Chapter 2

A Survey of Numerical Methods for the Solution of Ordinary and Partial Fractional Differential Equations

When working with problems stemming from "real-world" applications, it is only rarely possible to evaluate the solution of a given fractional differential equation in closed form, and even if such an analytic solution is available, it is typically too complicated to be used in practice. Therefore it is indispensable to have a number of numerical algorithms at hand so that one is able to compute numerical solutions with a sufficient accuracy in reasonable time. Thus, this and the following chapter will be devoted to a study of such algorithms. To be precise, in this chapter we will give a survey of the standard numerical methods that should give the reader an impression of what he or she may expect from today's state-of-the-art algorithms. In Chapter 3 we will then look at the most important algorithms more closely, giving a detailed account of their respective strengths and weaknesses.

Since almost all numerical methods for fractional differential equations are in some sense based on the approximation of fractional differential or integral operators by appropriate formulas, we shall begin our presentation in Section 2.1 with a look at this problem. The subsequent sections will be devoted to numerical methods for fractional ordinary differential equations with an emphasis on initial value problems; specifically Section 2.2 will deal with what we call *direct methods*, i.e. methods where the numerical discretization scheme is applied directly to the differential operator appearing in the equation under consideration, whereas in Section 2.3 we look at *indirect methods* where we apply some analytic manipulation to the differential equation before the numerical work begins. Then, Section 2.4 gives an overview of a particularly important class of methods that are based on

a generalization of classical methods for first-order equations, namely the so-called *linear multistep methods*. A brief survey of other methods is contained in Section 2.5. (It should be noted here that the classes of algorithms described in these sections are not disjoint; some methods fall into more than one category.) Section 2.6 is devoted to a study of numerical methods for a slightly different class of problems, the so-called terminal value problems. In Section 2.7 we then turn our attention towards formulas for the treatment of initial value problems for equations containing more than one differential operator. Finally, in Section 2.8, we will extend the concepts that we have established to the problem of fractional partial differential equations.

For the sake of simplicity, we shall always assume here and in Chapter 3 that the fractional operators are taken with starting point 0. This will allow us to drop the corresponding index from our notation, thus enhancing the readability of the formulas. Moreover, the parameter α that will be used to denote the order of the differential and integral operators under consideration will always satisfy the relation $\alpha > 0$ (and, unless stated otherwise, also $\alpha \notin \mathbb{N}$).

Finally we note that we shall only discuss the case of scalar problems explicitly. The extension of the numerical schemes described below to the vector-valued setting (i.e. to systems of differential equations) is straightforward and does not introduce any substantial difficulties. It is sufficient to interpret the scalar solutions as appropriate vectors.

2.1 The Approximation of Fractional Differential and Integral Operators

Let us recall from the introductory chapter the fractional integral in the sense of Riemann and Liouville, defined in Eqs. (1.3.8) and (1.3.1), *viz.*

$$J^\alpha f(x) = \frac{1}{\Gamma(\alpha)} \int_0^x (x-t)^{\alpha-1} f(t) dt. \tag{2.1.1}$$

Obviously, J^α is an integral operator. The Riemann-Liouville fractional derivative from Eqs. (1.3.9), (1.3.3) and (1.3.1), i.e.

$${}^{\mathrm{RL}}D^\alpha f(x) = \frac{d^n}{dx^n} J^{n-\alpha} f(x), \qquad n = \lceil \alpha \rceil, \tag{2.1.2}$$

where $\lceil\cdot\rceil$ denotes the ceiling function that rounds up to the nearest integer, is at first sight a combination of a classical differential operator and an integral operator, but for suitable functions f it can also be written as a pure integral operator, namely

$$ {}^{\mathrm{RL}}D^{\alpha}f(x) = \frac{1}{\Gamma(-\alpha)}\int_0^x (x-t)^{-\alpha-1}f(t)dt, \tag{2.1.3} $$

see, e.g., [202]. Here, the integral has a strong singularity (its order being $1+\alpha > 1$), and it must be interpreted according to Hadamard's finite-part concept [255]. Note that formally the representation (2.1.3) of the Riemann-Liouville derivative can be obtained from the representation (2.1.1) of the Riemann-Liouville integral if we simply replace α by $-\alpha$. Additionally, we recall the definition of the Caputo derivative from Eq. (1.3.36), namely

$$ {}^{\mathrm{C}}D^{\alpha}f(x) = J^{n-\alpha}D^n f(x), \qquad n = \lceil\alpha\rceil, \tag{2.1.4} $$

where D^n denotes the classical derivative of (integer) order n. As is well known, for a large class of functions this can be rewritten in the equivalent form

$$ {}^{\mathrm{C}}D^{\alpha}f(x) = {}^{\mathrm{RL}}D^{\alpha}(f - T_{n-1}[f])(x) \tag{2.1.5} $$

that establishes the connection between the Riemann-Liouville and the Caputo derivatives. Here once again $n = \lceil\alpha\rceil$ and

$$ T_{n-1}[f](u) = \sum_{k=0}^{n-1}\frac{u^k}{k!}f^{(k)}(0) $$

is the Taylor polynomial of degree $n-1$ for the function f, centered at the point 0. A useful consequence of Eq. (2.1.5) is that any numerical approximation method for Caputo derivatives immediately yields a corresponding algorithm for their Riemann-Liouville counterparts and vice versa. In the upcoming developments we therefore only need to describe the approaches for one of these two fractional derivatives explicitly; the other one can then be constructed easily with the help of Eq. (2.1.5).

2.1.1 *Methods based on quadrature theory*

It is evident from the representations (2.1.1), (2.1.3) and (2.1.4) that all fractional derivatives and integrals that we are interested in can be interpreted as integral operators in some sense. Therefore it seems to be very natural that the first approximation methods for such operators that we shall look at are based on principles from quadrature theory, i.e. the theory of numerical integration.

In the integrals appearing in Eqs. (2.1.1) and (2.1.3), the integrands always consist of two factors. The first factor is very easily described, it being the function $(x-t)^{\alpha-1}$ or $(x-t)^{-\alpha-1}$, respectively, and it has a singularity at one end point of the interval of integration. The other factor, $f(t)$, will typically have a much more complicated analytic representation, but is nonsingular in general (and, in most cases, smooth). Under these circumstances, it is well known from the theory of numerical integration [162] that the idea of product integration is a useful approach. This means that we replace the smooth but complicated factor f by an approximation $\tilde{f}$, say, that is easily computed and for which it is also a simple matter to determine the quantity $J^\alpha \tilde{f}$ or ${}^{\mathrm{RL}}D^\alpha \tilde{f}$, respectively. Of course, in view of relation (2.1.5), the approach for the Riemann-Liouville derivative ${}^{\mathrm{RL}}D^\alpha$ can — as mentioned above — easily be modified to construct a corresponding method for the Caputo derivative ${}^{\mathrm{C}}D^\alpha$.

In practical applications, the idea of using piecewise polynomial interpolation with constant step size h to construct $\tilde{f}$ from f has proven to be most useful. We shall restrict our attention to the two most important special cases, namely piecewise polynomials of degree 0 (i.e. step functions) and degree 1 (polygons). For the discretization on the fundamental interval $[0,b]$ we will use the grid points $x_j = jh$, $j = 1,2,\ldots,N$ where, for the sake of simplicity, we assume h to be chosen such that $b = x_N = Nh$ with some integer N.

For the Riemann-Liouville integral J^α, the discretization of f by step functions has been investigated in a detailed way in [183]. The main results were that the resulting approximation, the so-called *product rectangle formula*, can be written in the form

$$J^\alpha[f](x_k) \approx {}_hJ^\alpha_{\mathrm{Re}}[f](x_k) = h^\alpha \sum_{j=0}^{k-1} b_{j,k} f(x_j) \tag{2.1.6}$$

where

$$b_{j,k} = \frac{1}{\Gamma(1+\alpha)} \left((k-j)^{\alpha} - (k-1-j)^{\alpha} \right) \tag{2.1.7}$$

and that the approximation quality of this scheme can be described as follows (see Theorem 2.4 of [183] where, in particular, details of the proof may be found).

Theorem 2.1.

(a) *Let $f \in C^1[0,b]$. Then,*

$$|J^{\alpha}[f](x_k) - {}_h J^{\alpha}_{\mathrm{Re}}[f](x_k)| \leq \frac{1}{\alpha} \sup_{x \in [0,b]} |f'(x)| x_k^{\alpha} h.$$

(b) *Let $f(x) = x^p$ for some $p \in (0,1)$. Then,*

$$|J^{\alpha}[f](x_k) - {}_h J^{\alpha}_{\mathrm{Re}}[f](x_k)| \leq C^{\alpha,p}_{\mathrm{Re}} x_k^{\alpha+p-1} h$$

where $C^{\alpha,p}_{\mathrm{Re}}$ is a constant that depends only on α and p.

We thus conclude

Corollary 2.1.

(a) *Let $f \in C^1[0,b]$. Then,*

$$|J^{\alpha}[f](x_k) - {}_h J^{\alpha}_{\mathrm{Re}}[f](x_k)| = O(h)$$

uniformly on $[0,b]$.

(b) *Let $f(x) = x^p$ for some $p \in (0,1)$. Then,*

$$|J^{\alpha}[f](x_k) - {}_h J^{\alpha}_{\mathrm{Re}}[f](x_k)| = O(h)$$

pointwise on $[0,b]$. The convergence is uniform if and only if $\alpha + p \geq 1$.

For the engineer, physicist, *etc.* who wants to solve a concrete problem involving fractional operators numerically, an $O(h)$ convergence order is often not good enough. It is therefore useful to know that a slightly more sophisticated method can be employed to improve this rate of convergence. The most natural approach is then to use polygons (continuous piecewise

polynomials of degree 1) instead of step functions which leads us to the *product trapezoidal method*

$$J^{\alpha}[f](x_k) \approx {}_hJ^{\alpha}_{\mathrm{Tr}}[f](x_k) = h^{\alpha}\sum_{j=0}^{k} a_{j,k} f(x_j) \tag{2.1.8}$$

where

$$a_{j,k} = \frac{1}{\Gamma(2+\alpha)} \times \begin{cases} \left((k-1)^{1+\alpha} - (k-\alpha-1)k^{\alpha}\right) & \text{if } j = 0, \\ \left((k-j+1)^{1+\alpha} + (k-j-1)^{1+\alpha}\right. & \\ \qquad\qquad \left. -2(k-j)^{1+\alpha}\right) & \text{if } 1 \le j \le k-1, \\ 1 & \text{if } j = k. \end{cases} \tag{2.1.9}$$

whose approximation quality is described in the following results taken from Theorem 2.5 of [183].

Theorem 2.2.

(a) *Let $f \in C^2[0,T]$. Then there is a constant C^{α}_{Tr} depending only on α such that*

$$|J^{\alpha}[f](x_k) - {}_hJ^{\alpha}_{\mathrm{Tr}}[f](x_k)| \le C^{\alpha}_{\mathrm{Tr}} \sup_{x\in[0,b]} |f''(x)| x_k^{\alpha} h^2.$$

(b) *Let $f \in C^1[0,T]$ and assume that f' fulfils a Lipschitz condition of order μ for some $\mu \in (0,1)$. Then, there exist positive constants $B^{\alpha,\mu}_{\mathrm{Tr}}$ (depending only on α and μ) and $M(f,\mu)$ (depending only on f and μ) such that*

$$|J^{\alpha}[f](x_k) - {}_hJ^{\alpha}_{\mathrm{Tr}}[f](x_k)| \le B^{\alpha,\mu}_{\mathrm{Tr}} M(f,\mu) x_k^{\alpha} h^{1+\mu}.$$

(c) *Let $f(x) = x^p$ for some $p \in (0,2)$ and $\varrho = \min(2, p+1)$. Then,*

$$|J^{\alpha}[f](x_k) - {}_hJ^{\alpha}_{\mathrm{Tr}}[f](x_k)| \le C^{\alpha,p}_{\mathrm{Tr}} x_k^{\alpha+p-\varrho} h^{\varrho}$$

where $C^{\alpha,p}_{\mathrm{Tr}}$ is a constant that depends only on α and p.

The idea of piecewise linear interpolation that is the background of the product trapezoidal method (2.1.8) for the Riemann-Liouville integral has

also been used successfully for the Riemann-Liouville derivative [167]. Not surprisingly, the resulting formula has the form

$$ {}^{\mathrm{RL}}D^{\alpha}[f](x_k) \approx {}^{\mathrm{RL}}_{h}D^{\alpha}_{\mathrm{Tr}}[f](x_k) = h^{-\alpha}\sum_{j=0}^{k} A_{j,k} f(x_j) \tag{2.1.10} $$

with

$$ A_{j,k} = \frac{1}{\Gamma(2-\alpha)} \times \begin{cases} \left((k-1)^{1-\alpha} - (k+\alpha-1)k^{-\alpha}\right) & \text{if } j = 0, \\ \left((k-j+1)^{1-\alpha} + (k-j-1)^{1-\alpha}\right. & \\ \qquad\qquad \left. -2(k-j)^{1-\alpha}\right) & \text{if } 1 \le j \le k-1, \\ 1 & \text{if } j = k. \end{cases} \tag{2.1.11} $$

In other words, the product trapezoidal approximation for the Riemann-Liouville derivative, given by Eqs. (2.1.10) and (2.1.11), can be obtained from the corresponding expressions for the Riemann-Liouville integral, *viz.* Eqs. (2.1.8) and (2.1.9), by formally replacing α by $-\alpha$.

For the approximation quality of this latter method, we have from Lemma 2.2 of [167] (see also Theorem 2.3 of [168]):

Theorem 2.3. *Let $f \in C^2[0,T]$. Then there is a constant $\bar{C}^{\alpha}_{\mathrm{Tr}}$ depending only on α such that*

$$ \left|{}^{\mathrm{RL}}D^{\alpha}[f](x_k) - {}^{\mathrm{RL}}_{h}D^{\alpha}_{\mathrm{Tr}}[f](x_k)\right| \le \bar{C}^{\alpha}_{\mathrm{Tr}} \sup_{x\in[0,b]} |f''(x)| x_k^{-\alpha} h^2. $$

Since $x_k \ge x_1 = h$, we derive the uniform estimate

Corollary 2.2. *Let $f \in C^2[0,T]$. Then,*

$$ \left|{}^{\mathrm{RL}}D^{\alpha}[f](x_k) - {}^{\mathrm{RL}}_{h}D^{\alpha}_{\mathrm{Tr}}[f](x_k)\right| = O(h^{2-\alpha}) $$

uniformly for all k.

Thus we note that the quality of this approach deteriorates as α (the order of the differential operator) increases. In particular, we cannot expect any convergence at all if $\alpha \ge 2$. However, for $0 < \alpha < 1$ — the case that is relevant for the majority of the classical applications — we do observe quite good convergence properties.

Finally we remark that it is not difficult to generalize the concepts introduced above to the case that the mesh is not uniformly spaced.

2.1.2 *Grünwald-Letnikov methods*

Another very obvious approach for the discretization of fractional differential and integral operators is based on a straightforward generalization of concepts from classical calculus to the fractional case. Specifically, it is well known that an integer-order derivative can be written as a differential quotient, *viz.*

$$D^n f(x) = \lim_{h\to 0} \frac{1}{h^n} \sum_{k=0}^{n} (-1)^k \frac{\Gamma(n+1)}{\Gamma(k+1)\Gamma(n-k+1)} f(x-kh). \qquad (2.1.12)$$

Recalling that the Gamma function has a pole at the nonpositive integers, we deduce that the sum does not change if we replace the upper summation limit n by ∞: all terms that we add have the value zero, so the infinite series in fact collapses to the sum stated in (2.1.12). In the resulting formula, there is — at least from a formal standpoint — no obstacle that prevents us from replacing the integer number n by a real number $\alpha > 0$, thus giving

$$\lim_{h\to 0} \frac{1}{h^\alpha} \sum_{k=0}^{\infty} (-1)^k \frac{\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)} f(x-kh), \qquad (2.1.13)$$

the classical *Grünwald-Letnikov derivative* [252, 338, 347] that we had already met in Eq. (1.3.192).

However there are now two substantial difficulties that we have introduced by going from the integer-order case (2.1.12) to the fractional case in (2.1.13): First, we now have a truly infinite series, and so we must discuss its convergence properties, and second we need to evaluate f at every point in $(-\infty, x)$ in order to compute the expression given in Eq. (2.1.13) whereas in the integer order case it was sufficient to deal with values of f in an arbitrarily small neighborhood of x only. The latter point is a major problem because when dealing with differential equations one usually wants to work with functions defined on a finite interval $[0, b]$, say, with the initial condition given at the point 0, and not on an interval that extends to $-\infty$. In order to remove this problem, we can use a very simple idea, namely we indeed restrict ourselves to functions defined on $[0, b]$ as desired and extend these functions to the complete interval $(-\infty, b]$ by setting $f(x) = 0$ for $x < 0$. Evidently, this means that once again the infinite series reduces to a finite sum as long as h is fixed and positive, but the number of summands

grows in an unbounded way as $h \to 0$. This gives us the definition of the Grünwald-Letnikov differential operator that is appropriate in our context,

$$ {}^{\mathrm{GL}}D^{\alpha} f(x) = \lim_{h\to 0} \frac{1}{h^{\alpha}} \sum_{k=0}^{\lceil x/h \rceil} (-1)^k \frac{\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)} f(x-kh), \quad (2.1.14) $$

cf. Eq. (1.3.194) in combination with Eq. (1.3.196). It can be shown that this operator is closely related to the Riemann-Liouville derivative:

Theorem 2.4. *Let $\alpha > 0$ and $f \in C^{\lceil \alpha \rceil}[0,b]$. Then, for $x \in (0,b]$,*

$$ {}^{\mathrm{GL}}D^{\alpha} f(x) = {}^{\mathrm{RL}}D^{\alpha} f(x). $$

For a proof of this result we refer to [202] or Theorem 20.6 and pp. 228–229 of [501].

From the representation (2.1.14), it is evident that the corresponding difference quotient with step length h,

$$ {}^{\mathrm{GL}}_{h}D^{\alpha} f(x) = \frac{1}{h^{\alpha}} \sum_{k=0}^{\lceil x/h \rceil} (-1)^k \frac{\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)} f(x-kh), \quad (2.1.15) $$

is a candidate for an approximation of ${}^{\mathrm{RL}}D^{\alpha} f(x)$. We shall call this method the *Grünwald-Letnikov finite difference* of order α with step size h.

We note that, in the case $\alpha \in \mathbb{N}$, a formally simpler representation can be obtained because of the relation

$$ \frac{\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)} = \frac{\alpha!}{k!(\alpha-k)!} = \binom{\alpha}{k}. $$

For the sake of simplicity we shall use this binomial coefficient notation also in the case that $\alpha \notin \mathbb{N}$ in accordance with Eqs. (1.2.20) and (1.2.23).

It will turn out that the Grünwald-Letnikov method is a special case of the fractional linear multistep methods discussed in the next subsection. Therefore we will not discuss it any further here and refer to the general treatment below instead.

2.1.3 *Lubich's fractional linear multistep methods*

None of the methods described so far has exhibited very fast convergence. To overcome this difficulty we now describe a class of methods introduced by Lubich in a series of papers in the 1980s [355–358].

The starting point for Lubich's approach is a classical linear multistep method for first-order initial value problems of the form

$$y'(x) = f(x, y(x)), \qquad y(x_0) = y_0. \tag{2.1.16}$$

We recall some of the background of these methods. This and more information may be found in many standard textbooks on this topic, e.g. in Chapter III of [257].

Given a uniform discretization

$$x_j = x_0 + jh, \quad j = 0, 1, 2, \ldots, N$$

of the basic interval $[x_0, b]$ with a prescribed value $N \in \mathbb{N}$ and $h = (b - x_0)/N$, a *linear multistep method* for the problem (2.1.16) takes the form

$$\sum_{k=-1}^{p} \alpha_k y_{m-k} = h \sum_{k=-1}^{p} \beta_k f(x_{m-k}, y_{m-k}) \tag{2.1.17}$$

with some real constants α_k and β_k, $k = -1, 0, 1, \ldots, p$. To be precise, it is called a $(p+1)$-step method if $\alpha_p \neq 0$ or $\beta_p \neq 0$. In the following discussion we shall always assume that at least one of these two conditions is satisfied. The method is explicit if $\beta_{-1} = 0$ and implicit otherwise. It should be noted that Eq. (2.1.17) requires the data $y_m, y_{m-1}, \ldots, y_{m-p}$ to compute the new value y_{m+1}. Thus one has to assume the starting values $y_0, y_1, \ldots y_p$ to be given (they would typically be computed by some other numerical scheme), and then the values y_{m+1}, $m = p, p+1, \ldots, N-1$, will be determined by solving Eq. (2.1.17).

In the investigation of such methods, two polynomials play a fundamental role, namely the *first characteristic polynomial*

$$\rho(\zeta) = \sum_{k=-1}^{p} \alpha_k \zeta^{p-k} \tag{2.1.18}$$

and the *second characteristic polynomial*

$$\sigma(\zeta) = \sum_{k=-1}^{p} \beta_k \zeta^{p-k}. \tag{2.1.19}$$

It is obvious that we may immediately compute the characteristic polynomials for any given linear multistep method, and we may also construct the multistep method if the characteristic polynomials are given. Thus, it is common to speak of *the linear multistep method* (ρ, σ) instead of using the representation (2.1.17). From ρ and σ we may compute another function that is important for the analysis, namely the *generating function*

$$\omega(\zeta) = \frac{\sigma(1/\zeta)}{\rho(1/\zeta)} \tag{2.1.20}$$

of the linear multistep method. For the two-step backward differentiation formula

$$\frac{3}{2}y_{m+1} - 2y_m + \frac{1}{2}y_{m-1} = hf(x_{m+1}, y_{m+1}),$$

for example, we obtain the characteristic polynomials

$$\rho(\zeta) = \frac{3}{2}\zeta^2 - 2\zeta + \frac{1}{2} \quad \text{and} \quad \sigma(\zeta) = \zeta^2$$

and the generating function

$$\omega(\zeta) = \frac{1}{\frac{3}{2} - 2\zeta + \frac{1}{2}\zeta^2} = \sum_{m=0}^{\infty} \omega_m \zeta^m$$

with

$$\omega_m = 1 - 3^{-m-1} \qquad (m = 0, 1, \ldots).$$

We want to apply such a method to the particularly simple initial value problem

$$y'(x) = f(x), \quad y(x_0) = 0,$$

whose solution obviously is

$$y(x) = \int_{x_0}^{x} f(\xi) d\xi, \quad x \in [x_0, b].$$

In this way we produce an approximation for y, i.e. for the primitive of f, on the given grid $\{x_j : j = 0, 1, \ldots, N\}$. Thus we are effectively constructing a

quadrature formula. This quadrature formula can be written in a form that will be the basis for the extension of the concept to the fractional setting:

Lemma 2.1. *Consider the numerical solution of the initial value problem* $y'(x) = f(x)$, $y(x_0) = 0$, *on the interval* $[x_0, b]$ *with the linear* $(p+1)$*-step formula* (ρ, σ), *under the assumption that* $f(x) = 0$ *for* $x_0 \le x \le x_p$ *and using the exact initial values* $y_j = 0$ *for* $j = 0, 1, \ldots, p$. *This numerical solution can be written in the* convolution quadrature form

$$({}_hJf)(x_m) = y_m = h\sum_{j=0}^{m} \omega_{m-j} f(x_j), \quad m = p+1, p+2, \ldots, N,$$

where the convolution weights ω_m, $m = 0, 1, 2, \ldots$, *are the coefficients of the power series of the generating function* ω, *viz.*

$$\omega(\zeta) = \sum_{m=0}^{\infty} \omega_m \zeta^m.$$

This result follows from Lemma 2.1 of [353].

The generalization to the fractional setting is now a relatively simple matter. Indeed, to find an approximation for $J^\alpha f$ we proceed much as above; we only need to replace the coefficients of the power series of the generating function ω itself by the corresponding coefficients of the αth power of ω. To put it formally, we start from a classical multistep formula and use the function ω defined via the two characteristic polynomials ρ and σ as in Eq. (2.1.20). Then we write the power series expansion of ω^α, *viz.*

$$(\omega(\zeta))^\alpha = \sum_{m=0}^{\infty} \omega_m \zeta^m. \tag{2.1.21}$$

We stress that the coefficients ω_m in this representation depend on the order α of the fractional operator; however we have decided to follow the common standard of not mentioning this explicitly in our notation. Since α is kept fixed in most of the applications, this should not lead to any confusion. Based on this concept we can formulate a fractional generalization of Lemma 2.1.

Lemma 2.2. *For* $0 < \alpha < 1$, *consider the numerical solution of the initial value problem* ${}^{\mathrm{C}}D^\alpha y(x) = f(x)$, $y(x_0) = 0$, *on the interval* $[x_0, b]$ *with*

the fractional version of the linear $(p+1)$*-step formula* (ρ, σ)*, under the assumption that* $f(x) = 0$ *for* $x_0 \le x \le x_p$ *and using the exact initial values* $y_j = 0$ *for* $j = 0, 1, \ldots, p$*. This numerical solution can be written in the* convolution quadrature form

$$({}_hJ^\alpha f)(x_m) = y_m = h^\alpha \sum_{j=0}^{m} \omega_{m-j} f(x_j), \quad m = p+1, p+2, \ldots, N,$$

where the convolution weights ω_m*,* $m = 0, 1, 2, \ldots$*, are the coefficients of the power series of the* α*th power of the generating function* ω *as given in Eq.* (2.1.21)*. If the underlying classical linear multistep method* (ρ, σ) *is of order* r *then the fractional version also satisfies*

$$y(x_m) - y_m = O(h^r).$$

The assumption that $f(x) = 0$ for $x_0 \le x \le x_p$ is somewhat strange. Its main purpose is to enforce a certain regular behavior of the exact solution near the point x_0. For such a regular function, this convolution quadrature is perfectly appropriate. However, as we shall see later, in a typical application we cannot expect this condition to be satisfied. Rather we have to deal with functions whose asymptotic behavior near x_0 is of the form

$$\sum_{j=0}^{J} \sum_{k=0}^{K} z_{jk} (x - x_0)^{j+k\alpha} + O((x - x_0)^S)$$

with certain positive integers J, K and S. (This follows from the general theory of the solutions of such equations developed by Miller and Feldstein [401] and extended by Lubich [354].) In such a case it turns out that the convolution quadrature cannot handle the terms with exponents $j + k\alpha \notin \mathbb{N}$ very accurately. Therefore we have to introduce additional terms to correct these flaws. Such an approach is possible; it leads to the following concept.

Definition 2.1. Let $0 < \alpha < 1$. An approximation to the fractional differential equation ${}^{\mathrm{C}}D^\alpha y(x) = f(x)$, $y(x_0) = 0$, in the form

$$({}_hJ^\alpha f)(x_m) = h^\alpha \sum_{j=0}^{m} \omega_{m-j} f(x_j) + h^\alpha \sum_{j=0}^{s} w_{mj} f(x_j), \quad m = 0, 1, \ldots, N$$

with some fixed $s \in \mathbb{N}$ is called a *fractional convolution quadrature*. The weights ω_j are called the *convolution weights*, and the term

$$ {}_h\Omega^\alpha f(x_m) = h^\alpha \sum_{j=0}^{m} \omega_{m-j} f(x_j) $$

is the *convolution part* of ${}_hJ^\alpha$ with the corresponding *convolution quadrature error* ${}_hE^\alpha = J^\alpha - {}_h\Omega^\alpha$. The term

$$ {}_hS^\alpha f(x_m) = h^\alpha \sum_{j=0}^{s} w_{mj} f(x_j) $$

is called the *starting part* or *starting quadrature*, and the weights w_{mj} are the *starting weights*.

We have already seen above how we can find suitable convolution weights. The search for starting weights leading to highly accurate methods will be postponed until §2.4.

2.2 Direct Methods for Fractional Ordinary Differential Equations

We shall now apply the approximation methods introduced above in the context of the numerical solution of fractional differential equations. In this and the following two sections, we will look at ordinary differential equations of fractional order; Section 2.8 will then be devoted to an extension of the ideas to partial differential equations.

For the purpose of exposition of the methods for solving ordinary fractional differential equations, we believe a distinction of cases to be useful. Specifically, we shall classify the numerical methods under consideration into *direct* and *indirect* methods. The direct methods will be the topic of this section; we will discuss indirect methods in §2.3. Moreover we will look at *linear multistep methods* in §2.4. Many important special instances of the latter can actually be interpreted as either direct or indirect methods, but the general concept allows to describe a much larger class whose members may or may not fall into one of these categories.

2.2.1 *The basic idea*

So let us now start our investigations by introducing the direct methods. They are characterized by the fact that we take the given initial value problem

$$ {}^{\mathrm{C}}D^{\alpha}y(x) = f(x, y(x)), \qquad y^{(k)}(0) = y_0^{(k)} \quad (k = 0, 1, \ldots, \lceil \alpha \rceil - 1), \tag{2.2.1} $$

and apply one of the approximation algorithms of §2.1 directly to the operator ${}^{\mathrm{C}}D^{\alpha}$. Since we shall restrict our attention to linear schemes, i.e. methods of the form

$$ {}^{\mathrm{C}}D^{\alpha}y(x_n) \approx \sum_{k=0}^{n} a_{k,n} y(x_k) \tag{2.2.2} $$

(where, without loss of generality, the grid points are assumed to satisfy the relation $0 \le x_0 < x_1 < \cdots < x_n \le X$), this gives rise to numerical approaches for the initial value problem (2.2.1) that have the form

$$ \sum_{k=0}^{n} a_{k,n} y_k = f(x_n, y_n), \qquad n = 0, 1, \ldots \tag{2.2.3} $$

which can be seen by combining Eqs. (2.2.1) and (2.2.2) and by replacing the unknown values $y(x_k)$ by their respective approximations y_k. In the next subsections, we shall study some important special cases of these algorithms. Unless explicitly stated otherwise, we will always assume that a solution of (2.2.1) is sought on some interval $[0, X]$, and that we have a uniform grid

$$ x_j = \frac{j}{N} X \qquad (j = 0, 1, \ldots, N) \tag{2.2.4} $$

on the basic interval $[0, X]$ with some preassigned integer N. The step size is denoted by $h = X/N$. Our job will then be to construct approximate solution values $y_k \approx y(x_k)$ at the grid points. Of course, we will use the given initial value y_0 at the point $x_0 = 0$, so actual work needs to be done for $k = 1, 2, \ldots, N$.

2.2.2 *Quadrature-based direct methods*

We begin by looking at the method obtained by using a discretization of the differential operator based on quadrature theory, i.e. the formulas introduced in Eqs. (2.1.10) and (2.1.11). This leads to an algorithm of the form

$$h^{-\alpha}\sum_{j=0}^{k} A_{j,k}y_j = f(x_k, y_k) \qquad (k = 1, 2, \ldots, N) \tag{2.2.5}$$

with the $A_{j,k}$ given in Eq. (2.1.11). Since $A_{k,k} = 1/\Gamma(2-\alpha)$, we may rewrite this identity in the form

$$y_k = \Gamma(2-\alpha)h^{\alpha}f(x_k, y_k) - \Gamma(2-\alpha)\sum_{j=0}^{k-1} A_{j,k}y_j. \tag{2.2.6}$$

The usual procedure for the computation of the numerical solution is to calculate y_1 first, then y_2, then y_3, *etc.*, until y_N is reached. In this procedure, the currently computed value is the only unknown in the equation; all other data have already been calculated in previous steps and hence are known. It is then natural to ask whether Eq. (2.2.6) can be solved for y_k. The following result states that the answer is positive provided that the step size is sufficiently small.

Theorem 2.5. *Assume that the function f on the right-hand side of the differential equation* (2.2.1) *is continuous and satisfies a Lipschitz condition with respect to its second variable with Lipschitz constant L. Moreover let $h < (\Gamma(2-\alpha)L)^{-1/\alpha}$. Then, Eq.* (2.2.6) *has a unique solution y_k for all k.*

The assumption on f is not unnatural; it is just the standard assumption imposed in the usual analytical theorems [177] that assert the existence and uniqueness of the solution of the initial value problem, which of course is a prerequisite of any numerical work.

Proof. For an arbitrary but fixed $k \in \{1, 2, \ldots, N\}$ let us define

$$\Phi(u) = \Gamma(2-\alpha)h^{\alpha}f(x_k, u) - \Gamma(2-\alpha)\sum_{j=0}^{k-1} A_{j,k}y_j.$$

From Eq. (2.2.6) it is then evident that y_k is a fixed point of the operator $\Phi : \mathbb{R} \to \mathbb{R}$. In view of the Lipschitz condition on f we deduce for every $u_1, u_2 \in \mathbb{R}$

$$\begin{aligned}|\Phi(u_1) - \Phi(u_2)| &= \Gamma(2-\alpha)h^\alpha |f(x_k, u_1) - f(x_k, u_2)| \\ &\leq \Gamma(2-\alpha)h^\alpha L|u_1 - u_2|.\end{aligned}$$

By our assumption on h we have $\Gamma(2-\alpha)h^\alpha L < 1$, i.e. the mapping Φ is contractive. Thus, by Banach's fixed point theorem, it has a unique fixed point, i.e. Eq. (2.2.6) has a unique solution y_k. □

As a consequence of this proof we see that we may actually find the required numerical solution y_k from Eq. (2.2.6) by means of the Picard iteration sequence

$$y_{k,\ell+1} = \Phi(y_{k,\ell}) = \Gamma(2-\alpha)\left(h^\alpha f(x_k, y_{k,\ell}) - \sum_{j=0}^{k-1} A_{j,k} y_j\right) \quad (\ell = 0, 1, \ldots)$$

that will converge to y_k as $\ell \to \infty$ for any starting value $y_{k,0}$. A default choice for this starting value that is useful in most cases is $y_{k,0} = y_{k-1}$.

Remark 2.1. In the limit case $\alpha \to 1$ this method reduces to the classical first-order backward differentiation formula (BDF1). Therefore it seems reasonable to call this method a fractional backward differentiation formula. However we shall see shortly that other generalizations of the classical BDF1 exist as well. Similar problems arise with almost every other numerical method that we shall encounter: Most of these can be interpreted as a generalization of a well known routine for first-order initial value problems, but these basic routines can be extended to the fractional setting in many different ways, and all these generalizations have a certain right to inherit the name of their classical ancestor. Thus, the reader must be very careful when looking at the literature on numerical methods for fractional differential equations because quite frequently one will find different methods being denoted by identical names.

The convergence behavior of this fractional backward differentiation formula can be summarized in the following way.

Theorem 2.6. *The fractional backward differentiation formula* (2.2.6)*, with weights* A_{jk} *as in Eq.* (2.1.11)*, produces an approximate solution for the initial value problem* (2.2.1) *that satisfies the inequality*

$$|y(x_j) - y_j| = O(h^{2-\alpha})$$

uniformly for all j *if the solution* y *satisfies the smoothness condition* $y \in C^2[0, X]$.

In the case of a linear equation this has been shown in Corollary 1.2 of [167]; the extension to nonlinear problems can be done using standard methods.

Once again, as in Theorem 2.3 and in Corollary 2.2, it turns out that the approximation is quite good if α is close to zero but deteriorates as α increases, up to the point that no convergence at all can be expected if $\alpha \geq 2$.

2.3 Indirect Methods for Fractional Ordinary Differential Equations

2.3.1 *The basic idea*

In contrast to the direct methods that we had introduced in the previous section, indirect methods are constructed not by applying a discretization directly to the fractional differential equation, but by first performing some analytical manipulation on the initial value problem and by then applying the numerical method to the equation obtained as the result of this analytical operation. The most common idea in this context is to apply the Riemann-Liouville integral operator J^α to the initial value problem (2.2.1), thus creating the nonlinear and weakly singular Volterra integral equation of the second kind

$$y(x) = \sum_{k=0}^{\lceil\alpha\rceil-1} \frac{y_0^{(k)}}{k!} x^k + J^\alpha[f(\cdot, y(\cdot))](x). \tag{2.3.1}$$

The following result from Lemma 2.1 of [177] shows under very weak conditions on the given initial value problem that this operation does not change the exact solution.

Theorem 2.7. *If the function f is continuous, then the initial value problem* (2.2.1) *is equivalent to the Volterra integral equation* (2.3.1). *In other words, every continuous solution of* (2.3.1) *is also a solution of our original initial value problem* (2.2.1) *and vice versa.*

Thus, we can now use a numerical method to discretize the Riemann-Liouville integral operator J^α and not, as in the direct methods, the Caputo differential operator ${}^{\mathrm{C}}D^\alpha$. Once again we only consider linear schemes

$$J^\alpha z(x_n) \approx \sum_{k=0}^{n} a_{k,n} z(x_k) \tag{2.3.2}$$

hence producing numerical approaches for the Volterra equation (2.3.1) that have the form

$$y_n = \sum_{k=0}^{n} a_{k,n} f(x_k, y_k), \qquad n = 0, 1, \ldots. \tag{2.3.3}$$

In the next subsections, we shall again look at some special cases of this idea more closely, but before doing that we shall note one observation that catches the eye when comparing the direct method of Eq. (2.2.3) and the indirect method described in Eq. (2.3.3). Specifically, the computational complexity of the two methods appears to be somewhat different because each step of the direct method involves only one evaluation of the given function f, whereas the nth step of the indirect method requires $O(n)$ such evaluations. This can be an important aspect because the computation of f may, in some cases, be an extremely time-consuming job. Since in a practical application one would have to solve Eqs. (2.2.3) and Eq. (2.3.3), respectively, by some iterative procedure, the number of function calls would in both cases have to be multiplied by some factor. This feature is likely to make the difference even bigger. However, by using a careful implementation it is possible to store and re-use the values $f(x_j, y_j)$ in the indirect approach, thus obtaining a computational cost that effectively is almost identical to the cost of the direct method.

We shall once again assume that we are looking for an approximate solution of the fractional differential equation (2.2.1) or, equivalently, the Volterra integral equation (2.3.1), on the interval $[0, X]$ and that we

have discretized this interval in a uniform way with grid points given in Eq. (2.2.4), *viz.*

$$x_j = \frac{j}{N} X \qquad (j = 0, 1, \ldots, N).$$

Thus the step size is again $h = X/N$. The approximation for $y(x_j)$ will be denoted by y_j as usual.

2.3.2 *An Adams-type predictor-corrector method*

The first method that we shall look at in this section has been introduced by Diethelm and Freed [185, 186] and analyzed in a detailed way in [182, 183]. It is based on the approximation of the integral operator in Eq. (2.3.1) by the product trapezoidal method introduced in §2.1.1. This leads to the formula

$$y_k = \sum_{j=0}^{\lceil\alpha\rceil-1} \frac{x_k^j}{j!} y_0^{(j)} + h^\alpha \sum_{j=0}^{k} a_{jk} f(x_j, y_j) \tag{2.3.4}$$

for $k = 1, 2, \ldots, N$ where the weights a_{jk} are given in Eq. (2.1.9). In the limit case $\alpha \to 1$ this method reduces to the classical second-order Adams-Moulton formula (see, e.g., §III.1 of [257]); hence we shall call our method a *fractional Adams-Moulton formula* (but we explicitly recall the note of caution mentioned in Remark 2.1 with respect to the terminology).

It is evident from Eq. (2.3.4) that this method is implicit because the unknown value of the kth step, *viz.* the number y_k, appears not only on its own on the left-hand side of the equation but also inside the function f on the right-hand side. Thus we need to consider the question of solving Eq. (2.3.4) for y_k.

The first idea in this context is to create an iterative method for the solution of Eq. (2.3.4) for each k and to run these iterations until the difference between two successive iterations is sufficiently close (typically, a small multiple of machine epsilon). From an applied standpoint, this essentially amounts to computing the exact solution of Eq. (1.30). Such a process is indeed possible, and it results in the fractional Adams-Moulton formula in its pure form. We summarize the essential properties of this idea as follows.

Theorem 2.8. *Assume that the function f on the right-hand side of the differential equation* (2.2.1) *is continuous and satisfies a Lipschitz condition with respect to its second variable with Lipschitz constant L. Moreover let $h < (\Gamma(2+\alpha)/L)^{1/\alpha}$. Then, Eq.* (2.3.4) *has a unique solution y_k for all k.*

Moreover, the approximation computed in this way satisfies the error bound

$$|y(x_j) - y_j| = O(h^2)$$

uniformly for all j under either of the following conditions:

- ${}^{\mathrm{C}}D^{\alpha}y \in C^2[0,X]$,
- $\alpha > 1$ *and* $f \in C^3([0,X] \times [y_0^{(0)} - K, y_0^{(0)} + K])$ *with some* $K > 0$.

The first part of this result corresponds to Theorem 2.5 for the first-order backward differentiation formula of Diethelm; it can be proved in exactly the same way. The second part is an analogue of Theorem 2.6. It can be deduced from Theorem 2.2.

It must be noted however that the solution of the nonlinear equation (2.3.4) by the method of iteration until convergence will frequently require a rather large number of iteration steps, and hence the entire process is likely to be very time-consuming. Fortunately, it is actually not necessary to go into this expensive procedure. The reason is that y_k is only an approximation for the desired value $y(x_k)$ with a certain accuracy. Thus, if we only solve Eq. (2.3.4) up to an error of the same order, then the total error will still be of this order. Our goal is thus to try to find such a sufficiently good approximate solution of Eq. (2.3.4) using only a small number of arithmetical operations. To this end we recall some facts from the classical theory of first-order equations (see, e.g., Sections III.1 and III.8 of [257] and Section V.1 of [258]).

Specifically, a common procedure when dealing with such a method (that falls into the category of implicit multistep methods) whose order of convergence is $O(h^p)$ is to solve Eq. (2.3.4) by the following iterative procedure:

(1) Compute a first approximation of the solution y_k of Eq. (2.3.4), the so-called *predictor* $y_{k,0}$, by means of an explicit algorithm whose order of convergence is $O(h^{p-1})$.

(2) Perform one step of the iteration and take the result $y_{k,1}$ of this step, the so-called *corrector*, as the desired approximate solution, i.e. write

$$y_{k,1} = \sum_{j=0}^{\lceil\alpha\rceil-1} \frac{x_k^j}{j!} y_0^{(j)} + h^\alpha \sum_{j=0}^{k-1} a_{jk} f(x_j, y_j) + h^\alpha a_{kk} f(x_k, y_{k,0}). \quad (2.3.5)$$

and use this value in place of the true solution y_k in all future steps.

Such a procedure is known as a *predictor-corrector method* or, more precisely, as a PECE (predict-evaluate-correct-evaluate) method because we first compute the predictor $y_{k,0}$, then we evaluate the function f with arguments $(x_k, y_{k,0})$, then we use the result to compute the corrector $y_{k,1}$, and finally we evaluate f once again, this time with arguments $(x_k, y_{k,1})$, because we need this value for the process of computing y_{k+1}. The classical theory of numerical methods for first-order equation then tells us that the values $y_k = y_{k,1}$ approximate the true solution $y(x_k)$ of our initial value problem up to an error of magnitude $O(h^p)$.

Of course, it is possible to construct a similar process for the fractional setting. A natural candidate for a predictor for the fractional Adams-Moultion scheme is the *fractional Adams-Bashforth method*. This method is constructed in the same way as the Adams-Moulton method, except that we replace the product trapezoidal method by the product rectangle method that we had also introduced in §2.1.1. It is thus given by

$$y_k = \sum_{j=0}^{\lceil\alpha\rceil-1} \frac{x_k^j}{j!} y_0^{(j)} + h^\alpha \sum_{j=0}^{k-1} b_{jk} f(x_j, y_j) \quad (2.3.6)$$

for $k = 1, 2, \ldots, N$ where the weights b_{jk} are given in Eq. (2.1.7). In the limit case $\alpha \to 1$ this method reduces to the classical first-order Adams-Bashforth formula (see, e.g., §III.1 of [257]) that happens to coincide with the forward Euler method.

The Adams-Bashforth method can of course be considered as a numerical scheme for solving fractional differential equations in its own right. It is an explicit method since y_k does not appear on the right-hand side of Eq. (2.3.6); therefore we do not need to worry about conditions under which Eq. (2.3.6) can be solved for y_k. As a consequence of Corollary 2.1, the error can be estimated as follows:

Theorem 2.9. *The approximation computed by the Adams-Bashforth method satisfies the error bound*

$$|y(x_j) - y_j| = O(h)$$

uniformly for all j *if* ${}_{\mathtt{C}}D^{\alpha}y \in C^1[0, X]$.

However, our plan was not to use one of these methods on its own but rather to combine them to a predictor-corrector pair. Thus the following complete representation of the Adams-Bashforth-Moulton algorithm is to choose a certain value of $m \in \mathbb{N}$ (the number of corrector iterations; it is very common to use $m = 1$), write

$$y_{k,0} = \sum_{j=0}^{\lceil\alpha\rceil-1} \frac{x_k^j}{j!} y_0^{(j)} + h^{\alpha} \sum_{j=0}^{k-1} b_{jk} f(x_j, y_j), \tag{2.3.7a}$$

$$y_{k,\mu} = \sum_{j=0}^{\lceil\alpha\rceil-1} \frac{x_k^j}{j!} y_0^{(j)} + h^{\alpha} \sum_{j=0}^{k-1} a_{jk} f(x_j, y_j) + h^{\alpha} a_{kk} f(x_k, y_{k,\mu-1}) \qquad (\mu = 1, 2, \ldots, m) \tag{2.3.7b}$$

$$y_k = y_{k,m} \tag{2.3.7c}$$

for $k = 1, 2, \ldots, N$ and use the values y_k as the desired approximations. The weights a_{jk} and b_{jk} must be taken from Eqs. (2.1.9) and (2.1.7), respectively.

Looking at the error of this predictor-corrector method, we find a slight peculiarity of the fractional case in comparison to first-order equations. For the latter we know that each application of a corrector iteration increases the order of the approximation by 1 until the full order of the corrector is reached. Thus, if the predictor is an $O(h^{p_1})$ method and the pure corrector is $O(h^{p_2})$ with $p_2 > p_1$, then the predictor-corrector method with m corrector steps as described in Eq. (2.3.7) has got an order $O(h^{\min\{p_1+m, p_2\}})$. In the fractional case, we observe that we need more corrector steps to achieve such a goal. To be precise, each corrector iteration only increases the order by α (see, e.g., Section 6 of [169]). Combining the results of [169] and [183], the formal result reads as follows.

Theorem 2.10. *The relation* $|y(x_k) - y_k| = O(h^{\min\{2, 1+m\alpha\}})$ *holds uniformly for all* $j = 1, 2, \ldots, N$ *under either of the conditions mentioned in Theorem* 2.8.

Other error estimates under slightly different assumptions may be found in Section 3 of [183]. A full algorithmic description of the method in a pseudocode form has been given in [184].

2.3.3 *The Cao-Burrage-Abdullah approach*

A substantially different indirect method has recently been proposed by Cao, Burrage and Abdullah [129]. Also starting from the initial value problem

$$ {}^{\mathtt{C}}D^{\alpha}y(x) = f(x, y(x)), \qquad y(0) = y_0, \tag{2.3.8} $$

with $0 < \alpha < 1$, they decided not to apply the *integral* operator of order α but a *differential* operator of order $1 - \alpha$ to the differential equation. Interestingly, it is necessary here not to use a Caputo operator but a Riemann-Liouville operator, thus — using the formal relation ${}^{\mathtt{RL}}D^{1-\alpha}\,{}^{\mathtt{C}}D^{\alpha} = (D^1 J^{\alpha})(J^{1-\alpha}D^1) = D^1 J^1 D^1 = D^1$ — obtaining the problem

$$ y'(x) = {}^{\mathtt{RL}}D^{1-\alpha}\,{}^{\mathtt{C}}D^{\alpha}y(x) = {}^{\mathtt{RL}}D^{1-\alpha}[f(\cdot, y(\cdot))](x), \tag{2.3.9} $$

still augmented by the originally given initial condition $y(0) = y_0$.

Now, in a numerical approach we need to approximate the two differential operators D^1 and ${}^{\mathtt{RL}}D^{1-\alpha}$ on the right-hand side and the left-hand side of Eq. (2.3.9), respectively. Since the integer-order derivative D^1 can be handled numerically by very simple standard methods, the additional amount of work (compared to the approaches of the previous sections where we only had to deal with one differential or integral operator) is negligible. Moreover, the nature of the methods discussed earlier essentially forced us to use an equispaced mesh. In many cases such a straightforward approach is perfectly appropriate, but there are situations where one would rather prefer to use a non-uniform mesh. Examples include cases where the given data is only provided in tabulated form at some irregularly spaced abscissas, or when one has reason to believe that the solution behaves in an unpleasant (e.g., highly oscillatory) fashion in some subinterval, thus requiring small step sizes there, whereas the variation is small (and hence admits much more economical large stepsizes) elsewhere. It is precisely this

setting of nonuniform grids that Cao, Burrage and Abdullah had in mind when they developed their concept.

Specifically, let $0 = x_0 < x_1 < x_2 < \cdots < x_N$ be the mesh points. Then, the discretization at the point x_j, $1 \leq j \leq N$, takes the form

$$\frac{y_j - y_{j-1}}{x_j - x_{j-1}} = \sum_{k=0}^{j} a_{kj} f(x_k, y_k) \tag{2.3.10}$$

where the weights a_{kj} are taken from a suitable discretization of the Riemann-Liouville derivative. A number of possible choices are available here, for example a non-equispaced version of the product trapezoidal method from §2.1.1. Of course there remains the question which choice is good in this context. Unfortunately, almost no analytical results seem to be known in this connection. In addition, since the method has only recently been published, there is not much numerical experience available either.

The price that we have to pay for the high degree of flexibility in the choice of the mesh introduced by this approach is on the analytical side: At present we are not aware of any thorough error analysis of the algorithm. Moreover, there is a substantial difference between a discretization of Eq. (2.3.9) and the discretization that we had to perform in the earlier parts of this chapter. This difference is due to the fact that Eq. (2.3.9) will typically be valid only on the half-open interval $(0, x_N]$. The reason is that, as is well known [401], the solution y of the originally given fractional differential equation (2.3.8) in most cases behaves as $y(x) = y_0 + cx^\alpha$ as $x \to 0$ with some constant c. Thus, since $0 < \alpha < 1$, we have (except for the very rare special case where $c = 0$) that y is not differentiable at 0, and so Eq. (2.3.9) is undefined there. Moreover it is unclear what the influence of this discontinuity of y' is on the numerical approximation method. One certainly needs to take into account the possibility that the approximation in (2.3.10) may be poor for $j = 1$, giving a possibly very inaccurate value for y_1, and it is unknown how this potentially large error is propagated to the values $y_2, y_3, \ldots$. This is not to say that the method will be doomed to failure (indeed the numerical results in [129] look promising at least for the examples considered there), but one has to be warned that a convergence proof or error analysis under reasonably general assumptions is lacking at the moment.

2.4 Linear Multistep Methods

A particularly well understood class of methods is based on using the *linear multistep methods* described in Subsection 2.1.3 above for the discretization of the fractional differential and integral operators arising in our equations. In particular, we recall the concept of fractional convolution quadratures from Definition 2.1 and introduce the following notions.

Definition 2.2. Consider the fractional convolution quadrature

$$({}_hJ^\alpha f)(x_m) = h^\alpha \sum_{j=0}^{m} \omega_{m-j} f(x_j) + h^\alpha \sum_{j=0}^{s} w_{mj} f(x_j), \quad m = 0, 1, \ldots,$$

for some $\alpha > 0$.

- The quadrature is called *stable* for the integral J^α if

$$\omega_n = O(n^{\alpha-1}).$$

- It is called *consistent* of order p for J^α if

$$h^\alpha \omega(\exp(-h)) = 1 + O(h^p).$$

- It is called *convergent* of order p to J^α if

$${}_h\Omega^\alpha \pi_{z-1}(1) - J^\alpha \pi_{z-1}(1) = O(h^z) + O(h^p)$$

holds for all $z \in \mathbb{C} \setminus \{0, -1, -2, \ldots\}$ where we have used the notation $\pi_k(x) = x^k$.

Recall that the function ω used in this definition is related to the coefficients of the fractional convolution quadrature via the identity

$$\omega(z) = \sum_{k=0}^{\infty} \omega_k z^k.$$

It is important to note a special feature of Definition 2.2: The stability, consistency and convergence of a fractional convolution quadrature depend only on its convolution part. The starting quadrature does not play a role here. We shall come back to this observation in Theorem 2.13. First

however we note one important result that generalizes Dahlquist's well known result for first-order problems [158, 159] to the fractional setting. For a proof of this result we refer to Lubich's classical paper [356] or the detailed discussion in the book of Brunner and van der Houwen [110] or in Section 4.3 of Weilbeer's thesis [576].

Theorem 2.11. *Let α be a non-integer number and r_1 and r_2 be rational functions. Then, a fractional convolution quadrature whose convolution weights are defined by the generating function $\omega(\zeta) = (r_1(\zeta))^\alpha r_2(\zeta)$ is convergent of order p if and only if it is stable and consistent of order p.*

Whereas this result allows us to determine whether a given fractional convolution quadrature is convergent or not, another important result by Dahlquist [160] that can be generalized to the fractional setting (see Theorem 4.3.11 of [576]) allows us to construct fractional linear multistep methods in such a way that the convergence is guaranteed a priori. This construction may be based on any classical linear multistep method for first order problems that is known to be convergent.

Theorem 2.12. *Let (ρ, σ) denote an implicit classical inear multistep method for first order equations which is stable and consistent of order p. Moreover assume that all zeros of σ lie inside or on the boundary of the unit disc. Denote the generating function of this linear multistep method by ω. Then, for $\alpha > 0$, the fractional linear multistep method with generating function ω^α is convergent of order p to J^α.*

Now let us come back to the observation mentioned above that the starting quadrature does not play a role in the context of stability, consistency and convergence of the fractional convolution quadrature. Specifically, if we consider a fractional convolution quadrature ${}_hJ^\alpha$ as introduced in Definition 2.1 that converges to J^α with order p (as defined in Definition 2.2, i.e. without taking the starting part into consideration) then we can always find a set of starting weights such that for a sufficiently well behaved function f the error ${}_hJ^\alpha f - J^\alpha f$ is also of order p. The following theorem indicates what we mean by a sufficiently well behaved function in this context and shows how to obtain the required starting weights. It is based on a result given in [355].

Theorem 2.13. *Let (ρ, σ) be a convergent implicit linear multistep method of order $p \geq 1$ for first order differential equations, and let all zeros of σ lie inside or on the boundary of the unit disc. Moreover let $0 < \alpha < 1$ and*

$$\omega^{\alpha}(\zeta) = \left(\frac{\sigma(1/\zeta)}{\rho(1/\zeta)}\right)^{\alpha}.$$

Furthermore, set $A = \{\gamma = k + j\alpha : j, k \in \mathbb{N}_0, \gamma \leq p - 1\}$ and let $s + 1$ be the number of elements of A. Define the starting weights w_{nj} to be the solution of the linear system

$$\sum_{j=0}^{s} w_{nj} j^{\gamma} = \frac{\Gamma(\gamma + 1)}{\Gamma(\gamma + \alpha + 1)} n^{\gamma+\alpha} - \sum_{j=0}^{n} \omega_{n-j} j^{\gamma}, \quad \gamma \in A, \tag{2.4.1}$$

then we have $w_{nj} = O(n^{\alpha-1})$, $j = 0, 1, \ldots, s$.

Moreover, if y_n $(n = 0, 1, \ldots, N)$ is the numerical approximation of the solution y of the fractional initial value problem ${}^{\mathrm{C}}D^{\alpha} y(x) = f(x, y(x))$, $y(0) = y_0$, at the point $x_n = nh$, defined by

$$y_n = y_0 + h^{\alpha} \sum_{j=0}^{n} \omega_{n-j} f(x_j, y_j) + h^{\alpha} \sum_{j=0}^{s} w_{n,j} f(x_j, y_j)$$

where f is sufficiently smooth, then this numerical solution satisfies

$$\max_{n=0,1,\ldots,N} |y_n - y(x_n)| = O(h^{p-\epsilon})$$

with some $\epsilon \in [0, 1 - \alpha)$. In particular, $\epsilon = 0$ if $\alpha = q/(q + 1)$ with some $q \in \mathbb{N}$.

This means that we can always find a set of starting weights such that the error of the fractional convolution quadrature behaves in the same way as it does in the classical (first-order) case.

It thus turns out that, in theory, these formulas can be used to compute highly accurate numerical solutions for fractional differential equations. In practice however things are not that simple. In particular, it has been observed [175] that the computation of the starting weights can be a very ill-conditioned problem for certain choices of the order α of the differential operator whereas it behaves much better for other values of α. This is due

to the fact that the coefficient matrix of the linear system (2.4.1) that defines these weights may, depending on α, contain rows that almost coincide, i.e. the system may be very close to a singular system. In such a case, no accurate method for the solution of the system, i.e. for the computation of the starting weights, is available, and the use of a standard method for the solution of this system leads to highly inaccurate values for the starting weights and consequently to highly inaccurate numerical solutions of the fractional differential equations in question. We shall provide a detailed discussion about the cases where such problems may be expected and, on the other hand, where fractional linear multistep methods may be considered reliable in Subsection 3.1.3.

For the moment we conclude our discussion of this approach with a reference to the work of Ford and Connolly [221] who have provided a comparison of the performance of the linear multistep methods with the other algorithms described above. In particular, their goal was to display the accuracy that can be achieved by the methods in relation to their respective computational costs. The results of [221] indicate that the predictor-corrector method of Subsection 2.3.2 tends to be a very useful choice in many cases and that a third order backward differentiation scheme (which is a special case of a linear multistep method) might also be a good candidate, but the information presently known about the latter is not yet sufficient to fully decide on its usefulness.

2.5 Other Methods

Some authors have also suggested other approaches than those described so far. Most of these methods have not gained a substantial amount of attention. The primary exception is probably the so-called *Adomian decomposition* that is usually traced back to Adomian's books [8, 9] even though its roots can actually be found in a series of much older papers by Perron [447–450]. The idea of the method is to write the differential equation in the abstract form

$$M(x, y(x)) = g(x) \tag{2.5.1}$$

where g is a given function, y is the unknown solution and M is a suitable operator (in our case, usually a combination of a fractional differential

operator and some other, often nonlinear, functions applied to y). One then decomposes M according to

$$M = L_1 + L_2 + N \tag{2.5.2}$$

where L_1 and L_2 are linear operators chosen such that the inverse of L_1 can be found easily, while N denotes the remaining part of M that includes, in particular, all the nonlinearities. The method is based on the assumption that the exact solution y can be decomposed into a convergent series,

$$y = \sum_{n=0}^{\infty} y_n, \tag{2.5.3}$$

and the goal of the approach is to provide a reasonably simple method to compute the summands y_n of this series. To this end, we also decompose the nonlinearity N in the special series form

$$Ny(x) = \sum_{n=0}^{\infty} A_n(x)$$

with the so-called *Adomian polynomials*

$$A_n(x) = \frac{1}{n!}\left[\frac{d^n}{d\lambda^n} N\left(x, \sum_{j=0}^{n} \lambda^j y_j\right)\right]_{\lambda=0}.$$

Apparently A_n depends only on $y_0, y_1, \ldots, y_n$. One then starts with

$$y_0 = L_1^{-1} g$$

(which can be computed easily in view of our assumption on L_1) and proceeds by the recurrence relation

$$y_n = -L_1^{-1}(L_2 y_{n-1} - A_{n-1}), \qquad n = 1, 2, \ldots,$$

that defines the remaining summands of the series expansion (2.5.3) of y.

It is clear from the description above that the user of such an approach has a large amount of freedom in a concrete application. For example, the exact choice of the operators L_1, L_2 and N in Eq. (2.5.2) can be done in a number of different ways. Whereas it is obvious that the precise choice

of these features can have a significant influence on the behavior of the method, it seems that no general rules are available that give the user concrete hints as to which choice may be useful in the context of the specific problem at hand. As stated above, most authors dealing with this method *assume* that the series expansion of Eq. (2.5.3) converges, but they do not look for conditions under which this assumption is satisfied, i.e. they use a hypothesis in terms of the unknown solution rather than in terms of the given data. In addition they neither deal with the question for the appropriate notion of convergence in this context (Convergence with respect to which norm?) nor do they discuss whether the series converges rapidly or slowly. The values of the argument of y for which the series converges (i.e. the convergence radius of the series) is usually not investigated either, but some results indicate that the radius of convergence frequently is rather small [199, 216, 475]. Moreover the computation of the Adomian polynomials A_n is possible only if N, and hence M, is sufficiently smooth, and even in this case it requires an enormous amount of work if it is done analytically and is subject to severe numerical difficulties if done numerically. Therefore we believe that the precisely described and well understood methods of the previous sections should be preferred at least until more information about the decomposition becomes available.

Alternatively one may use the *variational iteration method* proposed by He [266] (see also the recent survey [267]). Like the Adomian method it is based on the abstract form (2.5.1) of the problem. The operator M is now decomposed in a slightly different way, namely as

$$M = L + N \tag{2.5.4}$$

where L is a linear operator. Starting from an initial approximation y_0 that is usually chosen to satisfy the initial conditions (and, if present, also the boundary conditions), one then constructs a sequence $(y_n)_{n=0}^{\infty}$ of approximations in recursive manner using the scheme

$$y_n(t) = y_{n-1}(t) + \int_0^t \lambda(\tau)\,(Ly_{n-1}(\tau) + N\tilde{y}_{n-1}(\tau) - g(\tau))\,d\tau \quad (n = 1, 2, \ldots). \tag{2.5.5}$$

Here, λ is a Lagrange multiplier that is characterized with the help of methods from variational calculus, and $\tilde{y}_{n-1}$ is a restricted variation, i.e.

$\delta\tilde{y}_{n-1} = 0$. Under certain conditions (that may be difficult to verify in practice) one can then show that $y_n \to y$ as $n \to \infty$ where y denotes the exact solution of the problem under consideration. We refer to [548] for more details on a related question; the approach presented there can be extended to our problem. Since, in practice, it is usually very difficult to determine the integrals on the right-hand side of Eq. (2.5.5), this method is of a rather limited value in concrete applications.

Similar remarks apply to a few other methods that have been proposed in the past like the *homotopy analysis method* [345] (that actually can be interpreted as a generalization of the Adomian decomposition), the *homotopy perturbation method* [406], and the *generalized differential transform method* [431]. In particular, the convergence properties of many of these methods tend to be rather poor [344].

Another completely different approach that is probably much more useful in practice is due to Yuan and Agrawal [591]. Their method is based on the observation that one can rewrite the Caputo derivative of the function y in the form

$$ {}^{\mathrm{C}}D^{\alpha}y(x) = \int_0^{\infty} \phi(w,x)\,dw. \tag{2.5.6} $$

where

$$ \phi(w,x) = (-1)^{\lfloor\alpha\rfloor}\frac{2\sin\pi\alpha}{\pi} w^{2\alpha-2\lceil\alpha\rceil+1} \int_0^x y^{(\lceil\alpha\rceil)}(\tau)e^{-(x-\tau)w^2}\,d\tau. \tag{2.5.7} $$

Then they note that this function ϕ is a solution of the *first-order* initial value problem

$$ \begin{aligned} \frac{\partial}{\partial x}\phi(w,x) &= -w^2\phi(w,x) \\ &\quad + (-1)^{\lfloor\alpha\rfloor}\frac{2\sin\pi\alpha}{\pi} w^{2\alpha-2\lceil\alpha\rceil+1} y^{(\lceil\alpha\rceil)}(x), \\ \phi(w,0) &= 0 \end{aligned} \tag{2.5.8} $$

for each fixed $w > 0$. We consider this to be an ordinary, not a partial, differential equation because w is assumed to be fixed. Based on these identities they then suggest to compute a numerical approximation for the Caputo derivative by replacing the integral on the right-hand side of

Eq. (2.5.6) by a Gauss-Laguerre quadrature. The required function values of ϕ in this quadrature are not computed by its definition as given in Eq. (2.5.7) but by solving the ordinary differential equation (2.5.8) numerically by a suitable algorithm.

Lu and Hanyga [352] and, in particular, Schmidt and Gaul [515] have criticized the poor approximation quality of this method. Later, Diethelm [170] has identified the mathematical background of this unsatisfactory behavior, it being the fact that the Gauss-Laguerre quadrature formula suggested by Yuan and Agrawal for the numerical approximation of the integral in Eq. (2.5.6) is really unsuitable because of the properties of the integrand. By replacing the Gauss-Laguerre method by a suitably transformed Gauss-Jacobi rule, much better results can be obtained. For details we refer to [170]. Birk and Song [94, 95] have successfully applied this modified version of the Yuan-Agrawal method in connection with the investigation of certain diffusion problems.

Alternatively, a very similar approach is due to Chatterjee and Singh [141, 522]. Instead of using the function ϕ given in Eq. (2.5.7), they define

$$\phi(w,x) = \frac{(-1)^{\lfloor\alpha\rfloor}\sin\pi\alpha}{\pi(\alpha-\lceil\alpha\rceil+1)}\int_0^x y^{(\lceil\alpha\rceil)}(\tau)\exp\left(-(x-\tau)w^{1/(\alpha-\lceil\alpha\rceil+1)}\right)d\tau \tag{2.5.9}$$

and show that this function also satisfies Eq. (2.5.6). Moreover, it can also be expressed as the solution of a first-order initial value problem similar to Eq. (2.5.8) that now takes the form

$$\begin{aligned}\frac{\partial}{\partial x}\phi(w,x) &= -w^{1/(\alpha-\lceil\alpha\rceil-1)}\phi(w,x)\\ &\quad + \frac{(-1)^{\lfloor\alpha\rfloor}\sin\pi\alpha}{\pi(\alpha-\lceil\alpha\rceil+1)}y^{(\lceil\alpha\rceil)}(x),\\ \phi(w,0) &= 0\end{aligned} \tag{2.5.10}$$

for a fixed $w > 0$. Either of the numerical techniques used for the original Yuan-Agrawal approach above is applicable to this version too. First steps in this direction are described in [170].

Moreover we want to mention the collocation method of Blank [99] that is much closer in spirit to the methods of the previous sections than the method of Adomian or the variational iteration method in the sense that the latter are really more analytical than numerical approximations. The

collocation scheme is based on choosing an N-dimensional linear space L of ansatz functions, for example a space of polynomial splines [163], and a set of N distinct points $x_1, \ldots, x_N$ in the interval where the solution of the differential equations is sought. One then tries to find a function $\phi \in L$ that satisfies the initial value problem (2.2.1) or, more conveniently, its Volterra integral formulation (2.3.1), at the points x_j, i.e.

$$\phi(x_j) = \sum_{k=0}^{\lceil\alpha\rceil-1} \frac{y_0^{(k)}}{k!} x_j^k + J^\alpha[f(\cdot, \phi(\cdot))](x_j), \quad j = 1, 2, \ldots, N. \tag{2.5.11}$$

This function ϕ is then taken as an approximation to the exact solution y of the initial value problem. In practice one chooses a basis $\{\phi_j : j = 1, 2, \ldots, N\}$ of the space L, represents the as yet unknown function ϕ as a linear combination of these basis function, *viz.*

$$\phi(x) = \sum_{k=1}^{N} a_k \phi_k(x),$$

and inserts this relation into each of the equations of the system (2.5.11). In this way we obtain an N-dimensional nonlinear system of equations with N unknowns that typically can be solved using appropriate methods.

In the special case that the differential equation is linear, the system becomes linear too and the solution theory and numerics become rather simple. Under this additional linearity assumption, Blank [99] has demonstrated that unique solutions exist and has developed a method to effectively compute these solutions.

2.6 Methods for Terminal Value Problems

Sometimes one is interested in the solution of a somewhat different class of problems that are occasionally known under the name *terminal value problems*, see, e.g., pp. 107–109 of [172]. Specifically we still want to solve a differential equation of the form

$${}^{\mathrm{C}}D^\alpha y(x) = f(x, y(x)) \tag{2.6.1a}$$

on some interval $[0, x^*]$, say. For the sake of simplicity we restrict our attention to the case $0 < \alpha < 1$. In contrast to the previously discussed

problem described in Eq. (2.2.1), our additional condition that asserts the existence and uniqueness of the solution now takes the modified form

$$y(x^*) = y^*, \tag{2.6.1b}$$

i.e. we prescribe the function value at the end point of the interval of interest (hence the name, terminal value problem) and not at the starting point of the differential operator. It has been shown in [181] that under reasonable assumptions this problem indeed is uniquely solvable; thus it is a natural question to ask for numerical methods to find an approximate solution.

A proposal for a potential numerical algorithm can be derived from the analysis provided in Theorem 6.18 of [172]. This result essentially states that the terminal value problem (2.6.1) can be rewritten as an equivalent integral equation of the form

$$y(x) = y^* + \frac{1}{\Gamma(\alpha)} \int_0^{x^*} G(x,t) f(t, y(t)) dt$$

where

$$G(x,t) = \begin{cases} -(x^* - t)^{\alpha-1} & \text{for } t > x, \\ (x-t)^{\alpha-1} - (x^* - t)^{\alpha-1} & \text{for } t \le x. \end{cases}$$

The significant difference to the initial value problems treated previously is that this integral equation is an equation of Fredholm and not Volterra type. Nevertheless it is a class of integral equations for which numerical methods are available (see, e.g., [250, 253, 302, 568]), so an application of one of these methods to the Fredholm equation immediately yields a numerical scheme for the terminal value problem.

An alternative way to construct a numerical method for terminal value problems has been briefly described in [181]. It is based on replacing the terminal condition (2.6.1b) by a standard-type initial condition

$$y(0) = y_0 \tag{2.6.2}$$

with an as yet unknown value y_0. One then uses the fact (see Corollary 6.16 of [172]) that the graphs of two solutions for the differential equation (2.6.1a) subject to two different initial conditions will, under standard assumptions, never intersect (the restriction $0 < \alpha < 1$ is relevant in this context though). Thus one starts with an arbitrary guess for the value y_0

and solves the initial value problem that consists of the differential equation (2.6.1a) and the initial condition (2.6.2) using any of the numerical schemes described above. Then one takes a look at the approximate solution at the point x^*. If this value is greater than the value y^* given in the terminal condition (2.6.1b) then one decreases y_0 and starts again; if it is smaller than y^* then y_0 has to be increased. This process is repeated in an iterative way, for example using a bisection strategy once one has reached a stage where an approximation with a too large value and an approximation with a too small value have been found, until one has determined an y_0 that leads to a numerical solution that matches the terminal condition up to the required accuracy.

In order to illustrate this scheme we recall here the example considered in [181], *viz.* the terminal value problem

$$ {}^{\mathrm{C}}D^{1/2}y(x) = \sin y(x), \qquad y(1) = 2.5. \tag{2.6.3}$$

We start by choosing an arbitrary initial value, in our case $y_0 = 1$, and compute the numerical approximation for $y(1)$ using the Adams-Bashforth-Moulton method with a step size of $1/200$. It turns out (see Table 2.1) that this leads to a too small value. Thus we restart the Adams method with $y_0 = 2$, which leads to a too large value. From here on we employ a simple bisection technique to find a new initial value, $y_0 = 1.5$, and compute $y(1)$ again. Proceeding in an iterative manner we find the required successive values for y_0 indicated in Table 2.1 and see that we can get as close to the desired exact solution as we like.

Table 2.1. Results of numerical algorithm for the terminal value problem (2.6.3).

y_0	1	2	1.5	1.75	1.625	1.6875	1.71875
$y(1)$	2.0556	2.63485	2.37728	2.51106	2.44567	2.47871	2.49496

2.7 Numerical Methods for Multi-Term Fractional Differential Equations and Multi-Order Fractional Differential Systems

Up to this point we have only discussed so-called *single-term* fractional differential equations, i.e. equations containing only one differential operator.

Whereas these equations form appropriate models for many problems in physics and other sciences, they are insufficient in some other cases. Indeed it is sometimes necessary to use differential equations involving differential operators of more than one order. Among the most prominent examples of such a situation we mention the *Bagley-Torvik equation* [558]

$$AD^2y(x) + B{}^{\mathsf{C}}D^{3/2}y(x) + Cy(x) = g(x), \quad y(0) = y_0, y'(0) = y_0', \quad (2.7.1)$$

where $A \neq 0$ and B and C are arbitrary real numbers. This equation can be used to model the motion of a rigid plate immersed in a Newtonian fluid; see, e.g., [558] or Section 8.3.2 of [453].

Other applications arise in the work of Koeller [321] who has given a theoretical explanation of why differential equations of the form

$$p_0\sigma(t) + p_1{}^{\mathsf{C}}D^{\alpha_1}\sigma(t) + p_2{}^{\mathsf{C}}D^{\alpha_2}\sigma(t) = q_0\epsilon(t) + q_1{}^{\mathsf{C}}D^{\alpha_1}\epsilon(t) + q_2{}^{\mathsf{C}}D^{\alpha_2}\epsilon(t)$$

may be used to model the mechanical behavior of viscoelastic materials with two relaxation modes. Here ϵ is strain and σ is stress. One of these quantities (in most cases ϵ) is usually known whereas the other one is the unknown solution of the multi-term equation. Koeller was particularly interested in the case $\alpha_2 = 2\alpha_1$, but this is by no means a must. Rossikhin and Shitikova [491] have recently pointed out that Koeller's equation is just a special case of a more general model for viscoelastic behavior introduced in a formally different but equivalent way by Rabotnov [473].

Our final example for an application of multi-term equations is the *Basset equation*

$$D^1y(x) + b{}^{\mathsf{C}}D^{\alpha}y(x) + cy(x) = f(x), \qquad y(0) = y_0,$$

where $0 < \alpha < 1$. This equation describes the forces that occur when a sphere sinks in a (relatively less dense) fluid; see, e.g., [369].

All these equations can be written in the form

$${}^{\mathsf{C}}D^{\alpha_n}y(x) = f(x, y(x), {}^{\mathsf{C}}D^{\alpha_1}y(x), \ldots, {}^{\mathsf{C}}D^{\alpha_{n-1}}y(x)) \quad (2.7.2)$$

with initial conditions

$$y^{(k)}(0) = y_0^{(k)}, \qquad k = 0, 1, \ldots, \lceil \alpha_n \rceil - 1, \quad (2.7.3)$$

where we assume that $0 < \alpha_1 < \alpha_2 < \cdots < \alpha_n$. An equation of the form (2.7.2) is called a *multi-term fractional differential equation*. Without loss of generality we shall assume that all the integers that are contained in the interval $(0, \alpha_n)$ are also members of the finite sequence $(\alpha_k)_{k=1}^n$. In other words, it is impossible for two consecutive elements of the finite sequence (α_k) to lie on opposite sides of an integer number. It is an immediate consequence of this assumption that we have $0 < \alpha_{j+1} - \alpha_j \leq 1$ for all $j = 1, 2, \ldots, n-1$.

In order to illustrate this assumption, we look at the Bagley-Torvik equation (2.7.1). Here we would choose $\alpha_1 = 1$, $\alpha_2 = 3/2$, $\alpha_3 = 2$, and $f(x, y(x), {}^{\mathrm{C}}D^{\alpha_1}y(x), {}^{\mathrm{C}}D^{\alpha_2}y(x)) = (g(x) - B{}^{\mathrm{C}}D^{\alpha_2}y(x) - Cy(x))/A$, thus bringing the equation into the form (2.7.2). An analytical theory for equations of the type (2.7.2), including existence and uniqueness theorems, has been given in [179]. That paper also contains first steps towards the development of a general-purpose numerical algorithm for multi-term equations. Additional information may be found in the survey article [171].

Apart from these multi-term equations, there is a different possibility to introduce more than one fractional derivative into a mathematical model. Specifically, we may use a system of fractional differential equations, each of which has an order that may or may not coincide with the orders of the other equations. To put it more formally, this leads to the model of the type

$$\begin{aligned} {}^{\mathrm{C}}D^{\alpha_1}y_1(x) &= f_1(x, y_1(x), \ldots, y_n(x)), \\ &\vdots \quad \vdots \\ {}^{\mathrm{C}}D^{\alpha_n}y_n(x) &= f_n(x, y_1(x), \ldots, y_n(x)). \end{aligned} \tag{2.7.4}$$

As we shall see it is sufficient for our purposes to assume that $0 < \alpha_k \leq 1$ for all k. This implies that the initial conditions for the differential equation system (2.7.4) are

$$\begin{aligned} y_1(0) &= y_{1,0}, \\ &\vdots \quad \vdots \\ y_n(0) &= y_{n,0}. \end{aligned} \tag{2.7.5}$$

A system of this class will be called *multi-order fractional differential system*. Such systems seem to be investigated less frequently than multi-term

equations, but we will now reveal some close connections between the two concepts, and thus the former deserve some attention at least in view of the fact that they can be very useful tools for the numerical treatment of the latter. The connections may be constructed in two different ways.

Specifically, given the equation (2.7.2), we may first write $\beta_1 = \alpha_1$, $\beta_j = \alpha_j - \alpha_{j-1}$ $(j = 2, 3, \ldots, n)$, $y_1 = y$ and $y_j = {}^{\mathrm{C}}D^{\alpha_{j-1}} y$, $j = 2, 3, \ldots, n$. Note that under our assumptions on the α_j it is clear that $0 < \beta_j \leq 1$ for all j. Then we can conclude:

Theorem 2.14. *The multi-term equation* (2.7.2) *with initial conditions* (2.7.3) *is equivalent to the system*

$$\begin{aligned} {}^{\mathrm{C}}D^{\beta_1} y_1(x) &= y_2(x), \\ {}^{\mathrm{C}}D^{\beta_2} y_2(x) &= y_3(x), \\ &\vdots \quad \vdots \\ {}^{\mathrm{C}}D^{\beta_{n-1}} y_{n-1}(x) &= y_n(x), \\ {}^{\mathrm{C}}D^{\beta_n} y_n(x) &= f(x, y_1(x), y_2(x), \ldots, y_n(x)) \end{aligned} \tag{2.7.6}$$

with the initial conditions

$$y_j(0) = \begin{cases} y_0^{(0)} & \text{if } j = 1, \\ y_0^{(k)} & \text{if } \alpha_{j-1} = k \in \mathbb{N}, \\ 0 & \text{else} \end{cases} \tag{2.7.7}$$

in the following sense:

(a) *Whenever the function $y \in C^{\lceil \alpha_n \rceil}[0, X]$ is a solution of the multi-term equation* (2.7.2) *with initial conditions* (2.7.3)*, the vector-valued function $Y = (y_1, \ldots, y_n)^{\mathrm{T}}$ with*

$$y_j(x) = \begin{cases} y(x) & \text{if } j = 1, \\ {}^{\mathrm{C}}D^{\alpha_{j-1}} y(x) & \text{if } j \geq 2, \end{cases} \tag{2.7.8}$$

is a solution of the multi-order system (2.7.6) *with initial conditions* (2.7.7)*.*

(b) *Whenever the vector-valued function $Y = (y_1, \ldots, y_n)^{\mathrm{T}}$ is a solution of the multi-order system* (2.7.6) *with initial conditions* (2.7.7)*, the function $y = y_1$ is a solution of the multi-term equation* (2.7.2) *with initial conditions* (2.7.3)*.*

A proof of this result may be found in [171] (see also [178]).

To describe the second possible method we assume, for the sake of simplicity, that the highest order differential operator is not an integer-order derivative. Otherwise some small formal modifications in the notation are necessary, but the basic concept and the main results remain unchanged. For a general formal description of this alternative approach it is advantageous to express the multi-term equation in the form

$$\begin{aligned} {}^{\mathrm{C}}D^{k+\delta_{k,n_k}} y(x) = f(x, {}^{\mathrm{C}}D^{0} y(x), {}^{\mathrm{C}}D^{\delta_{0,1}} y(x), \dots, {}^{\mathrm{C}}D^{\delta_{0,n_0}} y(x), \\ {}^{\mathrm{C}}D^{1} y(x), {}^{\mathrm{C}}D^{1+\delta_{1,1}} y(x), \dots, {}^{\mathrm{C}}D^{1+\delta_{1,n_1}} y(x), \dots, \\ {}^{\mathrm{C}}D^{k} y(x) \dots, {}^{\mathrm{C}}D^{k+\delta_{k,n_k-1}} y(x)), \end{aligned} \tag{2.7.9}$$

where $0 < \delta_{j,1} < \delta_{j,2} < \cdots < \delta_{j,n_j} < 1$ for all j. The corresponding initial conditions are then

$$y_j(0) = y_0^{(j)}, \qquad j = 0, 1, \dots, k. \tag{2.7.10}$$

In order to achieve our goal, we define

$$s(\mu, \sigma) = \sigma + \mu + 1 + \sum_{j=0}^{\mu-1} n_j \quad \text{and} \quad N = s(k, n_k) - 1 = k + \sum_{j=0}^{k} n_j.$$

Using this notation, we come to the following statement.

Theorem 2.15. *The multi-term equation* (2.7.9) *with initial conditions* (2.7.10) *is equivalent to the N-dimensional system*

$$\begin{aligned} {}^{\mathrm{C}}D^{\delta_{\mu,\sigma}} y_{s(\mu,0)}(x) &= y_{s(\mu,\sigma)}(x), \quad \mu = 0, 1, \dots, k, \quad \sigma = 1, 2, \dots, \widehat{\sigma}_\mu, \\ {}^{\mathrm{C}}D^{1} y_{s(\mu,0)}(x) &= y_{s(\mu+1,0)}(x), \quad \mu = 0, 1, \dots, k-1, \\ {}^{\mathrm{C}}D^{\delta_{k,n_k}} y_{s(k,0)}(x) &= f(x, y_1(x), y_2(x), \dots, y_N(x)) \end{aligned} \tag{2.7.11}$$

where $\widehat{\sigma}_\mu = n_\mu$ if $0 \le \mu < k$ and $\widehat{\sigma}_k = n_k - 1$, with the initial conditions

$$y_j(0) = \begin{cases} y_0^{(k)} & \text{if there exists } k \text{ such that } j = s(k, 0), \\ 0 & \text{else} \end{cases} \tag{2.7.12}$$

in the following sense:

(a) *Whenever the function* $y \in C^{k+1}[0, X]$ *is a solution of the multi-term equation* (2.7.9) *with initial conditions* [2.7.10]*, the vector-valued function* $Y = (y_1, \ldots, y_N)^{\mathrm{T}}$ *with*

$$y_{s(\mu,\sigma)}(x) = \begin{cases} {}^{\mathrm{C}}D^{\mu} y(x) & \text{for } \sigma = 0, \\ {}^{\mathrm{C}}D^{\mu+\delta_{\mu,\sigma}} y(x) & \text{for } \sigma = 1, 2, \ldots, \widehat{\sigma}_{\mu}, \end{cases} \qquad \mu = 0, 1, \ldots, k, \tag{2.7.13}$$

is a solution of the multi-order system (2.7.11) *with initial conditions* (2.7.12)*.*

(b) *Whenever the vector-valued function* $Y = (y_1, \ldots, y_N)^{\mathrm{T}}$ *is a solution of the multi-order system* (2.7.11) *with initial conditions* (2.7.12)*, the function* $y = y_1$ *is a solution of the multi-term equation* (2.7.9) *with initial conditions* (2.7.10)*.*

This result has been stated without proof for a subset of the class of equations described in (2.7.9) in [198]. A possible method of proof has been outlined in [171].

The key difference between these two approaches is that the method of Theorem 2.14 will convert the given multi-term equation into an n-dimensional multi-order system where all orders of the differential operators are strictly fractional (unless we have some j with $\alpha_j - \alpha_{j-1} = 1$), while the method of Theorem 2.15 will produce an n-dimensional system with $\lfloor \alpha_n \rfloor$ equations of order 1 and $n - \lfloor \alpha_n \rfloor$ equations of strictly fractional order. The latter may be advantageous from a numerical point of view because the first order equations involve local differential operators that may be discretized with a smaller computational cost.

A third method has been proposed by Diethelm and Ford [179]. That method however requires some number-theoretic assumptions on the orders α_j of the differential operators to be satisfied. In such a case the method provides a system of single-order equations that is equivalent to the given multi-term equation too. The formal advantage of this system is that all the equations are of the same order. This simplifies the analysis and implementation of the numerical method. However, there are also two disadvantages associated with the approach of [179]. Firstly, if the assumptions mentioned above are not satisfied then the method can be modified to provide a system of single-order equations that is approximately, but not exactly, equivalent to the given multi-term system. This is not a major drawback

because, in principle, the differences in the solutions can be made arbitrarily small, and since one normally can only approximately solve the system anyway, the additional error introduced in this way will not have significant effects. The second disadvantage may be more serious: The dimension of the system that will be constructed can be very large. This may lead to an unnecessarily high arithmetic complexity [169]. Therefore we shall not go into the details of this third approach here.

Ford and Simpson [223] have shown in passing that these techniques may also be used to reformulate a single-order fractional differential equation whose order is greater than 1 in the form of an equivalent multi-order system with orders less than or equal to 1. The latter is a somewhat easier object for numerical work as some algorithms tend to behave much worse when applied to equations of higher order. In addition, this concept may be considered an extension of the well known classical technique for the numerical solution of initial value problems of higher integer order which consists of rewriting the problem in the form of a first-order system and solving this system numerically with the help of an algorithm for first-order initial value problems.

An even more general problem is the question for numerical methods for so-called *distributed-order equations*, i.e. equations of the form

$$\int_0^m A(r, {}^{\mathrm{C}}D^r y(x))dr = f(x, y(x)) \tag{2.7.14}$$

with certain functions A and f and suitable initial conditions. Equations of this type arise in a number of applications, see, e.g., [41, 47, 48, 133–135, 263, 529, 562]. From a numerical point of view, the present knowledge about this class of problems is very limited. Therefore we shall restrict ourselves to drawing the reader's attention towards the works of Diethelm and Ford [176, 180] and the references cited therein.

We remark at this point that we also do not intend to discuss *variable-order equations*, i.e. equations of the form

$${}^{\mathrm{C}}D^{\alpha(x)}y(x) = f(x, y(x)) \tag{2.7.15}$$

where the order of the differential operator is not a constant but depends on the free variable x. For applications of models of this (or an even more general) type we refer, e.g., to [539].

2.8 The Extension to Fractional Partial Differential Equations

2.8.1 *General formulation of the problem*

In this Chapter, we have dealt only with ordinary differential equations of fractional order so far. However, in the modeling of various phenomena in finance, engineering, physics and other areas the use of partial differential equations with fractional differential operators is becoming more and more popular. A particularly important class of applications arising in this context are the so-called *time-fractional diffusion-wave equations*

$$ {}^{\mathtt{C}}D_t^{\alpha} y(x,t) + \phi(x,t) \sum_{j=1}^{n} \frac{\partial^2}{\partial x_j^2} y(x,t) = f(x,t) \tag{2.8.1} $$

for $x = (x_1, \ldots, x_n) \in G \subset \mathbb{R}^n$ and $t \in [0, T]$. Here, the notation ${}^{\mathtt{C}}D_t^{\alpha} y(x,t)$ indicates the partial derivative of Caputo type of order α of the function y with respect to t. It is clear that this equation unifies the classical concepts of the n-dimensional diffusion equation (heat equation) which is obtained for $\alpha = 1$ and the n-dimensional wave equation ($\alpha = 2$). In order to obtain a well-posed problem, it is natural to combine Eq. (2.8.1) with two boundary conditions and one (if $\alpha \leq 1$) or two (if $1 < \alpha \leq 2$) initial conditions. These conditions are usually given in the same form as in the classical case, *viz.*

$$ A(x,t) y(x,t) + B(x,t) \frac{\partial}{\partial n} y(x,t) = g(x,t) \quad \text{for } t \geq 0 \text{ and } x \in \partial G \tag{2.8.2} $$

(with $\partial/\partial n$ denoting the partial derivative in the direction of the outer normal of the boundary at the point x) and

$$ y(x,0) = f_1(x), \quad \frac{\partial}{\partial t} y(x,0) = f_2(x) \quad \text{for } x \in G \tag{2.8.3} $$

(the second condition of Eq. (2.8.3) of course being applicable only in the case $1 < \alpha \leq 2$). As the name *time-fractional diffusion-wave equations* indicates, we can obtain this class of equations by taking a classical diffusion or wave equation and replacing the first- or second-order derivative with respect to the time variable by a fractional derivative. From a formal point of view it would seem to be equally natural to replace the second-order

derivatives with respect to the space variables by their fractional-order generalizations. This has indeed been done occasionally; however, some authors raise strong objections against this idea because it leads to certain phenomena that violate fundamental principles of physics. For a thorough and well written discussion of this issue we refer to Section 2.3.2 of Hilfer's work [281].

A straightforward way to construct efficient numerical methods for this class of equations can be described in a very simple way. Specifically we can discretize the second order derivatives with respect to space by a classical second order central difference quotient whereas one of the numerical schemes mentioned in the previous sections can be used to discretize the fractional derivative with respect to the time variable. We shall demonstrate the details of this method by means of an example in one space dimension,

$$ {}^{\mathrm{C}}D_t^{\alpha} u(x,t) + \phi(x,t) \frac{\partial^2}{\partial x^2} u(x,t) = f(x,t) \tag{2.8.4} $$

for $t \in [0,T]$ and $x \in [a,b]$, combined with appropriate initial and boundary conditions that will be discussed more precisely later.

Let $\Delta x = (b-a)/N$ and $\Delta t = T/M$ denote the step size of the discretization in the space and time axis respectively, where the values N and M are assumed to be given values defining the size of the discretization grid

$$ \{(x_i, t_j) : i = 0, 1, \ldots, N, j = 0, 1, \ldots, M\} $$

where $x_0 = a$ and $x_N = b$. Using the discretization on the space axis, the second derivative $u_{xx}(x,t)$ can be approximated by the central difference of second order

$$ \frac{1}{\Delta x^2} \begin{pmatrix} d_1 & o_1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & o_2 & d_2 \end{pmatrix} \begin{pmatrix} u(a,t) \\ u(x_1,t) \\ \vdots \\ u(x_{N-1},t) \\ u(b,t) \end{pmatrix}, \tag{2.8.5} $$

where the values d_1, o_1, o_2, d_2 are determined by the initial and boundary conditions and will be given later in this section.

While the discretization in space is done as in the case of classical diffusion or wave equations the discretization of the time component in the diffusion-wave equation (2.8.4) contains, as mentioned in the previous sections, a more complex structure because of the non-local character of fractional derivatives.

We had seen above that, in general, the discretization of ${}^{\mathrm{C}}D_t^{\alpha}u(x,t)$ can be defined by

$$\frac{1}{\Delta t^{\alpha}}\begin{pmatrix}\omega_{0,0} & & \\ \vdots & \ddots & \\ \omega_{M,0} & \dots & \omega_{M,M}\end{pmatrix}\begin{pmatrix}u(x,t_0)\\ \vdots \\ u(x,t_M)\end{pmatrix} \tag{2.8.6}$$

with some weights $\omega_{k,j}$ defined by the approximation method used. At the time-step t_k, $k = 1,\dots,M$, the values for $u(x_i,t_j)$ for $i = 0,1,\dots,N$ and $j = 0,\dots,k-1$ are known. Thus at the time step t_k the sum

$$\sum_{\nu=0}^{k-1}\omega_{k,\nu}u(x_i,t_\nu), \quad i = 0,1,\dots,N$$

can be transferred to the right hand side of the discretization scheme. Therefore at the time step t_k a tridiagonal system has to be solved, where the coefficient matrix is given by

$$\phi(x_i,t_k)\mathbf{A} + \mathbf{B}, \quad i = 0,1,\dots,N.$$

Here the matrix $\mathbf{A}$ is defined by the discretization (2.8.5) and the matrix $\mathbf{B}$ is the matrix containing only the main diagonal of the discretization (2.8.6). With the above discretization the right hand side at time step t_k is given by

$$\begin{pmatrix} r_1 \\ f(x_1,t_k) \\ \vdots \\ f(x_{N-1},t_k) \\ r_2 \end{pmatrix} + \begin{pmatrix} 0 \\ {}^{\mathrm{C}}D_t^{\alpha}(T_{n-1}[u;0])(x_1,t_k) \\ \vdots \\ {}^{\mathrm{C}}D_t^{\alpha}(T_{n-1}[u;0])(x_{N-1},t_k) \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ \sum_{\nu=0}^{k-1}\omega_{k,\nu}u(x_1,t_\nu) \\ \vdots \\ \sum_{\nu=0}^{k-1}\omega_{k,\nu}u(x_{N-1},t_\nu) \\ 0 \end{pmatrix}.$$

The matrix entries d_1, o_1, o_2, d_2 and the vector entries r_1, r_2 are determined by the initial and boundary conditions using Taylor approximation.

Denoting $\xi_1 = a$ and $\xi_2 = b$ they are given for $i = 1, 2$ at time step t_k by:

$$
\begin{aligned}
d_i &= \frac{g_i(t_k)}{\Delta x} + h_i(t_k)(-1)^i \left[\frac{1}{\Delta x^2} - \frac{\omega_{k,k}}{2\phi(\xi_i, t_k)\Delta t^\alpha} \right], \\
o_i &= (-1)^{i+1} \frac{h_i(t_k)}{\Delta x^2}, \\
r_i &= \frac{r_i(t_k)}{\Delta x} + \frac{(-1)^{i+1} h_i(t_k)}{2\phi(\xi_i, t_k)} \\
&\quad \times \left[f(\xi_i, t_k) - \frac{1}{\Delta t^\alpha} \sum_{\nu=0}^{k-1} \omega_{k,\nu} u(\xi_i, t_\nu) + D_t^\alpha (T_{n-1}[u; 0])(\xi_i, t_k)) \right].
\end{aligned}
$$

In order to complete the description, it remains to say explicitly what the coefficients in the discretization of the fractional differential operator, i.e. the entries in the matrix of Eq. (2.8.6), look like. In view of the experience available from the classical situation $\alpha = 1$ it seems most natural to use backward differentiation formulas (BDF). For the beginning, only the most simple instances, i.e. BDFs having only first order accuracy, will be considered. There are two main ways to construct this. The first one is the approach of Subsection 2.2.2, the second one is based on the ideas discussed in Section 2.4.

In the above numerical method those two schemes differ only in the weights $\omega_{k,j}$ of (2.8.6). For the method of Subsection 2.2.2 they are given by

$$
\omega_{k,k-j} = \frac{1}{\Gamma(2-\alpha)} \times \begin{cases} 1 & \text{for } j = 0, \\ (j-1)^{1-\alpha} - 2j^{1-\alpha} + (j+1)^{1-\alpha} & \text{for } 1 \le k \le j-1, \\ (k-1)^{1-\alpha} - (\alpha-1)k^{-\alpha} - k^{1-\alpha} & \text{for } k = j \ge 1. \end{cases} \tag{2.8.7}
$$

according to Eq. (2.1.11).

For the Grünwald-Letnikov approach the weights are identical on the main diagonal and any off-diagonal. Thus the first row in (2.8.6) determines the whole set of weights and is given by (see [356, 454])

$$
\omega_{k,0} = (-1)^k \binom{\alpha}{k}. \tag{2.8.8}
$$

This approach gives a first order approximation if the function $u(x,t)$ is a causal function with respect to time (i.e. $u(x,t) \equiv 0$ if $t < 0$), and if $u(x,0) = 0$ (which usually is not the case unless a homogeneous initial condition is defined). For the cases where $u(x,0) \neq 0$, additional starting weights $w_{k,0}$ need to be added to the first row in (2.8.6), given by

$$\begin{aligned} w_{k,0} &= \frac{k^{-\alpha}}{\Gamma(1-\alpha)} - (-1)^k \frac{\Gamma(\alpha)}{\Gamma(\alpha-k)\Gamma(k+1)} \\ &= \frac{k^{-\alpha}}{\Gamma(k-\alpha)} - \sum_{j=0}^{k} \omega_{j,0}, \end{aligned} \tag{2.8.9}$$

while the rest of the matrix stays as defined in (2.8.8) above (see, e.g., [356]).

While the convergence order for the time-discretization of Diethelm's method is $\mathcal{O}(\Delta t^{2-\alpha})$, the order of convergence for Lubich's method is $\mathcal{O}(\Delta t^{\alpha})$ (see, e.g., [174]). Thus for fractional diffusion equations, Diethelm's method is superior, while for fractional wave equations Lubich's approach is advantageous. In the following example in each case the theoretically better suited method is used.

2.8.2 *Examples*

Consider the very simple time-fractional diffusion problem

$${}^{\mathrm{C}}D_t^{1/2} u(x,t) - \frac{\partial^2}{\partial x^2} u(x,t) = 0 \tag{2.8.10}$$

with the initial condition

$$u(x,0) = u_0(x) = x$$

and boundary conditions

$$u(a,t) = 0, \quad u(b,t) = 1.$$

The solution on $[a,b] \times [0,T] = [0,1] \times [0,1]$ is given by $u(x,t) = x$. The algorithm described above reproduced the exact solution up to machine precision with grid parameters $N = 67$ and $M = 80$ both for the choice

of weights indicated in Eq. (2.8.7) and for the weights given according to Eq. (2.8.8).

For the second example the differential equation

$$ {}^{\mathrm{C}}D_t^{\alpha} u(x,t) - \frac{\partial^2}{\partial x^2} u(x,t) = 0 \tag{2.8.11a} $$

is solved on $[0,\pi] \times [0,10]$ with the initial condition

$$ u(x,0) = \sin(x), \quad u_t(x,0) = 0 \tag{2.8.11b} $$

and boundary conditions $u(0,t) = 0$, $u(\pi,t) = 0$. This equation can either be viewed as the problem of temperature distribution in a bar generated by a point heat source kept in the middle (if $\alpha \in (0,1)$), or (if $\alpha \in (1,2)$) as the problem of the deflection of a string subjected to a point load at the center of the string in a string vibration setting. Of course the second initial condition mentioned in Eq. (2.8.11b) must be ignored in the case $0 < \alpha < 1$. (See also [17] for related results).

In Fig. 2.1 the numerical solution is plotted for the case $\alpha = 1/2$. The number of nodes in time and space were 100 and 30 respectively. In Fig. 2.2 the solution for the classical version of the same diffusion problem (i.e. $\alpha = 1$) is shown for comparison. The obvious difference between those two cases is that the fractional case exhibits fast diffusion in the beginning and slow diffusion later on. If α is chosen smaller, the diffusion process over time becomes even slower. This result was to be expected by the analytical background described in Section 2 of [187].

In addition, in Fig. 2.3 the numerical solution for $\alpha = 3/2$ is plotted, which exhibits a behavior that can be said to be of an intermediate form between classical diffusion and classical waves, but with the diffusive part dominating the character. If we increase the parameter α then the qualitative behavior becomes closer to that of the wave equation. This is evident from Fig. 2.4, where $\alpha = 1.75$. The solution of the classical wave equation (i.e. $\alpha = 2$) is plotted in Fig. 2.5 for comparison.

The described behaviors were also reproduced by Agrawal [17] for a similar problem, but using a completely different method (namely, the exact solution was computed explicitly in the form of an infinite series, and a finite partial sum of this series was then used as an approximation to the exact solution).

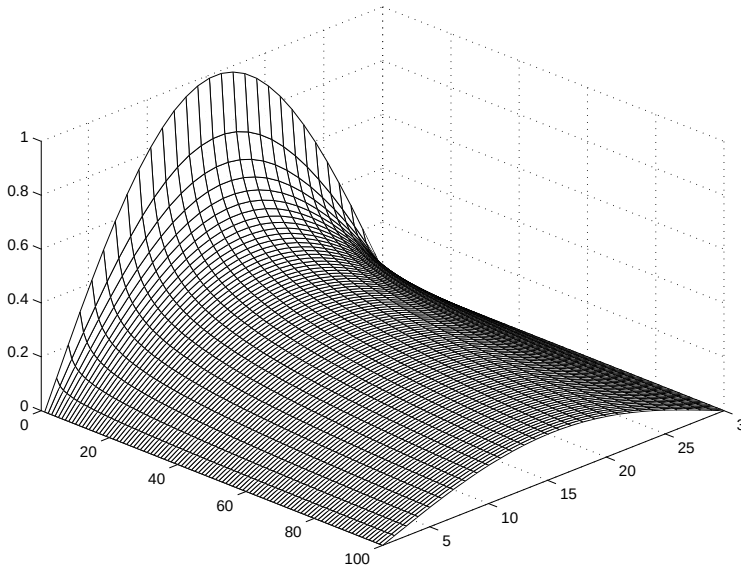

Fig. 2.1 Numerical solution of Eq. (2.8.11) for $\alpha = 1/2$.

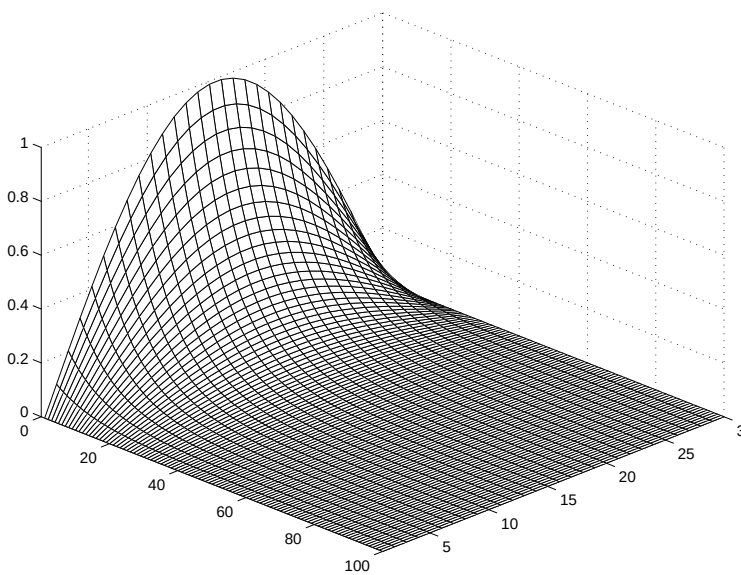

Fig. 2.2 Numerical solution of Eq. (2.8.11) for $\alpha = 1$.

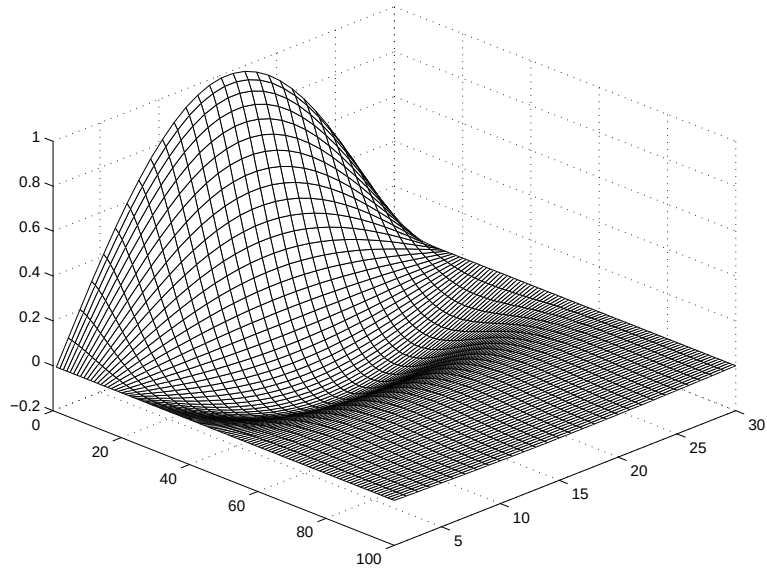

Fig. 2.3 Numerical solution of Eq. (2.8.11) for $\alpha = 3/2$.

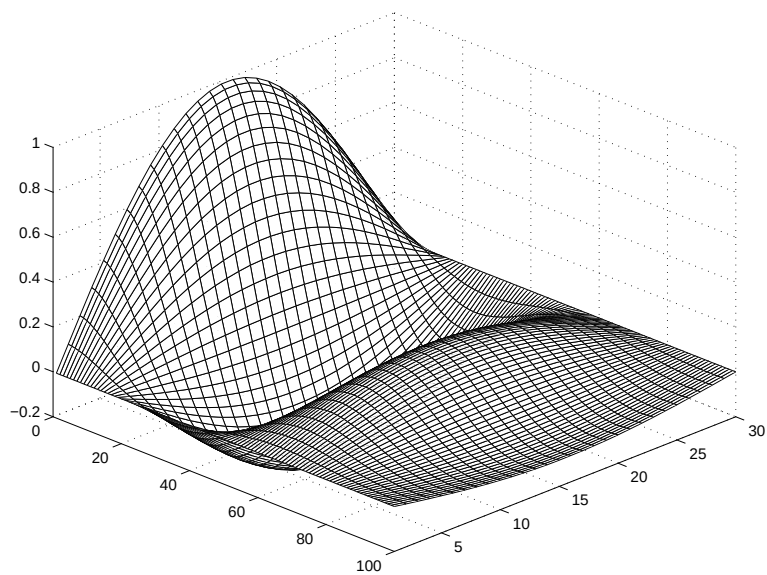

Fig. 2.4 Numerical solution of Eq. (2.8.11) for $\alpha = 7/4$.

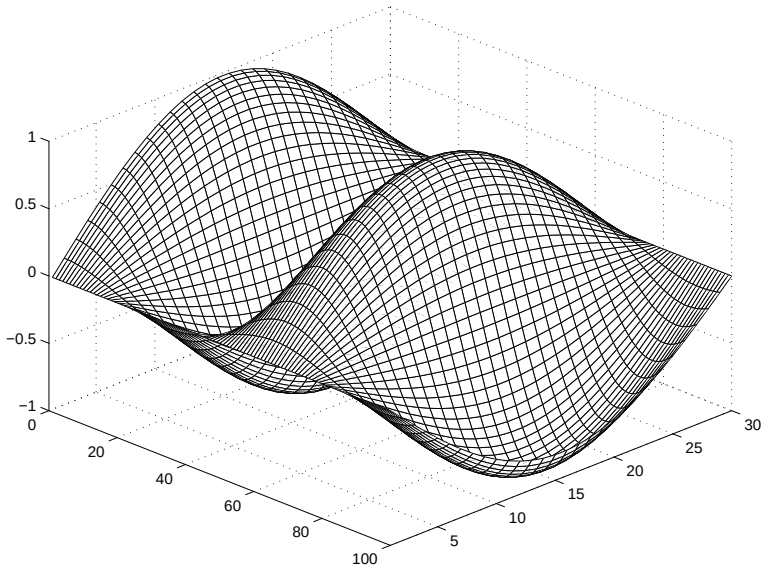

Fig. 2.5 Numerical solution of Eq. (2.8.11) for $\alpha = 2$.