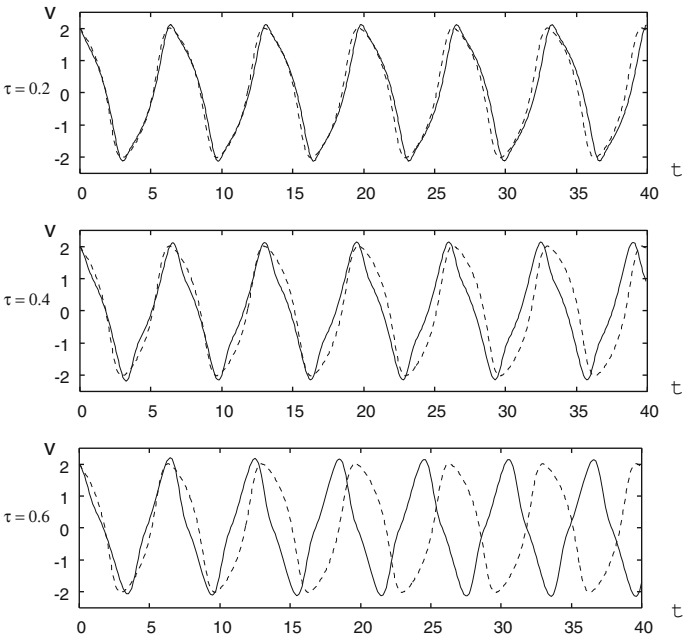


Fig. 1.7 Voltage variation of van der Pol circuit with delay

and $i(0) = 0$. In the delayed case, we calculated numerical solutions of (1.7) with $\varepsilon = 1$ for $\tau = 0.2, 0.4$ and 0.6, which are depicted in Fig. 1.7[1] in this order. The dashed curve shows the solution without delay. We can observe that when τ is becoming large, the solution oscillates with a shorter period.

Differential equations of this sort, which will be called delay differential equations, will be described in later chapters in more detail.

Other Problems for Differential Equations

There are several other problem formulations than the initial-value problem for ordinary differential equations. When we fix the time interval $[a, b]$ on which we are seeking the solution of the equation as

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{f}(t, \boldsymbol{x}) \qquad (a < t < b)$$

and are constrained by totally d conditions for $\boldsymbol{x}(a)$ and $\boldsymbol{x}(b)$, it is called a **boundary-value problem**. For example, assume that the second-order differential equation is given by

$$\frac{\mathrm{d}^2 x}{\mathrm{d}\, t^2} = f\left(t, x, \frac{\mathrm{d}x}{\mathrm{d}t}\right) \tag{1.8}$$

on (a, b) and the boundary condition

$$x(a) = A, \qquad x(b) = B$$

is assigned. Then, it is known that if $f(t, x, y)$ is differentiable with respect to both x and y and f_x and f_y are continuous on the domain

$$D = \{(t, x, y) : \ a \le t \le b, \ -\infty < x < \infty, -\infty < y < \infty\}$$

and furthermore $f_x \ge 0$ and $|f_y|$ is bounded on D, then the above boundary-value problem has a unique solution on $[a, b]$.

The above problem can be converted to seek the missing initial-value ζ. By imposing $\dfrac{\mathrm{d}x}{\mathrm{d}t}(a) = \zeta$ with a certain guess, we solve the initial-value problem of (1.8) and obtain the value $x(b)$. If $x(b) = B$ holds, we are happy to attain success with the exact guess ζ. If not, taking the difference $x(b) - B$ into account we modify the ζ, try again to solve the initial-value problem and repeat it. This is called **shooting**, which means to convert the boundary-value problem into the initial-value problem with a missing initial condition.

Next is the eigenvalue problem, which is explained by the following example. Assume that the boundary-value problem of the second-order differential equation

$$\frac{\mathrm{d}^2 x}{\mathrm{d}\, t^2} + (q(t) + \lambda r(t))\, x = 0 \quad (a < t < b), \quad x(a) = x(b) = 0$$

is given. Here λ is a parameter. It is obvious the trivial solution $x(t) \equiv 0$ satisfies it. However, for a certain non-zero λ it is known the equation has a non-trivial solution. This is called the **Sturm–Liouville-type eigenvalue problem**, which often arises in mathematical physics, and the parameter λ satisfying the problem is called its eigenvalue. We emphasize that the shooting principle is again applicable. Taking a non-trivial initial value for $\dfrac{\mathrm{d}x}{\mathrm{d}t}(a)$ and a guess λ we solve the initial-value problem and check whether the solution satisfies the condition $x(b) = 0$.

Henceforth you can understand that the solution method for the initial-value problem of ordinary differential equations has a big significance and its numerical solution is worth studying. In the following chapters we will explain the methods as well as

the ways of analysing them. Descriptions will be also given for differential equations with delay.

Exercises

1.1. Show that the function $x(t) = \cos(at + b)$ with the constants a and b satisfies the differential equation $\dfrac{\mathrm{d}^4 x}{\mathrm{d}t^4} = a^4 x$. Also confirm that another function $\cosh(at + b)$ satisfies the same equation.

1.2. Solve the initial-value problem

$$t\frac{\mathrm{d}x}{\mathrm{d}t} = x + t + 1, \qquad x(1) = -\frac{1}{2}$$

by determining the solution in the power series form as $\displaystyle\sum_{n=0}^{\infty} c_n (t-1)^n$. Derive a recurrence relation of the coefficients $\{c_n\}$ by substituting the series into the equation. If possible, determine the radius of convergence of the power series solution.

1.3. In the text the solution of (1.2) is given only for the case $\omega > \alpha$. Try to find solutions for other cases, that is, $\omega = \alpha$ and $\omega < \alpha$. Assume the solution to be in the form of $e^{\rho t}$ with undetermined ρ and, by substituting it into the differential equation, derive an algebraic equation for ρ (this is called the characteristic equation). What is the behaviour of the solution when t becomes large?

1.4. A body which is falling vertically in the atmosphere due to the gravitational force must encounter resistance from the air. Under the assumption that the resistance is proportional to the velocity v of the body, derive the differential equation which governs the falling motion of the body by taking v as the unknown function with the time t. Then, solve the equation with the initial condition $v(0) = v_0$.

1.5. For the solution $(\theta(t), \phi(t))$ of (1.1), let us introduce the functional H by

$$H(t) = \frac{1}{2} a\,(\varphi(t))^2 - g\cos(\theta(t)).$$

Confirm that the derivative of H along $(\theta(t), \varphi(t))$ vanishes so that H is invariant along it. In the context of a dynamical system, H is called the Hamiltonian of (1.1) and its invariance implies a closed orbit of $(\theta(t), \varphi(t))$.

1.6. Equation (1.4) is called *autonomous* when the function $\boldsymbol{f}$ does not depend on t. That is, assume that the equation

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x})$$

has a solution $\boldsymbol{x}(t)$. Then, show that $\boldsymbol{x}(t + c)$ is also a solution of the system for arbitrary c.

1.7. Prove the inequalities

$$\|\boldsymbol{x}\|_\infty \leq \|\boldsymbol{x}\|_2 \leq \sqrt{d}\,\|\boldsymbol{x}\|_\infty, \qquad \|\boldsymbol{x}\|_\infty \leq \|\boldsymbol{x}\|_1 \leq d\,\|\boldsymbol{x}\|_\infty$$

for any $\boldsymbol{x} \in \mathbb{R}^d$. Moreover, for arbitrary α and β ($\alpha, \beta \geq 1$) it has been shown that there exist positive constants m and M which satisfy

$$m\|\boldsymbol{x}\|_\alpha \leq \|\boldsymbol{x}\|_\beta \leq M\|\boldsymbol{x}\|_\alpha$$

for any $\boldsymbol{x} \in \mathbb{R}^d$, and this fact is called the norm equivalence of $\mathbb{R}^d$.

Chapter 2
Initial-Value Problems of Differential Equations: Theory

We already introduced the concept and examples of ordinary differential equations in Chap. 1. To deal with numerical solutions of the equation, several basic notions and analyses are required. Here we will briefly study a theory of ordinary differential equations and become acquainted with concepts such as continuous dependency with the initial value and stability of solution of the equation. The concepts will be extended to the case of differential equations with delay in the later chapters.

2.1 Existence and Uniqueness of Solution

As we already mentioned in Sect. 1.2, a theorem about the existence of a solution of the initial-value problem of the ordinary differential equation (1.4) is the start of our analysis, for numerical computations of an equation without solution are meaningless. Here, we try to give a proof of the theorem. You are required to keep Fig. 1.3 in mind.

Proof of Theorem 1.1 Using the constants ρ, R and M, we denote $\min(\rho, R/M)$ by Δ. We will note that the initial-value problem (1.4) has the equivalent integral expression

$$\boldsymbol{x}(t) = \boldsymbol{\xi} + \int_{t_0}^{t} \boldsymbol{f}(\tau, \boldsymbol{x}(\tau))\, \mathrm{d}\tau \tag{2.1}$$

on the interval $[t_0, t_0 + \Delta]$. This means the problem is converted to an integral equation for $\boldsymbol{x}(t)$ and we are to establish existence of its solution. To this end, we define a sequence of functions $\{\boldsymbol{x}_n(t)\}$ $(n = 1, 2, 3, \ldots)$ by the recurrence formula

T. Mitsui and G. Hu, *Numerical Analysis of Ordinary and Delay Differential Equations*, UNITEXT 145, https://doi.org/10.1007/978-981-19-9263-6_2

$$\boldsymbol{x}_n(t) = \boldsymbol{\xi} + \int_{t_0}^{t} \boldsymbol{f}(\tau, \boldsymbol{x}_{n-1}(\tau))\,\mathrm{d}\tau \quad \text{for } t \in [t_0, t_0 + \Delta\,] \quad (n = 1, 2, 3, \ldots) \tag{2.2}$$

with $\boldsymbol{x}_0(t) \equiv \boldsymbol{\xi}$. The sequence, which is often called the Picard sequence, is expected to converge to the function $\boldsymbol{x}(t)$ which satisfies (2.1) [and therefore (1.4)]. Since we assumed that the function $\boldsymbol{f}$ satisfies the condition (C) solely in the domain $\mathscr{D}$, we must show that $\|\boldsymbol{x}_n(t) - \boldsymbol{\xi}\| \le R$ for every $t \in [t_0, t_0 + \Delta\,]$. For $n = 1$, we can derive

$$\|\boldsymbol{x}_1(t) - \boldsymbol{x}_0(t)\| \le \int_{t_0}^{t} \|\boldsymbol{f}(\tau, \boldsymbol{\xi})\|\,\mathrm{d}\tau \le M\Delta \le R$$

for all $t \in [t_0, t_0 + \Delta\,]$. Thus, we assume that $\boldsymbol{x}_{n-1}(t)$ lies in $\mathscr{D}$. Then, on $[t_0, t_0 + \Delta\,]$ the estimation

$$\|\boldsymbol{x}_n(t) - \boldsymbol{\xi}\| \le \int_{t_0}^{t} \|\boldsymbol{f}(\tau, \boldsymbol{x}_{n-1}(\tau))\|\,\mathrm{d}\tau \le M\Delta \le R$$

holds again and by induction we can say that the whole sequence $\{\boldsymbol{x}_n(t)\}$ is in the domain $\mathscr{D}$.

Since the identity

$$\boldsymbol{x}_n(t) = \boldsymbol{\xi} + (\boldsymbol{x}_1(t) - \boldsymbol{x}_0(t)) + (\boldsymbol{x}_2(t) - \boldsymbol{x}_1(t)) + \cdots + (\boldsymbol{x}_n(t) - \boldsymbol{x}_{n-1}(t))$$

is obvious, the convergency of the function series $\displaystyle\sum_{k=1}^{\infty} (\boldsymbol{x}_k(t) - \boldsymbol{x}_{k-1}(t))$ will imply the existence of $\lim_{n\to\infty} \boldsymbol{x}_n(t)$. To obtain it, we introduce the difference $X_k(t) = \|\boldsymbol{x}_k(t) - \boldsymbol{x}_{k-1}(t)\|$ and estimate it as follows:

$$\begin{aligned} X_{k+1}(t) &= \left\| \int_{t_0}^{t} (\boldsymbol{f}(\tau, \boldsymbol{x}_k(\tau)) - \boldsymbol{f}(\tau, \boldsymbol{x}_{k-1}(\tau)))\,\mathrm{d}\tau \right\| \\ &\le L \int_{t_0}^{t} \|\boldsymbol{x}_k(\tau) - \boldsymbol{x}_{k-1}(\tau)\|\,\mathrm{d}\tau = L \int_{t_0}^{t} X_k(\tau)\,\mathrm{d}\tau \\ &\qquad \text{for} \quad k = 1, 2, \ldots \end{aligned}$$

Since the estimation

$$X_1(t) = \left\| \int_{t_0}^{t} \boldsymbol{f}(\tau, \boldsymbol{\xi})\,\mathrm{d}\tau \right\| \le M(t - t_0)$$

holds, we are to show the estimate

$$X_k(t) \le ML^{k-1} \frac{(t - t_0)^k}{k!}$$

for all k. Assuming this is true for up to k, we can calculate

$$X_{k+1}(t) \le L \int_{t_0}^{t} X_k(\tau)\,\mathrm{d}\tau \le L \int_{t_0}^{t} ML^{k-1}\frac{(\tau - t_0)^k}{k!}\mathrm{d}\tau = ML^k \frac{(t-t_0)^{k+1}}{(k+1)!}.$$

Our target is attained by induction. Furthermore, the assumption $t - t_0 \le \Delta$ derives

$$X_{k+1}(t) = \|\boldsymbol{x}_{k+1}(t) - \boldsymbol{x}_k(t)\| \le \frac{M}{L}\frac{(L\Delta)^{k+1}}{(k+1)!}.$$

Since the infinite series $\displaystyle\sum_{k=0}^{\infty}(L\Delta)^{k+1}/(k+1)!$ converges, the infinite sum $\displaystyle\sum_{k=0}^{\infty}(\boldsymbol{x}_{k+1}(t) - \boldsymbol{x}_k(t))$ converges **uniformly** on $[t_0, t_0 + \Delta]$. Therefore the Picard sequence $\{\boldsymbol{x}_n(t)\}$ has a limit $\boldsymbol{x}(t)$ which satisfies the integral equation (2.1). Furthermore, since a uniform limit of continuous functions is also continuous, $\boldsymbol{x}(t)$ satisfies the given differential equation and it fulfills the initial condition, too.

Next, we will show the uniqueness of the solution under the hypothesis of Theorem 1.1. The integral expression (2.1) serves the goal again. Assume that there are two solutions $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ (of course they are continuous on $[t_0, t_0 + \Delta]$). Namely,

$$\boldsymbol{x}(t) = \boldsymbol{\xi} + \int_{t_0}^{t} \boldsymbol{f}(\tau, \boldsymbol{x}(\tau))\,\mathrm{d}\tau \quad \text{and} \quad \boldsymbol{y}(t) = \boldsymbol{\xi} + \int_{t_0}^{t} \boldsymbol{f}(\tau, \boldsymbol{y}(\tau))\,\mathrm{d}\tau.$$

On the interval $[t_0, t_0 + \delta]$ (note δ, not Δ) we have

$$\begin{aligned}\|\boldsymbol{x}(t) - \boldsymbol{y}(t)\| &= \left\| \int_{t_0}^{t} (\boldsymbol{f}(\tau, \boldsymbol{x}(\tau)) - \boldsymbol{f}(\tau, \boldsymbol{y}(\tau)))\,\mathrm{d}\tau \right\| \\ &\le L \int_{t_0}^{t} \|\boldsymbol{x}(\tau) - \boldsymbol{y}(\tau)\|\,\mathrm{d}\tau \le L\delta \max_{t\in[t_0,t_0+\delta]} \|\boldsymbol{x}(t) - \boldsymbol{y}(t)\|.\end{aligned}$$

This means the inequality

$$\max_{t\in[t_0,t_0+\delta]} \|\boldsymbol{x}(t) - \boldsymbol{y}(t)\| \le L\delta \max_{t\in[t_0,t_0+\delta]} \|\boldsymbol{x}(t) - \boldsymbol{y}(t)\|$$

holds. However, a positive δ can be sufficiently small so that $L\delta$ becomes less than unity. Then the above inequality can be valid only for the case $\max_{t\in[t_0,t_0+\delta]} \|\boldsymbol{x}(t) - \boldsymbol{y}(t)\| = 0$. Thus, two solutions coincide on $[t_0, t_0 + \delta]$. The same assertion is repeated on $[t_0 + \delta, t_0 + 2\delta]$, $[t_0 + 2\delta, t_0 + 3\delta]$ and so on, and we can confirm the identity $\boldsymbol{x}(t) = \boldsymbol{y}(t)$ holds on $[t_0, t_0 + \Delta]$. □

We emphasize again that the condition (C) is a sufficient condition of unique existence of (local) solution of the initial-value problem (1.4). What happens if the condition breaks down?

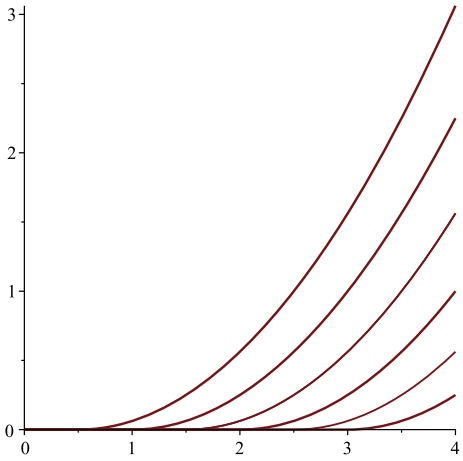


Fig. 2.1 Solutions of Example 2.1

Example 2.1 We think about the initial-value problem

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sqrt{x} \quad (t > t_0) \quad \text{with} \quad x(t_0) = 0.$$

It is easy to confirm that the function $x(t) = (1/4)(t - t_0)^2$ is **a** solution by substituting it into the differential equation. However, $x(t) \equiv 0$ is a solution, too. Moreover, the smooth function defined by

$$x(t) = \begin{cases} 0 & (t_0 \le t \le t_1) \\ (1/4)(t - t_1)^2 & (t > t_1) \end{cases}$$

is a solution again for any $t_1 (> t_0)$. Hence, we can see there are infinitely many solutions of the problem. Figure 2.1 shows several solution curves of the problem in (t, x)-plane. Indeed the right-hand side of the differential equation does not satisfy the condition (C), for the function $1/\sqrt{x}$ diverges when $x \downarrow 0$.

Then you may ask a question: 'Is the condition (C) a necessary condition for the unique existence of a solution?' Again, this is not true. It is known that an initial-value problem which has a unique solution exists with the right-hand side function $\boldsymbol{f}(t, \boldsymbol{x})$ not satisfying (C). Hence, you can understand that the situation is very complicated. Here we omit more detailed discussion and understand that most of the problems which we will handle satisfy the condition.

2.2 Dependence on the Initial Value

When the unique existence of a solution of the initial-value problem (1.4) is established, we will raise another question: What is the dependency of the solution on the initial value $\boldsymbol{\xi}$? That is, what is the variation of the solution $\boldsymbol{x}(t)$ of the same differential equation when the initial value varies as $\boldsymbol{\xi} + \boldsymbol{\delta}$? The first answer is:

Theorem 2.1 *Assume that the initial-value problem (1.4) satisfies the same condition of Theorem 1.1 on the domain $\mathscr{D}$. If the estimation $\|\boldsymbol{\xi} - \boldsymbol{\xi}'\| < \delta$ holds for a small positive δ (that is, $\boldsymbol{\xi}'$ is sufficiently close to $\boldsymbol{\xi}$), there is a positive constant K which satisfies*

$$\|\boldsymbol{x}(t;\boldsymbol{\xi}) - \boldsymbol{x}(t;\boldsymbol{\xi}')\| \le K\|\boldsymbol{\xi} - \boldsymbol{\xi}'\| \quad \forall t \in [t_0, t_0 + \overline{\rho}]$$

for the two solutions $\boldsymbol{x}(t;\boldsymbol{\xi})$ and $\boldsymbol{x}(t;\boldsymbol{\xi}')$ of (1.4). Here the symbols $\boldsymbol{x}(t;\boldsymbol{\xi})$ and $\boldsymbol{x}(t;\boldsymbol{\xi}')$ are introduced to give emphasis to the dependence on the initial value and $\overline{\rho}$ denotes the maximum length of the interval where the two solutions commonly exist.

Proof The integral expression of $\boldsymbol{x}(t;\boldsymbol{\xi})$ and $\boldsymbol{x}(t;\boldsymbol{\xi}')$ leads to

$$\boldsymbol{x}(t;\boldsymbol{\xi}) - \boldsymbol{x}(t;\boldsymbol{\xi}') = \boldsymbol{\xi} - \boldsymbol{\xi}' + \int_{t_0}^{t} \left(\boldsymbol{f}(\tau, \boldsymbol{x}(\tau;\boldsymbol{\xi})) - \boldsymbol{f}(\tau, \boldsymbol{x}(\tau;\boldsymbol{\xi}'))\right) \mathrm{d}\tau.$$

Thus, we have

$$\|\boldsymbol{x}(t;\boldsymbol{\xi}) - \boldsymbol{x}(t;\boldsymbol{\xi}')\| \le \|\boldsymbol{\xi} - \boldsymbol{\xi}'\| + L\int_{t_0}^{t} \|\boldsymbol{x}(\tau;\boldsymbol{\xi}) - \boldsymbol{x}(\tau;\boldsymbol{\xi}')\| \,\mathrm{d}\tau.$$

By applying Grönwall's lemma, whose statement and proof are given below, we obtain

$$\begin{aligned}\|\boldsymbol{x}(t;\boldsymbol{\xi}) - \boldsymbol{x}(t;\boldsymbol{\xi}')\| &\le \|\boldsymbol{\xi} - \boldsymbol{\xi}'\| + L\|\boldsymbol{\xi} - \boldsymbol{\xi}'\| \int_{t_0}^{t} \exp(L(t-\tau))\,\mathrm{d}\tau \\ &= \|\boldsymbol{\xi} - \boldsymbol{\xi}'\| \exp(L(t - t_0)),\end{aligned}$$

which completes the proof. □

The following lemma is instrumental in mathematical analysis.

Lemma 2.1 (TH. GRÖNWALL) *Assume that a nonnegative constant m, a continuous nonnegative function $u(t) : [t_0, T] \longmapsto [0, \infty)$ and a continuous positive function $w(t) : [t_0, T] \longmapsto [0, \infty)$ are given to satisfy*

$$u(t) \le m + \int_{t_0}^{t} w(s)u(s)\,\mathrm{d}s \quad \textit{for} \quad t \in [t_0, T].$$

Then $u(t)$ fulfills the inequality

$$u(t) \le m \exp\left(\int_{t_0}^{t} w(s)\,\mathrm{d}s\right) \quad \textit{for} \quad t \in [t_0, T].$$

Proof Let $U(t)$ be defined by

$$U(t) = m + \int_{t_0}^{t} w(s)u(s)\,\mathrm{d}s$$

for $t \in [t_0, T]$. Note that $u(t) \le U(t)$ holds for $t \in [t_0, T]$. Then, $U(t)$ is positive and differentiable for all $t \in [t_0, T]$, and satisfies

$$U'(t) = w(t)u(t) \le w(t)U(t) \quad (t \in [t_0, T]).$$

A division of the above inequality by $U(t)$ enables us to give

$$\frac{\mathrm{d}}{\mathrm{d}t} \log U(t) = \frac{U'(t)}{U(t)} \le w(t) \quad (t \in [t_0, T]).$$

By integration of both sides, we obtain

$$\log U(t) - \log m = \int_{t_0}^{t} \frac{\mathrm{d}}{\mathrm{d}t} \log U(s)\,\mathrm{d}s \le \int_{t_0}^{t} w(s)\,\mathrm{d}s$$

for all $t \in [t_0, T]$. Therefore, we obtain the inequality

$$\exp\left(\log U(t) - \log m\right) = \frac{U(t)}{m} \le \exp\left(\int_{t_0}^{t} w(s)\,\mathrm{d}s\right),$$

which confirms the conclusion. □

Our next question is how smoothly the solution depends on the initial value.

Theorem 2.2 *Again we assume that the initial-value problem (1.4) satisfies the same condition of Theorem 1.1 on the domain $\mathscr{D}$. Furthermore, we assume that the partial derivatives $\partial \boldsymbol{f}/\partial x_1, \partial \boldsymbol{f}/\partial x_2, \ldots, \partial \boldsymbol{f}/\partial x_d$ exist and are continuous on $\mathscr{D}$. Let its Jacobian matrix $J(t, \boldsymbol{x})$ be denoted by*

$$J(t, \boldsymbol{x}) = \left[\frac{\partial \boldsymbol{f}}{\partial x_1}, \frac{\partial \boldsymbol{f}}{\partial x_2}, \cdots, \frac{\partial \boldsymbol{f}}{\partial x_d}\right].$$

Then, the solution $\boldsymbol{x}(t; \boldsymbol{\xi})$ is continuously differentiable with respect to $\boldsymbol{\xi}$ in $\mathscr{D}$. Furthermore, the vector which consists of the derivatives $\partial \boldsymbol{x}/\partial \xi_k = [\partial x_1/\partial \xi_k, \partial x_2/\partial \xi_k, \cdots, \partial x_d/\partial \xi_k]^{\mathsf{T}}$ is the solution of the following initial-value problem of the linear differential equation

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} = J(t, \boldsymbol{x}(t; \boldsymbol{\xi}))\boldsymbol{y} \quad (t > t_0) \quad \textit{and} \quad \boldsymbol{y}(t_0) = \boldsymbol{e}_k, \tag{2.3}$$

where $\boldsymbol{e}_k$ is the d-dimensional vector whose k-th component is 1 and others are zero.

Proof of the theorem is very complicated and we omit it here. The significant point is that we obtain an expression of the derivative of the solution with respect to the initial value. Equation (2.3) is often referred to as the *first variational equation* of the original initial-value problem. Considering (2.3), we are aware of the importance of the solution of a **linear system** of differential equations.

Theorem 2.3 *Assume that the $d \times d$-dimensional matrix function $A(t)$ is continuous and the d-dimensional function $\boldsymbol{f}(t)$ is bounded on the interval $[t_0, T]$.*

(i) For any initial value $\boldsymbol{\xi}$ the problem

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = A(t)\boldsymbol{x} \quad (t > t_0) \quad \textit{and} \quad \boldsymbol{x}(t_0) = \boldsymbol{\xi} \tag{2.4}$$

which is called homogeneous, has a unique matrix function $\mathcal{R}(t; t_0)$ which gives its solution as

$$\boldsymbol{x}(t) = \mathcal{R}(t; t_0)\boldsymbol{\xi}. \tag{2.5}$$

(ii) For the inhomogeneous problem

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = A(t)\boldsymbol{x} + \boldsymbol{f}(t) \quad (t > t_0) \quad \textit{and} \quad \boldsymbol{x}(t_0) = \boldsymbol{\xi} \tag{2.6}$$

its solution is given by

$$\boldsymbol{x}(t) = \mathcal{R}(t; t_0)\boldsymbol{\xi} + \mathcal{R}(t; t_0) \int_{t_0}^{t} \mathcal{R}(\tau; t_0)^{-1} \boldsymbol{f}(\tau)\, d\tau. \tag{2.7}$$

The matrix $\mathcal{R}(t; t_0)$ is called the **fundamental matrix** of the differential equation in (2.4). The theorem can be derived by observing the following facts.

(i) In the homogeneous case, due to Theorem 1.1 a unique solution exists on $[t_0, T]$. Moreover, the solutions of the differential equation consist of a linear space. That is, when $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ are solutions, their linear combination is a solution, too: $(\mathrm{d}(\alpha\boldsymbol{x}(t) + \beta\boldsymbol{y}(t))/\mathrm{d}t = A(t)(\alpha\boldsymbol{x}(t) + \beta\boldsymbol{y}(t)))$.

(ii) This means we can define a linear map from the initial value to the solution for every $t \in [t_0, T]$ and the map is continuous on the interval. Hence we can introduce the matrix function $\mathcal{R}(t; t_0)$ representing the map on the interval.

(iii) The fundamental matrix $\mathcal{R}(t; t_0)$ consists of d linearly independent solutions $\mathrm{d}\boldsymbol{x}/\mathrm{d}t = A(t)\boldsymbol{x}$ as column vectors. Hence it is differentiable and nonsingular on the interval and satisfies

$$\frac{\mathrm{d}\mathcal{R}(t; t_0)}{\mathrm{d}t} = A(t)\mathcal{R}(t; t_0) \quad \text{and} \quad \mathcal{R}(t_0; t_0) = I_d.$$

(iv) In the inhomogeneous case, differentiation of the right-hand side of (2.7) confirms that $\boldsymbol{x}(t)$ satisfies the differential equation as well as the initial condition.

We can also obtain an estimation of the norm of $\boldsymbol{x}(t)$ in (2.7) as

$$\|\boldsymbol{x}(t)\| \leq \exp(L(t)) \left(\|\boldsymbol{\xi}\| + \int_{t_0}^{t} \exp(-L(\tau)) \|\boldsymbol{f}(\tau)\| \, \mathrm{d}\tau \right), \tag{2.8}$$

where $L(t) = \displaystyle\int_{t_0}^{t} \|A(\tau)\| \, \mathrm{d}\tau$.

When the matrix $A(t)$ does not depend on t and is equal to a constant matrix A, the above results become simpler. We introduce the exponential of A by

$$\exp A = \sum_{n=0}^{\infty} \frac{A^n}{n!}.$$

Then the fundamental matrix of the differential equation $\dfrac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = A\boldsymbol{x}$ can be expressed by $\mathcal{R}(t; t_0) = \exp((t - t_0)A)$ and the solution of the initial-value problem $\dfrac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = A\boldsymbol{x} + \boldsymbol{f}(t)$ and $\boldsymbol{x}(t_0) = \boldsymbol{\xi}$ is

$$\boldsymbol{x}(t) = \exp((t - t_0)A)\boldsymbol{\xi} + \int_{t_0}^{t} \exp((t - \tau)A)\boldsymbol{f}(\tau) \, \mathrm{d}\tau. \tag{2.9}$$

2.3 Stability of Solution

You may be aware that the results of the preceding sections (except linear cases) are about local properties of the solution. That is, we can only obtain properties of the solution limited by the initial value. On the other hand, we are also interested in the solution behaviour of (1.4) when t becomes large. This is called a **global** theory and we will focus on the stability property of the solution.

As in Definition 1.1, on the domain D ($\subset \mathbb{R}^d$) we consider the initial-value problem of a differential equation of general form

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{f}(t, \boldsymbol{x}) \quad (t \geq t_0), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \tag{2.10}$$

where $\boldsymbol{f}(t, \boldsymbol{x}) : [0, \infty) \times D \longmapsto \mathbb{R}^d$ is continuous in t and locally Lipschitz in $\boldsymbol{x}$ on $[0, \infty) \times D$. Assume that D contains the origin $\mathbf{0}$ of $\mathbb{R}^d$ and $\boldsymbol{f}$ satisfies the condition $\boldsymbol{f}(t, \mathbf{0}) = \mathbf{0}$ for every t (≥ 0). Then, $\boldsymbol{x}(t) = \mathbf{0}$ clearly satisfies the differential equation of (2.10). Hence we can say that the origin $\mathbf{0}$ is an *equilibrium point* of the system (2.10) if $\boldsymbol{f}(t, \mathbf{0}) = \mathbf{0}$ for every t (≥ 0). Then, we observe the solution

behaviour of (2.10) starting from the initial value *close* to the origin and give the following definition of stability.

Definition 2.1 The equilibrium point $\boldsymbol{x} = \mathbf{0}$ of the system (2.10) is:

(i) stable if, for each positive ε, there exists a positive $\delta = \delta(\varepsilon, t_0)$ such that $\|\boldsymbol{x}_0\| < \delta$ implies $\|\boldsymbol{x}(t)\| < \varepsilon$ for $t \geq t_0$ and
(ii) asymptotically stable if it is stable and moreover there is a positive constant $\gamma = \gamma(t_0)$ fulfilling the condition $\boldsymbol{x}(t) \to \mathbf{0}$ as $t \to \infty$ for all the initial values $\boldsymbol{x}_0$ with $\|\boldsymbol{x}_0\| < \gamma$.

This is generic, for an equilibrium point at the origin could be a translation of a nonzero equilibrium point or, more generally, a translation of a nonzero solution of the system (2.10).

For example, we examine the stability of the equation given by (1.3), for it is autonomous and therefore it has the origin as equilibrium point. When we select positive δ_1 and δ_2 and solve the equation from $t = 0$ with the initial value $(\delta_1, \delta_2\omega)$, we know its solution

$$y(t) = \delta_1 \cos(\omega t) + \delta_2 \sin(\omega t) \quad \text{and} \quad \frac{\mathrm{d}y}{\mathrm{d}t}(t) = \delta_2\omega \cos(\omega t) - \delta_1\omega \sin(\omega t).$$

Therefore, if we take $\delta_1^2 + \delta_2^2 < \varepsilon^2/(1 + \omega^2)$, the estimation

$$\sqrt{y^2(t) + \left(\frac{\mathrm{d}y}{\mathrm{d}t}(t)\right)^2} < \varepsilon$$

holds for any $t \in (0, \infty)$. This shows the stability of the equilibrium point of the equation in the sense of 2-norm.

However, the example is almost trivial, for we already know the general solution whose behaviour is obvious for $t \to \infty$. The essential problem is to make a criterion for stability without solving the initial-value problem. Mathematical theory for this direction is beyond the scope of the present volume and interested readers are recommended to consult references of qualitative theory of differential equations.

We can obtain more precise result about asymptotic stability for the linear system having a constant matrix.

Theorem 2.4 *The zero solution of the differential equation* $\dfrac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = A\boldsymbol{x}$ *is asymptotically stable if and only if all the eigenvalues of A have negative real part.*

Proof First we prove the case of diagonalizable A. The matrix A has a transformation matrix P, which gives $P^{-1}AP = \mathrm{diag}[\lambda_1, \ldots, \lambda_d]$. Then, we obtain

$$\exp((t-t_0)A) = P \begin{bmatrix} \exp((t-t_0)\lambda_1) & & & 0 \\ & \exp((t-t_0)\lambda_2) & & \\ & & \ddots & \\ 0 & & & \exp((t-t_0)\lambda_d) \end{bmatrix} P^{-1}.$$

Therefore, the estimation $\mathrm{Re}\,\lambda_j < 0$ implies $|\exp((t-t_0)\lambda_j)| \to 0$ $(t \to \infty)$ for all j and the conclusion holds. For non-diagonalizable A, we can employ the Jordan decomposition of A and the statement holds.

Assume that the matrix A has an eigenvalue λ whose real part is non-negative. Let $\boldsymbol{v}$ be the corresponding eigenvector and define $\boldsymbol{x}(t) = \exp(\lambda t)\,\boldsymbol{v}$. Then, we have

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \exp(\lambda t)\,\lambda\,\boldsymbol{v} = \exp(\lambda t)\,A\boldsymbol{v} = A\boldsymbol{x},$$

which asserts $\boldsymbol{x}(t)$ is a solution of the equation. However, because of $|\exp(\lambda t)| \geq 1$ for $t > 0$, the estimation $\|\boldsymbol{x}(t)\| \geq \|\boldsymbol{v}\|$ holds for $t \geq 0$ and $\boldsymbol{x}(t)$ never converges to $\boldsymbol{0}$ as $t \to \infty$. This means the condition is necessary for the asymptotic stability. □

Hereafter we often say that the linear system is asymptotically stable when its unique zero solution is asymptotically stable. You might suppose that in the linear time-varying case

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = A(t)\,\boldsymbol{x} \quad (t \geq t_0), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \tag{2.11}$$

a similar analysis will be possible as in the constant matrix case of Theorem 2.4, but it is not the case. We illustrate it through the following:

Example 2.2 Suppose that the linear time-varying equation (2.11) is given with the matrix

$$A(t) = \begin{bmatrix} -1 + \alpha\cos^2 t & 1 - \alpha\sin t\cos t \\ -1 - \alpha\sin t\cos t & -1 + \alpha\sin^2 t \end{bmatrix},$$

where α is a positive constant. When we fix t, the following eigenvalue-like identity

$$A(t)\boldsymbol{x}_\pm = \lambda_\pm \boldsymbol{x}_\pm$$

holds for certain $\boldsymbol{x}_\pm$ and each t with

$$\lambda_+ = \frac{\alpha - 2 + \sqrt{\alpha^2 - 4}}{2} \quad \text{and} \quad \lambda_- = \frac{\alpha - 2 - \sqrt{\alpha^2 - 4}}{2}.$$

Since the values $\lambda_\pm$ happen not to be t-dependent, we can introduce a matrix function, given as

$$\mathcal{R}(t,0) = \begin{bmatrix} \cos t\ \exp(\alpha-1)t & \sin t\ \exp(-t) \\ -\sin t\ \cos t\ \exp(\alpha-1)t & \cos t\ \exp(-t) \end{bmatrix},$$

which plays a similar role to the fundamental matrix in the constant-coefficient case. In fact, the function $\boldsymbol{x}(t) = \mathcal{R}(t,0)\boldsymbol{x}_0$ satisfies (2.11) for arbitrary t $(\geq t_0)$. Although both of $\lambda_\pm$ have a negative real part for every positive t when $0 < \alpha < 2$, every component of $\mathcal{R}(t,0)\boldsymbol{x}_0$ is unbounded in the case $\alpha > 1$, i.e., the system (2.11) is not stable generally.

Readers interested in the topic can refer to Section I.13 of [15].

Further Remarks

As mentioned in the last paragraph of Sect. 2.1, a necessary and sufficient condition for the unique existence of a solution of the initial-value problem, even though expected, is very difficult to state mathematically. In contrast to this, the Lipschitz condition *(C)* is rather easy to check and widely applied to actual problems. Even if a unique solution exists, its higher-order differentiability with respect to the independent variable t is another problem. Roughly speaking, when the function $\boldsymbol{f}(t,\boldsymbol{x})$ is sufficiently smooth with respect to t and $\boldsymbol{x}$, the solution is also smooth. In the sequel, we will often suppose that the solution can be expanded in the Taylor series even if it exists only locally.

Stability is also a significant issue. Here we introduced an analytical method for stability analysis. Another way to employ Lyapunov function is popular and often applied in engineering science. Furthermore, a topological approach is widely applied for stability analysis.

Interested readers who want to study more can refer to the classical textbooks by Arnold [1], Coddington–Levinson [11] or Robinson [31].

Exercises

2.1. Confirm that the differential equation of Example 2.1 with different initial conditions

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sqrt{x} \quad (t > t_0) \quad \text{and} \quad x(t_0) = 1$$

has a unique solution $x(t) = (t - t_0 + 4)^2/4$ $(t \geq t_0)$. Prove that the problem satisfies the Lipschitz condition around the initial point $(t_0, x(t_0))$.

2.2. Solve the initial-value problem of the scalar equation

$$\frac{\mathrm{d}\,x}{\mathrm{d}\,t} = \frac{\pi(1+x^2)}{2} \quad (t > 0), \qquad x(0) = 0$$

and confirm that the solution diverges when t is approaching 1. Hence the solution cannot exist globally.

2.3. Carry out the method of Picard iteration for the scalar initial value problem

$$\frac{\mathrm{d}x}{\mathrm{d}t} = kx \quad (t > 0), \qquad x(0) = 1,$$

where k is a constant.

2.4. Rewrite the equation (1.3) into a two-dimensional linear system to derive the coefficient matrix A with the parameter $\omega = \sqrt{(g/a)}$. Then, calculate the fundamental solution $\exp(tA)$ and confirm that it gives the general solution of the equation.
Hint. Try to make the powers $(tA)^2, (tA)^3, \ldots$ and to compare the components of $\left(I + tA + (tA)^2/2! + (tA)^3/3! + \cdots\right)[\theta_0, 0]^{\mathsf{T}}$ with the Maclaurin expansion of $\cos(\omega t)$ and $\sin(\omega t)$.

2.5. If the two matrices A and B are of the same size and commutative (i.e., $AB = BA$), prove that the identity $\exp(A + B) = (\exp A)(\exp B)$ holds. Moreover, show an example which gives $\exp(A + B) \neq (\exp A)(\exp B)$.

2.6. Solve the following initial-value problem of a linear system of differential equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 & -2 \\ 3 & -3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \exp(t)\begin{bmatrix} -1 \\ 2 \end{bmatrix}, \qquad \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

by means of the fundamental matrix.

(1) Denote the coefficient matrix by A, determine its two eigenpairs $(\lambda_1, \boldsymbol{p}_1)$ and $(\lambda_2, \boldsymbol{p}_2)$ and derive the diagonalization matrix $P = [\boldsymbol{p}_1, \boldsymbol{p}_2]$.
(2) Determine the diagonalized matrix Λ by calculating $P^{-1}AP$ and its exponential $\exp(t\Lambda)$.
(3) Derive the fundamental matrix by $\exp(tA) = P\exp(t\Lambda)P^{-1}$.
(4) Apply (2.9).

2.7. Show that the solution $(x(t), y(t))$ of a system of differential equations

$$\frac{\mathrm{d}x}{\mathrm{d}t} = y, \quad \frac{\mathrm{d}y}{\mathrm{d}t} = 2x$$

satisfies $2x^2 - y^2 = C$ with a certain constant C. Hence, confirm that no matter how small δ ($\delta > 0$) is taken, the trajectory $(x(t), y(t))$ of the solution starting from $(x(0), y(0)) = (\delta, 0)$ travels far from the origin $(0, 0)$ in the xy-plane. This means an asymptotic instability of the equation.

2.8. When two differentiable functions $x_1(t)$ and $x_2(t)$ are given, the determinant

$$W(x_1, x_2) \stackrel{\text{def}}{=} \det\begin{bmatrix} x_1(t) & x_2(t) \\ x_1'(t) & x_2'(t) \end{bmatrix}$$

is said to be the Wronskian or the Wronski determinant of (x_1, x_2). Prove that the necessary and sufficient condition of the linear independence of (x_1, x_2) is $W(x_1, x_2) \neq 0$. The Wronskian is useful to discriminate the linear independence of two solutions of linear second-order differential equation with variable coefficients

$$\frac{\mathrm{d}^2 x}{\mathrm{d}\, t^2} + p(t)\frac{\mathrm{d}\, x}{\mathrm{d}\, t} + q(t)x = 0,$$

because the Wronskian of its two solutions $x_1(t)$ and $x_2(t)$ is given by

$$W(x_1, x_2) = c\, \exp\left(-\int p(t)\, \mathrm{d}\, t\right).$$

Chapter 3
Runge–Kutta Methods for ODEs

Since an analytically-closed form solution of ordinary differential equations (ODEs) is hardly possible in real-world applications, their numerical solutions possess a definite role in science and engineering. Among numerical solutions, the discrete variable methods (DVMs) are important because of their flexibility and easiness in programming. This chapter starts with Runge–Kutta methods for ODEs, one of the representative class of DVMs and describes underlying concepts around the methods. An embedded Runge–Kutta pair is introduced for an efficient implementation of the method. As the stability of Runge–Kutta method is vital, we will analyse it in depth.

3.1 Runge–Kutta Methods for ODEs

Discrete Variable Method
We are trying numerically to solve the initial-value problem of ODEs given by

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(t, x) \ \ (t_0 < t < t_F), \qquad x(t_0) = x_0. \tag{3.1}$$

Hereafter, different from Chap. 1, we omit the bold-face letters for vectors, but assume the unknown function $x(t)$ to be $[t_0, t_F] \longmapsto \mathbb{R}^d$ and the given function f to be $[t_0, t_F] \times \mathbb{R}^d \longmapsto \mathbb{R}^d$. We again introduce the Lipschitz condition *(C)* for f to ensure unique existence of the analytical solution of (3.1) as shown in Chap. 1.

(C) The function $f(t, x)$ is continuous on $[t_0, t_F] \times \mathbb{R}^d$ and satisfies the global Lipschitz condition with respect to x. That is, there exists a non-negative constant L fulfilling the inequality

$$\|f(t, x) - f(t, y)\| \leq L\|x - y\| \tag{3.2}$$

T. Mitsui and G. Hu, *Numerical Analysis of Ordinary and Delay Differential Equations*,
UNITEXT 145, https://doi.org/10.1007/978-981-19-9263-6_3

for arbitrary $t \in [t_0, t_F]$ and $x, y \in \mathbb{R}^d$.

Then we divide the interval $[t_0, t_F]$ into sub-intervals and the discretized points of the variable t are denoted by

$$t_0 < t_1 < \cdots < t_n < t_{n+1} < \cdots < t_N = t_F.$$

We call t_n the step-point and the length $h_n = t_{n+1} - t_n$ the step-width or step-size. We denote the exact and approximate solution values at the nth step-point by $x(t_n)$ and x_n, respectively. First we assume the equidistant step-points, that is, the step-size does not depend on n and is given by the constant $h = (t_F - t_0)/N$. Note that in actual computations a variable step-size is often employed and its implementation is practically important.

A discrete variable method (DVM) stands for a computational way to obtain the sequence of approximate values $\{x_n\ ;\ n = 1, 2, \ldots\}$ by a certain *constructive* means. The simplest DVM is the **Euler method** given by the recurrence form

$$\boxed{x_{n+1} = x_n + hf(t_n, x_n).} \tag{3.3}$$

The pairs calculated by the Euler method

$$(t_0, x_0), (t_1, x_1), \ldots, (t_n, x_n), (t_{n+1}, x_{n+1}), \ldots$$

consist of polygonal curves in this order of the pairs in the (t, x) space. Thus it is also called the *polygonal method*. As shown in (3.3), the formula which describes the constructive way for calculation of the sequence $\{x_n\}$ by DVM is often called a **scheme**.

Runge–Kutta Methods

The Euler method is important in the theory of DVM and can be practically applied; however, it is not a recommended one from the viewpoint of computational efficiency. Hence two directions have been taken to improve the Euler method. One is to include more past values $\{x_{n-1}, x_{n-2}, \ldots\}$ into the right-hand side calculation of (3.3), while another is to mix up other off-step values of x and f between $[t_n, t_{n+1}]$ in (3.3). The former leads to linear multistep methods, which will be explained in Chap. 5, whereas the latter leads to Runge–Kutta methods.

Let s be a natural number and an array of real numbers be given by

$$\begin{array}{c|cccc} c_2 & a_{21} & & & \\ c_3 & a_{31} & a_{32} & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s\,s-1} \\ \hline & b_1 & b_2 & \cdots & b_{s-1}\ \ b_s \end{array} \quad . \tag{3.4}$$

The scheme to calculate x_{n+1} from x_n by

$$\left\{\begin{array}{l} k_1 = f(t_n, x_n), \\ k_2 = f\left(t_n + c_2\, h, x_n + h a_{21}\, k_1\right), \\ k_3 = f\left(t_n + c_3\, h, x_n + h a_{31}\, k_1 + h a_{32}\, k_2\right), \\ \quad \vdots \\ k_s = f\left(t_n + c_s\, h, x_n + h a_{s1}\, k_1 + \cdots + h a_{s,s-1}\, k_{s-1}\right), \\ x_{n+1} = x_n + h\left(b_1\, k_1 + b_2\, k_2 + \cdots + b_s\, k_s\right) \end{array}\right. \tag{3.5}$$

is called *a* Runge–Kutta method. Here, s is the number of stages of the Runge–Kutta method and means the number of function evaluations of f when proceeding from x_n to x_{n+1}. The parameters a_{ij}, b_i and c_i in the array are coefficients of the Runge–Kutta method.

Equation (3.5) can be expressed in a more compact form with

$$k_i = f\left(t_n + c_i\, h, x_n + h \sum_{j=1}^{i-1} a_{ij}\, k_j\right), \quad x_{n+1} = x_n + h \sum_{i=1}^{s} b_i\, k_i$$

by interpreting $c_1 = 0$. On the other hand, by introducing the intermediate variable X_i, another equivalent expression

$$X_i = x_n + h \sum_{j=1}^{i-1} a_{ij}\, f\left(t_n + c_j\, h, X_j\right), x_{n+1} = x_n + h \sum_{i=1}^{s} b_i\, f\left(t_n + c_i\, h, X_i\right) \tag{3.6}$$

is also possible and conveniently used, for in most cases X_i can be considered as an approximation to $x(t_n + c_i h)$.

The method whose coefficient array is given by

$$\begin{array}{c|cccc} 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

leads to the scheme

$$\boxed{\begin{array}{rcl} k_1 &=& f(t_n, x_n), \\ k_2 &=& f\left(t_n + \dfrac{h}{2},\, x_n + \dfrac{h}{2}\, k_1\right), \\ k_3 &=& f\left(t_n + \dfrac{h}{2},\, x_n + \dfrac{h}{2}\, k_2\right), \\ k_4 &=& f\left(t_n + h,\, x_n + h k_3\right), \\ x_{n+1} &=& x_n + \dfrac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \end{array}}$$

and is best known among the Runge–Kutta methods. Thus it is often referred to as the *classical* Runge–Kutta method, or, more simply '*the*' Runge–Kutta method. It has a distinctive feature that many parameters a_{ij} are zero, which enables us to program the method requiring economised intermediate memories. Its algorithmic description is shown in Algorithm 3.1.

Convergency of Runge–Kutta Methods

By taking the coefficients of the Runge–Kutta (hereafter RK) method freely, we can invent many methods. Crucial is what are the criteria for methods. They are convergency and stability. Here we first discuss the convergency of RK methods and introduce the concept of order of convergence. In the present subsection we assume the IVP (3.1) had a sufficiently smooth solution $x(t)$ on the whole interval $[t_0, t_F]$. Thus, every derivative of $x(t)$ is bounded on the interval. Since an application of RK methods from (t_n, x_n) to (t_{n+1}, x_{n+1}) is a single-step forwarding process, we introduce the function $\Phi(t, x; h)$ with the parameter h (step-size) and rewrite the general RK method as

$$x_{n+1} = x_n + h\Phi(t_n, x_n; h). \tag{3.7}$$

Note that the function Φ is generally dependent on f of IVP (3.1), and if $f \equiv 0$ then $\Phi \equiv 0$, too. In the Euler method, $\Phi(t_n, x_n; h)$ is nothing but $f(t_n, x_n)$.

Algorithm 3.1 Single-step integration by the classical RK method

Input $t_0, \boldsymbol{x}_0, h$
Arrange vectors $\boldsymbol{x}$, $\boldsymbol{saved\_x}$, $\boldsymbol{f}$ and $\boldsymbol{phi}$
$t \leftarrow t_0, \boldsymbol{x} \leftarrow \boldsymbol{x}_0$ and $\boldsymbol{saved\_x} \leftarrow \boldsymbol{x}$
{First stage:}
{Require the calling program to evaluate the rhs function:}
  $\boldsymbol{f} \leftarrow$ function$(t, \boldsymbol{x})$
  $\boldsymbol{phi} \leftarrow \boldsymbol{f}, \boldsymbol{x} \leftarrow \boldsymbol{saved\_x} + (h/2) * \boldsymbol{f}$ and $t \leftarrow t + (h/2)$
{Second stage:}
{Require the calling program to evaluate the rhs function:}
  $\boldsymbol{f} \leftarrow$ function$(t, \boldsymbol{x})$
  $\boldsymbol{phi} \leftarrow \boldsymbol{phi} + 2 * \boldsymbol{f}$ and $\boldsymbol{x} \leftarrow \boldsymbol{saved\_x} + (h/2) * \boldsymbol{f}$
{Third stage:}
{Require the calling program to evaluate the rhs function:}
  $\boldsymbol{f} \leftarrow$ function$(t, \boldsymbol{x})$
  $\boldsymbol{phi} \leftarrow \boldsymbol{phi} + 2 * \boldsymbol{f}, \boldsymbol{x} \leftarrow \boldsymbol{saved\_x} + h * \boldsymbol{f}$ and $t \leftarrow t + (h/2)$
{Fourth stage:}
{Require the calling program to evaluate the rhs function:}
  $\boldsymbol{f} \leftarrow$ function$(t, \boldsymbol{x})$
{Completion of the step:}
  $\boldsymbol{x} \leftarrow \boldsymbol{saved\_x} + (\boldsymbol{phi} + \boldsymbol{f}) * (h/6)$
Output $\boldsymbol{x}$ as $x(t_0 + h)$

Let us define $\widehat{x}_{n+1}$ by $\widehat{x}_{n+1} = x(t_n) + h\Phi(t_n, x(t_n); h)$ and assume its error τ_n (often called the *local truncation error*) satisfies the condition

$$\tau_n = x(t_{n+1}) - \widehat{x}_{n+1} = \mathcal{O}(h^{p+1}) \quad (h \downarrow 0 \text{ and } n = 0, 1, \ldots, N-1) \tag{3.8}$$

which is referred to as the method has the local order p. Furthermore, we assume the Lipschitz continuity of Φ by $\|\Phi(t, x; h) - \Phi(t, y; h)\| \le K\|x - y\|$. Then the global error $e_{n+1} \equiv x(t_{n+1}) - x_{n+1}$ of the method can be estimated as follows:

$$\|e_{n+1}\| = \| \left(x(t_{n+1}) - \widehat{x}_{n+1}\right) + \left(\widehat{x}_{n+1} - x_{n+1}\right) \| \le Ah^{p+1} + \|\widehat{x}_{n+1} - x_{n+1}\|$$

with a constant A over $[t_0, t_F]$ and

$$\|\widehat{x}_{n+1} - x_{n+1}\| \le \|x(t_n) - x_n\| + h\,\|\Phi(t_n, x(t_n); h) - \Phi(t_n, x_n; h)\| \le (1 + hK)\|e_n\|.$$

Summing up, we have the estimation

$$\|e_{n+1}\| \le (1 + hK)\|e_n\| + Ah^{p+1},$$

whose repeated application yields

$$\|e_{n+1}\| \le (1 + hK)^{n+1}\|e_0\| + ((1 + hK)^n + \cdots + 1)Ah^{p+1}.$$

An elementary identity

$$((1 + hK)^n + \cdots + 1)Ah^{p+1} = \frac{(1 + hK)^{n+1} - 1}{hK} Ah^{p+1} = \frac{(1 + hK)^{n+1} - 1}{K} Ah^p$$

and the inequality

$$(1 + hK)^{n+1} \le (e^{hK})^{n+1} \le e^{hNK} \le e^{(t_F - t_0)K}$$

imply

$$\|e_n\| \le C_0\|e_0\| + Ch^p$$

with two constants C_0 and C. Note that C depends on t_0, t_F and K, but not on h. Without loss of generality, we can assume $e_0 = x(t_0) - x_0$ vanishes and we obtain:

Theorem 3.1 *The function Φ is assumed to be Lipschitz continuous. Then the single-step method (3.7) is convergent with $\mathcal{O}(h^p)$ if the method has the local order p defined by (3.8).*

As the Euler method has the local order one, it is convergent of first order. Since the function Φ of an RK method is a linear combination of the stage values $\{k_i\}$ in (3.5), each of which is equal to an evaluated f, it is obvious that the Lipschitz constant K of Φ is a constant-multiple of L. Hence the theorem implies that usually we can check the order of convergence of an RK method by examining its local order. The assertion that an RK method is convergent of pth order implies that when we employ the halved step-size for the method, its global error reduces by 2^p for the same

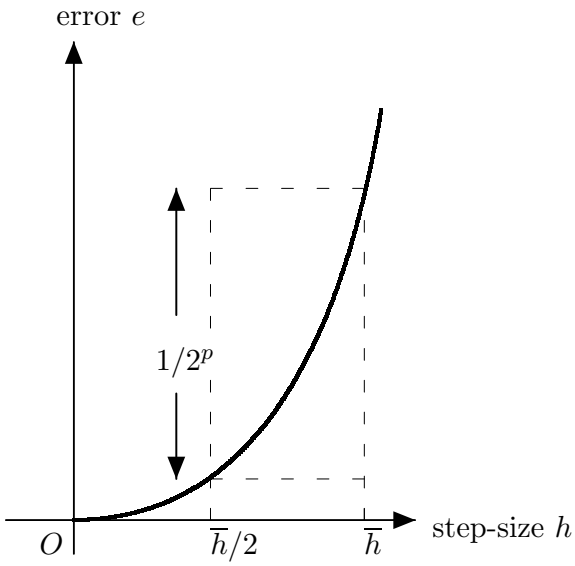


Fig. 3.1 Schematic explanation of order of convergence

initial-value problem, paying the doubled computational cost. See the schematic explanation in Fig. 3.1.

Order Condition of Runge–Kutta Method
Ordinarily we assume the coefficients of a Runge–Kutta method satisfy

$$\sum_{j=1}^{i-1} a_{ij} = c_i \quad (i = 2, 3, \ldots, s), \qquad \sum_{j=1}^{s} b_i = 1. \tag{3.9}$$

This is a natural requirement so that the scheme becomes equivalent for the non-autonomous equation $\mathrm{d}x/\mathrm{d}t = f(t, x)$ and the autonomous equation $\mathrm{d}x/\mathrm{d}t = f(x)$. Also we suppose that the solution $x(t)$ of the initial-value problem (3.1) is sufficiently smooth on $[t_0, t_F]$.

We shall start with the question what is the condition in the two-stage case given by

$$\begin{aligned} k_1 &= f(t_n, x_n), \quad k_2 = f\left(t_n + c_2\, h, x_n + h c_2\, k_1\right), \\ x_{n+1} &= x_n + h\left(b_1\, k_1 + b_2\, k_2\right). \end{aligned} \tag{3.10}$$

Assume $x_n = x(t_n)$ (exact value), then $k_1 = f(t_n, x_n) = x'(t_n)$ holds. Taylor series expansion of k_2 with respect to h yields

$$k_2 = f(t_n, x_n) + h\left(c_2 \frac{\partial f}{\partial t}(t_n, x_n) + c_2 \frac{\partial f}{\partial x}(t_n, x_n) x'(t_n)\right) + \mathcal{O}(h^2).$$

Thus we have

$$x_{n+1} = x_n + h(b_1 + b_2)x'(t_n) + h^2 b_2 c_2 \left(\frac{\partial f}{\partial t}(t_n, x_n) + \frac{\partial f}{\partial x}(t_n, x_n)x'(t_n) \right) + \mathcal{O}(h^3).$$

On the other hand, Taylor series expansion of the exact value $x(t_{n+1}) = x(t_n + h)$ gives

$$x(t_{n+1}) = x_n + hx'(t_n) + \frac{h^2}{2}x''(t_n) + \mathcal{O}(h^3)$$

and the second derivative fulfills

$$x''(t_n) = \frac{\mathrm{d}}{\mathrm{d}t} f(t, x(t))\big|_{t=t_n} = \frac{\partial f}{\partial t}(t_n, x_n) + \frac{\partial f}{\partial x}(t_n, x_n)x'(t_n).$$

Therefore we obtain

$$x(t_{n+1}) = x_n + hx'(t_n) + \frac{h^2}{2} \left(\frac{\partial f}{\partial t}(t_n, x_n) + \frac{\partial f}{\partial x}(t_n, x_n)x'(t_n) \right) + \mathcal{O}(h^3).$$

Consequently, by comparing coefficients of powers of h in x_{n+1} and $x(t_{n+1})$ the condition that the method (3.10) has the local order 2 is given by

$$b_1 + b_2 = 1 \quad \text{and} \quad b_2 c_2 = \frac{1}{2},$$

which have a one-parameter solution

$$a_{21} = c_2 = \gamma, \quad b_2 = \frac{1}{2\gamma}, \quad b_1 = 1 - \frac{1}{2\gamma}.$$

A special choice of $\gamma = 1$ leads to $a_{21} = c_2 = 1$ and $b_1 = b_2 = 1/2$ and we obtain the **Heun method**.

$$\begin{aligned} k_1 &= f(t_n, x_n), \quad k_2 = f(t_n + h, x_n + hk_1), \\ x_{n+1} &= x_n + \frac{h}{2}(k_1 + k_2) \end{aligned}$$

In principle, a general theory to establish the order of convergence of an RK method runs:

(i) assume the condition (3.9) and apply the scheme to the autonomous equation,
(ii) derive the local truncation error of the scheme and count its order.

For the local truncation error defined by (3.8), we introduce $\widehat{x}_{n+1}$ by

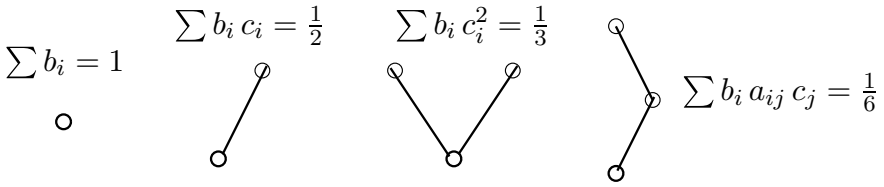


Fig. 3.2 Rooted trees with vertices ≤ 3

$$\overline{k}_i = f\left(x(t_n) + h\sum_{j=1}^{i-1} a_{ij}\,\overline{k}_j\right), \qquad \widehat{x}_{n+1} = x(t_n) + h\sum_{i=1}^{s} b_i\overline{k}_i, \tag{3.11}$$

next derive the Taylor series expansion of $\widehat{x}_{n+1}$ with respect to h and compare it with its counterpart $x(t_{n+1})$.

General derivation of the higher order conditions along this line is tedious and prone to errors, for the number of conditions increases rapidly. The *rooted tree analysis*, which was invented by J. C. BUTCHER, makes the derivation process very transparent and develops deeper insight. Its detailed description is, however, out of our scope. Interested readers can refer to [9, 10] and [15].

For example, Fig. 3.2[1] lists all the rooted trees possessing vertices less than or equal to three, which corresponds to the conditions given in (3.12).

$$\sum_{i=1}^{s} b_i = 1, \quad \sum_{i=1}^{s} b_i c_i = \frac{1}{2}, \quad \sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}, \quad \sum_{i=1}^{s}\sum_{j=1}^{i-1} b_i a_{ij} c_j = \frac{1}{6} \tag{3.12}$$

When the parameters satisfy the first, the first and second and all the four equations in (3.12), the order increases as $p = 1, 2$ and 3, respectively. (Remember the case for the Heun method!) Conditions such as (3.12) are called the *order conditions*. The following is known.

Theorem 3.2 *Under the assumption (3.9), the order conditions for order p are equivalently given by the rooted trees with vertices less than or equal to p.*

From the viewpoint of the order attained versus number of stages, the classical Runge–Kutta method is particularly advantageous, for it attains fourth-order convergency. All the five-stage RKs can never attain fifth order. The question of what is the least number of stages s required to attain pth order is very hard to solve. The already known results are listed in Table 3.1.

[1] Reproduced from '*Introduction to Computational Science with Differential Equations*' (in Japanese) by T. MITSUI, T. KOTO and Y. SAITO, Kyoritsu Shuppan, Tokyo, 2004 (ISBN: 4-320-01753-6).

Table 3.1 Least number of stages to attain first to eighth order

Order	1	2	3	4	5	6	7	8
Least stages	1	2	3	4	6	7	9	11

3.2 Embedded Pair of Runge–Kutta Schemes

Because of the single-step nature of a Runge–Kutta method, it is easy to change the step-size from the present step $[t_n, t_{n+1}]$ to the next step $[t_{n+1}, t_{n+2}]$. We only replace the current step-size h by the new step-size h' in (3.5). Note that for the new step we are only required with (t_{n+1}, x_{n+1}) which become available at the end of the present step.

Assume that we are given an error tolerance ε_{TOL} which we expect to obtain at each step and try to develop an RK whose local error is within ε_{TOL} and whose total computational cost is as small as possible. To this end, an *a-posteriori* error estimation is crucial. The mathematical expression of the local error term obtained by the comparison of Taylor series expansions in the previous section is impractical, for it includes many higher-order partial derivatives of $f(t, x)$. One plausible idea is to integrate the step $[t_n, t_{n+1}]$ once with the step-size h and twice with the step-size $h/2$ by the same RK of the local order p. Assuming the principal local error term as $C(x_n, h)h^{p+1}$, we denote the numerical solution with h by x_{n+1}, while denoting that with $h/2$ by $\widehat{x}_{n+1}$. Then, we approximately obtain

$$x(t_{n+1}) - x_{n+1} \approx C(x_n, h)h^{p+1} \quad \text{and} \quad x(t_{n+1}) - \widehat{x}_{n+1} \approx 2C(x_n, h/2)\left(\frac{h}{2}\right)^{p+1}$$

when h is sufficiently small and the solution $x(t)$ is sufficiently smooth. This implies that the estimation

$$C(x_n, h)h^{p+1} \approx \frac{\widehat{x}_{n+1} - x_{n+1}}{1 - 1/2^p}$$

holds by assuming $C(x_n, h) \approx C(x_n, h/2)$ and we have an *a-posteriori* error estimation of x_{n+1}. However, the computational cost is tripled and the technique is hard to implement.

Another idea was proposed by E. FEHLBERG in 1970. Consider an s-stage RK of pth order of convergence whose scheme is given in (3.5) for x_{n+1}. Then, by employing the same stage values $\{k_i\}$ but with different weights $\{b_i^*\}$ we derive another scheme of $(p-1)$th order to give x_{n+1}^*. That is,

$$x_{n+1}^* = x_n + h\left(b_1^* k_1 + b_2^* k_2 + \cdots + b_s^* k_s\right),$$

which derives an *a-posteriori* error estimate

$$T_{n+1} = h\left((b_1 - b_1^*)\,k_1 + (b_2 - b_2^*)\,k_2 + \cdots + (b_s - b_s^*)\,k_s\right). \tag{3.13}$$

We denote the coefficient of the principal local error term of x_{n+1}^* by $C^*(x_n, h)$ and assume the estimation $\|T_{n+1}\| \le \varepsilon_{TOL}$ holds at the present step. This means $\|C^*(x_n, h)h^p\| \le \varepsilon_{TOL}$ holds *approximately*. Then, for the next step integration the step-size $\widehat{h}$ is expected to satisfy $\|C^*(x_{n+1}, \widehat{h})\,\widehat{h}^p\| \approx \varepsilon_{TOL}$. By assuming $C^*(x_{n+1}, \widehat{h}) \approx C^*(x_n, \widehat{h})$ and taking the estimation

$$\|C^*(x_n, h)\| \approx \frac{\|T_{n+1}\|}{h^p}$$

into account, for $\widehat{h}$ we obtain

$$\frac{\|T_{n+1}\|}{h^p}\,\widehat{h}^p = \varepsilon_{TOL},$$

which leads to

$$\widehat{h} = h\left(\frac{\varepsilon_{TOL}}{\|T_{n+1}\|}\right)^{1/p}$$

as an estimation of $\widehat{h}$ for the next step. Since all the estimations here are crude, it is better to use a safer estimation by

$$\widehat{h} = \alpha\, h\left(\frac{\varepsilon_{TOL}}{\|T_{n+1}\|}\right)^{1/p} \tag{3.14}$$

with the safety parameter α between 0.8 and 0.9.

The pair of the parameters $\{a_{ij}, b_i, b_i^*, c_i\}$ is called an embedded Runge–Kutta scheme, for the $(p-1)$th method for x_{n+1}^* is embedded into the pth order method for x_{n+1}. E. FEHLBERG invented a six-stage RK pair of fifth and fourth orders whose parameters are given in Table 3.2. It is often referred to as the **RKF45 method** and its implementation is given in Algorithm 3.2. Note that the implementation determines the relaxed step-size $\widehat{h}$ if the *a-posteriori* error estimate holds, while it reduces the step-size by the power of 2 until the error condition be attained if the error estimate does not hold. Since at the exit of the algorithm the output step-size will be changed from the input one, its driver program must manage to carry out numerical solution on the whole interval.

Also note that the computational cost of RKF45 is the same as the other six-stage RKs. The analysis of embedded RKs described here includes many approximations and crude estimations, but the estimated *a-posteriori* error is practically efficient under the assumption that the ODE solution is sufficiently smooth and the selected step-sizes are relatively small. RKF45 provides a basis of the sophisticated imple-

Table 3.2 Parameters of RKF45 method

0						
$\frac{1}{4}$	$\frac{1}{4}$					
$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$				
$\frac{12}{13}$	$\frac{1932}{2197}$	$-\frac{7200}{2197}$	$\frac{7296}{2197}$			
1	$\frac{216}{439}$	-8	$\frac{3680}{513}$	$-\frac{845}{4104}$		
$\frac{1}{2}$	$-\frac{8}{27}$	2	$-\frac{3544}{2565}$	$\frac{1859}{4104}$	$-\frac{11}{40}$	
b_i	$\frac{16}{135}$	0	$\frac{6656}{12825}$	$\frac{28561}{56430}$	$-\frac{9}{50}$	$\frac{2}{55}$
b_i^*	$\frac{25}{216}$	0	$\frac{1408}{2565}$	$\frac{2197}{4104}$	$-\frac{1}{5}$	0

mentation of MATLAB package `ode45`. Another example of the well-known seven-stage embedded RK pair is DOPRI (5, 4) by J. R. DORMAND and P. J. PRINCE (1980). Its FORTRAN code is listed in the Appendix of [15].

3.3 Linear Stability of Runge–Kutta Methods for ODEs

In Sect. 2.3 we already emphasized the significance of the stability concept in the analytical theory of ODEs. In parallel, the stability concept is another important criterion for Runge–Kutta methods. Roughly speaking, when the numerical solution by a discrete variable method is contaminated with a certain error during the computation process, the method is said to be stable if the introduced error is not magnified or, more hopefully, is diminishing in the following process. Since an error contamination like computational rounding errors is inevitable in the practical situation, the stability has a vital role in numerical analysis. In other words, a method whose stability is not guaranteed cannot be employed in the actual applications.

Syntax Diagram of Linear Stability

To start the stability consideration for DVMs by taking note of the general concept introduced in Sect. 2.3, we will take the *syntax* of a stability definition proposed

by J. D. LAMBERT (Chap. 2 of [27]). It can be broken down into the following components. (Refer to the syntax diagram of stability analysis shown in Fig. 3.3.)

1. We impose certain conditions C_P upon the initial-value problem (1.4) which force the exact solution $x(t)$ ($t \in [t_0, t_F]$) to exhibit a certain stable behaviour.
2. We apply a DVM to the problem which is assumed to satisfy C_P.
3. We ask what conditions C_m must be imposed on the method in order that the numerical solutions $\{x_n : n = 0, 1, \ldots\}$ show a stability property analogous to the exact solution.

Algorithm 3.2 Single-step integration by RKF45 method

Input x_0, $\boldsymbol{y}_0$, H, $h_{\min}$ and ε_{TOL}
Allocate the constant α
Arrange vectors $\boldsymbol{y}, \boldsymbol{k}_1, \boldsymbol{k}_2, \boldsymbol{k}_3, \boldsymbol{k}_4, \boldsymbol{k}_5, \boldsymbol{k}_6$ and $\boldsymbol{err}$
$h \leftarrow H$, $x \leftarrow x_0$ and $\boldsymbol{y} \leftarrow \boldsymbol{y}_0$
state $\leftarrow$ stepping
repeat
  { First stage: } $\boldsymbol{k}_1 \leftarrow \text{func}(x, \boldsymbol{y})$
  $\boldsymbol{k}_6 \leftarrow \boldsymbol{y} + (h/4) * \boldsymbol{k}_1$
  { Second stage: } $\boldsymbol{k}_2 \leftarrow \text{func}(x + (h/4), \boldsymbol{k}_6)$
  $\boldsymbol{k}_6 \leftarrow \boldsymbol{y} + (h/32) * (3 * \boldsymbol{k}_1 + 9 * \boldsymbol{k}_2)$
  { Third stage: } $\boldsymbol{k}_3 \leftarrow \text{func}(x + (3/8) * h, \boldsymbol{k}_6)$
  $\boldsymbol{k}_6 \leftarrow \boldsymbol{y} + (h/2197) * (1932 * \boldsymbol{k}_1 - 7200 * \boldsymbol{k}_2 + 7296 * \boldsymbol{k}_3)$
  { Fourth stage: } $\boldsymbol{k}_4 \leftarrow \text{func}(x + (12/13) * h, \boldsymbol{k}_6)$
  $\boldsymbol{k}_6 \leftarrow \boldsymbol{y} + h * ((439/216) * \boldsymbol{k}_1 - 8 * \boldsymbol{k}_2 + (3680/513) * \boldsymbol{k}_3 - (845/4104) * \boldsymbol{k}_4)$
  { Fifth stage: } $\boldsymbol{k}_5 \leftarrow \text{func}(x + h, \boldsymbol{k}_6)$
  $\boldsymbol{k}_2 \leftarrow \boldsymbol{y} + h * ((-8/27) * \boldsymbol{k}_1 + 2 * \boldsymbol{k}_2 - (3544/2565) * \boldsymbol{k}_3$
    $+(1859/4104) * \boldsymbol{k}_4 - (11/40) * \boldsymbol{k}_5)$
  { Sixth stage: } $\boldsymbol{k}_6 \leftarrow \text{func}(x + h/2, \boldsymbol{k}_2)$
  { Estimated error: }
    $\boldsymbol{err} \leftarrow h * ((1/360) * \boldsymbol{k}_1 - (128/4275) * \boldsymbol{k}_3$
      $-(2197/75240) * \boldsymbol{k}_4 + (1/50) * \boldsymbol{k}_5 + (2/55) * \boldsymbol{k}_6)$
  if $\|\boldsymbol{err}\| > \varepsilon_{TOL}$ **then**
    begin $h \leftarrow h/2$; **if** $h < h_{\min}$ **then** state $\leftarrow$ h_too_small **end**
  else
    begin
      $\boldsymbol{y} \leftarrow \boldsymbol{y} + h * ((16/135) * \boldsymbol{k}_1 + (6656/12825) * \boldsymbol{k}_3$
        $+(28561/56430) * \boldsymbol{k}_4 - (9/50) * \boldsymbol{k}_5 + (2/55) * \boldsymbol{k}_6)$
      $h \leftarrow \alpha * h * (\varepsilon_{TOL}/\|\boldsymbol{err}\|)^{1/5}$, $x \leftarrow x + h$, state $\leftarrow$ stepping_end
    end
until state $\neq$ stepping
Output x, $\boldsymbol{y}$, h

The condition C_P has several possibilities. Here we are focusing on the asymptotic stability of the linearized system, for this is a generic idea of stability as explained just below Definition 2.1. Henceforth we handle the d-dimensional constant-coefficient linear system

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \Lambda x, \qquad x(t_0) = x_0, \tag{3.15}$$

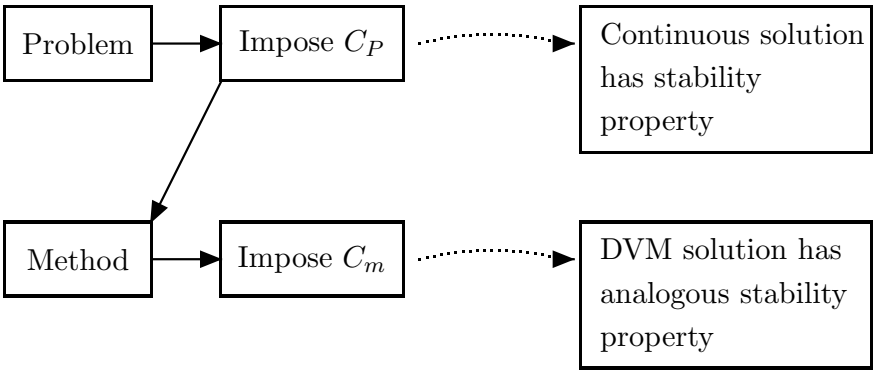


Fig. 3.3 Syntax diagram of stability analysis

where Λ is a d-dimensional constant matrix and all of its eigenvalues λ_i $(i = 1, 2, \ldots, d)$ satisfy the condition of negative real part $\mathrm{Re}\,\lambda_i < 0$ (the condition C_P). This implies that the solution of (3.15) approaches 0 as $t \to \infty$. More specifically, we take the *scalar* equation $\dfrac{\mathrm{d}x}{\mathrm{d}t} = \lambda x (\mathrm{Re}\,\lambda < 0)$ with $x(t_0) = x_0$ (this is called the *test equation* of stability) and ask for the condition C_m when a DVM is applied to it. This is called the *linear stability analysis* of numerical solution. We emphasize that we will analyse the bahaviour of $\{x_n\}$ in keeping the step-size h positive, in contrast to the case of the convergency analysis where $h \downarrow 0$.

Stability Function and Stability Region of RK

An application of RK to the test equation with the notations $\widehat{k_i} = hk_i$ and $z = h\lambda$ leads to

$$\widehat{k_i} = z\left(x_n + \sum_{j=1}^{i-1} a_{ij}\widehat{k_j}\right) \quad \text{and} \quad x_{n+1} = x_n + \sum_{i=1}^{s} b_i \widehat{k_i}.$$

Recurrent substitutions give

$$\widehat{k_1} = zx_n, \qquad \widehat{k_2} = z\left(x_n + a_{21}\widehat{k_1}\right) = (z + a_{21}z^2)x_n,$$
$$\widehat{k_3} = z\left(x_n + a_{31}\widehat{k_1} + a_{32}\widehat{k_2}\right) = \left[z + (a_{31} + a_{32})z^2 + a_{31}a_{32}z^3\right]x_n, \quad \ldots$$

and finally we arrive at

$$x_{n+1} = R(z)x_n. \qquad (R(z)\text{: polynomial of degree } s \text{ w.r.t. } z)$$

An n-step forwarding by the RK gives $x_n = \{R(z)\}^n x_0$ so that the factor $R(z)$ decides its stability behaviour. Hence, we call $R(z)$ the *stability function* of the RK and the set in the complex plane given by

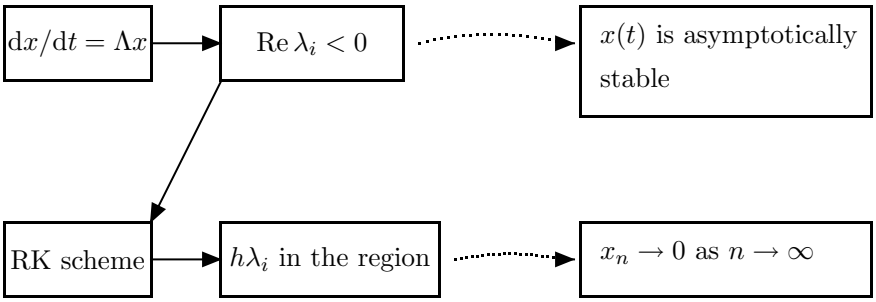


Fig. 3.4 Syntax diagram of linear stability analysis of RK

$$\mathcal{R} = \{z \in \mathbb{C} : |R(z)| < 1\} \tag{3.16}$$

its *stability region*. For Runge–Kutta methods, we can obtain the condition C_m as "$h\lambda_i$ is included in the stability region $\mathcal{R}$" (see the syntax diagram given by Fig. 3.4).

The exact solution of the test equation is expressed as $x(t) = e^{\lambda(t-t_n)}x(t_n)$, which yields its local truncation error by the RK as $x(t_{n+1}) - \widehat{x}_{n+1} = [e^z - R(z)]x(t_n)$. Therefore, for the RK of pth order, $R(z)$ should be

$$R(z) = 1 + z + \frac{1}{2!}z^2 + \cdots + \frac{1}{p!}z^p + \gamma_{p+1}z^{p+1} + \cdots + \gamma_s z^s,$$

for the Maclaurin expansion of $R(z)$ coincides with that of e^z up to the term z^p (note that $s \geq p$). Here $\gamma_{p+1}, \ldots, \gamma_s$ are constants derived by the formula parameters of the RK. Then, the region $\mathcal{R}$ is not empty when the method is at least of first order, for we have the following:

Theorem 3.3 *Assume that all the eigenvalues of the matrix Λ in (3.15) satisfy $\mathrm{Re}\,\lambda_k < 0$ and the Runge–Kutta method is at least of first order. Then, there exists an $h_0 > 0$ and for any $h < h_0$ the product $h\lambda_k$ falls in $\mathcal{R}$.*

Proof Since the RK is at least of first order, the estimation $R(h\lambda) = 1 + h\lambda + \mathcal{O}(h^2)$ holds. Then we have

$$|R(h\lambda)|^2 = 1 + 2h\,\mathrm{Re}\,\lambda + \mathcal{O}(h^2).$$

Thus, for $\mathrm{Re}\,\lambda < 0$ and a sufficiently small $h > 0$ $|R(h\lambda)|$ should be smaller than unity. □

The stability region $\mathcal{R}$ characterises the efficiency of the employed RK method for the numerical integration. Refer to the syntax diagram in Fig. 3.4. Hence an RK method which has a broader $\mathcal{R}$ in $\mathbb{C}^-$ is superior with respect to the performance sense so that it can stably deal with the ODE by a bigger step-size h. For instance, the stability functions of Runge–Kutta methods of 1-stage first-order (Euler method!),

2-stage second-order, 3-stage third-order and 4-stage fourth-order (the classical RK!) are listed as

$$R(z) = 1 + z, \quad 1 + z + \frac{z^2}{2}, \quad 1 + z + \frac{z^2}{2} + \frac{z^3}{6}, \quad 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24},$$

respectively. Their stability region on the xy-plane ($z = x + \mathrm{i}\, y$) is shown in Fig. 3.5 (from top left to bottom right). You can confirm that the classical RK has the broadest stability region among them.

3.4 Implicit Runge–Kutta Methods

The above discussion confirms that the stability region of RK scheme (3.5) is bounded from the left-hand side, too. This means that when we meet an asymptotically stable ODE system whose eigenvalues are distributed in a wide range of $\mathbb{C}^-$, we must keep the step-size very small for a long integration interval of numerical solution. Such a system is called *stiff* and practical ODE problems often derive stiff systems. To cope with stiff systems, we generalise the RK formulation to allow the s-dimensional array $A = (a_{ij})$ in the formulation (3.4) be a full matrix. Then, we are required to solve a simultaneous system of (nonlinear) equations for $\{k_1, k_2, \ldots, k_s\}$ in (3.5) or for $\{X_1, X_2, \ldots, X_s\}$ in (3.6) at every step t_n. Hence, this formulation is called an *implicit* Runge–Kutta method.

In spite of the computational cost in solving simultaneous systems of nonlinear equations, the implicit RK becomes significant in both a theoretical and a practical sense. By modifying (3.6) as

$$X_i = x_n + h \sum_{j=1}^{s} a_{ij}\, f\left(t_n + c_j\, h, X_j\right), x_{n+1} = x_n + h \sum_{i=1}^{s} b_i\, f\left(t_n + c_i\, h, X_i\right) \tag{3.17}$$

we introduce the scheme of the implicit RK method. Note that the summation in the first equation of (3.17) runs from 1 to s. This means we assume non-zero a_{ij} even for $j \geq i$. Application of (3.17) to the scalar test equation of stability $\dfrac{\mathrm{d}x}{\mathrm{d}t} = \lambda x$ ($\mathrm{Re}\,\lambda < 0$) gives the following $(s+1)$-dimensional linear system

$$\begin{bmatrix} 1 - za_{11} & -za_{12} & \cdots & -za_{1s} & 0 \\ -za_{21} & 1 - za_{22} & \cdots & -za_{2s} & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ -za_{s1} & -za_{s2} & \cdots & 1 - za_{ss} & 0 \\ -zb_1 & -zb_2 & \cdots & -zb_s & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_s \\ x_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ x_n \\ \vdots \\ x_n \\ x_n \end{bmatrix} \tag{3.18}$$

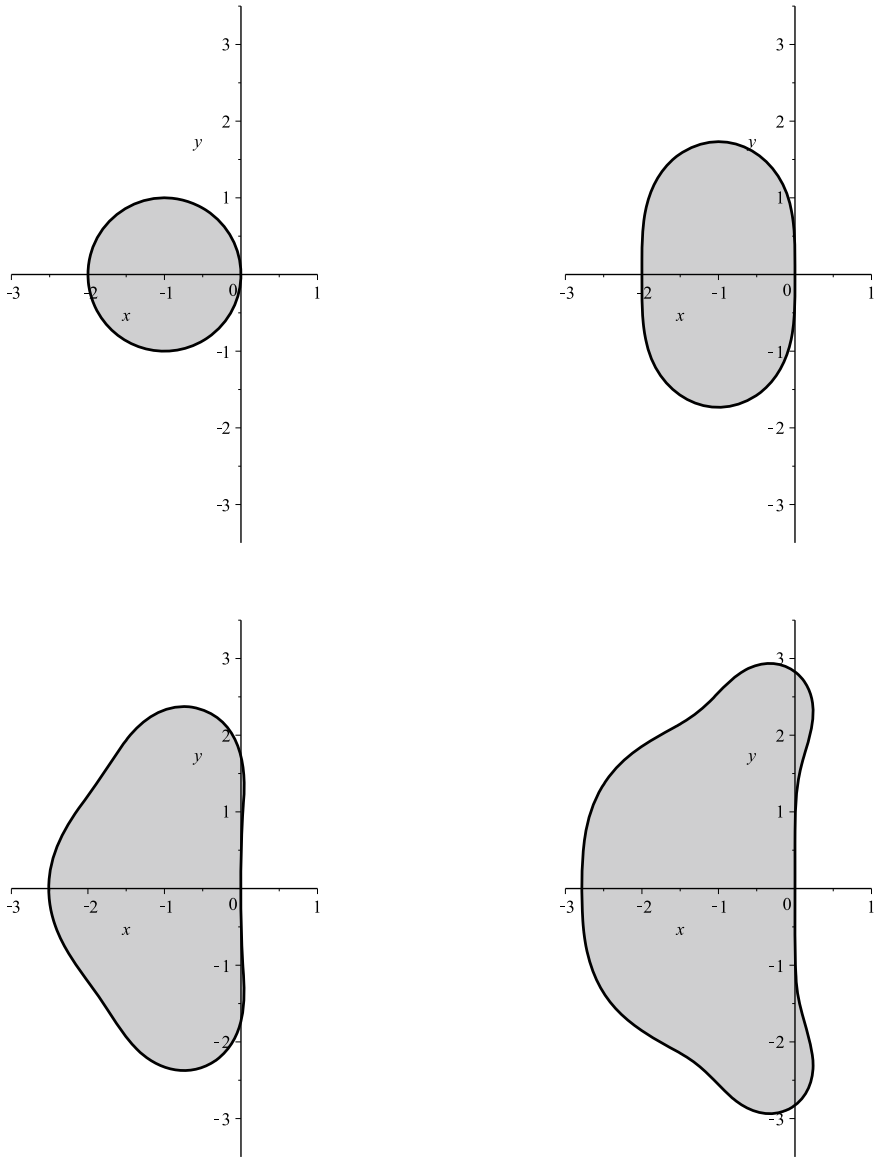


Fig. 3.5 Stability region of the Runge–Kutta methods

with $z = h\lambda$. We introduce the vector symbols $\boldsymbol{b} = (b_1, \ldots, b_s)^\mathsf{T}$ and $\boldsymbol{e} = (1, 1, \ldots, 1)^\mathsf{T}$, and obtain:

Lemma 3.1 *The stability function $R(z)$ of the implicit RK method (3.17) is given by*

$$R(z) = \frac{\det(I - zA + z\boldsymbol{e}\boldsymbol{b}^\mathsf{T})}{\det(I - zA)} \tag{3.19}$$

when the step-size h is sufficiently small to guarantee the invertibility of the matrix $I - zA = I - h\lambda A$.

Proof We solve (3.18) of the vector $[X_1, X_2, \ldots, X_s, x_{n+1}]^\mathsf{T}$ by Cramer's formula. Then, its last component x_{n+1} must be $N(z)x_n/D(z)$ with the polynomials $N(z)$ and $D(z)$ of degree s and the stability function $R(z)$ is given by $N(z)/D(z)$. The denominator $D(z)$ is equal to the determinant of the matrix which is given by excluding the last column and row from the left-hand side matrix of (3.18). Hence, it is nothing but $\det(I - zA)$. On the other hand, the numerator $N(z)x_n$ is equal to the determinant of the matrix

$$\begin{bmatrix} 1 - za_{11} & -za_{12} & \cdots & -za_{1s} & x_n \\ -za_{21} & 1 - za_{22} & \cdots & -za_{2s} & x_n \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -za_{s1} & -za_{s2} & \cdots & 1 - za_{ss} & x_n \\ -zb_1 & -zb_2 & \cdots & -zb_s & x_n \end{bmatrix} = \begin{bmatrix} & & & x_n \\ & I - zA & & \vdots \\ & & & x_n \\ -zb_1 & \cdots & & -zb_s\ x_n \end{bmatrix}.$$

An equivalent transformation of the determinant runs

$$\det \begin{bmatrix} & & & x_n \\ & I - zA & & \vdots \\ & & & x_n \\ -zb_1 & \cdots & & -zb_s\ x_n \end{bmatrix} = \det \begin{bmatrix} & & & 0 \\ & I - zA + z\boldsymbol{e}\boldsymbol{b}^\mathsf{T} & & \vdots \\ & & & 0 \\ -zb_1 & \cdots & & -zb_s\ x_n \end{bmatrix}$$

and we attain the conclusion. □

Note that the explicit RK case yields a strictly lower triangular matrix A and implies $\det(I - zA) = \det I = 1$. Hence the result coincides with that of the previous section. On the other hand, the genuine implicit case yields a polynomial of z to $\det(I - zA)$ and its stability function becomes a rational function of z. In fact the following strong result is known.

Theorem 3.4 *An s-stage implicit Runge–Kutta method can attain $(2s)$-order of convergence and there are implicit Runge–Kutta methods whose stability region includes the left-half of the complex plane $\mathbb{C}$.*

A discrete variable method is called A-stable when its stability region covers the left-half of $\mathbb{C}$. An A-stable numerical method has no restriction on the step-size to

numerically integrate ODE systems from the viewpoint of stability. For instance, the two-stage implicit Runge–Kutta method whose coefficients are given in the following table is of fourth-order convergence as well as A-stable:

$$\begin{array}{c|cc} \dfrac{3-\sqrt{3}}{6} & \dfrac{1}{4} & \dfrac{3-2\sqrt{3}}{12} \\ \dfrac{3+\sqrt{3}}{6} & \dfrac{3+2\sqrt{3}}{12} & \dfrac{1}{4} \\ \hline & \dfrac{1}{2} & \dfrac{1}{2} \end{array}$$

It is called the two-stage Gauß–Legendre formula because of its method of derivation. It was revealed that a systematic way to develop implicit RK schemes is closely related to the collocation methods based on the numerical quadrature on the interval [0, 1]. They are called Gauß, Radau and Lobatto types due to their origins. More details can be given in Sect. IV.5 of [16]. Numerical solution of (3.17) with respect to $[X_1, X_2, \ldots, X_s, x_{n+1}]^{\mathsf{T}}$ is another issue and many studies and experiments have been performed. Refer to Sect. IV.8 of [16].

Further Remarks

The history of Runge–Kutta methods is not as long as that of numerical analysis. It started with the famous paper by Carl RUNGE in *Mathmatische Annalen* vol. 46 (1895). Since then, many theoretical and practical studies have been carried out and now several sophisticated and well-tested programs are available based on the formulation of the method. This volume cannot cover all the details and interested readers are recommended to refer to, e.g., [9, 15, 16]. At the same time, the application field of RK methods extends to other objects related to ODEs. They are, e.g., differential-algebraic, delay-differential (the topic will be described later), stochastic differential equations and so on. Explanation of each topic can be found in other specific textbooks.

Exercises

3.1. Prove that the function Φ given in (3.7), often called the increment function of RK method, satisfies Lipschitz continuity if the function f is Lipschitz continuous. Hence, the assumption just below (3.8) is valid.

3.2. In the initial-value problem (3.1), assume that the function f is twice continuously differentiable and set

$$g(t,x) = \frac{\partial f}{\partial t}(t,x) + \frac{\partial f}{\partial x}(t,x) \cdot f(t,x).$$

We introduce the following single-step method:

$$x_{n+1} = x_n + h\left(f(t_n, x_n) + \frac{h}{2}\, g\left(t_n + \frac{h}{3}, x_n + \frac{h}{3} f(t_n, x_n)\right)\right).$$

Prove that this is a third-order method.

3.3. Suppose that the second-order ordinary differential equation $\dfrac{\mathrm{d}^2x}{\mathrm{d}\,t^2} = g(t, x)$, not including the first derivative explicitly, is given with the initial condition $x(t_0) = x_0$ and $\dfrac{\mathrm{d}\,x}{\mathrm{d}\,t}(t_0) = x'(t_0)$. To solve the problem numerically, we employ the following scheme with the step-size h:

$$x_{n+1} = x_n + h\left[x'_n + \frac{1}{4}h(k_1 + k_2)\right], \quad x'_{n+1} = x'_n + \frac{1}{4}h(k_1 + 3k_2),$$

$$k_1 = g(t_n, x_n), \quad k_2 = g\left(t_n + \frac{1}{3}h, x_n + \frac{2}{3}hx'_n + \frac{1}{3}h^2k_1\right).$$

Prove that the scheme is of third order. Note that you should prove that x'_{n+1} is a second-order approximation of $\dfrac{\mathrm{d}x}{\mathrm{d}\,t}(t_{n+1})$, too. This is one of the Runge–Kutta–Nyström schemes.

3.4. When the function f in (3.1) does not depend on x, the problem reduces to the numerical quadrature to approximate the definite integral $x_0 + \displaystyle\int_{t_0}^{t_F} f(t)\,\mathrm{d}t$. In this case, show that a single-step application of the classical Runge–Kutta method is equivalent to Simpson's rule for $x_n + \displaystyle\int_{t_n}^{t_{n+1}} f(t)\,\mathrm{d}t$.

3.5. We suppose that the very simple initial values problem of scalar differential equation $\dfrac{\mathrm{d}x}{\mathrm{d}t} = -x + t$ $(t > 0)$ with $x(0) = 1$ is solved by the Euler method with constant step-size. Try to employ the step-size sequence $h_k = 2^{-k}$ $(k = 1, 2, \ldots)$ and calculate the numerical solutions at $t = 1$. Let the exact error $e_k = x_{2^k} - x(1)$ $(k = 1, 2, \ldots)$ and observe that the ratio e_k/h_k approaches a constant. P. HENRICI analyzed this by the magnified error function of the numerical solution.

3.6. We now apply the RKF45 method to the scalar constant-coefficient differential equation $\dfrac{\mathrm{d}x}{\mathrm{d}t} = at + bx + c$ $(t > 0)$ with the initial condition $x(0) = 0$. Calculate the first step by the step-size h from $t = 0$ to $t = h$ and compare the actual local error with the estimated local error given by (3.13). You can carry out the comparison neglecting $\mathcal{O}(h^7)$ terms, for the approximation by RKF45 is at most of fifth order.

3.7. Calculate the stability function of the five-stage fourth-order Runge–Kutta method whose coefficient is given in Table 3.3 and draw its stability region. The coefficient is the fourth-order part of RKF45 given in Sect. 3.2.

3.8. The following single-step method can be considered as an implicit Runge–Kutta method:

Table 3.3 Parameters of Exercise 3.4

0					
$\frac{1}{4}$	$\frac{1}{4}$				
$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$			
$\frac{12}{13}$	$\frac{1932}{2197}$	$-\frac{7200}{2197}$	$\frac{7296}{2197}$		
1	$\frac{216}{439}$	-8	$\frac{3680}{513}$	$-\frac{845}{4104}$	
	$\frac{25}{216}$	0	$\frac{1408}{2565}$	$\frac{2197}{4104}$	$-\frac{1}{5}$

$$x_{n+1/3} = x_n + \frac{h}{6}\left(f(t_n, x_n) + f\left(t_n + \frac{h}{3}, x_{n+1/3}\right)\right),$$
$$x_{n+1} = x_n + \frac{h}{4}\left(3f\left(t_n + \frac{h}{3}, x_{n+1/3}\right) + f(t_n + h, x_{n+1})\right)$$

Prove that this is of second order.

Chapter 4
Polynomial Interpolation

Prior to describing the linear multistep method, another major class of numerical solution of ODEs, we will study interpolation, which is one of the common tools in function approximation. When we are given only a finite number of sample data of a hidden function with continuous variable, we have to approximate the function by the data and apply the approximation to interpolate other function values. The interpolating function must take the same value at all the sample points. Even when the approximated function is known in its mathematical expression, the interpolation is still useful to handle the function which requires infinitesimal calculus or a very complicated computational process. When the interpolating object is a polynomial, it is called a *polynomial interpolation*, which can be dated back to the age of Isaac NEWTON. This chapter provides essentials of the theory of polynomial interpolation and prepares its application to numerical solution of differential equations.

4.1 Polynomial Interpolation and Its Algorithms

Assume that we are given a set of $(n+1)$ data of an unknown (hidden) function $f(x)$ as

$$\{(x_i, f_i) \quad (i = 0, 1, \ldots, n)\}. \tag{4.1}$$

We will call the points $\{(x_i, f_i)\}$ *support points*, the location x_i *support abscissae*, and the values $\{f_i\}$ *support ordinates*. A function $\widehat{f}(x)$ is said to interpolate $f(x)$ if it satisfies the condition

$$\widehat{f}(x_0) = f_0, \ \widehat{f}(x_1) = f_1, \ \cdots, \ \widehat{f}(x_n) = f_n.$$

When $\widehat{f}(x)$ is a polynomial of x, this is called the polynomial interpolation (Fig. 4.1).

T. Mitsui and G. Hu, *Numerical Analysis of Ordinary and Delay Differential Equations*,
UNITEXT 145, https://doi.org/10.1007/978-981-19-9263-6_4

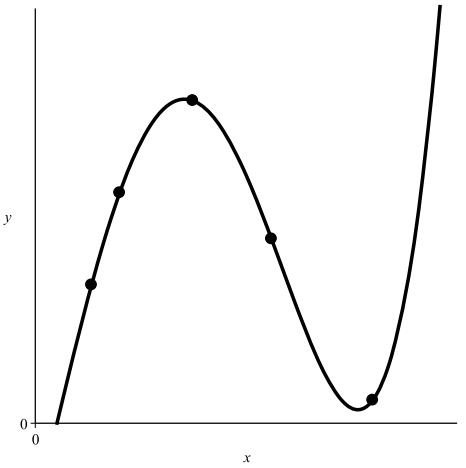


Fig. 4.1 Polynomial interpolation by five points

The following theorem is obvious.

Theorem 4.1 *The interpolating polynomial* $P_n(x)$ *of degree n*

$$P_n(x) = a_0 + a_1 x + \cdots + a_n x^n$$

uniquely exists if and only if the support abscissae are distinct.

Proof The coefficients of the interpolating polynomial $P_n(x)$ are determined by the simultaneous linear equations

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}.$$

Then, the coefficient matrix is Vandermonde, which is non-singular for distinct abscissae (see Exercise 4.1). □

Note that the theorem can provide no practical methods to compute $P_n(x)$. You might think that the above $(n + 1)$-dimensional equation is easy to give the solution $(a_0, a_1, \ldots, a_n)$, but it is not the case. The Vandermonde matrix becomes much ill-conditioned when n becomes large. Hence, we look for other computationally efficient methods to polynomial interpolation. One of the established methods is the Lagrange formulation of polynomial interpolation.

Lagrange Formulation

Now we introduce a set of $(n+1)$ polynomials of degree n by

$$L_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{(x-x_j)}{(x_i-x_j)} \tag{4.2}$$
$$= \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}$$
$$(i=0,1,\ldots,n).$$

Note that they satisfy the identities

$$L_i(x_j) = \begin{cases} 0 & \text{for } j \neq i, \\ 1 & \text{for } j = i. \end{cases} \tag{4.3}$$

Then, the interpolating polynomial on the support points is given by

$$P_n(x) = \sum_{i=0}^{n} f_i\, L_i(x), \tag{4.4}$$

which is called the Lagrange formulation of interpolating polynomial. When we define the polynomial $\omega_n(x)$ of degree $(n+1)$ by

$$\omega_n(x) = \prod_{i=0}^{n}(x-x_i) = (x-x_0)(x-x_1)\cdots(x-x_n),$$

the identity

$$L_i(x) = \frac{\omega_n(x)}{(x-x_i)\,\omega_n'(x_i)} \tag{4.5}$$

holds. Hence the Lagrange formulation is also expressed with

$$P_n(x) = \sum_{i=0}^{n} f_i\, \frac{\omega_n(x)}{(x-x_i)\,\omega_n'(x_i)}. \tag{4.6}$$

Although the Lagrange formulation is linear with respect to the support ordinates $\{f_i\}$, it is not convenient when the support points are supplemented. All $L_i(x)$'s must be calculated again. The defect can be overcome by introducing another expression of polynomial utilising the divided difference.

To obtain an instinctive understanding of the Lagrange formulation, Fig. 4.2 depicts graphs of the polynomials (4.2) in the case of $n=4$ and $x_0=0, x_1=1/4, x_2=1/2, x_3=3/4$ and $x_4=1$. That is, they are the basis of the Lagrange formulation by the equidistant support points on the interval [0, 1]. The solid,dotted,

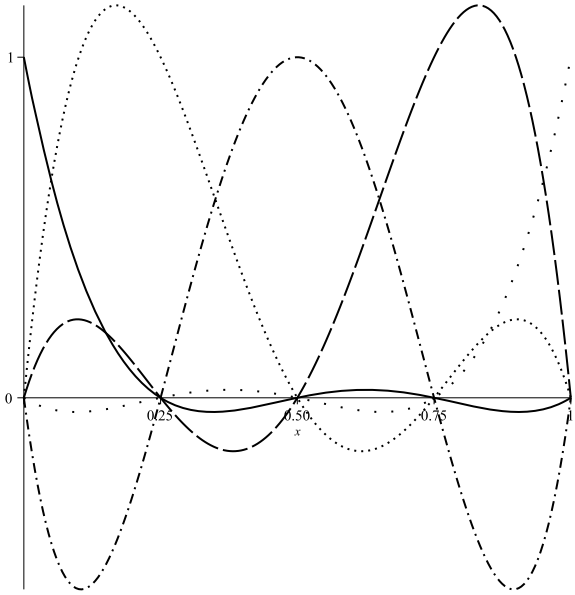


Fig. 4.2 Basis of the Lagrange formulation

dash-dot, dashed and space-dot curves represent $L_0(x)$, $L_1(x)$, $L_2(x)$, $L_3(x)$ and $L_4(x)$, respectively.

Divided Difference and Neville's Algorithm
Note that the polynomial $P_n(x)$ in (4.4) can also be expressed with

$$P_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \cdots + c_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}).$$

Our problem is how to determine the new coefficients $\{c_i\}$ by the support points.

Lemma 4.1 *Define* $p(x; i)$ $(i = 0, 1, \ldots, n)$ *be polynomial of degree* 0 *satisfying* $p(x; i) = f_i$ *and* $p(x; i_0, i_1, \ldots, i_k)$ *be interpolating polynomial of degree* k *on the partial set* $\{(x_{i_j}, f_{i_j})\ (j = 0, 1, \ldots, k)\}$ *of the supporting points. Then the following recursive identity holds.*

$$p(x; i_0, i_1, \ldots, i_k) = \frac{(x - x_{i_0})p(x; i_1, i_2, \ldots, i_k) - (x - x_{i_k})p(x; i_0, i_1, \ldots, i_{k-1})}{x_{i_k} - x_{i_0}}$$

Proof is easily obtained by substituting x_{i_j} into x in both sides and by checking the fact that the left-hand side is higher by one degree than $p(x; i_1, i_2, \ldots, i_k)$ and $p(x; i_0, i_1, \ldots, i_{k-1})$ in the right-hand side. Note that when we accomplish the recursive process of the lemma, the obtained $p(x; 0, 1, \ldots, n)$ is the desired $P_n(x)$.

Furthermore, we note that the difference

$$p(x; i_0, i_1, \ldots, i_{k-1}, i_k) - p(x; i_0, i_1, \ldots, i_{k-1})$$

Table 4.1 Table for calculating divided differences

x_0	$f[0]$					
		$f[0,1]$				
x_1	$f[1]$		$f[0,1,2]$			
		$f[1,2]$		$\ddots$		
x_2	$f[2]$		$f[1,2,3]$		$\ddots$	
		$f[2,3]$				$\ddots$
x_3	$f[3]$		$f[2,3,4]$			$f[0,1,2,\cdots,n]$
$\vdots$	$\vdots$	$f[3,4]$	$\vdots$		$\iddots$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\iddots$		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\iddots$		
x_{n-1}	$f[n-1]$		$f[n-2,n-1,n]$			
		$f[n-1,n]$				
x_n	$f[n]$					

vanishes at $x = x_{i_0}, x_{i_1}, \ldots, x_{i_{k-1}}$. It implies that the identity

$$\begin{aligned} &p(x; i_0, i_1, \ldots, i_{k-1}, i_k) - p(x; i_0, i_1, \ldots, i_{k-1}) \\ &\quad = (\text{const}) \times (x - x_{i_0})(x - x_{i_1}) \cdots (x - x_{i_{k-1}}) \end{aligned}$$

holds with a certain constant. Since the set $\{(x_j, f_j)\ (j = i_0, i_1, \ldots, i_k)\}$ determines the constant, we denote it by $f[i_0, i_1, \ldots, i_k]$ and call it the kth *divided difference*. Then we have the following:

Lemma 4.2 *The kth divided difference is derived from the* $(k-1)$*th through*

$$f[i_0, i_1, \ldots, i_k] = \frac{f[i_1, i_2, \ldots, i_k] - f[i_0, i_1, \ldots, i_{k-1}]}{x_{i_k} - x_{i_0}}.$$

To sum up, we have the following result.

Theorem 4.2 *The interpolating polynomial on the support points is given by*

$$\begin{aligned} P_n(x) = f[0] + f[0,1](x - x_0) + f[0,1,2](x - x_0)(x - x_1) \\ + \cdots + f[0,1,\ldots,n](x - x_0)(x - x_1) \cdots (x - x_{n-1}). \end{aligned} \tag{4.7}$$

The expression (4.7) is called the *Newton formulation* of an interpolating polynomial. Note that we do not care about the order of the support abscissae. Hence we need not assume $x_0 < x_1 < \ldots < x_n$. When the support points are given, a systematic way to calculate all the necessary divided differences is shown in Table 4.1. It is also clear that when a new data (x_{n+1}, f_{n+1}) is appended, we are only required to calculate $f[n+1], f[n, n+1], \ldots, f[0, 1, \cdots, n, n+1]$ additionally.

Lemma 4.1 also suggests that an evaluation of the intermediate polynomial $p(x; i_0, i_1, \ldots, i_k)$ at $\overline{x}$ can be carried out as

$$p(\overline{x}; i_0, i_1, \ldots, i_k) = \frac{(\overline{x} - x_{i_0})p(\overline{x}; i_1, i_2, \ldots, i_k) - (\overline{x} - x_{i_k})p(\overline{x}; i_0, i_1, \ldots, i_{k-1})}{x_{i_k} - x_{i_0}}.$$

Iterating the procedure, we can evaluate $P_n(\overline{x})$ without construction of the functional form (4.7). This can be described more clearly by introducing the double-indexed array $\{T_{i,k}\ (k = 1, 2, \ldots, i;\ i = 0, 1, 2, \ldots, n)\}$ as follows:

$$T_{i,0} = p(\overline{x}; i) = f_i \quad \text{and} \quad T_{i+k,k} = p(\overline{x}; i, i+1, \ldots, i+k).$$

Then we have the recursive procedure given by

$$\begin{aligned} T_{i,k} &= \frac{(\overline{x} - x_{i-k})T_{i,k-1} - (\overline{x} - x_i)T_{i-1,k-1}}{x_i - x_{i-k}} \\ &= T_{i,k-1} + \frac{(T_{i,k-1} - T_{i-1,k-1})(\overline{x} - x_i)}{x_i - x_{i-k}} \\ &\qquad (k = 1, 2, \ldots, n;\ i = k, k+1, \ldots, n) \end{aligned} \tag{4.8}$$

The procedure is called *Neville's algorithm*, which can be expressed in the table-form given in Table 4.2, appearing similar to Table 4.1, and in Algorithm 4.1.

Algorithm 4.1 Neville's algorithm

Input the value $\overline{x}$, the integer n, the support points $\{(x_i, f_i)\}$
Arrange one-dimensional array $\boldsymbol{t}$
for $j = 0$ **step** 1 **until** n **do** $\boldsymbol{t}_j \leftarrow f_j$
for $k = 1$ **step** 1 **until** n **do**
  for $i = n$ **step** -1 **until** k **do** $\boldsymbol{t}_i \leftarrow \boldsymbol{t}_i - (\boldsymbol{t}_i - \boldsymbol{t}_{i-1}) * (\overline{x} - x_i)/(x_i - x_{i-k})$
Output $\boldsymbol{t}_n$ as $P_n(\overline{x})$

Equi-distant Interpolation

So far, we only assume the support abscissae $\{x_l\}$ are distinct. In the actual situation, however, we often meet the case that they are distributed equi-distantly. That is, they satisfy the condition

$$x_l = x_0 + lh \quad (l = 0, 1, \ldots, n) \tag{4.9}$$

Table 4.2 Table-form of Neville's algorithm

x_0	$f_0 = T_{0,0}$					
		$T_{1,1}$				
x_1	$f_1 = T_{1,0}$		$T_{2,2}$			
		$T_{2,1}$		$\ddots$		
x_2	$f_2 = T_{2,0}$		$T_{3,2}$		$\ddots$	
		$T_{3,1}$				$\ddots$
x_3	$f_3 = T_{3,0}$		$T_{4,2}$			$T_{n,n}$
$\vdots$	$\vdots$	$T_{4,1}$	$\vdots$		$\iddots$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\iddots$		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\iddots$		
x_{n-1}	$f_{n-1} = T_{n-1,0}$	$\vdots$	$T_{n,2}$			
		$T_{n,1}$				
x_n	$f_n = T_{n,0}$					

with a certain positive constant h, which is referred to as the *step-size*. We introduce two difference operators Δ and ∇ associated with h. The *forward difference operator* Δ acts on $\{f_l\}$ as

$$\Delta f_l = f_{l+1} - f_l, \quad \Delta^k f_l = \Delta^{k-1} f_{l+1} - \Delta^{k-1} f_l,$$

while the *backward difference operator* acts as

$$\nabla f_l = f_l - f_{l-1}, \quad \nabla^k f_l = \nabla^{k-1} f_l - \nabla^{k-1} f_{l-1}.$$

Note that we define $\Delta^0 f_l = f_l$ and $\nabla^0 f_l = f_l$.

Then, the expression of the interpolating polynomial given in Theorem 4.2 is much simplified by re-scaling the independent variable with h. When we take x_0 as the origin and θ as the variable, we obtain

$$N_n^F(x_0 + \theta h) \equiv \sum_{l=0}^{n} \binom{\theta}{l} \Delta^l f_0, \tag{4.10}$$

which is called the *forward Newton interpolation formula*. On the other hand, when we take x_n as the origin and θ as the variable, we obtain

$$N_n^B(x_n + \theta h) \equiv \sum_{l=0}^{n} (-1)^l \binom{-\theta}{l} \nabla^l f_n, \tag{4.11}$$

which is called the *backward Newton interpolation formula*. Here the symbol $\binom{x}{k}$ for an integer k and a continuous variable x stands for the generalized binomial coefficient, that is, the k-degree polynomial $x(x-1)\cdots(x-k+1)/k!$.

4.2 Error in Polynomial Interpolation

As we see in the previous section, the interpolating polynomial of the same support points is unique, even if there are different expressions. The problem in the present section is its error, that is, how far it is from the interpolated function.

Given a set of points $\xi_1, \xi_2, \ldots, \xi_m$ on the real line, we shall denote the minimum closed interval including all the points by $J[\xi_1, \xi_2, \ldots, \xi_m]$.

Theorem 4.3 *Assume that the function $f(x)$ is in C^{n+1} and the support points are given by $f_i = f(x_i)$ $(i = 0, 1, \ldots, n)$ upon distinct abscissae $\{x_i\}$. Let $P_n(x)$ be the interpolating polynomial by the support points. Then, for any $\overline{x}$ in the domain of $f(x)$, there exists $\xi \in J[x_0, x_1, \ldots, x_n, \overline{x}]$ depending on $\overline{x}$, which satisfies the equation*

$$f(\overline{x}) - P_n(\overline{x}) = \frac{\omega_n(\overline{x})}{(n+1)!} f^{(n+1)}(\xi). \tag{4.12}$$

Proof We can assume $\overline{x} \neq x_i$ $(i = 0, 1, \ldots, n)$, for otherwise (4.12) is trivial with any ξ. Define $F(x)$ by

$$F(x) = f(x) - P_n(x) - K\omega_n(x)$$

with the constant $K = (f(\overline{x}) - P_n(\overline{x}))/\omega_n(\overline{x})$. Then we have

$$F(x) = 0 \quad \text{for} \quad x = x_0, x_1, \ldots, x_n, \overline{x}.$$

This implies that the C^{n+1}-class function $F(x)$ has at least $(n+2)$ zeros on the interval $J[x_0, x_1, \ldots, x_n, \overline{x}]$. Rolle's theorem deduces that $F'(x)$ has $(n+1)$ zeros on the interval. Then, $F''(x)$ has at least n zeros on the same interval. Repeating the deduction, we can assert that $F^{(n+1)}(x)$ has at least one zero ξ in $J[x_0, x_1, \ldots, x_n, \overline{x}]$. Since $P_n(x)$ and $\omega_n(x)$ are polynomials of degree n and $(n+1)$, respectively, we have $P_n^{(n+1)}(x) = 0$ and $\omega_n^{(n+1)}(x) = (n+1)!$. This means $F^{(n+1)}(\xi) = f^{(n+1)}(\xi) - K(n+1)! = 0$, which concludes $K = f^{(n+1)}(\xi)/(n+1)!$. □

Remark 4.1 Although the theorem appears fine, we must interpret it carefully. First, it is rather hard to know if the interpolated function $f(x)$ is in C^{n+1}. It is an unknown hidden function to be approximated. Thus, it is harder to obtain the functional form of its $(n+1)$th derivative. The right-hand side of (4.12) is an *a-priori* estimate of the error, not *a-posteriori* one. Second, when n becomes large, i.e., more support points are included, the error seems to becoming small, for the factorial $(n+1)!$ tends to

infinity very quickly. However, it is not the case. We never neglect the term $f^{(n+1)}(\xi)$ which may increase its magnitude as $n \to \infty$. Indeed an example which derives an oddly behaving interpolating polynomial is known (see Runge's phenomenon below). Furthermore, you can imagine a hazardous situation when the point $\overline{x}$ is selected outside of $J[x_0, x_1, \ldots, x_n]$. Then, the magnitude of $\omega_n(\overline{x})$ becomes big and it may violate the interpolation property. The use of the interpolating polynomial $P_n(x)$ outside of the interval $J[x_0, x_1, \ldots, x_n]$ must be avoided if possible.

In the case of equidistant support abscissae, the error estimation (4.12) can be simplified.

Corollary 4.1 *Under the same assumption of Theorem 4.3 for f and $\{x_i\}$ satisfying (4.9), there exist $\xi \in J[x_0, x_n, x_0 + \theta h]$ and $\overline{\xi} \in J[x_0, x_n, x_n + \theta h]$ depending on θ which satisfy the equations*

$$f(x_0 + \theta h) - N_n^F(x_0 + \theta h) = h^{n+1} \binom{\theta}{n+1} f^{(n+1)}(\xi) \tag{4.13}$$

and

$$f(x_n + \theta h) - N_n^B(x_n + \theta h) = (-h)^{n+1} \binom{-\theta}{n+1} f^{(n+1)}(\overline{\xi}), \tag{4.14}$$

respectively.

Runge's Phenomenon

To demonstrate the note mentioned in Remark 4.1, we numerically study a simple interpolation problem with equidistant support abscissae. Let $I_{2n}(x)$ be the polynomial of degree $(2n)$ which interpolates the function $f(x) = 1/(1 + 25x^2)$ on the interval $[-1, 1]$ by the support abscissae $x_l = lh$ $(l = 0, \pm 1, \pm 2, \ldots, \pm n)$ with $h = 1/n$. Note that the function $f(x)$ is quite smooth and bounded on the interval. Figure 4.3 depicts the function plot of $f(x)$, $I_8(x)$ and $I_{16}(x)$ (with solid, dotted and dash dot curves, respectively) on the xy-plane. The polynomials $I_8(x)$ and $I_{16}(x)$ certainly interpolate $f(x)$, but the maximum value of the difference from $f(x)$ becomes bigger when n increases. You can easily imagine $\max_{[-1,1]} |f(x) - I_{2n}(x)|$ will explode as $n \to \infty$, even though a certain smaller interval J $(\subset I)$ exists and $I_{2n}(x) \to f(x)$ holds pointwisely on J. This is called Runge's phenomenon named after German mathematician Carl RUNGE, who pointed out the phenomenon.

Lebesgue Constant

Why does Runge's phenomenon happen? At the first glance it contradicts the following theorem due to K. T. W. WEIESRSTRASS.

Weiesrstrass' Approximation Theorem

Assume that f is any continuous function on the finite closed interval $[a.b]$, e.g., *$f \in C[a, b]$. Then for any positive ε there exists a polynomial $p_n(x)$ of degree n, which depends on ε, satisfying*

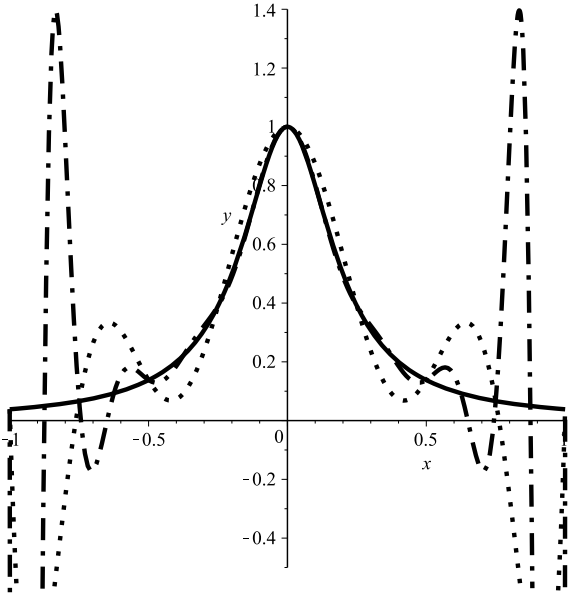


Fig. 4.3 Runge's phenomenon for $f(x) = 1/(1 + 25x^2)$

$$\max_{x\in[a,b]} |f(x) - p_n(x)| \le \varepsilon.$$

Note that the theorem tells neither about a way to calculate the approximating polynomial nor about the equidistant polynomial interpolation. Thus, the phenomenon may happen and can be explained by several ways of higher analysis. Here, we restrict ourselves to describe it briefly. We consider the set of all continuous functions on the interval $[-1, 1]$ and denote it by $C[-1, 1]$ equipped with the norm $\|f\| = \max_{-1\le x\le 1} |f(x)|$. It is known that $C[-1, 1]$ becomes a Banach space with the norm. We define the linear operator Π_n which maps $f \in C[-1, 1]$ to its interpolation polynomial (4.4): $\Pi_n f(x) \ : \ f(x) \to L_n(x) = \sum_{i=0}^{n} f_i\, L_i(x)$. The operator norm of Π_n on $C[-1, 1]$ is given by

$$\|\Pi_n\| = \sup_{\|f\|\le 1} \|\Pi_n f\|.$$

Then, it can be shown the estimation

$$(\|\Pi_n\| - 1)\|f\| \le \sup_{f} \|f - \Pi_n f\| \le (\|\Pi_n\| + 1)\|f\|$$

holds. This implies that the relative interpolation error $\|f - \Pi_n f\|/\|f\|$ has a bound closely equal to $\|\Pi_n\|$. Moreover, it can be proved the identity

$$\|\Pi_n\| = \Lambda_n \equiv \max_{-1\le x\le 1} \sum_{k=0}^{n} |L_i(x)| \tag{4.15}$$

holds. The constant Λ_n, which is determined only by the distribution of the support abscissae (see the definition of $L_i(x)$!), is called the Lebesgue constant. In fact, for the equidistant case the estimation $\Lambda_n \simeq \mathcal{O}(2^n)$ is known as $n \to \infty$. This means there should be an interpolated function in $C[-1, 1]$ whose error with polynomial interpolation explodes when n increases.

The constant Λ_n has another significance in the polynomial interpolation. Suppose the interpolated function $f(x)$ is slightly perturbed to $\widehat{f}(x)$. Since the interpolation operator Π_n is linear, the estimation

$$\|\Pi_n f - \Pi_n \widehat{f}\| \leq \Lambda_n \|f - \widehat{f}\|$$

holds. This means the perturbation error can be magnified by Λ_n in the interpolation result. The polynomial interpolation with a large Λ_n can be said to be *ill-conditioned*.

Further Remarks

We have shown that the equidistant polynomial interpolation has a limit, even though it is easy to handle. Its remedy has several methods. The first one is to look for support abscissae whose Lebesgue constant is small. The Chebyshev interpolation fits to the purpose for functions in $C[-1, 1]$. It employs the Chebyshev points $x_k = \cos\theta_k, \theta_k = (2k-1)\pi/(2n)$ $(k = 1, \ldots, n)$ which are the roots of the Chebyshev polynomial $T_n(x)$. Its Lebesgue constant is known to behave $\log n$ as $n \to \infty$. Other orthogonal polynomials like such as Legendre, Laguerre etc. are useful in the so-called *orthogonal polynomial interpolation*.

The second method is to utilise piece-wise interpolation. That is, we divide the interval of interpolation into several small ones and apply lower degree polynomials to interpolate on each subinterval and join them at the end-points of subinterval as smooth as possible. This leads to *spline interpolation*, which is widely used in science and engineering. The third one is to employ rational functions as the interpolants. When we can obtain the information of derivatives of the interpolated function, they should be included. This is called *Hermite interpolation*. When the interpolated function is assumed to be periodic, interpolation by the trigonometric functions are powerful. Historically, this is developed as *Fourier analysis*. Readers who want to know more about these extensions are recommended to refer to [35].

Sergei N. BERNSTEIN, a Russian mathematician of the 20th century, gave a clever proof of the Weierstrass theorem by employing the series of polynomials of degree n given by

$$B_k^n(x) = \binom{n}{k} \frac{(b-x)^{n-k}(x-a)^k}{(b-a)^n} \quad (k = 0, 1, \ldots, n).$$

For instance, the series of the polynomials in the case of $a = 0, b = 1$ and $k = 4$ are shown in Fig. 4.4. Compare it with Fig. 4.2. Then, you can see much difference between the two bases. A course to employ the Bernstein basis for function approxi-

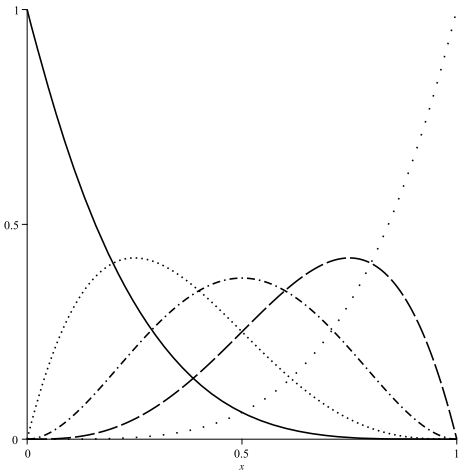


Fig. 4.4 Basis by the Bernstein polynomials

mation obtained the Bézier curves, which are widely applied in CAD (computer-aided design). However, they are no longer an interpolation, for they do not necessarily pass the support points. Refer to [25].

Exercises

4.1. For the Vandermonde matrix appearing in Theorem 4.1, prove that its determinant is equal to $\prod_{j>i}(x_j - x_i)$. Therefore, when all the support abscissae are distinct, the matrix is non-singular.

4.2. Let $L_i(x)$ $(i = 0, 1, \ldots, n)$ be given by (4.2) upon the distinct support abscissae $\{x_i\}$. Show that the identity $\sum_{i=0}^{n} L_i(x) = 1$ holds.

Hint. Try to think of an application of (4.4).

4.3. Prove the identity (4.5).

4.4. When the equidistant support points are given as

x	1.0	2.0	3.0	4.0	5.0
y	3.60	1.80	1.20	0.90	0.72

calculate the interpolation polynomial $P_4(x)$ in the forward and backward Newton formulas and compute the interpolated value $P_4(2.5)$. The support points are derived from the function $f(x) = 3.6/x$. Evaluate $P_4(3.5)$ by Neville's algorithm and compare it with the exact value $f(3.5)$.

4.5. When the interpolated function is a polynomial of degree n given by $a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0$, prove that the nth difference quotient must be equal to a_n whatever distinct support points are selected. Applying this fact, determine the

actual degree of the polynomial whose degree is known to be less than or equal to 5 and support points are given as follows:

x	-2	-1	0	1	2	3
y	-5	1	1	1	7	25

4.6. We are trying to interpolate the Bessel function of order 0 which has the integral expression

$$J_0(x) = \frac{1}{\pi} \int_0^{\pi} \cos(x \sin t)\, \mathrm{d}t$$

by the interpolation polynomial of second degree on the interval $[0, 1]$ with three equidistant support abscissae x_0, x_1 and x_2 of step-size h. Estimate how small h should be taken to obtain the interpolation error to be less than 10^{-6} for arbitrary selection of x_0, x_1, x_2 upon $[0, 1]$. You can apply Theorem 4.3 and the estimation $|J_0''(x)| \leq 1$ for $0 \leq x \leq \pi$.

4.7. For the function $f(x) = \sqrt{x}$, carry out the polynomial interpolation by equidistant support abscissae upon the interval $[0, 1]$ and confirm that a similar phenomenon such as Runge's appears when the number of points is increased.

4.8. One of the typical problems of Hermite interpolation is to find the polynomial $H_n(x)$ of degree $(2n + 1)$ satisfying the condition

$$H(x_i) = f_i, \quad H'(x_i) = f_i' \quad (i = 0, 1, \ldots, n)$$

for the support points $\{(x_i, f_i, f_i')\}$. We assume the interpolated function $f(x)$ and the identities $f_i = f(x_i)$ and $f_i' = f'(x_i)$. Then, prove that the Hermite interpolation polynomial $H_n(x)$ uniquely exists if and only if the support abscissae $\{x_i\}$ are distinct.

Chapter 5
Linear Multistep Methods for ODEs

Linear multistep methods are another representative class of discrete variable methods for ODEs. The class is contrastive with Runge–Kutta methods in the previous chapter. Here we introduce linear multistep methods (LM methods) and discuss their theory about convergency and stability. We will see that LM methods are carefully applied for a practical implementation. Predictor and corrector combination is one of such application.

5.1 Linear Multistep Methods for ODEs

Now we return back to the initial-value problem of ODEs formulated by (3.1) with the same framework of discretization as used there. As explained in Sect. 3.1, the Euler method (3.3), which proceeds the present step-value x_n to the next step-value x_{n+1}, can be improved by extending more past values $\{x_{n-1}, x_{n-2}, \ldots\}$ involved in. To this end, we rearrange the step numbers and introduce the linear multistep method as follows.

Let k be a natural number and introduce in total $(2k+2)$ real constants α_j, β_j $(j = 0, 1, \ldots, k)$ satisfying

$$\alpha_k \neq 0 \quad \text{and} \quad |\alpha_0| + |\beta_0| > 0.$$

We define the linear k-step method as the discrete variable method given by

$$\sum_{j=0}^{k} \alpha_j \, x_{n+j} = h \sum_{j=0}^{k} \beta_j \, f(t_{n+j}, x_{n+j}). \tag{5.1}$$

T. Mitsui and G. Hu, *Numerical Analysis of Ordinary and Delay Differential Equations*,
UNITEXT 145, https://doi.org/10.1007/978-981-19-9263-6_5

When $k = 1$, $\alpha_1 = 1$, $\alpha_0 = -1$, $\beta_1 = 0$ and $\beta_0 = 1$, it only reduces to the Euler method (3.3). In the case $k = 2$, taking

$$\alpha_2 = 1, \quad \alpha_1 = 0, \quad \alpha_0 = -1, \quad \beta_2 = 0, \quad \beta_1 = 2, \quad \beta_0 = 0$$

implies the scheme of the LM method given by

$$\boxed{x_{n+2} = x_n + 2hf(t_{n+1}, x_{n+1}).} \tag{5.2}$$

For $\mathrm{d}x/\mathrm{d}t = f(t, x)$ at $t = t_{n+1}$, the scheme (5.2) employs the replacement of the derivative term in the left-hand side with the central difference $(x_{n+2} - x_n)/2h$ and of the right-hand side term with $f(t_{n+1}, x_{n+1})$. Consequently it is called the *mid-point rule*. Note that we need not only x_0 (the initial value) but also x_1 at the start of the scheme (5.2), for it is a three-term recurrence formula. To this end, we can apply, e.g., the Euler method as $x_1 = x_0 + hf(t_0, x_0)$. Afterwards, the scheme (5.2) gives x_2, x_3, ... one after another.

When β_k is not zero in (5.1), we have to solve the equation, possibly nonlinear,

$$\alpha_k x_{n+k} = h\beta_k f(t_{n+k}, x_{n+k}) + G_n$$

with respect to x_{n+k}. Thus, it is called *implicit*, otherwise it is *explicit*.

Generally a k-step LM method, which becomes a $(k + 1)$-term recurrence formula, requires the step-values $x_1, x_2, \ldots, x_{k-1}$ together with the initial value for its start-up. These values are called the *starting values*, which are conventionally provided from outside of the method (e.g., by an RK method).

Convergency of LMs

For the general formula of the LM scheme (5.1), we first suppose the *localization assumption*. This means that the back values $x_n, x_{n+1}, \ldots, x_{n+k-1}$ in (5.1) are assumed to be all equal to those of the exact solution $x(t)$ of (1.4), i.e., $x_j = x(t_j)$ holds for $j = n, n + 1, \ldots, n + k - 1$.

Definition 5.1 (i) Under the localization assumption, let $\widehat{x}_{n+k}$ denote the numerical solution by (5.1). Then the difference

$$e_n = x(t_{n+k}) - \widehat{x}_{n+k} \tag{5.3}$$

is called the local truncation error of LM.

(ii) A linear multistep method whose order of the local truncation error with respect to the step-size h is more than or equal to one is called consistent.

(iii) Associated with the LM scheme (5.1), we introduce the following algebraic equation:

$$\zeta^k + \alpha'_{k-1}\zeta^{k-1} + \cdots + \alpha'_1\zeta + \alpha'_0 = 0 \quad (\alpha'_j = \alpha_j/\alpha_k). \tag{5.4}$$

If all the roots of the above equation are less than or equal to unity in magnitude and furthermore the roots with the unit modulus are all simple, the method is said to be zero-stable.

The quantity e_n stands for the error generated at the present step-point under the localization assumption. For single-step methods, (5.4) reduces to $\zeta - 1 = 0$. Hence all single-step methods are unconditionally zero-stable. For the mid-point rule, we can deduce the equation $\zeta^2 - 1 = 0$ implies $\zeta = \pm 1$ and hence the mid-point rule is zero-stable, too. The significance of zero-stability is given by the following:

Theorem 5.1 (G. DAHLQUIST, 1956) *A linear multistep method is convergent if and only if it is consistent and zero-stable.*

The counterpart of LM to Theorem 3.1 is:

Theorem 5.2 *The function f of IVP of ODE is assumed to satisfy the Lipschitz condition. Then, a linear multistep method is convergent with $\mathcal{O}(h^p)$ if the following two conditions hold:*

(i) The method is zero-stable.
(ii) For sufficiently small step-size h, its local truncation error (5.3) has the estimation

$$\max_{0 \le n \le N-k} \|e_n\| \le Ch^{p+1}, \tag{5.5}$$

where the constant C is independent on h.

Proof of the above theorems is found in III.4 of [15]. Theorem 5.2 suggests that an analysis for the local truncation error is again crucial in the LM case. Hence, we are going to analyse it.

The localization assumption implies that $\widehat{x}_{n+k}$ fulfills

$$\begin{aligned}\alpha_k \widehat{x}_{n+k} + \sum_{j=0}^{k-1} \alpha_j x(t_{n+j}) &= h\beta_k f(t_{n+k}, \widehat{x}_{n+k}) + h \sum_{j=0}^{k-1} \beta_j f\left(t_{n+j}, x(t_{n+j})\right) \\ &= h\beta_k f(t_{n+k}, \widehat{x}_{n+k}) + h \sum_{j=0}^{k-1} \beta_j x'(t_{n+j}).\end{aligned}$$

For the differentiable function $x(t)$ we introduce the linear operator $\mathcal{L}$ by

$$\mathcal{L}(x, t; h) = \sum_{j=0}^{k} \left[\alpha_j x(t + jh) - h\beta_j x'(t + jh)\right]. \tag{5.6}$$

Then the solution $x(t)$ of IVP satisfies the identity

$$\alpha_k \left(x(t_{n+k}) - \widehat{x}_{n+k}\right) - h\beta_k \left[f(t_{n+k}, x(t_{n+k})) - f(t_{n+k}, \widehat{x}_{n+k})\right] = \mathcal{L}(x, t_n; h),$$

which leads to

$$(\alpha_k I_d - h\beta_k F)\,(x(t_{n+k}) - \widehat{x}_{n+k}) = \mathcal{L}(x, t_n, h), \tag{5.7}$$

where $F = \int_0^1 J\,(t_{n+k}, \theta x(t_{n+k}) + (1-\theta)\widehat{x}_{n+k})\,\mathrm{d}\theta$. As in Theorem 2.2, the symbol J stands for the Jacobian matrix of f with respect to x. When x is differentiable an arbitrary number of times, (5.7) derives the power series expansion of $\mathcal{L}(x, t_n; h)$ with respect to h as

$$\begin{aligned}\mathcal{L}(x, t_n; h) &= \sum_{j=0}^{k}\left(\sum_{q\geq 0}\alpha_j \frac{j^q h^q}{q!}x^{(q)}(t_n) - h\sum_{r\geq 0}\beta_j \frac{j^r h^r}{r!}x^{(r+1)}(t_n)\right)\\ &= \left(\sum_{j=0}^{k}\alpha_j\right)x(t_n) + \sum_{q\geq 1}\frac{h^q}{q!}\left(\sum_{j=0}^{k}\alpha_j j^q - q\sum_{j=0}^{k}\beta_j j^{q-1}\right)x^{(q)}(t_n).\end{aligned}$$

Thus the local order problem reduces to that of how many coefficients of derivatives $x^{(q)}(t_n)$ can vanish in the right-hand side. This implies the p-th order condition of LM as

$$\sum_{j=0}^{k}\alpha_j = 0 \quad \text{and} \quad \sum_{j=0}^{k}\alpha_j j^q = q\sum_{j=0}^{k}\beta_j j^{q-1} \quad (q = 1, 2, \ldots, p). \tag{5.8}$$

Therefore the system of $(p+1)$ linear equations for α_j and β_j given above consists of the p-th order condition.

For the mid-point rule, its parameters satisfy (5.8) with $p = 2$. Thus it is of second order. Generally, since a linear k-step method has $(2k+1)$ free parameters $\{\alpha_j, \beta_j\}$, it can fulfill maximally $p = 2k$ order conditions. However, as shown in Theorem 5.2, the method should be zero-stable. Then we have the following theorem about the attainable order of a linear k-step method with zero-stability.

Theorem 5.3 (First Dahlquist barrier) *Under the constraint of the zero-stability, we have the following. For odd k, we can attain $p = k+1$, whereas for even k, $p = k+2$. Furthermore, when $\beta_k = 0$ (explicit LM), we only attain $p = k$ irrelevant to odd or even k.*

Its proof is indeed not easy. Refer to III.3 of [15].

Adams-Type Linear Multistep Methods

There are several ways to derive a series of LM methods systematically. We explain the representative one. An integration of (3.1) over the interval $[t_{n+k}, t_{n+k-1}]$ yields

$$x(t_{n+k}) - x(t_{n+k-1}) = \int_{t_{n+k-1}}^{t_{n+k}} f(s, x(s))\,\mathrm{d}s. \tag{5.9}$$

The integrand $f(s, x(s))$ of the right-hand side is unknown. However, we have the information of back-values $(t_n, x_n), (t_{n+1}, x_{n+1}), \ldots, (t_{n+k-1}, x_{n+k-1})$, which lead to the approximate values f_{n+j} for $f(t_{n+j}, x(t_{n+j}))$ $(j = 0, 1, \ldots, k-1)$, too. Therefore the unique interpolating polynomial $\varphi(s)$ of $f(s, x(s))$ is available over $[t_n, t_{n+k-1}]$. Most manageable is the backward Newton interpolating formula explained in Sect. 4.1 and we have

$$\begin{aligned}\varphi(t_{n+k-1} + \theta h) = f_{n+k-1} + \theta \nabla f_{n+k-1} + \frac{\theta(\theta+1)}{2!}\nabla^2 f_{n+k-1} + \cdots \\ + \frac{\theta(\theta+1)\cdots(\theta+k-1)}{k!}\nabla^k f_{n+k-1} \quad (-(k-1) \le \theta < 0),\end{aligned} \tag{5.10}$$

where the symbol ∇ denotes the backward difference operator. Substituting $\varphi(s)$ into the integrand $f(s, x(s))$ and replacing $x(t_j)$ with x_j, we obtain

$$\begin{aligned}&x_{n+k} - x_{n+k-1} \\ &= h\int_0^1 \left(f_{n+k-1} + \theta \nabla f_{n+k-1} + \cdots + \frac{\theta(\theta+1)\cdots(\theta+k-1)}{k!}\nabla^k f_{n+k-1} \right) \mathrm{d}\theta.\end{aligned}$$

The integrand merely becomes a polynomial of θ that is easily integrated. Consequently, we have an explicit formula of k-step as

$$x_{n+k} - x_{n+k-1} = h\sum_{j=0}^{k} \gamma_j^* \nabla^j f_{n+k-1}, \tag{5.11}$$

where the coefficients $\{\gamma_j^*\}$ are given by

$$\gamma_j^* = \int_0^1 \frac{\theta(\theta+1)\cdots(\theta+j-1)}{j!}\,\mathrm{d}\theta. \tag{5.12}$$

Moreover, due to the theory developed in Sect. 4.2 its local truncation error is explicitly given as

$$e_n = \gamma_{k+1}^* h^{k+1} x^{(k+2)}(\xi^*) \qquad (t_n < \xi^* < t_{n+k}) \tag{5.13}$$

when the exact solution is of C^{k+2}-class. The scheme, which is naturally explicit and of order k, is often referred to as the k-step Adams–Bashforth method.

In (5.9), let us consider the case that $f(s, x(s))$ is interpolated with f_{n+k}, f_{n+k-1}, $\ldots$, f_n as

$$\begin{aligned}\widehat{\varphi}(t_{n+k-1} + \theta h) = f_{n+k} + (\theta-1)\nabla f_{n+k} + \frac{(\theta-1)\theta}{2!}\nabla^2 f_{n+k} + \cdots \\ + \frac{(\theta-1)\theta\cdots(\theta+k-2)}{k!}\nabla^k f_{n+k} \qquad (-k \le \theta < 1).\end{aligned} \tag{5.14}$$

Then we obtain

$$x_{n+k} - x_{n+k-1} = h \int_0^1 \left(f_{n+k} + (\theta - 1)\nabla f_{n+k} + \cdots + \frac{(\theta - 1)\theta \cdots (\theta + k - 2)}{k!} \nabla^k f_{n+k} \right) \mathrm{d}\theta,$$

which leads to an implicit k-step formula,

$$x_{n+k} - x_{n+k-1} = h \sum_{j=0}^{k} \gamma_j \nabla^j f_{n+k} \tag{5.15}$$

and

$$\gamma_j = \int_0^1 \frac{(\theta - 1)\theta \cdots (\theta + j - 2)}{j!} \mathrm{d}\theta. \tag{5.16}$$

The local truncation error is given as

$$e_n = \gamma_{k+1} h^{k+2} x^{(k+3)}(\xi^*) \qquad (t_n < \xi^* < t_{n+k}) \tag{5.17}$$

when the exact solution is of C^{k+3}-class. The scheme, which is implicit and of order $k + 1$, is often referred to as the k-step Adams–Moulton method.

To sum up, the linear multistep method which satisfies the condition $\alpha_k = 1$, $\alpha_{k-1} = -1$ and other α_j's zero, that is

$$x_{n+k} = x_{n+k-1} + h \sum_{j=0}^{k} \beta_j \, f(t_{n+j}, x_{n+j}), \tag{5.18}$$

is called an Adams-type formula. John Couch ADAMS was an English astronomer in the 19th century.

5.2 Implementation Issues of Linear Multistep Methods

The present section is devoted to several issues to implement the linear multistep methods.

Predictor-Corrector Method

Since the explicit formula is computationally simple but less stable and with big error constant γ_k^*, and the implicit formula is computationally hard but more stable and with small error constant, their combination is actually implemented. This is explained by the fourth-order pair of explicit and implicit Adams formulae as follows. First we obtain the predicted value x_{n+k}^{P} by

$$x_{n+4}^{\mathrm{P}} = x_{n+3} + \frac{h}{24}\left(55 f_{n+3} - 59 f_{n+2} + 37 f_{n+1} - 9 f_n\right) \tag{5.19}$$

with the symbol $f_{n+j} = f(t_{n+j}, x_{n+j})$. This is called the prediction mode (P mode). Then, the evaluation mode (E mode) calculates

$$f_{n+4}^{\mathrm{E}} = f(t_{n+4}, x_{n+4}^{\mathrm{P}}) \tag{5.20}$$

by calling the function f. Next, the corrected value for x_{n+4}^{C} is computed by

$$x_{n+4}^{\mathrm{C}} = x_{n+3} + \frac{h}{24}\left(9 f_{n+4}^{\mathrm{E}} + 19 f_{n+3} - 5 f_{n+2} + f_{n+1}\right), \tag{5.21}$$

which stands for the correction mode (C mode). This is an example of the predictor-corrector (PC) method based on LM formula and its implementation for a single step forwarding is given in Algorithm 5.1. A driver program can repeatedly call the same algorithm for $n = 0, 1, 2, \ldots$ to output x_{n+4} as far as the step-size h is kept constant. On the other hand, by taking x_{n+4}^{C} as a better approximation we can apply (5.20) again to obtain more accurate solution for the implicit formula. This is expressed as PECEC mode and can be realised by a slight modification of Algorithm 5.1. Of course, more EC modes can be supplemented by paying more computational costs of f.

Algorithm 5.1 Single-step integration by the fourth-order Adams PC pair of PEC-mode

Input $t_{n+3}, h, \boldsymbol{x}_{n+3}, \boldsymbol{f}_{n+3}, \boldsymbol{f}_{n+2}, \boldsymbol{f}_{n+1}, \boldsymbol{f}_n$
Arrange vectors $\boldsymbol{x}, \boldsymbol{f}$
$t \leftarrow t_{n+3} + h$
{Prediction:}
$\boldsymbol{x} \leftarrow \boldsymbol{x}_{n+3} + (h/24) * (55 * \boldsymbol{f}_{n+3} - 59 * \boldsymbol{f}_{n+2} + 37 * \boldsymbol{f}_{n+1} - 9 * \boldsymbol{f}_n)$
{Evaluation: Require the calling program to evaluate the rhs function}
$\boldsymbol{f} \leftarrow$ function$(t, \boldsymbol{x})$
{Correction:}
$\boldsymbol{x} \leftarrow \boldsymbol{x}_{n+3} + (h/24) * (9 * \boldsymbol{f} + 19 * \boldsymbol{f}_{n+3} - 5 * \boldsymbol{f}_{n+2} + \boldsymbol{f}_{n+1})$
{Update the step-values for the next-step integration:}
$\boldsymbol{x}_{n+3} \leftarrow \boldsymbol{x}, \boldsymbol{f}_n \leftarrow \boldsymbol{f}_{n+1}, \boldsymbol{f}_{n+1} \leftarrow \boldsymbol{f}_{n+2}, \boldsymbol{f}_{n+2} \leftarrow \boldsymbol{f}_{n+3}, \boldsymbol{f}_{n+3} \leftarrow \boldsymbol{f}$
Output $\boldsymbol{x}$ as $x(t_{n+4})$

A-posteriori Error Estimation

The expression of the local truncation error of an LM method, e.g., given by (5.13) or (5.17), is again not practical, even though it is simpler than the RK case, for the higher derivatives of the exact solution are not readily available. Instead the PC pair given by (5.19) and (5.21) can suggest an implementable way for *a-posteriori* error estimation.

Assume that the exact solution $x(t)$ of the problem is sufficiently smooth and that the localization assumption holds. Then, the local truncation error of the predictor and the corrector is expressed as

$$x(t_{n+4}) - x^{\mathrm{P}}_{n+4} = \gamma^*_5 h^5 x^{(5)}(t_{n+3}) + \mathcal{O}(h^6) = \frac{251}{720} h^5 x^{(5)}(t_{n+3}) + \mathcal{O}(h^6)$$

and

$$x(t_{n+4}) - x^{\mathrm{C}}_{n+4} = \gamma_5 h^5 x^{(5)}(t_{n+3}) + \mathcal{O}(h^6) = -\frac{19}{720} h^5 x^{(5)}(t_{n+3}) + \mathcal{O}(h^6),$$

respectively, provided the corrector equation is solved exactly. Neglect of the higher order terms and subtraction of each term give an approximation

$$x^{\mathrm{C}}_{n+4} - x^{\mathrm{P}}_{n+4} \approx \frac{3}{8} h^5 x^{(5)}(t_{n+3}),$$

which yields

$$x(t_{n+4}) - x^{\mathrm{C}}_{n+4} \approx -\frac{19}{270} \left(x^{\mathrm{C}}_{n+4} - x^{\mathrm{P}}_{n+4} \right). \tag{5.22}$$

That is, the difference between the corrected and the predicted values gives an error estimation of the corrected value. This is an application of **Milne's device** for *a-posteriori* error estimation of PC pairs based on LM methods. Hence, one can expect that the modified value

$$x^{\mathrm{C}}_{n+4} - \frac{19}{270} \left(x^{\mathrm{C}}_{n+4} - x^{\mathrm{P}}_{n+4} \right) = \frac{1}{270} \left(251 x^{\mathrm{C}}_{n+4} + 19 x^{\mathrm{P}}_{n+4} \right)$$

can give a more accurate approximation than x^{C}_{n+4}. The process is called the modification mode (M mode) and can be attached to the PC method with a small computational cost.

Adaptive Step-Size Control

Once an *a-posteriori* error estimation is available, we may think of an adaptive step-size control of a PC method. Contrary to the RK case with the embedded pairs, this is not easy to implement in the LM case. The first big problem is re-evaluation of the back values $x_n, x_{n+1}, \ldots, x_{n+k-1}$ and $f(t_n, x_n)$, $f(t_{n+1}, x_{n+1}), \ldots, f(t_{n+k-1}, x_{n+k-2})$ in (5.1). They are assumed to be approximations along with the equidistant step-points $t_n, \ldots, t_{n+k-1}$ and when the step-size is changed, the back values must be re-evaluated, too. The process together with a strategy of adaptive step-size control costs much and a sophisticated implementation is required. See, e.g., III.7 of [15].

5.3 Linear Stability of Linear Multistep Methods for ODEs

Stability Polynomial of LM Method

We will analyse the stability of linear multistep methods from the same standpoint as for Runge–Kutta methods described in Sect. 3.3. An application of the LM scheme to the test equation of stability $\dfrac{\mathrm{d}x}{\mathrm{d}t} = \lambda x$ ($\mathrm{Re}\,\lambda < 0$) results in

$$\sum_{j=0}^{k} \alpha_j x_{n+j} = z \sum_{j=0}^{k} \beta_j x_{n+j} \quad (z = \lambda h), \tag{5.23}$$

which becomes a (constant-coefficient) linear difference equation and governs the stability through its solution behaviour. Hence, we introduce polynomials $\rho(\zeta)$ and $\sigma(\zeta)$ by

$$\rho(\zeta) = \sum_{j=0}^{k} \alpha_j \zeta^j, \qquad \sigma(\zeta) = \sum_{j=0}^{k} \beta_j \zeta^j,$$

and the characteristic equation of (5.23) is given by

$$\pi(\zeta; z) \stackrel{\text{def}}{=} \rho(\zeta) - z\sigma(\zeta) = \sum_{j=0}^{k} (\alpha_j - z\beta_j)\zeta^j = 0, \tag{5.24}$$

which is a complex-coefficient algebraic equation with respect to ζ of degree k. The polynomial $\pi(\zeta; z)$ is often referred to as the stability polynomial of the LM scheme.

It is known that all the solutions of difference equation (5.23) fulfill the condition $x_n \to 0$ as $n \to \infty$ if and only if all the roots of the stability polynomial satisfy $|\zeta| < 1$. This naturally suggests the definition of the stability region $\mathcal{R}$ of the LM, as follows:

$$\mathcal{R} = \{z \in \mathbb{C} \,:\, \text{all the roots of } \pi(\zeta; z) \text{ satisfy } |\zeta| < 1\}. \tag{5.25}$$

The implication of $\mathcal{R}$ is similar to the RK case and the syntax diagram for linear stability analysis of LM methods is shown in Fig. 5.1. Equation (5.24) of the mid-point rule (5.2) reduces to the quadratic one $\zeta^2 - 2z\zeta - 1 = 0$. Its two roots ζ_1 and ζ_2 keep the identity $\zeta_1\,\zeta_2 = -1$, which means $|\zeta_1||\zeta_2| = 1$. Whenever $z \neq 0$ ($h \neq 0$), the two roots do not coincide in magnitude and for any z the case $|\zeta_1| < 1$ and $|\zeta_2| < 1$ never happens. Thus the stability region of the mid-point rule is empty and the method is *absolutely unstable*.

Schur Algorithm

Our present problem is how to draw the stability region of an LM method on the z-plane. Here we introduce a method based on the Schur algorithm [20]. A complex-coefficient polynomial of degree n

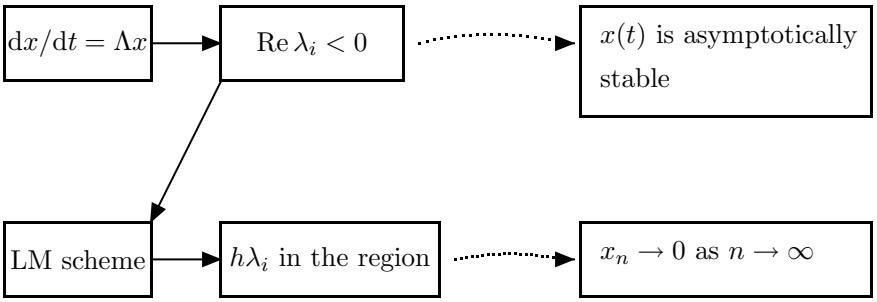


Fig. 5.1 Syntax diagram of linear stability analysis of LM

$$P(\zeta) = c_n\zeta^n + c_{n-1}\zeta^{n-1} + \cdots + c_1\zeta + c_0$$

with $c_n c_0 \neq 0$ is said to be Schur when all of its roots are less than unity in magnitude. The dual polynomial $\widehat{P}(\zeta)$ of $P(\zeta)$ is defined by

$$\widehat{P}(\zeta) = c_0^*\zeta^n + c_1^*\zeta^{n-1} + \cdots + c_{n-1}^*\zeta + c_n^*,$$

where the symbol $*$ denotes the complex conjugate. We denote the polynomial whose coefficients are complex conjugates of $P(\zeta)$ by $P^*(\zeta)$, that is,

$$P^*(\zeta) = c_n^*\zeta^n + c_{n-1}^*\zeta^{n-1} + \cdots + c_1^*\zeta + c_0^*.$$

Then, the identity $\widehat{P}(\zeta) = \zeta^n P^*\left(\dfrac{1}{\zeta}\right)$ is obvious. Noting the fact that the polynomial $\widehat{P}(0)P(\zeta) - P(0)\widehat{P}(\zeta)$ of degree n has the root $\zeta = 0$, the polynomial $P_1(\zeta) \stackrel{\text{def}}{=} \left(\widehat{P}(0)P(\zeta) - P(0)\widehat{P}(\zeta)\right)/\zeta$ becomes of degree $(n-1)$ and the following lemma holds.

Lemma 5.1 *The polynomial $P(\zeta)$ is Schur if and only if $P_1(\zeta)$ is Schur and the condition $|\widehat{P}(0)| > |P(0)|$ (equivalently $|c_n| > |c_0|$) holds.*

Proof can be derived by applying Rouché's theorem in complex analysis (2.7 of [20]).

The analysis of whether $P_1(\zeta)$ is Schur can be carried out by taking its dual polynomial $\widehat{P}_1(\zeta)$ and checking the Schur property of $P_2(\zeta) = \left(\widehat{P}_1(0)P_1(\zeta) - P_1(0)\widehat{P}_1(\zeta)\right)/\zeta$, whose degree reduces by one. By recursively repeating the process, we can obtain a series of conditions which guarantee the original polynomial $P(\zeta)$ is Schur. For example, the one-step implicit Adams method $x_{n+1} - x_n = (h/2)(f_{n+1} + f_n)$ has its stability polynomial as $\pi(\zeta; z) = (1 - z/2)\,\zeta - (1 + z/2)$. Hence the Schur criterion gives the condition $|(1 + z/2)/(1 - z/2)| < 1$, which obviously implies $\mathrm{Re}\, z < 0$. The method, which has the whole left-hand side half

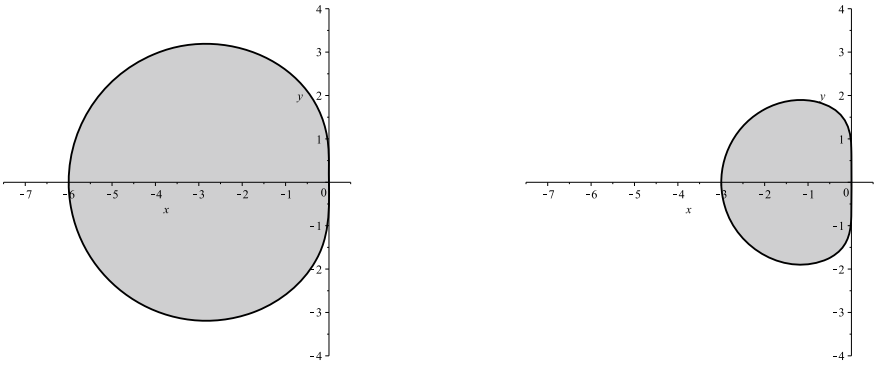


Fig. 5.2 Stability region of implicit Adams methods (left: $k = 2$, right: $k = 3$)

plane of z as its stability region, i.e., which exhibits A-stability (see Sect. 3.3), is often called the *trapezoidal rule*.

The Schur criterion for two- and three-step implicit Adams methods leads to the condition

$$\left|\frac{1}{6}\,|z|^2 - \frac{5}{6}\,\mathrm{Re}\,z + 1\right| > \left|\frac{1}{3}\,|z|^2 - \frac{7}{12}\,z + \frac{5}{12}\,z^* - 1\right|$$

and

$$\left|1 - \frac{3}{2}\,\mathrm{Re}z + \frac{17}{32}\,|z|^2 + \frac{9}{64}\,z^2 + \frac{9}{64}\,z^{*2} - \frac{37}{216}\mathrm{Re}\,z\,|z|^2 + \frac{1}{144}\,|z|^4\right| >$$
$$\left|\frac{11}{144}\,|z|^4 - \frac{101}{432}\,z\,|z|^2 - \frac{65}{432}\,z^*\,|z|^2 + \frac{11}{32}\,|z|^2 + \frac{15}{64}\,z^2 - \frac{9}{64}\,z^{*2} - \frac{1}{4}\,z + \frac{3}{4}\,z^* - 1\right|,$$

respectively and their region of stability is shown in Fig. 5.2.

We can observe that contrary to the RK case the region of the Adams-type method becomes smaller when the order p increases. Also the region of implicit method is generally broader than that of the explicit one of the same order.

Backward Differentiation Formula (BDF)

As we have seen above, Adams-type LM methods have a restriction from the stability point of view. To overcome this, we slightly change the derivation principle of LM methods. We take the interpolation polynomial $p(t)$ of the solution $x(t)$ itself instead of its derivative $x'(t) = f(t, x(t))$. That is, for the interpolation data (or the support points in the terminology of Chap. 4) (t_{n+1}, x_{n+1}), (t_n, x_n), (t_{n-1}, x_{n-1}), ... we introduce the interpolating polynomial of $(k+1)$-th degree

$$p(t) = p(t_{n+1} + \theta h) = \sum_{j=0}^{k} \binom{-\theta}{j} \nabla^j x_{n+1}$$

Table 5.1 Parameters of BDF for $k = 1-6$

k	1	2	3	4	5	6
$\alpha_0^{(k)}$	-1	$\frac{1}{3}$	$-\frac{2}{11}$	$\frac{3}{25}$	$-\frac{12}{137}$	$\frac{10}{147}$
$\alpha_1^{(k)}$		$-\frac{4}{3}$	$\frac{9}{11}$	$-\frac{16}{25}$	$\frac{75}{137}$	$-\frac{24}{49}$
$\alpha_2^{(k)}$			$-\frac{18}{11}$	$\frac{36}{25}$	$-\frac{200}{137}$	$\frac{75}{49}$
$\alpha_3^{(k)}$				$-\frac{48}{25}$	$\frac{300}{137}$	$-\frac{400}{147}$
$\alpha_4^{(k)}$					$-\frac{300}{137}$	$\frac{150}{49}$
$\alpha_5^{(k)}$						$-\frac{120}{49}$
$\beta^{(k)}$	1	$\frac{2}{3}$	$\frac{6}{11}$	$\frac{12}{25}$	$\frac{60}{137}$	$\frac{20}{49}$

and assign the condition $p'(t_{n+1}) = f_{n+1}$. Note that a different notation is employed here from the standard one in (5.1). Since the identity

$$p'(t) = \frac{1}{h}\sum_{j=0}^{k}(-1)^j \left(\frac{\mathrm{d}}{\mathrm{d}\theta}\binom{-\theta}{j}\right) \nabla^j x_{n+1}$$

holds, we arrive at

$$\frac{1}{h}\sum_{j=0}^{k}(-1)^j \left(\frac{\mathrm{d}}{\mathrm{d}\theta}\binom{-\theta}{j}\right)\Bigg|_{\theta=0} \nabla^j x_{n+1} = f_{n+1}.$$

Due to the identity

$$(-1)^j \left(\frac{\mathrm{d}}{\mathrm{d}\theta}\binom{-\theta}{j}\right)\Bigg|_{\theta=0} = \frac{1}{j} \quad (j = 1, 2, \ldots),$$

the scheme

$$\alpha_0^{(k)} x_{n-k+1} + \alpha_1^{(k)} x_{n-k+2} + \cdots + \alpha_{k-1}^{(k)} x_n + x_{n+1} = h\beta^{(k)} f(t_{n+1}, x_{n+1}) \tag{5.26}$$

is obtained for each k. The parameters $\{\alpha_j^{(k)}\}$ and $\beta^{(k)}$ are given in Table 5.1.

The method, which is called the *backward differentiation formula* (BDF), is implicit with respect to x_{n+1} and has the following property:

(i) The k-step BDF is convergent and of k-th order; however, it is not stable for $k \geq 7$.

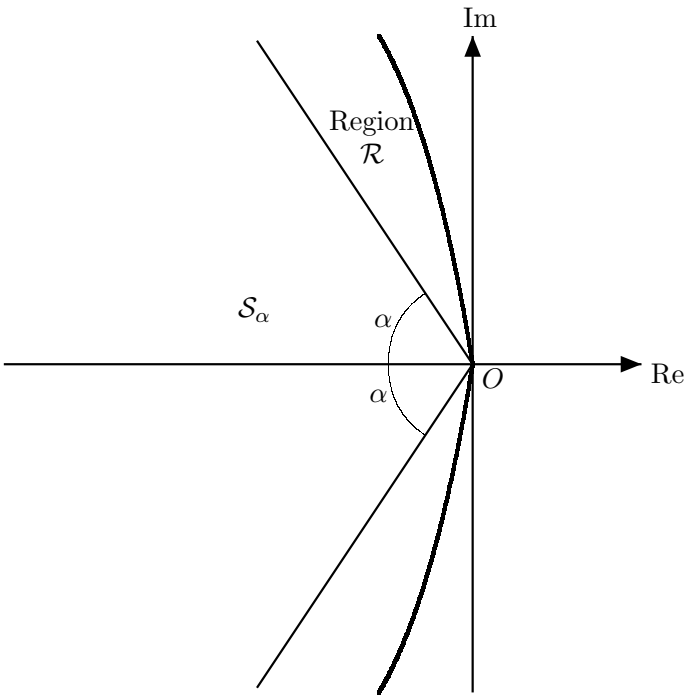


Fig. 5.3 Concept of $A(\alpha)$-stability

(ii) The one-step and two-step BDFs, that is,

$$x_{n+1} - x_n = hf(t_{n+1}, x_{n+1}) \quad \text{and} \quad x_{n+1} - \frac{4}{3}x_n + \frac{1}{3}x_{n-1} = \frac{2}{3}hf(t_{n+1}, x_{n+1})$$

are A-stable. The former is also known as the implicit Euler method.

(iii) BDFs of $k = 3$ to 6, though they are not A-stable, have an angle α (> 0) for $A(\alpha)$-stability.

Here the $A(\alpha)$-stability is introduced in:

Definition 5.2 A convergent linear multistep method is $A(\alpha)$-stable for $0 < \alpha < \pi/2$ if its stability region $\mathcal{R}$ fulfills the condition

$$\mathcal{R} \supset \mathcal{S}_\alpha \stackrel{\text{def}}{=} \{z \in \mathbb{C};\ |\arg(-z)| < \alpha, \quad z \neq 0\}.$$

If the complex parameter λ of the test equation of stability in Sect. 3.3 lies in the sector $\mathcal{S}_\alpha$, a method possessing $A(\alpha)$-stability effectively works like an A-stable method. See Fig. 5.3.

Further Remarks

The introduction of linear multistep methods started when J. C. ADAMS and F. BASHFORTH numerically studied capillary action in the second half of the 19th century. A systematic and comprehensive study was, however, performed by G. DAHLQUIST in the 1950s. Since then many developments have been carried out and now much practical software is available for numerical solution of ODEs. Interested readers are again recommended to refer to, e.g., [9, 15, 16].

Exercises

5.1 Study the order and the zero-stability of the linear two-step method

$$x_{n+2} - \alpha x_{n+1} - \beta x_n = h(\gamma f_{n+1} + \delta f_n)$$

with coefficients α, β, γ and δ when they vary. Furthermore, study the absolute stability of the case $\alpha = 2, \beta = -1, \gamma = 1$ and $\delta = -1$.

5.2 The following four-step implicit method is called Quade's method:

$$x_{n+4} - \frac{8}{19}(x_{n+3} - x_{n+1}) - x_n = \frac{6}{19}h(f_{n+4} + 4f_{n+3} + 4f_{n+1} + f_n).$$

Prove this is zero-stable and derive its order of convergence.

5.3 Prove that for Adams-type LM methods the series of coefficients $\{\gamma_j^*\}$ and $\{\gamma_j\}$ given by (5.12) and (5.16), respectively, fulfill the identities

$$\gamma_j^* + \frac{1}{2}\gamma_{j-1}^* + \cdots + \frac{1}{j+1}\gamma_0^* = 1$$

and

$$\gamma_j + \frac{1}{2}\gamma_{j-1} + \cdots + \frac{1}{j+1}\gamma_0 = 0$$

with $\gamma_0^* = \gamma_0 = 1$. *Hint.* With the definition (5.12), introduce a function $G^*(z)$ by the series $G^*(z) = \sum_{j=0}^{\infty} \gamma_j^* z^j$. Then, confirm the identity

$$G^*(z) = \sum_{j=0}^{\infty} (-z)^j \left(\int_0^1 \binom{-\theta}{j} \mathrm{d}\theta \right) = \frac{-z}{(1-z)\log(1-z)}$$

and derive the desired recursive identity by expanding both sides of

$$G^*(z) \left(\frac{-\log(1-z)}{z} \right) = \frac{1}{1-z}$$

into power series of z.

5.4 Discuss the order of convergence, zero-stability and linear stability of the following two-step method:

$$x_{n+2} - x_n = \frac{h}{3}(f_{n+2} + 4f_{n+1} + f_n).$$

This is one of the Milne–Simpson methods.

5.5 When we replace (5.9) with the integration over $[t_{n+k-2}, t_n]$

$$x(t_{n+k}) - x(t_{n+k-2}) = \int_{t_{n+k-2}}^{t_{n+k}} f(s, x(s))\,\mathrm{d}s$$

and approximate the integrand of the right-hand side by an interpolation polynomial, we obtain the Nyström-type LM formula. Derive several explicit formulae.

5.6 Employing the two polynomials $\rho(\zeta)$ and $\sigma(\zeta)$ introduced in Sect. 5.3, we define another function $\Pi(\xi)$ by

$$\Pi(\xi) = \rho(1+\xi) - \sigma(1+\xi)\log(1+\xi).$$

When the LM method with $\rho(\zeta)$ and $\sigma(\zeta)$ is of p-th order, prove that the estimation $\Pi(\xi) = \mathcal{O}(\xi^{p+1})$ holds if $|\xi|$ is sufficiently small.

5.7 Utilizing a computer for curve plotting, draw the region of absolute stability of the three-step BDF given by

$$x_{n+1} - \frac{18}{11}x_n + \frac{9}{11}x_{n-1} - \frac{2}{11}x_{n-2} = \frac{6}{11}hf(t_{n+1}, x_{n+1})$$

and confirm that it is $A(\alpha)$-stable.

Chapter 6
Analytical Theory of Delay Differential Equations

We will extend differential equations to include a time delay. Then, its theoretical analysis as well as its numerical solution become more difficult than the ODE case. This chapter briefly introduces an analytical study of delay differential equation (DDE) and describes results on the unique existence of its solution as well as stability properties. We will introduce concepts of delay-independent and delay-dependent stability and then derive a practical way to obtain delay-dependent criteria of stability by applying a lovely property of entire functions of complex variables.

6.1 Differential Equation with Delay

Now we turn our attention to a differential equation including the time-delay term as

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = f(t, x(t), x(t-\tau)) \quad (t_0 < t < t_F) \tag{6.1}$$

with the initial condition

$$x(t) = \phi(t) \quad (t_0 - \tau \leq t \leq t_0). \tag{6.2}$$

Here, as in Sect. 3.1, we assume the unknown function $x(t)$ be $[t_0, t_F] \longmapsto \mathbb{R}^d$, but the given function f be $[t_0, t_F] \times \mathbb{R}^d \times \mathbb{R}^d \longmapsto \mathbb{R}^d$ and τ be a positive constant. Note that the initial condition is given with a *function*, not merely at the initial point t_0. When the variable t is in $(t_0, t_0 + \tau)$, the reference value $t - \tau$ varies between $t_0 - \tau$ and t_0. Thus, we need the function to start equation (6.1).

Equation (6.1) is called the delay differential equation (DDE) with constant delay. The function $\phi(t)$ in (6.2) is usually continuous as $[t_0 - \tau, t_0] \longmapsto \mathbb{R}^d$.

T. Mitsui and G. Hu, *Numerical Analysis of Ordinary and Delay Differential Equations*, UNITEXT 145, https://doi.org/10.1007/978-981-19-9263-6_6

Phenomenologically speaking, (6.1) means that the present-time derivative depends not only on the present state but also on the past state. As already stated in Sect. 1.2, the electronic circuit with tunnel-diode is modeled by the differential equation with delay (1.7). Another typical example occurs in biomathematics. When we consider the population variation x and y of two species in a closed environment, we can first model it as the two-dimensional system of ordinary differential equations

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = (\alpha_1 - \gamma x(t) - \beta_1 y(t))\, x(t), \quad \frac{\mathrm{d}y}{\mathrm{d}t}(t) = (-\alpha_2 + \beta_2 x(t))\, y(t),$$

where $x(t)$ and $y(t)$ express the population of the prey and the predator, respectively, and α_i, β_i $(i = 1, 2)$ and γ are positive constants. Although the model, which is often referred to as the Lotka–Volterra model, exhibits an interesting mathematical problem, we can further introduce another assumption that the present proportional growth rate $(\mathrm{d}y/\mathrm{d}t)(t)/y(t)$ of the predator is assumed to depend on the supply of prey x at an earlier time τ units before. Then, we have the system of delay differential equations given by

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = (\alpha_1 - \gamma x(t) - \beta_1 y(t))\, x(t), \quad \frac{\mathrm{d}y}{\mathrm{d}t}(t) = (-\alpha_2 + \beta_2 x(t-\tau))\, y(t).$$

The paper [28] discusses periodic solutions of the system.

In the same way, DDE can mathematically formulate many models in real-world applications, e.g., demography, control theory and engineering. Interested readers can refer to Sect. 2 of [18] or Chap. 1 of [26]. You may think the introduction of a delay term into the equation is a small jump from an ordinary differential equation, but it is not the case. As is shown later, the DDE is much more difficult and rewarding at least in its mathematical analysis.

Method of Steps

We start with a generic idea to solve the delay differential equation with a constant delay (6.1) and (6.2). When $t_0 \le t \le t_0 + \tau$, we have $t_0 - \tau \le t - \tau \le t_0$ so that we can substitute $\phi(t - \tau)$ into the term $x(t - \tau)$ in the right-hand side of (6.1). This implies that on the interval $[t_0, t_0 + \tau]$ we are required to solve the initial-value problem of the ordinary differential equation

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = f(t, x(t), \phi(t-\tau)) \quad \text{and} \quad x(t_0) = \phi(t_0). \tag{6.3}$$

This can be carried out by numerical solution, if necessary. Hence, denoting its solution by $\xi_0(t)$, on the interval $[t_0 + \tau, t_0 + 2\tau]$ we can convert (6.1) into

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = f(t, x(t), \xi_0(t-\tau)) \quad \text{and} \quad x(t_0 + \tau) = \xi_0(t_0 + \tau). \tag{6.4}$$

Again this is an initial-value problem with respect to $x(t)$ and we can obtain its solution $\xi_1(t)$. Repeating the process, we can get the sequence of solutions $\{\xi_\ell(t)\}$, each of which is the solution of

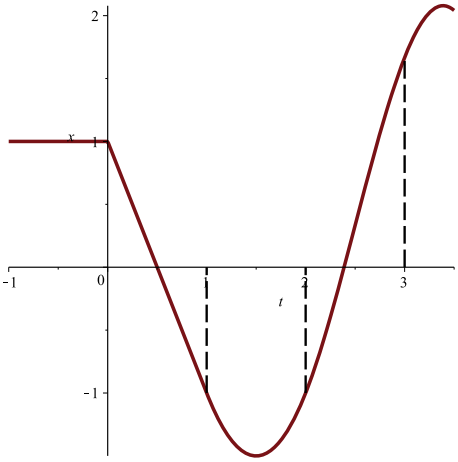


Fig. 6.1 Solution of simple DDE by the method of steps

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = f(t, x(t), \xi_{\ell-1}(t-\tau)) \quad \text{and} \quad x(t_0 + \ell\tau) = \xi_{\ell-1}(t + \ell\tau) \tag{6.5}$$

on the interval $[t_0 + \ell\tau, t_0 + (\ell+1)\tau]$ $(\ell = 0, 1, \ldots)$. On the whole interval we can introduce the function $\Xi(t)$ which joins all the ξ_ℓ's by

$$\Xi(t) = \xi_\ell(t) \quad \text{on} \quad [t_0 + \ell\tau, t_0 + (\ell+1)\tau]$$

as the solution of the original problem (6.1) and (6.2). This is called the *method of steps* or *continuation method*. We emphasize that the core of the method is to solve the initial-value problem of the ordinary differential equation (6.5) recursively.

Example 6.1 When a simple scalar DDE is given as

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = -2x(t-1) \quad \text{with} \quad x(t) = 1 \ (-1 \le t \le 0),$$

its solution by the method of steps is displayed in the (t, x)-plane of Fig. 6.1. In fact, the analytical expression of the solution is

for $0 \le t \le 1$, $x(t) = -2t + 1$;
for $1 \le t \le 2$, $x(t) = 2t^2 - 6t + 3$;
for $2 \le t \le 3$, $x(t) = (-4/3)t^3 + 10t^2 - 22t + 41/3$;
for $3 \le t \le 4$, $x(t) = (2/3)t^4 - (28/3)t^3 + 46t^2 - 94t + (203/3)$; $\ldots$

One can observe derivative discontinuities of the solution at every node point $t = 0, 1, 2, \ldots$. The left- and the right-limits of derivatives are easily calculated as follows:

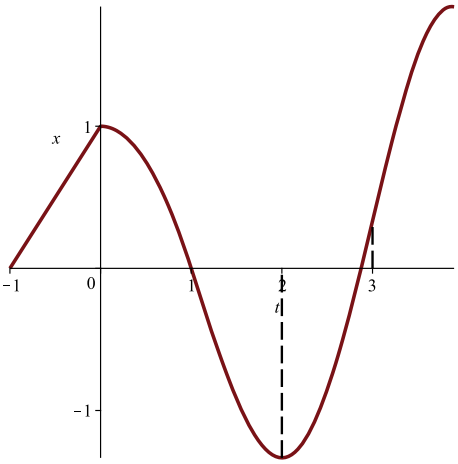


Fig. 6.2 Solution of Example 6.2 by the method of steps

$$\frac{\mathrm{d}x}{\mathrm{d}t}(0)^- = 0,\ \frac{\mathrm{d}x}{\mathrm{d}t}(0)^+ = -2;$$
$$\frac{\mathrm{d}x}{\mathrm{d}t}(1)^- = \frac{\mathrm{d}x}{\mathrm{d}t}(1)^+ = -2;\quad \frac{\mathrm{d}^2x}{\mathrm{d}t^2}(1)^- = 0,\ \frac{\mathrm{d}^2x}{\mathrm{d}t^2}(1)^+ = 4;$$
$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2}(2)^- = \frac{\mathrm{d}^2x}{\mathrm{d}t^2}(2)^+ = 4;\quad \frac{\mathrm{d}^3x}{\mathrm{d}t^2}(2)^- = 0,\ \frac{\mathrm{d}^3x}{\mathrm{d}t^2}(2)^+ = -8;$$
$$\frac{\mathrm{d}^3x}{\mathrm{d}t^3}(3)^- = \frac{\mathrm{d}^3x}{\mathrm{d}t^3}(3)^+ = -8;\quad \frac{\mathrm{d}^4x}{\mathrm{d}t^4}(3)^- = 0,\ \frac{\mathrm{d}^4x}{\mathrm{d}t^4}(3)^+ = 16;\quad \ldots$$

The *lack of continuity* in solution derivatives and their order-to-order recovery by the step-forwarding are common in DDE solutions. This suggests that we should be careful in solving DDEs by the discrete variable methods.

The above-described method of steps also suggests a sufficient condition of unique existence of the solution of (6.1) and (6.2). At each step the initial-value problem must have a unique solution. Hence, we assume that the function $f(t, x, y)$ in (6.1) is Lipschitz continuous with respect to (t, x, y).

Difficulties in DDEs

Moreover, the following example alludes to a theoretical problem to study DDEs.

Example 6.2 We try to solve the same DDE as Example 6.1, but with a different initial function:

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = -2x(t-1) \quad \text{with} \quad x(t) = 1 + t \quad (-1 \le t \le 0).$$

The method of steps is applicable to give analytical expressions of the solution as:

$$\text{for } 0 \le t \le 1,\ x(t) = -t^2 + 1;$$
$$\text{for } 1 \le t \le 2,\ x(t) = \frac{2}{3}t^3 - 2t^2 + \frac{4}{3};$$
$$\text{for } 2 \le t \le 3,\ x(t) = -\frac{1}{3}t^4 + \frac{8}{3}t^3 - 6t^2 + \frac{8}{3}t + \frac{4}{3};$$
$$\text{for } 3 \le t \le 4,\ x(t) = \frac{2}{15}t^5 - 2t^4 + \frac{32}{3}t^3 - 24t^2 + \frac{62}{3}t - \frac{61}{15}; \quad \ldots$$

Figure 6.2 displays the solution in the (t, x)-plane. One can observe the two solutions in Examples 6.1 and 6.2 are completely different, even though at $t = 0$ two solution values coincide. The difference is caused by the difference of initial *functions*. Mathematically, this means that the variety of solution of a DDE is same with that of initial functions and the function space composed by solutions of the DDE is no longer finite-dimensional, contrary to the case of ODE.

Considering the phase portrait analysis of ordinary differential equations described in Sect. 1.2, we draw an intuitive picture of its solution in Fig. 6.3. In the figure, starting from the initial t_0, the $\mathbb{R}^d$-plane is moving along the t-axis. Then, the solution of the initial-value problem of ODEs (1.4) is forming a curve connecting the points $\{x(t);\ t_0 \le t \le t_f\}$ on $\mathbb{R}^d$. Similarly, the solution curve of the initial-value problem of DDEs (6.1) and (6.2) is depicted in Fig. 6.4. Note that the function space

$$C_t(\mathbb{R}^d; \tau) = \{x : (t-\tau, t) \longmapsto \mathbb{R}^d,\ \text{continuous}\}$$
$$\text{with} \quad \|x\|_\tau \stackrel{\text{def}}{=} \max_{t-\tau \le s \le t} \|x(s)\|, \tag{6.6}$$

which contains the solution and is gliding along the t-axis, is of infinite dimension. We will take these facts into consideration when discussing delay differential equations.

We have described the DDE with a constant delay τ and will study conditions of unique existence of the equation. However, in practical applications, a variable delay case $\tau = \tau(t)$ appears and in the next chapter, where numerical solutions will be discussed for DDEs, we try to deal with the variable delay by introducing a formulation of the discrete variable method. Furthermore, more complicated and generalized frameworks of DDEs are possible. The subsection '**Further remarks**' of this chapter will touch on the topic.

6.2 Analytical Solution of DDEs

Here we briefly introduce known results about the analytical solution of DDEs. They are the conditions of unique existence of solution of DDEs and its stability.

Picard Iteration

In the case of the constant-delay problem (6.1) and (6.2), the method of steps can open in a similar way to Sect. 2.1 to study the solution of the problem. At the start we concentrate on (6.3) for the first step $[t_0, t_0 + \tau]$. Suppose that the function f be continuous on the domain

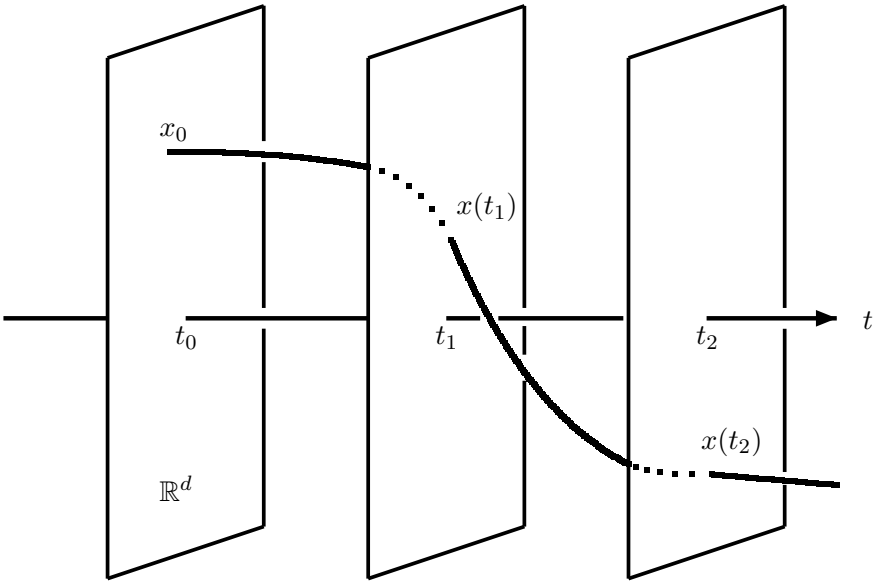


Fig. 6.3 Intuitive picture of IVP of ODEs

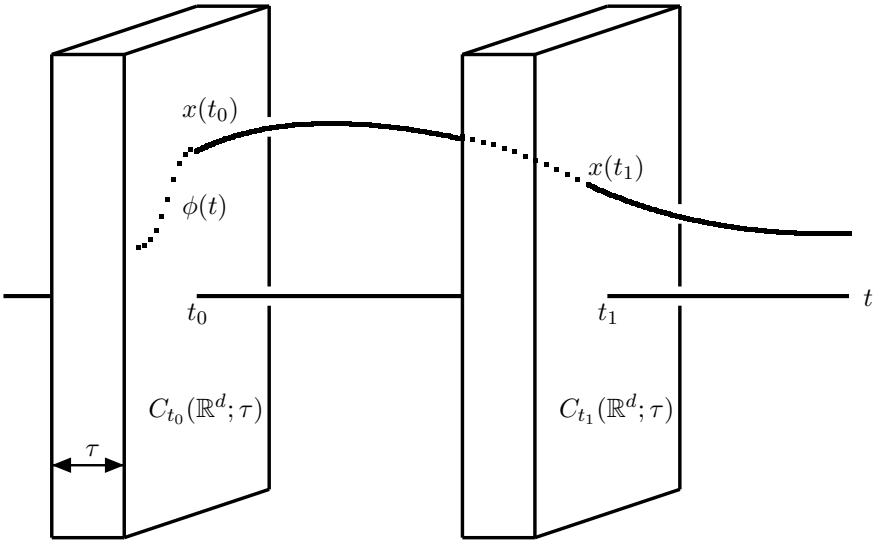


Fig. 6.4 Intuitive picture of IVP of DDEs

$$\mathscr{D}' = \{(t, x, y) \in \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d;\ |t - t_0| \le \tau,\ \|x\| \le R,\ \|y\| \le R'\} \tag{6.7}$$

and bounded $\|f(t, \boldsymbol{x}, y)\| \le M$ there, and furthermore satisfies the Lipschitz condition

$$\|f(t, x, y) - f(t, \tilde{x}, \tilde{y})\| \le L(\|x - \tilde{x}\| + \|y - \tilde{y}\|) \tag{6.8}$$

in $\mathscr{D}'$. Then, the transformation into the integral equation

$$x(t) = \begin{cases} \phi(t) & (t_0 - \tau \le t \le t_0), \\ \phi(t_0) + \displaystyle\int_{t_0}^{t} f(s, x(s), \phi(s - \tau))\,\mathrm{d}s & (t_0 < t < t_0 + \tau) \end{cases} \tag{6.9}$$

readily gives an equivalent form of (6.3). By defining

$$x_0(t) = \begin{cases} \phi(t) & (t_0 - \tau < t \le t_0), \\ \phi(t_0) & (t_0 < t \le t_0 + \tau), \end{cases}$$

the Picard iteration sequence $\{x_n(t)\}$ given by

$$x_n(t) = \begin{cases} \phi(t) & (t_0 - \tau < t \le t_0), \\ \phi(t_0) + \displaystyle\int_{t_0}^{t} f(s, x_{n-1}(s), \phi(s - \tau))\,\mathrm{d}s & (t_0 < t < t_0 + \tau) \\ \qquad (n = 1, 2, 3, \ldots) \end{cases}$$

can be shown to be uniformly convergent and guarantees a unique continuous solution of (6.9), denoted by $\xi_0(t)$, on $[t_0 - \tau, t_0 + \tau']$, where $\tau' \le \tau$, in the same line of Sect. 2.1. Indeed the same assertion seems to be applicable when we replace $\phi(t)$ by $\xi_0(t - t_0)$ and shift the interval to $[t_0, t_0 + 2\tau]$. However, we must be careful that the proof by the Picard iteration has a local nature. That is, the guaranteed interval of existence might be shorter than $[t_0 - \tau, t_0 + \tau]$. A continuation of the solution over $t_0 + \tau'$ requires more study of the solution. Furthermore, even if $\xi_0(t)$ can be substituted in place of $\phi(t)$ on the right-hand side of (6.9), we are not yet sure whether it falls into the same domain $\mathscr{D}'$. It requires another proof.

Unique Existence of Solution

More precise results on unique existence of the (local) solution as well as on its continuous dependence to the initial function can be found in Chap. 2 of [4] or in Chap. 2 of [17]. For reference convenience, we give the following theorem, which is stated as Theorem 2.2.1 in [4], without proof.

Theorem 6.1 (Local existence) *For the delay differential equation with a variable delay*

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = f(t, x(t), x(t - \tau(t))) \quad (t_0 < t < t_F), \quad x(t_0) = x_0$$

we assume that $f(t, x, y)$ is continuous on a certain domain in $[t_0, t_F) \times \mathbb{R}^d \times \mathbb{R}^d$ and locally Lipschitz continuous with respect to (x, y). Moreover, we assume $\tau(t)$ is nonnegative, continuous on $[t_0, t_F)$, $\tau(t_0) = 0$ and, for a certain $\xi > 0$, $t - \tau(t) > 0$ holds on $(t_0, t_0 + \xi]$. Then, there exists a positive δ such that the above initial-value problem has a unique solution on $[t_0, t_0 + \delta)$ and the solution depends continuously on the initial data.

We emphasize again that the Lipschitz continuity of f is a key assumption.

6.3 Linear Stability of DDEs

Asymptotic Stability of DDEs

First we ask what is the counterpart to Definition 2.1 of the ODE case for DDEs. We focus ourselves on the constant-delay case and, to deal with initial functions on $[-\tau, 0]$, introduce the function space $C_0(\mathbb{R}^d; \tau)$ as the special case $t = 0$ of $C_t(\mathbb{R}^d; \tau)$ defined in (6.6). That is, the space $C_0(\mathbb{R}^d; \tau)$ consists of a d-dimensional continuous function $\phi(t)$ on $[-\tau, 0]$ and is equipped with the norm $\|\phi\|_\tau = \max_{-\tau \le s \le 0} \|\phi(s)\|$. Recall Fig. 6.4. As in Sect. 6.1, we consider the delay differential equation (6.1) in $\mathscr{D}'$ together with the initial condition (6.2) and assume f satisfies the Lipschitz condition (6.8) locally. We denote the solution of (6.1) and (6.2) by $x(t; t_0, \phi)$ to stress its dependence on the initial condition. Note that the domain $\mathscr{D}'$ contains the origin $x = 0$. Thus, the origin is said to be an *equilibrium point* of the system (6.1) if $f(t, 0, 0) = 0$ holds for all $t \ge t_0$.

Definition 6.1 The equilibrium point $x = 0$ of (6.1) is:

(i) stable if, for each positive ε, there exists a positive $\delta = \delta(\varepsilon, t_0)$ which satisfies the condition that $\|\phi\|_\tau < \delta$ implies $\|x(t; t_0, \phi)\| < \varepsilon$ for $t \ge t_0$; and
(ii) asymptotically stable if it is stable and moreover there is a positive constant $\gamma = \gamma(t_0)$ fulfilling the condition $x(t; t_0, \phi)$ tends to 0 as $t \to \infty$ for all the initial function ϕ such as $\|\phi\|_\tau < \gamma$.

Several analytical criteria of the stability are known. Interested readers can refer to Chap. VIII of [13] or Chap. 5 of [17]. In the following we focus ourselves to the asymptotic stability of linear delay differential equation.

Linear Stability of DDEs

Considering Definition 6.1 as well as the syntax diagram of stability in the ODE case shown by Fig. 3.3, we ask what stands for the linear test equation in the DDE case. For the constant delay case we naturally arrive at

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = Lx(t) + Mx(t - \tau), \tag{6.10}$$

where L and M are d-dimensional square matrices and τ is a positive constant. Note that the origin is the equilibrium point of (6.10). As the simplest case of (6.10), we

try to consider $\dfrac{\mathrm{d}x}{\mathrm{d}t}(t) = \lambda x(t) + \mu x(t-\tau)$ for the scalar $x(t)$, λ and μ. We assume its solution in the form $x(t) = \exp(zt)$ and substitute it in the equation. Then, we obtain $(z - \lambda - \mu \exp(-z\tau)) \exp(zt) = 0$ and realize that the root of the nonlinear equation

$$z - \lambda - \mu \exp(-z\tau) = 0$$

gives a solution of the DDE in the form $\exp(zt)$. Hence the condition for λ, μ and τ to give z satisfying $\mathrm{Re}\, z < 0$ becomes a sufficient one of (linear) stability. A serious difference from the ODE case is that we are not able to assume that both L and M in (6.10) reduce to scalar constants, for a simultaneous triangularization of the matrices is generally not guaranteed. That is, in general there is no nonsingular matrix S which reduces $S^{-1}LS$ and $S^{-1}MS$ to triangular form. Henceforth we are required directly to seek conditions for asymptotic stability of the solution of (6.10).

Fortunately, a computable condition of asymptotic stability is known. Let $z \in \mathbb{C}$ and introduce the equation

$$\det\left[zI - L - M\exp(-z\tau)\right] = 0 \tag{6.11}$$

for (6.10) through the matrix determinant. It is called the *characteristic equation* of DDE (6.10) and its root is the *characteristic root*. Note that it is no longer an algebraic equation, so the number of roots may be infinite, even though they are countable. The characteristic roots have a similar significance as in the ODE case described in Sect. 2.3. In fact, Sect. 1.4 of [17] tells us the following. As we observe above, when L and M are merely scalar numbers in (6.10), we suppose that z is a root of multiplicity m of the characteristic equation (6.11). Then each of the functions $t^k \exp(zt)$ $(k = 0, 1, 2, \ldots, m-1)$ is a solution of (6.10). Since (6.10) is linear, any finite sum of such solutions is also a solution. Infinite sums are also solutions under suitable conditions to ensure convergence. The idea can be extended to the matrix case.

Henceforth we attain the following criterion of asymptotic stability.

Lemma 6.1 *The trivial solution $x(t) = 0$ of (6.10) is asymptotically stable if and only if the characteristic equation (6.11) has no zeros in the right half-plane $\mathbb{C}^+ \stackrel{\text{def}}{=} \{s \in \mathbb{C};\ \mathrm{Re}\, s \geq 0\}$.*

Therefore, we can take the condition that all the characteristic roots have negative real part as the necessary and sufficient condition of the asymptotic stability of (6.10). This implies it also becomes the condition C_P in the syntax diagram of stability.

Some Knowledge from Linear Algebra

Let $\sigma(A)$ and $\rho(A)$ denote the spectrum and the spectral radius, respectively, of d-dimensional matrix A. Moreover, the symbol $\lambda_j(A)$ will be employed to specify the jth eigenvalue of A, hence $\lambda_j(A) \in \sigma(A)$ and $\rho(A) = \max_j |\lambda_j(A)|$. Hereafter we adopt a natural norm for matrices, that is, the following identity holds for a d-dimensional matrix A:

$$\|A\| = \sup_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\|A\boldsymbol{x}\|}{\|\boldsymbol{x}\|}.$$

When we employ the p-norm introduced in Remark 1.1 to $\mathbb{R}^d$, for $p = 1, 2$ and ∞ we can directly calculate the natural matrix norm of A as follows:

$$\|A\|_1 = \max_j \sum_{i=1}^{d} |a_{ij}|, \quad \|A\|_2 = \sqrt{\rho(A^{\mathsf{T}}A)}, \quad \|A\|_\infty = \max_i \sum_{j=1}^{d} |a_{ij}| \tag{6.12}$$

Delay-Independent and Delay-Dependent Criteria of Linear Stability
Now the problem of linear stability analysis becomes one of looking for conditions that all the characteristic roots of (6.11) have negative real part. Since the characteristic equation includes the delay τ as a parameter, a criterion depending on τ is desirable. However, a major part of the established results is delay-independent, because of difficulty in analysis of delay-dependent case. For instance, the authors of the present volume derived the following result [22].

Theorem 6.2 *If for $\xi \in \mathbb{C}$ satisfying $|\xi| = 1$ the eigenvalue estimation $\operatorname{Re}\lambda_i[L + \xi M] < 0$ holds for every $i = 1, \ldots, d$, then the system (6.10) is asymptotically stable.*

Several delay-independent stability criteria can be found in [4], too.

Here, we turn our interest to delay-dependent stability criteria and present a computable approach for asymptotic stability of test DDE system.

Let $P(z;\tau)$ be equal to $\det\left[zI - L - M\exp(-z\tau)\right]$ in (6.11) with the parameter τ and rewrite the characteristic equation (6.11) as $P(z;\tau) = 0$. Then, we have

Theorem 6.3 *Let z be an unstable characteristic root of (6.11) of system (6.10), i.e., $\operatorname{Re} z \geq 0$, then the estimation*

$$|z| \leq \beta \stackrel{\text{def}}{=} \|L\| + \|M\| \tag{6.13}$$

holds.

Proof Since we are discussing an unstable root, we assume $\operatorname{Re} z \geq 0$ throughout the proof. Introduction of

$$W(z) = L + M\exp(-\tau z)$$

rewrites the characteristic equation as

$$P(z;\tau) = \det\left(zI - W(z)\right) = 0.$$

This implies the z is an eigenvalue of the matrix $W(z)$ and there exists an integer j $(1 \leq j \leq d)$ such that

$$z = \lambda_j \quad \text{where} \quad \lambda_j \in \sigma(W(z)).$$

Algorithm 6.1 Algorithm to distinguish delay-dependent stability of DDE system

Input matrices L and M and the delay τ of (6.10),
sufficiently large $n \in \mathbb{N}$, error tolerances δ_1, δ_2;
Arrange the half-disk D_β in $\mathbb{C}$, the flag `stable`;
{Initial step:}
Calculate $\|L\|$ and $\|M\|$;
Calculate $\beta \leftarrow \|L\| + \|M\|$;
Determine the half-disk D_β and its boundary Γ_β by β;
{First step:}
Distribute n node points $\{z_j\}$ $(j = 1, \ldots, n)$ on Γ_β;
Evaluate $P(z_j; \tau)$ by computing the determinant

$$P(z_j; \tau) = \det[z_j I - L - M \exp(-\tau z_j)];$$

{Second step:}
Check whether $|P(z_j; \tau)| < \delta_1$ holds for each j;
If yes, `stable` $\leftarrow$ NO and terminate the algorithm
otherwise, go to the next step;
{Third step:}
Check whether $|\triangle_{\Gamma_\beta} \arg P(z; \tau)| < \delta_2$ holds by calculating $\arg P(z_j; \tau)$;
If yes, `stable` $\leftarrow$ YES, otherwise `stable` $\leftarrow$ NO;
Output `stable`

Thus we can estimate

$$|z| = |\lambda_j| \leq \|W(z)\| = \|L + M \exp(-\tau z)\| \leq \|L\| + \|M\| = \beta.$$

□

Theorem 6.3 means that there is a bounded region in the right-half complex plane $\mathbb{C}^+$ which includes *all* the unstable characteristic roots of (6.11). Hence, we give the following:

Definition 6.2 Assume that the conditions of Theorem 6.3 hold and the bound β is obtained. The closed half-disk D_β in the z-plane given by

$$D_\beta = \{z : \mathrm{Re}\, z \geq 0 \quad \text{and} \quad |z| \leq \beta\}$$

is said to be the region of instability of (6.10). Moreover, the boundary of D_β is denoted by Γ_β (see Fig. 6.5).

The Definition provides the region of instability D_β which includes all the unstable characteristic roots of (6.10). Thus, by applying the *argument principle* of complex functions (see, e.g., [8]) we can give a necessary and sufficient condition of asymptotic stability of DDEs (6.10). Hereafter, for a positively-oriented simple closed contour $\Gamma \in \mathbb{C}$, we introduce the symbol $\triangle_\Gamma \arg f(z)$ as the change of the argument of complex function $f(z)$ along Γ.

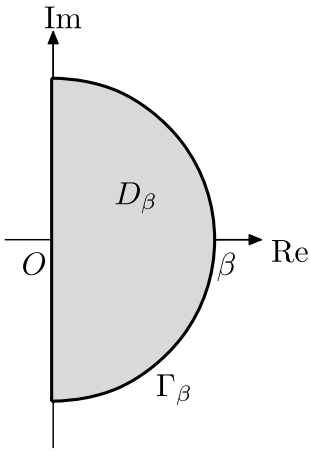


Fig. 6.5 Region of instability D_β

Theorem 6.4 *Assume that the conditions of Theorem 6.3 hold and the bound β is obtained. The DDE system (6.10) is asymptotically stable if and only if the two conditions*

$$P(z;\tau) \neq 0 \quad \textit{for} \quad z \in \Gamma_\beta \tag{6.14}$$

and

$$\triangle_{\Gamma_\beta} \arg P(z;\tau) = 0 \tag{6.15}$$

hold on the boundary of the region of instability D_β.

Proof Our analysis up to here guarantees that all the unstable characteristic roots, if they exist, must be included in the region of instability D_β. Due to the definition, note that the complex function $P(z;\tau)$ has no finite pole in the z-plane. By the argument principle of complex functions, the condition (6.15) implies that there are no roots of $P(z;\tau)$ within the region D_β. Hence, the two conditions (6.14) and (6.15) show that D_β together with its boundary does not include any characteristic root. And vice versa. □

Therefore, we can describe an algorithm to distinguish delay-dependent stability of DDE system based on Theorem 6.4. See Algorithm 6.1. Note that for computational efficiency the algorithm evaluates the determinant $P(z_j;\tau)$ by the elementary row (or column) operations and avoids the Laplace expansion which develops a matrix determinant by calculating all of its sub-determinants. Further results along this line can be found in [21]. Our way of analysis will be applied to the stability of numerical solutions of DDEs in Sect. 7.3.

In closing the present section, we stress that approximation of characteristic roots of (6.11) is significant in practical application of DDEs to engineering problems, particularly since the characteristic equation is a transcendental one. Many authors discuss the problem and interested readers can refer to, e.g., [6, 7, 24, 29].

Further Remarks

Together with DDEs, we can also study another type of differential equation including another time-delay term

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = g\left(t, x(t), x(t-\tau), \frac{\mathrm{d}x}{\mathrm{d}t}(t-\tau)\right) \quad (t_0 < t < t_F), \tag{6.16}$$

where $g : [t_0, t_F] \times \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \longmapsto \mathbb{R}^d$. Equation (6.16) is traditionally called a neutral delay-differential equation (NDDE). When the initial condition (6.2) is imposed on an NDDE (6.16), at least its C^1-continuity on the same interval is assumed and the condition $\dfrac{\mathrm{d}x}{\mathrm{d}t}(t) = \phi'(t)$ is imposed on the interval, too.

In a very generalized framework, a DDE or NDDE can be regarded as a class of retarded functional differential equations (RFDEs), which further includes a delay integro-differential equation

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = f\left(t, x(t), \int_{t-\sigma(t)}^{t-\tau(t)} K(t, t-s, x(s))\,\mathrm{d}s\right).$$

Such combining of several classes of differential equation enables us to give a unified theory about unique existence, continuous dependency on the initial condition and stability analysis of its solution. The functional differential equation framework can cover more complicated delay cases; multiple constant delay case $[f(t, x(t), x(t-\tau_1), \ldots, x(t-\tau_m))$ in (6.1)], multiple variable delay case $[f(t, x(t), x(t-\tau_1(t)), \ldots, x(t-\tau_m(t)))$ in (6.1)] and the case when the delays is vanishing as $\lim_{t\to\infty} \tau_j(t) \to 0$. Since they cannot be covered in the present book, interested readers can refer to other textbooks of DDE theory, e.g., [17, 26].

Exercises

6.1 Solve the scalar delay differential equation

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = \alpha\, x(t-1) \quad (t > 0), \quad x(t) = \phi(t) \quad (-1 \le t \le 0)$$

by the method of steps when $\phi(t) = 1$. Here α is a non-zero constant.

6.2 In the previous problem we change the initial function and specify the parameter α as follows:

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = -\frac{\pi}{2}\, x(t-1) \quad (t > 0), \quad x(t) = \phi(t) \stackrel{\text{def}}{=} \cos\left(\frac{\pi}{2}t\right) \quad (-1 \le t \le 0).$$

Determine its solution.

6.3 In the previous two problems, we try to extend the initial function $\phi(t)$ as it is beyond the interval of the defining domain. That is, on $[-1, 1]$ $\phi(t) = 1$ for

6.1 and $\phi(t) = \cos\left(\frac{\pi}{2}t\right)$ for 6.2. Then, confirm on the extended interval the function $\phi(t)$ satisfies the differential equation in 6.2, while it does not in 6.1.

6.4 Given a simple scalar DDE $\frac{\mathrm{d}x}{\mathrm{d}t}(t) = -x(t-\tau)$, derive its characteristic equation and give the condition of τ for its asymptotic stability. To make a guess, try to solve the equations $z + \exp(-z) = 0$ and $z + \exp(-2z) = 0$ for $z \in \mathbb{C}$.

6.5 Given the scalar DDE

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = -x(t) + \mu x(t-1)\ (t > t_0), \quad x(t) = \phi(t)\ (t_0 - 1 \le t \le t_0),$$

solve the problem by the method of steps for the cases of $\mu = 1/2$ and 2, and confirm their solution behaviour as $t \to \infty$. The characteristic equation becomes $z + 1 - \mu e^{-z} = 0$ and it is known the condition of asymptotic stability is equivalent to that the inequality $\arccos(1/\mu) - \sqrt{\mu^2 - 1} > 0$ holds (9.2 of [4]).

6.6 Prove the identities (6.12) about the natural matrix norm.

6.7 Assume that the complex function $f(z)$ is the polynomial given by

$$f(z) = (z - \xi_1)(z - \xi_2)\cdots(z - \xi_n)$$

where $\{\xi_i\}$ are all distinct. By applying the residue theorem, calculate the contour integral $\frac{1}{2\pi\,\mathrm{i}}\oint_C \frac{f'(z)}{f(z)}\,\mathrm{d}z$ where C is the circle given by $C = \{z;\ z = R\exp(\mathrm{i}\,\theta)\ (0 \le \theta \le 2\pi)\}$ and $R > \max|\xi_j|$ is satisfied. Hence, in this case the contour integral is equal to $\triangle_C \arg f(z)$.

6.8 Determine the asymptotic stability of the linear system (6.10) with $\tau = 1$, that is

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = Lx(t) + Mx(t-1)$$

in the following two cases.

Case (i)

$$L = \begin{bmatrix} 0.9558 & -2.0549 \\ 1.1432 & -0.5033 \end{bmatrix} \quad \text{and} \quad M = \begin{bmatrix} 0.293 & -0.464 \\ -0.7762 & 0.0725 \end{bmatrix}$$

Case (ii)

$$L = \begin{bmatrix} 0.6294 & -0.746 \\ 0.8116 & 0.8268 \end{bmatrix} \quad \text{and} \quad M = \begin{bmatrix} 0.2647 & -0.443 \\ -0.8049 & 0.0938 \end{bmatrix}$$

Chapter 7
Numerical Solution of DDEs and Its Stability Analysis

We will discuss application of discrete variable methods to delay-differential equations (DDEs). Since it is hard to implement the class of linear multistep methods for DDEs, we will restrict our description to Runge–Kutta methods and introduce their continuous extension to cope with DDEs. As in the case of ODEs, the stability of Runge–Kutta methods is also significant when they are applied to DDEs. Following the stability analysis of analytical solution of DDEs given in the preceding chapter, we will develop its counterpart of numerical solutions and show examples of delay-dependent stability.

7.1 Numerical Solution of DDEs

In Example 6.1 of Sect. 6.1 we observe that the discontinuity of solution derivatives propagates forwardly as the method of steps proceeds, even though lower order derivatives recover their continuity at the step-point one-by-one. The phenomenon is known to be common in the solution of DDEs. This implies that it is difficult to discuss the order of convergence of numerical solutions over the whole interval of integration. Note that our discussion about convergency of DVM for ODEs, e.g., in Sect. 3.1, assumes a sufficiently smooth solution of the differential equation on the whole interval.

Taking the formulation (6.5) into account, we can assert the following in the constant-delay case.

Proposition 7.1 *Assume that the function f of (6.1) is of C^p-class and satisfies the global Lipschitz continuity on $[t_0, t_F] \times \mathbb{R}^d \times \mathbb{R}^d$. When we apply a discrete variable method of order p to (6.1) and (6.2) with the step-size $h = \tau/m$ ($m \in \mathbb{N}$) from t_0 to $t_0 + \ell\tau$, there exists an m_0 and a constant C which fulfills*

T. Mitsui and G. Hu, *Numerical Analysis of Ordinary and Delay Differential Equations*, UNITEXT 145, https://doi.org/10.1007/978-981-19-9263-6_7

$$\max_{1\le n\le \ell m} \|x_n - x(t_n)\| \le Ch^p$$

if $m \ge m_0$.

Note that the error only at the step-points is estimated. Hereafter, we will restrict ourselves to concentrate this sort of local convergence for discrete variable methods applied to DDEs.

Numerically Solving DDEs

We are going to apply discrete variable methods, which have been developed for ODEs, to DDEs. The first difficulty is how to handle the delayed values in the numerical scheme. Suppose we try to apply an s-stage RK method formulated in (3.6) to the DDE (6.1) through the method of steps, then we are required to evaluate the intermediate values

$$X_i = x_n + h\sum_{j=1}^{i-1} a_{ij}\, f\left(t_n + c_j\, h,\ X_j,\ x(t_n + c_j\, h - \tau)\right)$$

for the ith stage at the nth step-point. Of course we assume $c_1 = 0$ and $c_i = \sum_{j=1}^{i-1} a_{ij}$ $(i = 2, \ldots, s)$. The question is how to evaluate or obtain $x(t_n + c_j\, h - \tau)$ for every j approximately.

If we can take the step-size h of DVM to be integral fraction of τ as $h = \tau/(\text{natural number})$ and employ the constant step-size strategy, the required values are already computed along with the RK process. However, this does not hold in the case of the variable step-size strategy as well as in the variable delay case $\tau = \tau(t)$. For instance, the following example implies how complicated a variable delay case is even in formulation of the method of steps.

Example 7.1 (Example 6.3.4 of [4])

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = \lambda\frac{t-1}{t}x\,(t - \log(t) - 1)\cdot x(t), \quad t \ge 1 \quad \text{and} \quad x(t) = 1, \quad 0 \le t \le 1, \tag{7.1}$$

where λ is a non-zero constant. First, note that $t - \log(t) - 1 \le t$ holds for $t \ge 1$. Hence, this is indeed a delayed argument. Let T_1 be the (unique) solution of the equation $t - \log(t) - 1 = 1$ for $t > 1$. Then, for $1 \le t \le T_1$ we have the initial-value problem

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = \lambda\frac{t-1}{t}x(t) \quad \text{with} \quad x(1) = 1,$$

whose solution is $x(t) = \exp(\lambda(t - \log(t) - 1))$ and $x(T_1) = e^{\lambda}$. On the next interval over T_1, we first solve the equation $t - \log(t) - 1 = T_1$ to obtain its solution T_2. For $T_1 \le t \le T_2$, the problem turns out to be

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = \lambda \frac{t-1}{t} \exp\left(\lambda(t - \log(t) - 1 - \log(t - \log(t) - 1) - 1)\right) \cdot x(t)$$
$$\text{with} \quad x(T_1) = e^{\lambda}.$$

This is certainly a scalar variable-coefficient linear ODE whose solution can be expressed as

$$x(t) = \exp\left\{\lambda\left(\int_{T_1}^{t} \frac{s-1}{s} \exp\left(\lambda(s - \log s - \log(s - \log s - 1) - \log\lambda - 2)\right) \mathrm{d}s + 1\right)\right\}.$$

For the third interval, we solve the equation $t - \log(t) - 1 = T_2$ to obtain T_3 and solve the initial-value problem formulated according to (7.1) on $[T_2, T_3]$. However, as one can easily imagine, it is hard to carry out in practice.

When we are given a DDE with variable delay

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(t, x(t), x(t - \tau(t))) \quad (t \geq t_0) \tag{7.2}$$

and the initial condition

$$x(t) = \phi(t) \quad (t_0 - \tau(t_0) \leq t \leq t_0), \tag{7.3}$$

Example 7.1 suggests the following.

(1) The method of steps is theoretically possible, but practically insufficient and a numerical solution should be called for.
(2) The step sequence $t_0 < T_1 < T_2 < \cdots$ can be assigned by solving the equation $T_i - \tau(T_i) = T_{i-1}$ recursively. This must derive non-equidistant step-sizes $T_i - T_{i-1}$.

Note that up to here we use the words 'step' and 'step-size' in the method of steps differently from the terminology of discrete variable methods. To avoid a confusion, hereafter we call them the interval and the interval-length of the method of steps. Therefore, we are required to cope with the initial-value problem

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(t, x(t), \Xi(t - \tau(t))) \quad \text{and} \quad x(T_{\ell-1} + \tau(T_{\ell-1})) = \xi_{\ell-1}(T_{\ell-1} + \tau(T_{\ell-1})) \tag{7.4}$$

on the interval $[T_{\ell-1}, T_\ell]$ for obtaining $\xi_\ell(t)$ in place of (6.5). As before, the function $\Xi(t)$ is the joining of $\{\xi_\ell(t)\}$. Note that the interval-length $T_\ell - T_{\ell-1}$ is varying with ℓ and we are faced with the difficulty already described in the present section.

7.2 Continuous Extension of Runge–Kutta Methods for DDEs

To overcome the difficulties explained in the previous section, we introduce the *continuous Runge–Kutta method*, a modified version of (3.6) to (3.1).

Continuous Extension in the ODE Case
Returning back to the initial-value problem of ODEs with the same notations in Sect. 3.1, we slightly rewrite (3.6) to

$$X_{n,i} = x_n + h \sum_{j=1}^{i-1} a_{ij}\, f\left(t_n + c_j\, h, X_{n,j}\right), x_{n+1} = x_n + h \sum_{i=1}^{s} b_i\, f\left(t_n + c_i\, h, X_{n,i}\right) \tag{7.5}$$

to emphasize the dependency of the intermediate values on the step-point. Then, we arrange a set of s polynomials $w_i(\theta)$ $(i = 1, 2, \ldots, s)$ of variable θ which satisfy the condition

$$w_i(0) = 0, \quad w_i(1) = b_i \tag{7.6}$$

and, in addition to x_n and x_{n+1}, we introduce

$$\psi_n(t_n + \theta\, h) = x_n + h \sum_{i=1}^{s} w_i(\theta) f(x_n + c_i\, h, X_{n,i}) \quad (0 \le \theta \le 1) \tag{7.7}$$

as an approximation of $x(t_n + \theta\, h)$. Since $\psi_n(t_n) = x_n$ and $\psi_n(t_{n+1}) = x_{n+1}$ hold, $\psi_n(t)$ is a polynomial interpolation of x_n and x_{n+1} upon $[t_n, t_{n+1}]$. We call the combination of (7.5) and (7.7) the continuous Runge–Kutta method and the function $\Psi(t)$ defined by

$$\Psi(t) = \psi_n(t) \quad \text{for} \quad t \in [t_n, t_{n+1}] \quad (n = 0, 1, \ldots)$$

the *continuous extension* of the RK. Historically, it was called the dense output of the RK, too. The Runge–Kutta method given by (7.5) is often called the underlying RK.

Our next problem is how to choose the polynomials $\{w_i(\theta)\}$ associated with the parameters of RK. The simplest way is to select $w_i(\theta) = b_i\theta$, but this derives

$$\psi_n(t_n + \theta h) = (1 - \theta)x_n + \theta x_{n+1},$$

which is only the linear interpolation between x_n and x_{n+1} and does not reflect the accuracy of the underlying RK approximation. Motivated by this, we give the following:

Definition 7.1 Given the (local) solution of the initial-value problem

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(t, x), \quad x(t_n) = x_n$$

by $\varphi_n(t)$, the integer q satisfying the estimation

$$\sup_{\theta\in[0,1]} \|\psi_n(t_n+\theta h)-\varphi_n(t_n+\theta h)\| = \mathcal{O}(h^{q+1})$$

is said to be the uniform order of the continuous Runge–Kutta method which generates $\{\psi_n(t_n+\theta h)\}$.

The inequality $1 \le q \le p$ is obvious, where p stands for the order of convergence of the underlying RK method. For the Heun method of second order, we can attach its continuous extension by taking

$$w_1(\theta) = \theta - \frac{1}{2}\theta^2, \quad w_2(\theta) = \frac{1}{2}\theta^2. \tag{7.8}$$

This has the uniform order $q = 2$. Hence, we hope to derive a three-stage third-order continuous RK which has uniform order three, but it can be shown to be impossible. Similarly, a four-stage fourth-order continuous RK cannot attain uniform order four. This is analysed, e.g., in Sect. 5.2 of [4], where the order conditions are also explained for continuous RKs. The classical Runge–Kutta method has the continuous counterpart with the coefficients

$$w_1(\theta) = \theta - \frac{3}{2}\theta^2 + \frac{2}{3}\theta^3,\ w_2(\theta) = w_3(\theta) = \theta^2 - \frac{2}{3}\theta^3,\ w_4(\theta) = -\frac{1}{2}\theta^2 + \frac{2}{3}\theta^3 \tag{7.9}$$

of uniform order three. The fact that the uniform order q becomes less than the convergence order p appears to be disappointing, but it can be overcome.

Theorem 7.1 *Suppose we have the continuous Runge–Kutta method of convergence order p and uniform order q and $p \ge q+1$ holds. Furthermore, we assume that the function f belongs to C^{q+1}-class of $[t_0, t_F]\times\mathbb{R}^d \longmapsto \mathbb{R}^d$. When we employ the method with the constant step-size $h = (t_0 - t_F)/N$ $(N \in \mathbb{N})$ and obtain the continuous extension $\Psi(t)$, it is convergent to $x(t)$ and of $(q+1)$th order, that is, there exist an $N_0 \in \mathbb{N}$ and a constant C fulfilling the estimate*

$$\sup_{t\in[t_0,t_F]} \|\Psi(t) - x(t)\| \le Ch^{q+1}$$

if $N \ge N_0$.

Sketch of Proof On the interval $[t_n, t_{n+1}]$, by inserting the local solution $\varphi_n(t)$ given in Definition 7.1 we have

$$\Psi(t) - x(t) = (\psi_n(t) - \varphi_n(t)) + (\varphi_n(t) - x(t))\,.$$

Since we assume the uniform order q, the first difference in the r.h.s. has the estimate

$$\|\psi_n(t) - \varphi_n(t)\| = \mathcal{O}(h^{q+1}) \quad \text{on} \quad [t_n, t_{n+1}].$$

On the other hand, since the underlying RK is of pth order, we have $\varphi(t_n) - x(t_n) = x_n - x(t_n) = \mathcal{O}(h^p)$. Then, the continuous dependency of solution of the differential equation on the initial value implies

$$\|\varphi_n(t) - x(t)\| = \mathcal{O}(h^p) \quad \text{on} \quad [t_n, t_{n+1}].$$

As the two estimates hold for all of the interval $[t_n, t_{n+1}]$ and we assume $p \geq q + 1$, we attain the desired conclusion. □

Hence, when we can take q equal to $p - 1$, the continuous extension $\Psi(t)$ has the order p on the whole interval $[t_0, t_F]$ as $h \downarrow 0$. This means we can obtain a continuous numerical solution possessing the same order of convergence for arbitrary t within $[t_0, t_F]$ by the continuous Runge–Kutta methods. In fact, in modern software packages, e.g., MATLAB, continuous Runge–Kutta methods have been implemented and the users can employ this facility to obtain a continuous solution, not only discrete data $\{x_n\}$ of step-wise approximations [34].

Remark 7.1 In this section we restrict ourselves to the case that the continuous RK has the same number of stages as the underlying RK. It is, however, possible to append more intermediate values $X_{n,i}$ for $i = s + 1, \ldots, s^*$ and to give

$$\psi_n^*(t_n + \theta\, h) = x_n + h \sum_{i=1}^{s^*} w_i(\theta)\, f(x_n + c_i\, h, X_{n,i})$$

as a new continuous RK. This extends the possibility of continuous RKs. Interested readers can refer to [4, 15].

Continuous Runge–Kutta Method for DDEs

We will explain a numerical solution process of the initial-value problem of the delay-differential equation given by (7.2) and (7.3) as an application of the continuous Runge–Kutta method to the problem based on the principle of the method of steps.

The function $\tau(t)$ in (7.2) is assumed to be sufficiently smooth and positive on the whole interval $[t_0, t_F]$ and when defined as

$$\underline{\tau} = \inf_{t \in [t_0, t_F]} \tau(t) \quad \text{and} \quad \overline{\tau} = \sup_{t \in [t_0, t_F]} \tau(t),$$

we further assume $\underline{\tau}$ is positive and $\overline{\tau}$ is finite. This means we exclude the infinite delay ($\overline{\tau} = \infty$) and the vanishing delay ($\underline{\tau} = 0$) cases. Also we do not handle a state-dependent delay, i.e., $\tau(t, x(t))$. As easily imagined, the state-dependent delay case is too complicated and we omit it.

By considering (7.4), the solution process can be described in the following recursive steps.

Solution process of DDE by the continuous RK

Input: The function $f(t, x(t), y(t))$, the delay function $\tau(t)$, the initial function $\phi(t)$, the interval $[t_0, t_F]$ and $m \in \mathbb{N}$.

Arrange: Array of the interval-points $\{T_\ell\}$ and the current step-size h_ℓ.
Goal: The array $\{x_n\}$ of the approximate solution on the step-points and its continuous extension $\Xi(t)$.
First preliminary step: Introduce the continuous RK by specifying $\{a_{ij}.b_i, c_i\}$ and $\{w_i(\theta)\}$. Assign $\phi(t)$ as the (-1)th component of $\Xi(t)$ for $t_0 - \tau(t_0) \le t \le t_0$.
Second preliminary step: Compute the interval-point T_ℓ by solving the equation $T_\ell - \tau(T_\ell) = T_{\ell-1}$ recursively for $\ell = 1, 2, \ldots$ with $T_0 = t_0$. Set T_L to be the first value satisfying $T_L \ge t_F$.
0th interval: On the interval $[t_0, T_1]$, solve the IVP of ODE

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(t, x(t), \phi(t - \tau(t))), \quad x(t_0) = \phi(t_0)$$

by the continuous RK of the step-size $h_0 = (T_1 - t_0)/m$ and obtain $x_{0,1}, x_{0,2}, \ldots, x_{0,m}$ $(x_{0,n} \approx x(t_0 + nh_0))$ together with the continuous approximations $\psi_{0,1}(t)$, $\psi_{0,2}(t), \ldots, \psi_{0,m}(t)$. Assign $\Psi_0(t) = \{\psi_{0,n}(t)\}$ as the 0th component set of $\Xi(t)$ for $t_0 \le t \le T_1$.
For $\ell = 1$ to $L - 1$, repeat
ℓth interval: On the interval $[T_\ell, T_{\ell+1}]$, solve the IVP of ODE

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(t, x(t), \Xi(t - \tau(t))), \quad x(T_\ell) = \Xi(T_\ell)$$

by the continuous RK of the step-size $h_\ell = (T_{\ell+1} - T_\ell)/m$ and obtain $x_{\ell,1}, x_{\ell,2}, \ldots, x_{\ell,m}$ $(x_{\ell,n} \approx x(T_\ell + nh_\ell))$ together with the continuous approximations $\psi_{\ell,1}(t), \psi_{\ell,2}(t), \ldots, \psi_{\ell,m}(t)$. Append $\Psi_\ell(t) = \{\psi_{\ell,n}(t)\}$ as the ℓth component set of $\Xi(t)$ for $T_\ell \le t \le T_{\ell+1}$.

Note that, differently from the method of steps given in Sect. 6.1, the ℓth interval employs h_ℓ, not τ, as the step-size of the continuous RK method for the integration. Hence the reference to $\Xi(t - \tau(t))$ will play the key role of the solution process. On the nth step of the ℓth interval $[T_\ell, T_{\ell+1}]$ we are required to evaluate $\Xi(T_\ell + (n + c_i)h_\ell - \tau(T_\ell + (n + c_i)h_\ell))$ $(i = 1, 2, \ldots, s)$, whose argument decides which component of $\Xi(t)$ is actually called for. Also note that in the constant delay case ($\tau(t) = \tau$: positive constant) the solution process is much simplified and indeed we can take $T_\ell = \ell\tau$ and the equal step-size $h = \tau/m$ for the continuous RK method, and skip **Second preliminary step**.

7.3 Linear Stability of Runge–Kutta Methods for DDEs

We will follow the course of linear stability analysis developed in Sect. 3.3 and restrict ourselves to the continuous extension of Runge–Kutta methods discussed in Sect. 7.2 together with a certain integrally fractional step-size of τ, that is, $h =$

τ/m ($m \in \mathbb{N}$). Then, in the case of the linear test equation (6.10) we can rewrite the nth step forwarding of the s-stage continuous RK in the following form:

$$X_{n,i} = hL\left(x_n + \sum_{j=1}^{i-1} a_{ij} X_{n,j}\right) + hM\left(x_{n-m} + \sum_{j=1}^{i-1} a_{ij} X_{n-m,j}\right) \quad (i = 1, 2, \ldots, s) \tag{7.10}$$

and

$$x_{n+1} = x_n + \sum_{i=1}^{s} b_i X_{n,i}, \tag{7.11}$$

where the symbol $X_{\ell,i}$ denotes the d-dimensional vector of the ith intermediate value of the RK at the ℓth step-point. This formulation of the continuous RK is often called natural [4]. Considering (7.10) and (7.11) as a linear difference system, its characteristic *polynomial* can be given in the following

Lemma 7.1 *The characteristic polynomials $P_{RK}(z)$ of (7.10) and (7.11) are given by*

$$P_{RK}(z) = \det\left\{\begin{bmatrix} I_{sd} - h(A \bigotimes L) & 0 \\ -\boldsymbol{b}^{\mathsf{T}} \bigotimes I_d & I_d \end{bmatrix} z^{m+1} - \begin{bmatrix} 0 & h(\boldsymbol{e} \bigotimes L) \\ 0 & I_d \end{bmatrix} z^m - \begin{bmatrix} h(A \bigotimes M) & 0 \\ 0 & 0 \end{bmatrix} z - \begin{bmatrix} 0 & h(\boldsymbol{e} \bigotimes M) \\ 0 & 0 \end{bmatrix}\right\}, \tag{7.12}$$

where the s-dimensional vector $\boldsymbol{e}$ is defined by $\boldsymbol{e} = (1, 1, \ldots, 1)^{\mathsf{T}}$ and the symbol $\bigotimes$ stands for the Kronecker product of matrices.

Proof by determinant calculations can be found in [23]. Note that $P_{RK}(z)$ depends on τ through the identity $h = \tau/m$ and the number of roots of (7.12) is finite, because they are polynomials of z.

Delay-Independent Criteria of Linear Stability

As in Sect. 6.3 the problem of linear stability analysis of the Runge–Kutta method becomes one of looking for conditions that all the characteristic roots of (7.12) have negative real parts.

Definition 7.2 A numerical method is called asymptotically stable for the system (6.10) if there exists a positive integer m such that the step-size $h = \tau/m$ and the numerical solution x_n with h satisfies the condition

$$x_n \to 0 \quad \text{as} \quad n \to \infty$$

for any initial function.

For a delay-independent stability criterion of a natural Runge–Kutta method which has the coefficient array $(A, \boldsymbol{b})$, we can refer to the following result [22] as one of the examples.

Theorem 7.2 *Assume that the step-size h satisfies $h = \tau/m$ for a certain $m \in \mathbb{N}$ and the following conditions hold:*

(i) The real part of all the eigenvalues of $L + \xi M$ satisfying $|\xi| \leq 1$ is negative (This is the assumption of Theorem 6.2).
(ii) The underlying RK scheme is A-stable and for its coefficient matrix A, all $\lambda_i \in \sigma(A)$ have positive real parts.

Then, the natural Runge–Kutta method for DDEs is asymptotically stable.

Problems in the delay-independent criteria are that they often derive very restrictive estimation for a stable step-size h. Many efforts have been devoted to obtain relaxed conditions, see, e.g., Chap. 10 of [4]. Thus, we turn our interest to delay-dependent stability criteria of natural Runge–Kutta methods.

Now in Algorithm 7.1 we can describe an RK counterpart to Algorithm 6.1. Compare Fig. 7.1 to Fig. 3.4 for the syntax diagram of linear stability analysis.

Algorithm 7.1 Algorithm to distinguish delay-dependent stability of RK scheme

Input matrices L and M and the delay τ of (6.10),
sufficiently large $n \in \mathbb{N}$, error tolerances δ_1, δ_2,
the coefficient matrix A and vector $\boldsymbol{b}$ of s-stage RK;
Arrange n nodes $\{z_0, z_1, \ldots, z_{n-1}\}$ upon the unit circle μ of z-plane
so as $\arg z_\ell = (2\pi)\ell/n$,
the flag `stable`;
{Initial step:}
For ℓ from 0 to $n-1$ do
Evaluate $P_{RK}(z_\ell)$ by computing the determinants

$$P_{RK}(z_\ell) = \det\left\{\begin{bmatrix} I_{sd} - h(A \bigotimes L) & 0 \\ -\boldsymbol{b}^T \bigotimes I_d & I_d \end{bmatrix} {z_\ell}^{m+1} - \begin{bmatrix} 0 & h(\boldsymbol{e} \bigotimes L) \\ 0 & I_d \end{bmatrix} {z_\ell}^{m} - \begin{bmatrix} h(A \bigotimes M) & 0 \\ 0 & 0 \end{bmatrix} z_\ell - \begin{bmatrix} 0 & h(\boldsymbol{e} \bigotimes M) \\ 0 & 0 \end{bmatrix}\right\};$$

Decompose $P_{RK}(z_\ell)$ into its real and imaginary parts;
end do;
{First step:}
Check whether $P_{RK}(z_\ell) = 0$ holds for each z_ℓ $(\ell = 0, 1, \ldots, n-1)$
by $|P_{RK}(z_\ell)| \leq \delta_1$;
If yes, `stable` $\leftarrow$ NO and terminate the algorithm
otherwise, go to the next step;
{Second step:}
If $\left|\dfrac{1}{2\pi}\triangle_\mu \arg P_{RK}(z_\ell) - d(s+1)(m+1)\right| \leq \delta_2$ holds,
then `stable` $\leftarrow$ YES, otherwise `stable` $\leftarrow$ NO;
Output `stable`

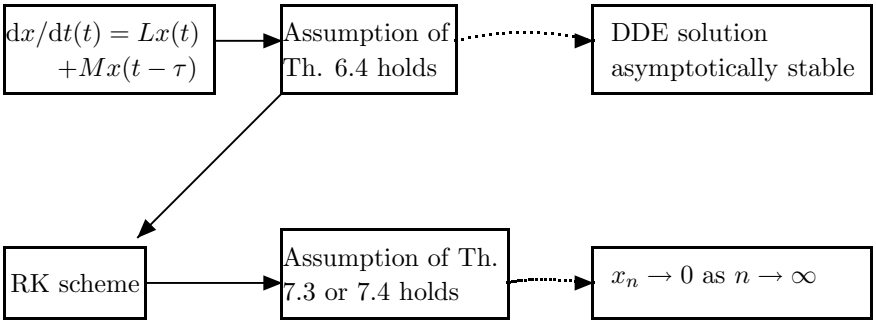


Fig. 7.1 Syntax diagram of linear stability analysis of RK for DDEs

Delay-Dependent Criteria of Linear Stability

As shown above, the delay-independent criteria of system (6.10) are very conservative which impose restrictions on the parameter matrices L and M. Hence, we proceed to investigate numerical stability based on the delay-dependent criteria of system (6.10). By applying the solution process described in Sect. 7.2, we can produce a sequence of approximate values $\{x_0, x_1, \ldots, x_n, \ldots\}$ of $\{x(t_0), x(t_1), \ldots, x(t_n), \ldots\}$ of (6.10) on certain equidistant step-points $\{t_n(= nh)\}$ with the step-size h. Then, we give the following:

Definition 7.3 Assume that the DDE system is asymptotically stable for given matrices L, M and a delay τ. A numerical method is called weakly delay-dependently stable for system (6.10) if there exists a positive integer m such that the step-size $h = \tau/m$ and the numerical solution x_n with h satisfies the condition

$$x_n \to 0 \quad \text{as} \quad n \to \infty$$

for any initial function.

Since we already established the characteristic polynomial $P_{RK}(z)$ of the natural RK scheme in (7.12), an application of the argument principle gives a sufficient condition of its weak delay-dependent stability in the following theorem.

Theorem 7.3 *Assume that:*

(i) the DDE system is asymptotically stable for given matrices L, M and delay τ (therefore, Theorem 6.4 holds);
(ii) the RK method is of s stages, explicit and natural with the step-size $h = \tau/m$;
(iii) the characteristic polynomial $P_{RK}(z)$ never vanishes on the unit circle $\mu = \{z : |z| = 1\}$ and its change of argument satisfies the identity

$$\frac{1}{2\pi} \triangle_\mu \arg P_{RK}(z) = d(s+1)(m+1). \tag{7.13}$$

Then the RK method for the DDE system is weakly delay-dependently stable.

Proof The difference system (7.10) and (7.11) is asymptotically stable if and only if all the characteristic roots of $P_{RK}(z) = 0$ lie within the unit circle. Note that the coefficient matrix of the term z^{m+1} in $P_{RK}(z)$ is

$$\begin{bmatrix} I_{sd} - h(A \bigotimes L) & 0 \\ -\boldsymbol{b}^T \bigotimes I_d & I_d \end{bmatrix}.$$

Since the Runge–Kutta method is explicit, $\lambda_i(A) = 0$ holds for $i = 1, \ldots, s$. It means that all the eigenvalues of matrix $h(A \bigotimes L)$ vanish because of

$$\text{eigenvalue of } (hA \bigotimes L) = h\lambda_i(A)\lambda_j(L) = 0$$

with $i = 1, \ldots, s$ and $j = 1, \ldots, d$. Thus the matrix $I_{sd} - h(A \bigotimes L)$ as well as the matrix

$$\begin{bmatrix} I_{sd} - h(A \bigotimes L) & 0 \\ -\boldsymbol{b}^T \bigotimes I_d & I_d \end{bmatrix}$$

are nonsingular. Since the degree of $P_{RK}(z)$ is $d(s+1)(m+1)$, it has the same number of roots in total by counting their multiplicity. By the argument principle, the condition (iii) implies that the condition $|z| < 1$ holds for all the $d(s+1)(m+1)$ roots of $P_{RK}(z) = 0$. □

For the implicit RK case applied to the DDE system, we can derive the following result, whose proof is similar to the previous Theorem 7.3.

Theorem 7.4 *Assume that:*

(i) the DDE system is asymptotically stable for given matrices L, M and delay τ;
(ii) the RK method is of s stages, implicit and natural with the step-size $h = \tau/m$;
(iii) the product $h\lambda_i(A)\lambda_j(L)$ never equals unity for all i $(1 \le i \le s)$ and j $(1 \le j \le d)$;
(iv) the characteristic polynomial $P_{RK}(z)$ never vanishes on the unit circle $\{z : |z| = 1\}$ and its change of argument satisfies

$$\frac{1}{2\pi} \triangle_\mu \arg P_{RK}(z) = d(s+1)(m+1). \tag{7.14}$$

Then the RK method for the DDE system is weakly delay-dependently stable.

Note that the condition (iii) ensures the matrix

$$\begin{bmatrix} I_{sd} - h(A \bigotimes L) & 0 \\ -\boldsymbol{b}^T \bigotimes I_d & I_d \end{bmatrix}$$

is nonsingular since the matrix $I_{sd} - h(A \bigotimes L)$ is nonsingular. Thus the degree of the polynomial $P_{RK}(z)$ becomes $d(s+1)(m+1)$.

Examples

To demonstrate the approach explained in the present section, we will present two examples of stability analysis. The underlying Runge–Kutta method is the classical four-stage one given in Sect. 3.1. Then, as we give in the previous section, its continuous extension can be attached with the uniform order three. When we employ Algorithm 7.1 to distinguish the delay-dependent stability of the RK method, we need the number of nodes n upon the unit circle of the z-plane. In the following examples we take $n = 3.2 \times 10^5$ commonly.

Example 7.2 Consider the two-dimensional linear delay system with the parameter matrices given by

$$L = \begin{bmatrix} -2 & 0 \\ 0 & -0.9 \end{bmatrix} \quad \text{and} \quad M = \begin{bmatrix} -1 & 0 \\ -1 & -1 \end{bmatrix} \tag{7.15}$$

with the initial vector function

$$u(t) = \begin{bmatrix} \sin t - 2 \\ t + 2 \end{bmatrix} \quad \text{for} \quad t \in [-\tau, 0]. \tag{7.16}$$

We can calculate $\beta = \|L\|_2 + \|M\|_2 = 3.618$ and obtain the region of instability D_β and its boundary Γ_β introduced in Definition 6.2.

The case of $\tau = 1.1$. We can apply Algorithm 6.1 to check stability of the system. Since our computation gives $\Delta_{\Gamma_\beta} \arg P(z;\tau) = 0$, according to Theorem 6.4, we can confirm that the system with the given parameter matrices is asymptotically stable.

Now we employ Algorithm 7.1 to check the delay-dependent stability of the RK method for the system. For the case of $m = 10$, we obtain $\Delta_\mu \arg P_{RK}(z)/(2\pi) = 110 = d(s+1)(m+1) = 2(4+1)(10+1) = 110$. Theorem 7.3 asserts that the RK method is weakly delay-dependently stable. The numerical solution, which is converging to 0, is depicted in Fig. 7.2a. Conversely, when $m = 1$, we obtain $\Delta_\mu \arg P_{RK}(z)/(2\pi) = 19 \neq d(s+1)(m+1) = 2(4+1)(1+1) = 20$ and the theorem does not hold. The numerical solution is divergent and its behaviour is shown in Fig. 7.2b.

The case of $\tau = 3$. Since we can calculate $\Delta_{\Gamma_\beta} \arg P(z;\tau) = 0$, the DDE system is asymptotically stable by the same reasoning as the previous case. Then, for the case of $m = 10$, the identity $\Delta_\mu \arg P_{RK}(z)/(2\pi) = 110 = d(s+1)(m+1) = 2(4+1)(10+1) = 110$ holds and the RK method is weakly delay-dependently stable. In fact the numerical solution is converging to 0 as depicted in Fig. 7.3a. On the other hand, the case $m = 3$ gives $\Delta_\mu \arg P_{RK}(z)/(2\pi) = 39 \neq d(s+1)(m+1) = 2(4+1)(3+1) = 40$ and the numerical solution is not stable. Its behaviour is shown in Fig. 7.3b.

The case of $\tau = 9$. Then, our calculation gives $\Delta_{\Gamma_\beta} \arg P(z;\tau)/(2\pi) = 2$ and the system with the given parameter matrices is not asymptotically stable. Therefore, the

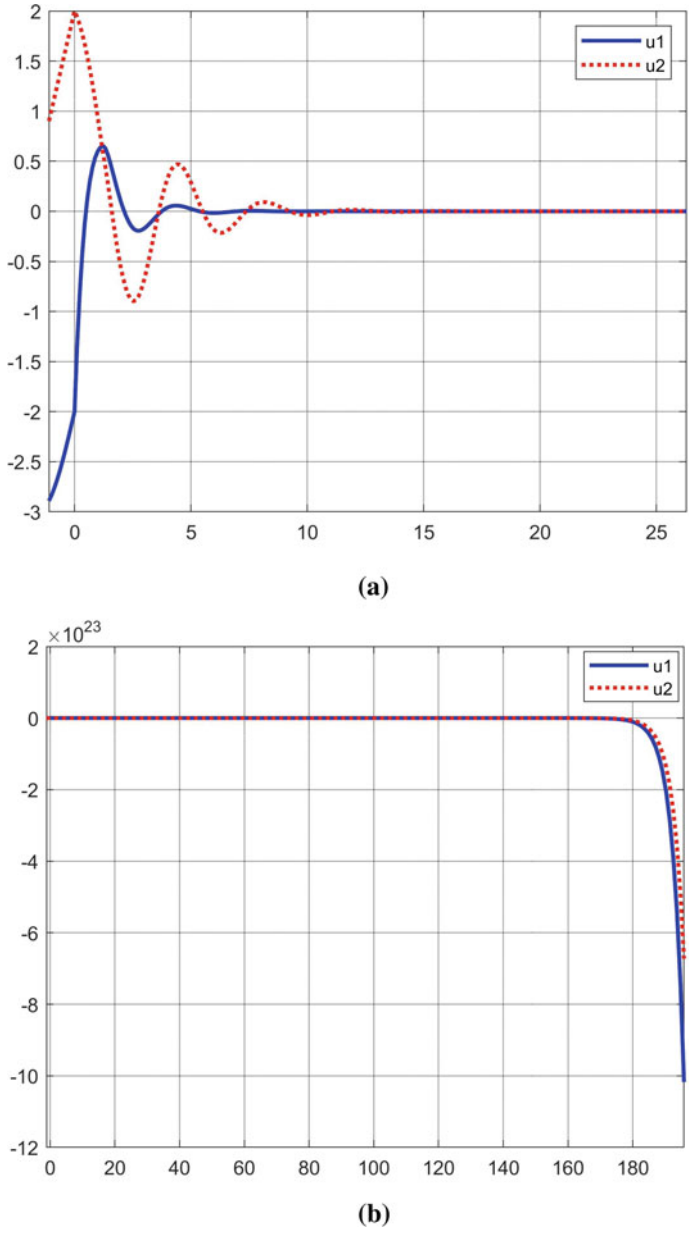


Fig. 7.2 Numerical solutions for $\tau = 1.1$ of Example 7.2. **a** when $m = 10$ (upper), **b** when $m = 1$ (lower)

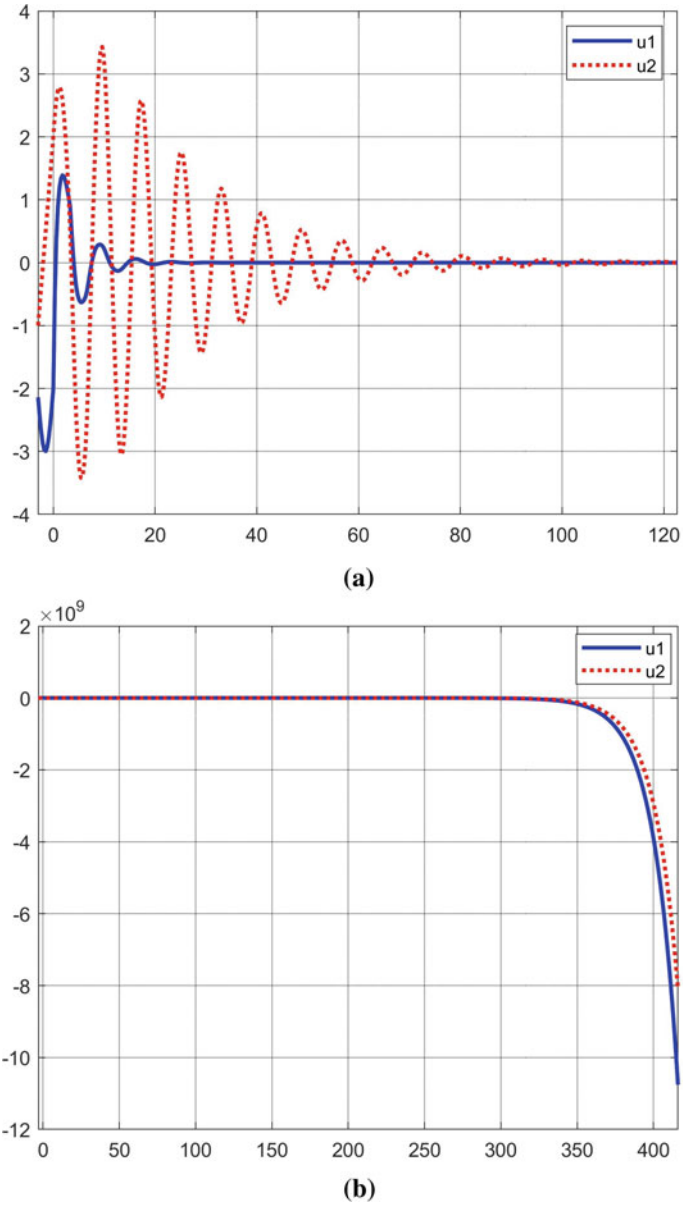


Fig. 7.3 Numerical solutions for $\tau = 3$ of Example 7.2. **a** when $m = 10$ (upper), **b** when $m = 3$ (lower)

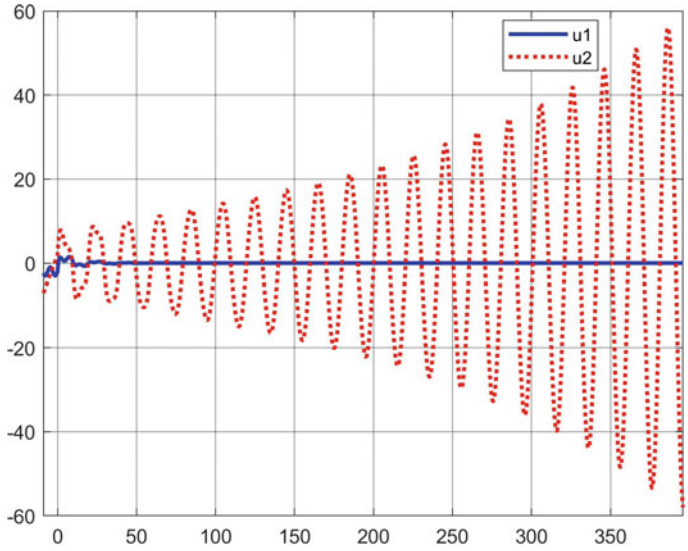


Fig. 7.4 Numerical solutions when $\tau = 9$ and $m = 100$ in Example 7.2

assumptions of the theorem do not hold and the numerical solution is not guaranteed to be asymptotically stable. In fact, the numerical solution given in Fig. 7.4 shows a divergence even for $m = 100$. We also carried out several numerical experiments for $m > 100$, whose numerical solutions are still divergent.

Example 7.3 This example is a four-dimensional one with the matrices

$$L = \begin{bmatrix} 0 & 1 & -1 & 0 \\ -3.346 & -2.715 & 2.075 & -2.007 \\ -4 & 0 & -2 & 0 \\ -3 & 0 & 0 & -6 \end{bmatrix} \quad \text{and} \quad M = \begin{bmatrix} -1 & 2 & 2 & -1 \\ 3 & 3 & -2 & 0 \\ 1 & 2 & -1 & 1 \\ 2 & 3 & 1 & -3 \end{bmatrix} \tag{7.17}$$

and the initial function

$$u(t) = \begin{bmatrix} 2\sin t + 1 \\ \cos t - 2 \\ t + 2\sin t \\ 3t + \cos t \end{bmatrix} \quad \text{for} \quad t \in [-\tau, 0]. \tag{7.18}$$

Our calculation gives the constant $\beta = \|L\|_2 + \|M\|_2 = 14.38$, which defines the region of instability D_β and its boundary Γ_β.

The case of $\tau = 0.1$. The procedure goes similarly to Example 7.2. Since our computation gives $\Delta_{\Gamma_\beta} \arg P(z; \tau) = 0$, we can confirm that the system with the given parameter matrices is asymptotically stable. For $m = 100$, our computation

gives $\Delta_\mu \arg P_{RK}(z)/(2\pi) = 2020 = d(s+1)(m+1) = 4(4+1)(100+1) = 2020$. Hence the RK method is weakly delay-dependently stable. In fact, the numerical solution is convergent as shown in Fig. 7.5a. Even for $m = 1$, we compute $\Delta_\mu \arg P_{RK}(z)/(2\pi) = 40 = d(s+1)(m+1) = 4(4+1)(1+1) = 40$. Thus the RK method is again weakly delay-dependently stable and the numerical solution is convergent as shown in Fig. 7.5b.

The case of $\tau = 0.3$. Since our computation gives $\Delta_{\Gamma_\beta} \arg P(z;\tau)/(2\pi) = 2$, we can confirm that the system with the given parameter matrices is not asymptotically stable. Then, the numerical solution is divergent for $m = 100$, whose computation results are shown in Fig. 7.6. We also carried out several other numerical experiments for $m > 100$, but the numerical solutions are still divergent.

Further Remarks

A comprehensive description about numerical solutions of DDEs can be found in the monograph [4]. The authors of [4] together with two other experts gave a further survey work [3] for numerical solutions of RFDEs. A concise survey of numerical solutions of RFDEs is also given by [2].

Known program packages for numerical solution of DDEs are compiled at the Web page whose URL is http://www.cs.kuleuven.ac.be/~twr/research/software/delay/software.shtml.

In particular, MATLAB includes the functions `dde23` (constant delay) and `ddesd` (variable and state-dependent delay). Chapter 15 of [12] explains in detail about the functions and gives a helpful discussion about numerical methods of DDEs. Section 12.4 of [19] also explains application of `dde23` with program list. Moreover, the book [7] provides a comprehensive description of MATLAB programming for DDEs and the authors have enabled downloading of the MATLAB codes from their web page.

As you can see, a numerical solution of NDDEs is more complicated than that of DDEs. Hence, Shampine [33] derives a convenient way to approximate the derivative term $\dfrac{\mathrm{d}x}{\mathrm{d}t}(t-\tau)$ in (6.16) by a difference quotient with a small positive amount of δ. That is, we employ the approximation by $\dfrac{\mathrm{d}x}{\mathrm{d}t}(t-\tau) \approx \dfrac{x(t-\tau) - x(t-\tau-\delta)}{\delta}$. Then, we need not a reference of the past values of $\dfrac{\mathrm{d}x}{\mathrm{d}t}(t)$. The idea was implemented in the MATLAB package `ddeNsd`. A detailed discussion about discrete variable solutions of NDDEs is found in [4].

The description on the delay-dependent stability criteria is based on the authors' recent study [23], which also develops analysis for linear stability of NDDEs and their Runge–Kutta solutions.

Exercises

7.1 Prove that the continuous extension of the Heun method attached to (7.8) has the uniform order two.

7.2 The initial-value problem of a scalar differential equation is given by

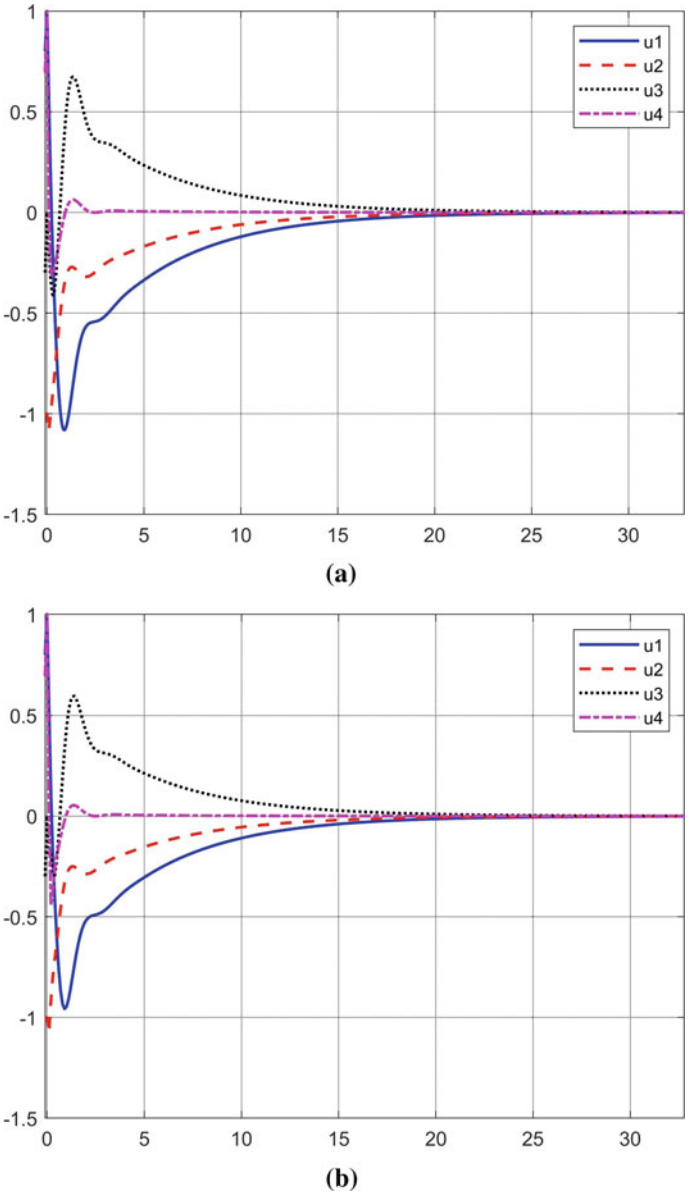


Fig. 7.5 Numerical solutions for $\tau = 0.1$ of Example 7.3. **a** when $m = 100$ (upper), **b** when $m = 1$ (lower)

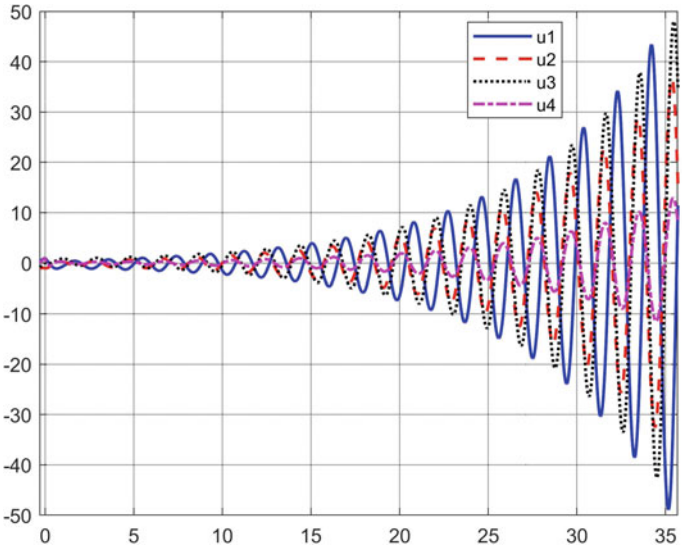


Fig. 7.6 Numerical solutions for $\tau = 0.3$ of Example 7.3 ($m = 100$)

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = a\,x(t) + b\,x(rt) \quad (0 \le t \le T), \quad x(0) = x_0,$$

where a, b and r are constants and $0 < r < 1$ holds. This is one of several retarded functional differential equations and is called the pantograph equation for a historical reason. Show that it has the power-series solution

$$x(t) = x_0 \left(1 + \sum_{k=1}^{\infty} \frac{\prod_{\ell=0}^{k-1} (a + br^{\ell})}{k!}\, t^k \right).$$

Derive the radius of convergence of the right-hand side power-series.

7.3 We take the two examples of a linear system of DDEs given in Exercise 6.8. Write a computer program for a Runge–Kutta method with a constant step-size applicable to the system and determine the stability of the Runge–Kutta method for each case.

7.4 Using your favourite programming tool, write a computer program for a constant step-size Runge–Kutta method applicable to DDEs and solve the following initial-value problem of a scalar nonlinear DDE for $0 \le t \le 3$:

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = (3 - 2x(t-1))\,x(t) \quad (t \ge 0), \quad x(t) = 1 \quad (-1 \le t \le 0).$$

Correction to: Introduction

Correction to:
Chapter 1 in: T. Mitsui and G. Hu, *Numerical Analysis of Ordinary and Delay Differential Equations*, UNITEXT 145, https://doi.org/10.1007/978-981-19-9263-6_1

The original version of the book was updated with the belated figure correction in Chapter 1 Figure 1.3. The correction chapter and the book have been updated with the changes.

The updated version of this chapter can be found at https://doi.org/10.1007/978-981-19-9263-6_1

T. Mitsui and G. Hu, *Numerical Analysis of Ordinary and Delay Differential Equations*, UNITEXT 145, https://doi.org/10.1007/978-981-19-9263-6_8

Bibliography

1. Arnold, V. I. (1971). *Geometric methods in the theory of ordinary differential equations*. MIT Press.
2. Baker, Ch. T. H. (2000). Retarded differential equation. *Journal of Computational and Applied Mathematics, 125*, 309–335.
3. Bellen, A., Guglielmi, N., Maset, S., & Zennaro, M. (2009). Recent trends in the numerical simulation of retarded functional differential equations. *Acta Numerica, 18*, 1–110.
4. Bellen, A., & Zennaro, M. (2003). *Numerical methods for delay differential equations*. Oxford UP.
5. Braun, M. (1978). *Differential equations and their applications: Short version*. Springer.
6. Breda, S., Maset, S., & Vermiglio, R. (2004). Pseudospectral differencing methods for characteristic roots of delay differential equations. *SIAM Journal on Scientific Computing, 27*, 482–495.
7. Breda, S., Maset, S., & Vermiglio, R. (2015). *Stability of linear delay differential equations—A numerical approach with MATLAB*. Springer.
8. Brown, J. W., & Churchill, R. V. (2014). *Complex variables and applications* (9th ed.). McGraw-Hill.
9. Butcher, J. C. (2003). *Numerical methods for ordinary differential equations*. Wiley.
10. Butcher, J. C. (2021). *B-series: Algebraic analysis of numerical methods*. Springer Nature.
11. Coddington, E. A., & Levinson, N. (1955). *Theory of ordinary differential equations*. McGraw-Hill.
12. Corless, R. M., & Fillion, N. (2013). *A graduate introduction to numerical methods*. Springer.
13. Driver, R. D. (1977). *Ordinary and delay differential equations*. Springer.
14. El'sgol'ts, L. E., & Norkin, S. B. (1973). *Introduction to the theory and application of differential equations with deviating arguments*. Academic Press.
15. Hairer, E., Nørsett, S. P., & Wanner, G. (1993). *Solving ordinary differential equations I, Nonstiff problems* (2nd rev ed.). Springer.
16. Hairer, E., & Wanner, G. (1996). *Solving ordinary differential equations II, Stiff and differential-algebraic systems* (2nd rev ed.). Springer.
17. Hale, J. K., & Verduyn Lunel, S. M. (1993). *Introduction to functional differential equations*. Springer.
18. Hartung, F., Krisztin, T., Walther, H.-O., & Wu, J. (2006). Functional differential equations with state-dependent delays: Theory and applications. In A. Canada et al. (Eds.), *Handbook of differential equations: Ordinary differential equations* (Chap. 5, Vol. 3, 1st ed.) North Holland.
19. Higham, D. J., & Higham, N. J. (2005). *MATLAB guide* (2nd ed.). SIAM.

T. Mitsui and G. Hu, *Numerical Analysis of Ordinary and Delay Differential Equations*, UNITEXT 145, https://doi.org/10.1007/978-981-19-9263-6

20. Householder, A. S. (1970). *The numerical treatment of a single nonlinear equation*. McGraw-Hill.
21. Hu, G. D. (2021). Separation property of observer-based stabilizing controller for linear delay systems. *Siberian Mathematical Journal, 62*, 763–772.
22. Hu, G. D., & Mitsui, T. (1995). Stability analysis of numerical methods for systems of neutral delay-differential equations. *BIT, 35*, 504–515.
23. Hu, G. D., & Mitsui, T. (2017). Delay-dependent stability of numerical methods for delay differential systems of neutral type. *BIT, 57*, 731–752.
24. Insperger, T., & Stépán, G. (2011). *Semi-discretization for time-delay systems*. Springer Science and Business Media.
25. Kahaner, D., Moler, C., & Nash, S. (1989). *Numerical methods and software*. Prentice Hall.
26. Kolmanovskii, V. B., & Myshkis, A. (1992). *Applied theory of functional differential equations*. Kluwer Academic Publishers.
27. Lambert, J. D. (1991). *Numerical methods for ordinary differential systems: The initial value problem*. Wiley.
28. Leung, A. (1977). Periodic solutions for a prey-predator differential delay equation. *The Journal of Differential Equations, 26*, 391–403.
29. Michiels, W., & Niculescu, S.-I. (Eds.). (2014). *Stability, control, and computation for time-delay systems: An eigenvalue-based approach* (2nd ed.). SIAM.
30. Mitsui, T. (1985). *Introduction to numerical analysis with emphasis on ordinary differential equations*. Asakura Publishing (in Japanese).
31. Robinson, C. (1998). *Dynamical systems: Stability, symbolic dynamics, and chaos* (2nd ed.). CRC Press.
32. Sell, G. R. (1977). What is a dynamical system? In J. Hale (Ed.), *Studies in ordinary differential equations*. Mathematical Association of America.
33. Shampine, L. F. (2008). Dissipative approximations to neutral DDEs. *Applied Mathematics and Computation, 203*, 641–648.
34. Shampine, L. F., & Reichelt, M. W. (1997). The MATLAB ODE suite. *SIAM Journal on Scientific Computing, 18*, 1–12.
35. Stoer, J., & Bulirsch, R. (2002). *Introduction to numerical analysis* (3rd ed.). Springer.

Index

T. Mitsui and G. Hu, *Numerical Analysis of Ordinary and Delay Differential Equations*, UNITEXT 145, https://doi.org/10.1007/978-981-19-9263-6