SIAM J. Sci. Comput.
Vol. 41, No. 2, pp. A927–A947

FAST AND PARALLEL RUNGE–KUTTA APPROXIMATION OF FRACTIONAL EVOLUTION EQUATIONS[*]

MARINA FISCHER[†]

Abstract. We consider a linear inhomogeneous fractional evolution equation which is obtained from a Cauchy problem by replacing its first-order time derivative with Caputo's fractional derivative. The operator in the fractional evolution equation is assumed to be sectorial. By using the inverse Laplace transform a solution to the fractional evolution equation is obtained which can be written as a convolution. Based on L-stable Runge–Kutta methods a convolution quadrature is derived which allows a stable approximation of the solution. Here, the convolution quadrature weights are represented as contour integrals. On discretizing these integrals, we are able to give an algorithm which computes the solution after N time steps with step size h up to an arbitrary accuracy ε. For this purpose the algorithm only requires $\mathcal{O}(N)$ Runge–Kutta steps for a large number of scalar linear inhomogeneous ordinary differential equations and the solutions of $\mathcal{O}(\log(N)\log(\frac{1}{\varepsilon}))$ linear systems which can be done in parallel. In numerical examples we illustrate the algorithm's performance.

Key words. convolution quadrature, inverse Laplace transform, Runge–Kutta methods, subdiffusion equation, parallelizable algorithm

AMS subject classifications. 65R20, 65M15, 65Y05, 26A33

DOI. 10.1137/18M1175616

1. Introduction.

Over the last few decades the interest in linear fractional differential equations has grown, not least due to the fact that they model phenomena in applied mathematics and physics such as anomalous diffusion in viscoelastic materials. See the references in [8, 14, 20] for an overview of applications and examples. In this paper, we study the inhomogeneous fractional evolution equation

$$\mathrm{D}_t^\alpha u(t) = Au(t) + g(t) \quad \text{for} \quad t \in I := (0,T) \quad \text{and} \quad u(0) = u_0 \in X, \tag{1.1}$$

where $A : D(A) \subset X \to X$ is a closed linear densely defined operator in a Banach space $(X, \|.\|)$ and the Caputo fractional derivative D_t^α of order $\alpha \in (0,1)$ is given by

$$\mathrm{D}_t^\alpha u(t) := \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\tau)^{-\alpha}\, u'(\tau)\, \mathrm{d}\tau; \quad \text{cf. [31, 34].}$$

Although it is possible to attribute physical meaning to the initial conditions for other definitions of the fractional derivative, such as that of Riemann–Liouville, using Caputo's derivative enables us to take into account initial values whose physical interpretation is easier to see; cf. [18, 34].

We further assume the operator A in (1.1) to be *sectorial* (cf. [12, 17]), which means that

$$\begin{gathered} \text{The resolvent } R(\lambda, A) \text{ is analytic in a sector} \\ \Sigma_{\theta_0} := \{\lambda \in \mathbb{C} \setminus \{0\} \mid |\arg(\lambda)| < {}^{\pi}\!/_{2} + \theta_0\} \text{ with } \theta_0 \in (0, {}^{\pi}\!/_{2}] \\ \text{and is bounded there by } \|R(\lambda, A)\| \le {}^{C}\!/_{|\lambda|} \text{ for some real } C > 0. \end{gathered} \tag{1.2}$$

The assumption includes unbounded operators such as the Laplacian $A = \Delta$ on $\mathbb{R}^d$ or on a domain $\Omega \subset \mathbb{R}^d$ together with Dirichlet or Neumann boundary conditions [17, 33].

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section March 14, 2018; accepted for publication January 14, 2019; published electronically March 28, 2019.
http://www.siam.org/journals/sisc/41-2/M117561.html

[†] Mathematisches Institut, Heinrich-Heine-Universität, 40225 Düsseldorf, Germany (marina.fischer@uni-duesseldorf.de).

Typically, the spatial discretization of sectorial operators with finite differences or finite elements also satisfies the bound (1.2) in L^p-norms, where the constant C and the angle θ_0 are independent of the spatial discretization parameter; see, e.g., [2, 4].

On the one hand, fractional evolution equations such as (1.1) with sectorial operators have already been studied under theoretical aspects—for example, in order to give existence and uniqueness theorems concerning their solutions; cf. [6, 7, 22]. In particular, [6] should be mentioned as one of the most prominent sources of this paper's theoretical basis as it focuses on the homogeneous version of (1.1) as well as on the evolution equation using the Riemann–Liouville derivative.

On the other hand, there is a huge number of numerical methods for the time discretization of (1.1) proposed by various authors [3, 5, 8, 10, 14, 19, 20, 24, 30, 32, 36, 38, 39]. In almost every approach integrals occur which have to be approximated throughout the numerical treatment of (1.1). Some of these are contour integrals which, for example, can be derived using the Laplace transform of the fractional equation; see, e.g., [39]. Other approaches make use of the integral representation of fractional derivatives or fractional integration [14, 20]. Often the different approaches lead to convolution integrals. Here, among methods like product integration rules [14] or the recently formulated sum-of-exponentials approximation [20], convolution quadrature based on multistep [27] or Runge–Kutta methods [29] has to be mentioned in particular, as it provides a theoretical background for the numerical treatment of fractional equations such as (1.1).

In this paper, we give the solution $u(t)$ to (1.1) which can partly be represented as a convolution integral. Based on the fast Runge–Kutta approximation of inhomogeneous parabolic equations, which is presented in [23], we propose an algorithm that computes $u(t)$ at a fixed time $t = Nh$ after N steps of Runge–Kutta convolution quadrature and with step size h, up to an arbitrary accuracy ε. In order to do so the algorithm requires

$$\mathcal{O}\left(N\right) \text{ steps}$$

of an implicit L-stable Runge–Kutta time discretization for ordinary differential equations of the form $y'(t) = \lambda y(t) + g(t)$, which can be done in parallel for $\mathcal{O}\left(\log(1/\varepsilon)\right)$ complex parameters λ and for the entries of the inhomogeneity g. Furthermore, the algorithm requires the solution of only

$$\mathcal{O}\left(\log(N)\log(1/\varepsilon)\right) \text{ linear systems}$$

of the form $(\lambda^\alpha \mathrm{Id} - A)x = y$, all of which can be treated in parallel. If the number of steps N is large, the number of linear systems is thus reduced noticeably.

Therefore, the algorithm especially offers a fast approximation to the solution of (1.1) on a short subinterval around the fixed time t or at a relatively small number of selected time points. However, it is not useful for computing all values $u_1, \dots, u_N$; see, e.g., [14, 19, 20, 28, 29, 30, 36] for algorithms with this purpose.

Before focusing on numerical considerations related to (1.1), section 2 is concerned with finding the solution of (1.1) itself. Using some theoretical results on solution operators, we find that the solution contains a convolution of an operator-valued function with the inhomogeneity g. Hence, in section 3 we review convolution quadrature based on Runge–Kutta methods. Using Cauchy's integral formula, we give a contour integral representation of the occurring convolution weights whose discretization along hyperbolas is studied in section 4. Based on the results of these sections, we describe the fast and parallel algorithm in section 5 and give an extension to systems with a mass matrix. Section 6 concludes the paper by illustrating the algorithm's performance in some numerical experiments.

2. Solution to the fractional evolution equation.

In this first basic section we take a closer look at the fractional evolution equation and deal with its strong solution. To this end, by making use of the resolvent, we define two operators S_α and P_α that enable us to write the solution with the help of a convolution which is essential for the algorithm we propose here.

2.1. Preliminaries.

We first have a look at the operator-valued function $R(\lambda^\alpha, A)$, which plays an important role in giving the solution to (1.1): Define the angle θ_1 depending on the order α of the fractional derivative and on the angle θ_0 as

$$\theta_1 := \min\left\{\frac{\pi(1-\alpha)+2\theta_0}{2\alpha}, \frac{\pi}{2}\right\} \in \left(0, \frac{\pi}{2}\right]. \tag{2.1}$$

With this choice, on taking the αth power of $\lambda \in \Sigma_{\theta_1}$ we obtain that λ^α is contained in the resolvent's sector of analyticity Σ_{θ_0}. Therefore, $R(\lambda^\alpha, A)$ is analytic in the sector Σ_{θ_1}. Here the relation $\theta_1 \geq \theta_0$ yields $\Sigma_{\theta_0} \subset \Sigma_{\theta_1}$. Taking into account the bound (1.2), $R(\lambda^\alpha, A)$ thus satisfies

$$\|R(\lambda^\alpha, A)\| \leq \frac{C}{|\lambda|^\alpha} \quad \text{and} \quad \left\|\lambda^{\alpha-1} R(\lambda^\alpha, A)\right\| \leq \frac{C\,|\lambda|^{\alpha-1}}{|\lambda|^\alpha} = \frac{C}{|\lambda|} \tag{2.2}$$

for some real $C > 0$ and for $\lambda \in \Sigma_{\theta_1}$.

2.2. Homogeneous fractional evolution equation.

In the next two subsections we aim at giving the solution to the evolution equation (1.1) whose approximation is studied in the main part of this paper. Here, for brevity, we introduce the function $\Phi_\beta(t)$ which we define for a parameter $\beta \geq 0$ as

$$\Phi_\beta(t) := \begin{cases} \frac{1}{\Gamma(\beta)}\, t^{\beta-1}, & t > 0, \\ 0, & t \leq 0, \end{cases}$$

where $\Gamma(\beta)$ is the Gamma function [31]. It can be verified that this function satisfies the semigroup property $(\Phi_\alpha * \Phi_\beta)(t) = \Phi_{\alpha+\beta}(t)$ for $\alpha,\ \beta \geq 0$.

Definition 2.1. *A function $u \in C(I, X)$ that satisfies $u \in C(I, D(A))$ and $\Phi_{1-\alpha} * (u - u_0) \in C^1(I, X)$ and that fulfils* (1.1) *is called a strong solution to* (1.1)*; cf.* [6].

In order to give a representation of the solution we first consider the homogeneous version of the fractional evolution equation (1.1) which reads as

$$\mathrm{D}_t^\alpha u(t) = Au(t) \quad \text{for} \quad t \in I \quad \text{with} \quad u(0) = u_0. \tag{2.3}$$

Applying the Riemann–Liouville fractional integral [31], which means convolving each side of the evolution equation with Φ_α, we find that (2.3) is equivalent to the Volterra equation

$$u(t) = u_0 + (\Phi_\alpha * Au)(t) \quad \text{for} \quad t \in I.$$

Now we can give the following important definition; cf. [6, 35].

Definition 2.2. *A family $\{S_\alpha(t)\}_{t\geq 0} \subset \mathcal{B}(X)$ is called a solution operator for (2.3) if the following conditions are satisfied:*

(a) *$S_\alpha(t)$ is strongly continuous for $t \geq 0$ and $S_\alpha(0) = \mathrm{Id}$.*

(b) *$S_\alpha(t)D(A) \subset D(A)$ and $AS_\alpha(t)u_0 = S_\alpha(t)Au_0$ for all $u_0 \in D(A)$ and $t \geq 0$.*

(c) *The resolvent equation*

$$S_\alpha(t)u_0 = u_0 + (\Phi_\alpha * AS_\alpha u_0)(t) \tag{2.4}$$

holds for all $u_0 \in D(A)$ and $t \geq 0$.

If $S_\alpha(t)$ moreover admits an analytic extension to a sector $\Sigma_{\rho_0 - \pi/2}$ for some $\rho_0 \in (0, \pi/2]$ and if for each $\rho < \rho_0$ and $\omega > 0$ there is a constant $C > 0$ such that the bound

$$\|S_\alpha(t)\| \leq C\,\mathrm{e}^{\omega \mathrm{Re}(t)} \quad \text{for} \quad t \in \Sigma_{\rho - \pi/2}$$

is fulfilled, then the solution operator $S_\alpha(t)$ is of analyticity type $(\rho_0, 0)$.

In [35, Prop. 1.1] it is shown that (2.3) is well-posed in the sense of [6, Def. 2.2] iff it admits a solution operator. Furthermore, if a solution operator exists, then it is unique. In [6, Thm. 2.14] the existence of solution operators for (2.3) is characterized as follows.

Theorem 2.3. *A linear closed densely defined operator A generates an analytic solution operator $S_\alpha(t)$ of type $(\rho_0, 0)$ with $\rho_0 \in (0, \pi/2]$ iff $\lambda^\alpha \in \rho(A)$ for each $\lambda \in \Sigma_{\rho_0}$ and for any $\omega > 0$, $\rho < \rho_0$, there is a constant $C > 0$ such that*

$$\left\|\lambda^{\alpha-1} R\left(\lambda^\alpha,\, A\right)\right\| \leq \frac{C}{|\lambda|} \quad \text{for} \quad \lambda \in \Sigma_\rho.$$

With the findings in the preliminaries, we can thus deduce that (2.3) has a unique solution operator $\{S_\alpha(t)\}_{t\geq 0}$ which is of analyticity type $(\theta_1, 0)$ and which, as proven in [6], can be written as the inverse Laplace transform [11, 37] of the operator-valued function $R(\lambda^\alpha,\, A)$. This means that it is

$$\begin{aligned} S_\alpha(t) &:= \mathscr{L}^{-1}\left\{\lambda^{\alpha-1} R\left(\lambda^\alpha,\, A\right),\ t\right\} \\ &= \frac{1}{2\pi i}\int_\Gamma \mathrm{e}^{\lambda t}\,\lambda^{\alpha-1} R\left(\lambda^\alpha,\, A\right)\,\mathrm{d}\lambda, \end{aligned}$$

with a contour Γ in Σ_{θ_1}, going to infinity with an acute angle to the negative real half-axis and oriented counterclockwise. It can be verified that we have $S_\alpha(t)u_0 \in D(A)$ for any $u_0 \in X$ and $t > 0$. In addition, the bound

$$\|AS_\alpha(t)\| \leq C\,\mathrm{e}^{\omega t}\left(1 + t^{-\alpha}\right)$$

holds for $t > 0$, $C > 0$, and $\omega > 0$ [6, Prop. 2.15]. Now it is easy to see that $S_\alpha(t)u_0$ satisfies the conditions in Definition 2.1 and hence is a strong solution to (2.3) for $t \in I$ and for each $u_0 \in X$.

2.3. Inhomogeneous fractional evolution equation. After having discussed the homogeneous case, we now focus on the inhomogeneous evolution equation

$$\mathrm{D}_t^\alpha u(t) = Au(t) + g(t) \quad \text{for} \quad t \in I \quad \text{with} \quad u(0) = 0. \tag{2.5}$$

In order to give its strong solution we define another operator $P_\alpha(t) \in \mathcal{B}(X)$ for $t \geq 0$ as

$$\begin{aligned} P_\alpha(t) &:= \mathscr{L}^{-1}\left\{R\left(\lambda^\alpha,\ A\right),\ t\right\} \\ &= \frac{1}{2\pi i}\int_\Gamma \mathrm{e}^{\lambda t}\, R\left(\lambda^\alpha,\ A\right)\,\mathrm{d}\lambda, \end{aligned} \tag{2.6}$$

where again Γ is a counterclockwise oriented contour contained in Σ_{θ_1}, going to infinity with an acute angle to the negative real half-axis. This definition is possible due to the bound (2.2) which also yields $\|P_\alpha(t)\| \leq Ct^{\alpha-1}\mathrm{e}^{\omega t}$ for $t > 0$ and $\omega > 0$; cf. [11].

Although $P_\alpha(t)$ is not a solution operator in the sense of Definition 2.2, it has some properties in common with $S_\alpha(t)$ which we list next.

Lemma 2.4. *The following assertions hold for the operator $P_\alpha(t)$:*
(a) *$P_\alpha(t)$ is strongly continuous for $t \geq 0$.*
(b) *$P_\alpha(t)u_0 \in D(A)$ for $t > 0$ and for all $u_0 \in X$.*
(c) *$AP_\alpha(t)u_0 = P_\alpha(t)Au_0$ for all $u_0 \in D(A)$.*

Here, properties (a) and (b) are shown in [6]. Helpful for a proof of property (c) is Proposition 1.1.7 in [1].

Theorem 2.5. *Assume that $g \in C^1(I, D(A))$. Then the unique strong solution to* (2.5) *is given by*

$$u(t) = \left(P_\alpha * g\right)(t) \quad \textit{for} \quad t \in I. \tag{2.7}$$

Proof. (cf. [22, 33]) *Uniqueness.* Let $u_1,\ u_2 \in C(I, X)$ be two different strong solutions to (2.5). Then $w := u_1 - u_2 \in C(I, X)\,,\ w(0) = 0$, and $\mathrm{D}_t^\alpha w(t) = Aw(t)$ for $t \in I$. The unique strong solution is thus given by $w(t) = S_\alpha(t)w(0) = 0$, which yields $u_1 = u_2$.

Existence. We show that $u(t)$ satisfies the properties in Definition 2.1. It follows from Lemma 2.4 and [1, Prop. 1.3.4] that $P_\alpha * g \in C(I, D(A))$. Using the convolution theorem for Laplace transforms [11] the equality $\Phi_{1-\alpha} * P_\alpha * g = S_\alpha * g$ can easily be verified. On using [35, Prop. 1.2] and since $g \in C^1(I, D(A))$, we have that $S_\alpha * g$, and therefore $\Phi_{1-\alpha} * P_\alpha * g$, is in $C^1(I, D(A))$. The strong continuity of $P_\alpha(t)$ implies $u(0) = 0$.

Laplace transforming the solution $u(t)$ and using the convolution theorem yields for $\mathrm{Re}(\lambda) > 0$

$$\mathscr{L}\left\{u(t),\ \lambda\right\} = \mathscr{L}\left\{P_\alpha(t),\ \lambda\right\}\mathscr{L}\left\{g(t),\ \lambda\right\} = R\left(\lambda^\alpha,\ A\right)G(\lambda),$$

where $G(\lambda)$ denotes the Laplace transform of $g(t)$. Because of the equivalence between (2.5) and the Volterra equation $u(t) = \left(\Phi_\alpha * g\right)(t) + \left(\Phi_\alpha * Au\right)(t)$, we know from [35, Prop. 1.2] that (2.5) is satisfied by

$$\widetilde{u}(t) = \left(\Phi_\alpha * S_\alpha g(0)\right)(t) + \left(\Phi_\alpha * S_\alpha * g'\right)(t) \quad \text{for} \quad t \in I,$$

such that for $\mathrm{Re}(\lambda) > 0$ we have

$$\begin{aligned} &\mathscr{L}\left\{\left(\Phi_\alpha * S_\alpha g(0)\right)(t) + \left(\Phi_\alpha * S_\alpha * g'\right)(t),\ \lambda\right\} \\ &= \mathscr{L}\left\{\Phi_\alpha(t),\ \lambda\right\}\mathscr{L}\left\{S_\alpha(t),\ \lambda\right\}g(0) + \mathscr{L}\left\{\Phi_\alpha(t),\ \lambda\right\}\mathscr{L}\left\{S_\alpha(t),\ \lambda\right\}\mathscr{L}\left\{g'(t),\ \lambda\right\} \\ &= \lambda^{-1}R\left(\lambda^\alpha,\ A\right)g(0) + \lambda^{-1}R\left(\lambda^\alpha,\ A\right)\left(-g(0) + \lambda G(\lambda)\right) \\ &= R\left(\lambda^\alpha,\ A\right)G(\lambda). \end{aligned}$$

It follows from the uniqueness of the Laplace transform that $u(t) = \widetilde{u}(t)$ and hence $u(t)$ satisfies (2.5). ◻

Combining the results concerning the strong solutions of (2.3) and (2.5), we can thus deduce that the strong solution to (1.1) is given by

$$u(t) = S_\alpha(t)u_0 + (P_\alpha * g)(t) \quad \text{for} \quad t \in I.$$

Remark 2.6.

(a) In the case $A \in \mathcal{B}(X)$ this corresponds to the well-known formula

$$u(t) = \mathrm{E}_{\alpha,1}(At^\alpha)u_0 + \int_0^t (t-\tau)^{\alpha-1}\mathrm{E}_{\alpha,\alpha}(A(t-\tau)^\alpha)g(\tau)\,\mathrm{d}\tau \quad (\text{cf. [34]}),$$

where $\mathrm{E}_{\alpha,\beta}(t)$ with $\alpha, \beta > 0$ is the Mittag–Leffler function; see, e.g., [31, 34].

(b) Instead of assuming $g \in C^1(\mathbb{R}_+, D(A))$ in Theorem 2.5, it is even sufficient to consider $g \in W^{1,1}(I, D(A))$; cf. [35, Prop. 1.2].

3. Approximation of the convolution.

In this section we provide the basis for the computations in the fast algorithm. In order to do so, we briefly review Runge–Kutta schemes and Runge–Kutta based convolution quadrature and give two representations of the convolution quadrature weights.

3.1. Runge–Kutta schemes.

In the following we consider an implicit Runge–Kutta method with s stages and coefficients $a_{ij},\ b_j,\ c_i$ for $i, j = 1, \ldots, s$, which is of (classical) order $p \geq 1$ and stage order $\widetilde{p} \leq p$. We denote the Runge–Kutta matrix by $\mathcal{A} = (a_{ij})_{i,j=1}^s$ and the row vector of the weights by $b^T = (b_j)_{j=1}^s$. Hence, the stability function is given by

$$r(z) := 1 + z\,q(z)\mathbb{1},$$

where $\mathbb{1} := (1, \ldots, 1)^T$ and the row vector $q(z)$ is defined as

$$q(z) := b^T\left(\mathrm{Id} - z\mathcal{A}\right)^{-1}.$$

As is well known for a pth order Runge–Kutta scheme, the stability function is a rational approximation of order p to the exponential function [16].

We assume the Runge–Kutta scheme to be L-stable, which means that the stability function satisfies the criterion for A-stability, i.e.,

$$|r(z)| \leq 1 \quad \text{for all } \mathrm{Re}(z) \leq 0,$$

as well as

$$\lim_{\mathrm{Re}(z)\to-\infty} r(z) = 0.$$

Here the first property guarantees that the whole negative half-plane is contained in the region of stability.

Additional to the L-stability, we make the following extra assumptions on the Runge–Kutta scheme which we list next.

Assumptions 3.1. *The L-stable Runge–Kutta scheme fulfills the following:*

(a) *The row vector of the weights equals the last row in the Runge–Kutta matrix.*
(b) *The Runge–Kutta matrix $\mathcal{A}$ is invertible.*
(c) *The eigenvalues of the Runge–Kutta matrix $\mathcal{A}$ have positive real part.*

Here assumptions (a) and (b) imply

$$r(z) = b^T \mathcal{A}^{-1} \left(\mathrm{Id} - z\mathcal{A}\right)^{-1} \mathbb{1} = e_s^T \left(\mathrm{Id} - z\mathcal{A}\right)^{-1} \mathbb{1},$$

whereas (c) means that the spectrum $\sigma(\mathcal{A})$ is contained in Σ_{θ_1}. The assumptions listed above are in particular satisfied by the Radau IIA family of Runge–Kutta methods; see [16]. It is well known that an s-stage Radau IIA method is of order $2s-1$ and stage order s [16]. In the numerical experiments in section 6 we make use of these methods of orders 1, 3, and 5.

3.2. Runge–Kutta based convolution quadrature. From now on we restrict our considerations to the inhomogeneous evolution equation with zero initial data (2.5) whose strong solution is then solely given as a convolution. Because every equation in the form (1.1) can be easily transformed into an equation with zero initial data, this does not limit the applicability of our algorithm.

To construct a convolution quadrature we insert the contour integral representation (2.6) of P_α into (2.7). Interchanging the integrals yields

$$u(t) = \frac{1}{2\pi i} \int_\Gamma R\left(\lambda^\alpha,\, A\right) y_\lambda(t)\, \mathrm{d}\lambda, \tag{3.1}$$

where $y_\lambda(t) := \int_0^t \mathrm{e}^{\lambda(t-\tau)} g(\tau)\, \mathrm{d}\tau$ is the solution to the inhomogeneous differential equation $y'(t) = \lambda y(t) + g(t)$ with $y(0) = 0$; see [33]. Here, following the idea of [29], we discretize this initial value problem by a Runge–Kutta method that satisfies Assumptions 3.1, insert the approximation into (3.1), and simplify the result using Cauchy's integral formula. Thus, we obtain the Runge–Kutta based convolution quadrature via a generating function as it is formulated in [29].

Proposition 3.2. *Consider an s-stage Runge–Kutta method that satisfies Assumptions* 3.1*. Then at time $t_N = Nh$, where N is the number of steps and h is the step size with $Nh \le T$, we get the approximation to the solution of* (2.5) *by*

$$u_N = h\, \left(e_s^T \otimes \mathrm{Id}\right) \sum_{n=0}^{N-1} W_n\, G_{N-1-n},$$

where $G_n = (g(t_n + c_k h))_{k=1}^s$ is a column vector and W_n is the convolution quadrature weight given as the nth coefficient of the generating function

$$h \sum_{n=0}^{\infty} W_n \zeta^n = \frac{h}{2\pi i} \int_\Gamma \left(\Delta(\zeta) - h\lambda \mathrm{Id}\right)^{-1} \otimes R\left(\lambda^\alpha,\, A\right)\, \mathrm{d}\lambda = R\left(\left(\frac{\Delta(\zeta)}{h}\right)^\alpha,\, A\right) \tag{3.2}$$

with

$$\Delta(\zeta) := \left(\mathcal{A} + \frac{\zeta}{1-\zeta} \mathbb{1} b^T\right)^{-1}.$$

As is shown in [29], the convolution quadrature weights satisfy, for $h \le h_0$ with a sufficiently small h_0 and constants $C > 0,\ \gamma \ge 0$, the estimate

$$\|W_n\| \le C\, (nh)^{\alpha-1}\, e^{\gamma\, nh} \quad \text{for} \quad n \ge 1,$$

where for $n = 0$ the same bound holds as for $n = 1$, which implies the stability of the approximation. Furthermore, under some assumptions on the function g and its derivatives, the convolution quadrature is convergent of order $\min(p,\ \widetilde{p} + 1 + \alpha)$.

THEOREM 3.3. *Assume* (1.2) *and consider a Runge–Kutta method of order* p *and stage order* $\widetilde{p}$ *that satisfies Assumptions* 3.1. *Then the error of the convolution quadrature at* $t_N = Nh$ *is bounded for* $h \le h_0$ *with a sufficiently small* h_0 *by*

$$\|u_N - u(t_N)\| \le C\, h^p \sum_{l=0}^{\widetilde{p}} \left(1 + t_N^{\alpha + l - p}\right) \left\| g^{(l)}(0) \right\| \\ + C\left(h^p + h^{\widetilde{p}+1+\alpha} \left|\log(h)\right|\right) \left(\sum_{l=\widetilde{p}+1}^{p-1} \left\| g^{(l)}(0)\right\| + \max_{0 \le \tau \le t_n} \left\| g^{(p)}(\tau)\right\| \right).$$

The constants C *and* h_0 *depend only on the Runge–Kutta method, on the constants in* (1.2)*, and on the length of the time interval. In particular, they are affected neither by* N *and* h *with* $Nh \le T$ *nor by* $g \in C^p(\mathbb{R}_+, X)$.

The proof of Theorem 3.3 can be found in [29]. It is also shown there that the equation

$$(\Delta(\zeta) - z\mathrm{Id})^{-1} = \mathcal{A}\left(\mathrm{Id} - z\mathcal{A}\right)^{-1} + \sum_{n=1}^{\infty} r(z)^{n-1} \left(\mathrm{Id} - z\mathcal{A}\right)^{-1} \mathbb{1} b^T \left(\mathrm{Id} - z\mathcal{A}\right)^{-1} \zeta^n$$

holds under Assumptions 3.1. On inserting this into Cauchy's integral formula (3.2), we get a representation of the last "row" of the convolution quadrature weights, i.e.,

$$w_n := \left(e_s^T \otimes \mathrm{Id}\right) W_n = \frac{1}{2\pi i} \int_\Gamma r(h\lambda)^n\, q(h\lambda) \otimes R\left(\lambda^\alpha,\ A\right)\, \mathrm{d}\lambda \quad \text{for} \quad n \ge 0. \tag{3.3}$$

The norm of the weights w_n is bounded by the same term as the one of the weights W_n except for a different constant C. Together with the fact that $r(h\lambda)^n$ is an approximation to $\exp(nh\lambda)$, this shows that (3.3) can be interpreted as a discrete analogue of the operator P_α (2.6).

Now we are able to reformulate Proposition 3.2 using the contour integral representation instead of generating functions.

PROPOSITION 3.4. *For an* s*-stage Runge–Kutta method that satisfies Assumptions* 3.1 *we get at time* $t_N = Nh$ *the approximation*

$$u_N = h \sum_{n=0}^{N-1} w_n\, G_{N-1-n} = h \sum_{n=0}^{N-1} \left(\frac{1}{2\pi i} \int_\Gamma r(h\lambda)^n\, q(h\lambda) \otimes R\left(\lambda^\alpha,\ A\right)\, \mathrm{d}\lambda \right) G_{N-1-n},$$

where $\Gamma \in \Sigma_{\theta_1}$ *is chosen such that* $h\Gamma \cap \sigma\left(\mathcal{A}^{-1}\right) = \emptyset$.

In the fast algorithm both Propositions 3.2 and 3.4 will be applied.

Remark 3.5. Instead of basing convolution quadrature on Runge–Kutta methods, it is also possible to construct it using multistep methods, e.g., backward differentiation formulas. For further details see [27, 36].

4. Discretization of the contour integrals.

Relevant to the fast algorithm is the question of how to discretize the contour integrals (3.3) along suitable complex contours. In this section this issue will be discussed following the approach in [23], [26], and [36].

4.1. Quadrature using hyperbolas. Since it is neither useful to discretize the integrals w_n for every n with the same contour nor efficient to define a different contour for each integral, we consider the sequence of fast-growing intervals

$$I_\ell = \left[\Lambda^{\ell-1}h,\, \Lambda^\ell h\right), \quad \ell \geq 1,$$

where $\Lambda > 1$ is an integer, and fix one contour Γ_ℓ for each such interval. If $nh \in I_\ell$, then (3.3) is approximated by a quadrature along Γ_ℓ. Here, e.g., $\Lambda = 5$ showed up as a good choice in the numerical experiments.

Recalling that the operator A is sectorial, we choose the contours Γ_ℓ as hyperbolas which are parameterized by the maps

$$\gamma_\ell :\ \mathbb{R} \to \Gamma_\ell, \quad x \mapsto \gamma_\ell(x) = \mu_\ell\,(1 + \sin(ix - \varphi)), \quad \ell \geq 1, \tag{4.1}$$

where $\varphi > 0$ is the angle between the upper imaginary axis and the asymptote on its left. Assuming $\varphi \in (0, \theta)$ with $\theta < \theta_1$ the hyperbolas Γ_ℓ hence are all contained in the sector of analyticity Σ_{θ_1}. In contrast to the angle, the scale parameter $\mu_\ell > 0$ depends on ℓ. Furthermore, the hyperbolas are chosen such that the singularities of $r(h\lambda)^n\, q(h\lambda)$ lie to the right of the contour.

Since we consider an L-stable Runge–Kutta scheme, the integrand in (3.3) decays rapidly for $\mathrm{Re}(\lambda) \to -\infty$, which makes the integral well suited for approximations by the trapezoidal rule; see [39]. Applying this rule to (3.3) with contour Γ_ℓ and choosing the number of quadrature points on Γ_ℓ, independent of ℓ, as $2K + 1$ yields

$$w_n \approx \sum_{k=-K}^{K} \omega_k^{(\ell)}\, r(h\lambda_k^{(\ell)})^n\, q(h\lambda_k^{(\ell)}) \otimes R\left((\lambda_k^{(\ell)})^\alpha,\, A\right) \quad \text{for} \quad nh \in I_\ell, \tag{4.2}$$

where the weights $\omega_k^{(\ell)}$ and quadrature points $\lambda_k^{(\ell)}$ are given by

$$\omega_k^{(\ell)} = \frac{\tau}{2\pi i}\, \gamma_\ell'(x_k) \quad \text{and} \quad \lambda_k^{(\ell)} = \gamma_\ell(x_k),$$

with $x_k = k\tau$ and the step length parameter τ.

4.2. Theoretical error bound. To determine an appropriate number of quadrature points we take a look at the error $E(\tau, K, h, n)$ of the contour integral approximation in (4.2). For the case of the hyperbola, the following theoretical error bound, which shows exponential convergence, can be found in [36].

THEOREM 4.1. *There are positive constants $C, d, c_0, \ldots, c_4$ and b, so that, if $1 \leq b\mu t \leq n$, the quadrature error in* (4.2) *for a hyperbola* (4.1) *at $t = nh \leq T$ is bounded by*

$$\|E(\tau, K, h, n)\| \leq C\, t^{\alpha-1}\, (\mu t)^{1-\alpha} \left(\frac{\mathrm{e}^{c_0 \mu t}}{\mathrm{e}^{2\pi d/\tau} - 1} + \mathrm{e}^{(c_1 - c_2 \cosh(K\tau))\mu t} + \mathrm{e}^{c_3 \mu t} \left(1 + \frac{c_4 \cosh(K\tau)\mu t}{n/2}\right)^{-n/2} \right). \tag{4.3}$$

Here, the first summand in the estimate corresponds to the error in the discretization of the integral (3.3) which we obtain by an infinite quadrature series over all integers k and where the contour Γ_ℓ is parameterized over the real line with an integrand holomorphic in a horizontal strip $\{z \in \mathbb{C} \mid |\mathrm{Im}(z)| \le d\}$; see [25]. Truncating the series to the terms $-K \le k \le K$ leads to the other two terms in the sum (4.3).

By choosing the step size τ so small that $c_0\mu t - ({}^{2\pi d}/_{\tau}) \le \log(\varepsilon)$, the first term in (4.3) becomes $\mathcal{O}\left(\varepsilon\, t^{\alpha-1}\right)$. In order to do so we require an asymptotic proportionality $\frac{1}{\tau} \sim \log\left({}^{1}/_{\varepsilon}\right) + \mu t$. If we additionally choose μ such that

$$\frac{a_1}{\Lambda}\log\left(\frac{1}{\varepsilon}\right) \le \mu t \le a_1 \log\left(\frac{1}{\varepsilon}\right) \tag{4.4}$$

for an arbitrary $a_1 > 0$ and with $\Lambda > 1$, and if $c_1 - c_2\cosh(K\tau) \le -{}^{\Lambda}/_{a_1}$, then the next summand as well is $\mathcal{O}\left(\varepsilon\, t^{\alpha-1}\right)$. Here, the latter condition holds with $\cosh(K\tau) = a_2$ for a sufficiently large constant a_2. With the above choice of τ this yields $K \sim \log\left({}^{1}/_{\varepsilon}\right)$. For $n \ge a_3 \log({}^{1}/_{\varepsilon})$ and a sufficiently large constant a_3, the third term then becomes smaller than $\varepsilon\, t^{\alpha-1}$.

All in all, we get the following bound for the required number of quadrature points on the hyperbola; cf. [36].

THEOREM 4.2. *If* (4.4) *is satisfied for* $a_1 > 0$ *and* $\Lambda > 1$*, then in* (4.2) *a quadrature error bounded in norm by* $\varepsilon\, t^{\alpha-1}$ *is obtained for*

$$K = \mathcal{O}\left(\log\left(\frac{1}{\varepsilon}\right)\right).$$

The estimate holds for $n \ge a_3\log({}^{1}/_{\varepsilon})$ *with a sufficiently large constant* $a_3 > 0$*. Here the number* K *is independent of* ℓ*,* n*, and* h *with* $nh \le T$*.*

Indeed we do not obtain a small error bound for the first few $n < a_3 \log({}^{1}/_{\varepsilon})$ as shown in numerical experiments in [36]. Therefore, these first quadrature weights are computed in a different way that is described in section 5.1.

4.3. Choice of the parameters.

Concerning the inverse Laplace transform and its discretization along hyperbolas, strategies for choosing parameters are derived in [26] and [39] under the assumption $t \in [t_0, \Lambda t_0)$. An error estimate is obtained there that is explicit in all constants involved. Achieving such an error estimate for the discretization (4.2) is much more complicated as the convolution quadrature weights w_n are not given as inverse Laplace transforms but can just be interpreted as a discrete analogue of those; cf. section 3.2. Since for large n and small h with $nh \le T$ the estimate in Theorem 4.1 tends to an expression of the same type as the error estimate for the approximation of the inverse Laplace transform in [26], we can nevertheless in practice choose the parameters according to [26].

Therefore, for the approximation interval I_ℓ and $2K+1$ nodes on the hyperbola as in Theorem 4.2, we use the following strategy to determine parameters for the discretization:

1. Choose $\varphi = d = {}^{\theta}/_{2}$ with $\theta < \theta_1$.
2. Minimize for $0 < \rho < 1$ the expression

$$\texttt{eps}\, \epsilon_K(\rho)^{\rho-1} + \epsilon_K(\rho)^{\rho},$$

where

$$\epsilon_K(\rho) = \exp\left(\frac{-2\pi d}{a(\rho)}K\right), \quad a(\rho) = \operatorname{arccosh}\left(\frac{\Lambda}{(1-\rho)\sin(\varphi)}\right),$$

and `eps` is the machine precision.

3. Take

$$\tau = \frac{1}{K} a(\rho_{\mathrm{opt}}) \quad \text{and} \quad \mu_\ell = \frac{2\pi d\, K\, (1-\rho_{\mathrm{opt}})}{\Lambda^\ell h\, a(\rho_{\mathrm{opt}})}.$$

With this choice of parameters we are now able to draw the connection between Theorem 4.2 and the intervals I_ℓ.

If $t = nh \in I_\ell$, then $\Lambda^{\ell-1} \le n < \Lambda^\ell$, and with μ_ℓ as above and $a_1 = \frac{2\pi d(1-\rho_{\mathrm{opt}})}{a(\rho_{\mathrm{opt}})} > 0$ we obtain

$$\frac{a_1}{\Lambda} K = \frac{a_1\,\Lambda^{\ell-1}}{\Lambda^\ell} K \le \frac{a_1\, n}{\Lambda^\ell} K = \mu_\ell\, t < \frac{a_1\, \Lambda^\ell}{\Lambda^\ell} K = a_1\, K.$$

Thus, (4.4) holds. With $K = \mathcal{O}\left(\log\left({}^1\!/\!_\varepsilon\right)\right)$ Theorem 4.2 implies that the quadrature error is bounded by $\varepsilon\, t^{\alpha-1}$ for $nh \in I_\ell$ independent of ℓ, except for the first few n. Here, our choice of intervals I_ℓ comes in useful.

5. The fast and parallel algorithm.

With the results of the previous section we are now able to describe the fast algorithm for computing the solution (2.7) after N time steps with step size h.

5.1. Computing the first summands of the approximation.

We already know that the error bound in Theorem 4.2 does not hold for all convolution weights and the approximation properties of the discretized contour integral are poor for the first few weights w_n with $n \le \kappa$ (e.g., $\kappa = 20$ in our numerical experiments, or $\kappa = a_3 \log({}^1\!/\!_\varepsilon)$ asymptotically). Therefore, for $n \le \kappa$ we consider the weights W_n from Proposition 3.2 and make use of their representation as an integral over a circle with radius ρ, i.e.,

$$hW_n = \frac{1}{2\pi i} \int_{|\zeta|=\rho} \zeta^{-n-1} R\left(\left(\frac{\Delta(\zeta)}{h}\right)^\alpha, A\right) \mathrm{d}\zeta \quad \text{(see [29])},$$

whose approximation by the trapezoidal rule yields

$$hW_n \approx \frac{\rho^{-n}}{J} \sum_{j=0}^{J-1} R\left(\left(\frac{\Delta(\zeta_j)}{h}\right)^\alpha, A\right) \mathrm{e}^{-2\pi i n j/J} \quad \text{for} \quad n = 0, \dots, \kappa \tag{5.1}$$

with $\zeta_j = \rho\, \mathrm{e}^{2\pi i j/J}$ and the parameter J which we choose according to [28, 29]: Assuming that the values of the Laplace transform are computed with an accuracy ε and choosing $J = \kappa$ and $\rho^J = \sqrt{\varepsilon}$ an error of $\mathcal{O}\left(\sqrt{\varepsilon}\right)$ is obtained in (5.1). If we even choose $J \ge \kappa \log({}^1\!/\!_\varepsilon)$ and $\rho = \mathrm{e}^{-\gamma h}$ with $\gamma > 0$, then the error becomes $\mathcal{O}\left(\varepsilon\right)$.

Differently from [29], we do not use this formula to compute each W_n in an extra step but consider the sum

$$h \sum_{n=0}^{\kappa} W_n\, G_{N-1-n} \approx \sum_{n=0}^{\kappa} \frac{\rho^{-n}}{J} \sum_{j=0}^{J-1} R\left(\left(\frac{\Delta(\zeta_j)}{h}\right)^\alpha, A\right) G_{N-1-n}\, \mathrm{e}^{-2\pi i n j/J},$$

where we use the eigenvalue decomposition of the $s \times s$ matrices ${}^{\Delta(\zeta_j)}\!/\!_h$ into the product ${}^{\Delta(\zeta_j)}\!/\!_h = U_j\, \mathcal{D}_j\, U_j^{-1}$ with diagonal matrix $\mathcal{D}_j$ to obtain

$$h \sum_{n=0}^{\kappa} W_n\, G_{N-1-n} \approx \sum_{j=0}^{J-1} \left(U_j \otimes \mathrm{Id}\right) x_j. \tag{5.2}$$

Here x_j is the solution of the decoupled system

$$((\mathcal{D}_j)^{\alpha} \otimes \mathrm{Id} - \mathrm{Id}_s \otimes A)\, x_j = \left(U_j^{-1} \otimes \mathrm{Id}\right) \sum_{n=0}^{\kappa} \frac{\rho^{-n}}{J}\, G_{N-1-n}\, \mathrm{e}^{-2\pi i n j/J}. \tag{5.3}$$

In order to compute (5.2), we need to solve sJ linear systems, which can be done in parallel. Here, the J right-hand sides of (5.3) are computed in $\mathcal{O}\left(J \log(J)\right)$ operations using fast Fourier transform.

5.2. Fast convolution approximation. In order to give the fast approximation of the quadrature weights w_n for $n > \kappa$, we modify the intervals I_ℓ from section 4.1 so that they do not contain the first $\kappa + 1$ quadrature weights. This means we now consider for $\Lambda > 1$ the sequence

$$\widetilde{I}_\ell = \left[\Lambda^{\ell-1}(\kappa+1)h,\, \Lambda^{\ell}(\kappa+1)h\right), \quad \ell \geq 1,$$

where L is the smallest integer such that $N \leq \Lambda^L(\kappa + 1)$. Thus, we can split the approximation u_N into the following $L + 1$ sums:

$$u_N = h \sum_{n=0}^{N-1} w_n\, G_{N-1-n} = u_N^{(0)} + u_N^{(1)} + \cdots + u_N^{(L)},$$

where

$$u_N^{(0)} := h \sum_{n=0}^{\kappa} w_n\, G_{N-1-n} = (e_s \otimes \mathrm{Id})\ h \sum_{n=0}^{\kappa} W_n\, G_{N-1-n} \tag{5.4}$$

is approximated via (5.2). For $\ell = 1, \ldots, L$ we define the sums $u_N^{(\ell)}$ corresponding to the intervals $\widetilde{I}_\ell$ and hyperbolas Γ_ℓ given in (4.1) as

$$u_N^{(\ell)} := h \sum_{nh \in \widetilde{I}_\ell} w_n\, G_{N-1-n} = h \sum_{nh \in \widetilde{I}_\ell} \frac{1}{2\pi i} \int_{\Gamma_\ell} \left(r(h\lambda)^n\, q(h\lambda) \otimes R\left(\lambda^{\alpha},\, A\right)\right) G_{N-1-n}\, \mathrm{d}\lambda,$$

where we now choose $\mu_\ell = \frac{2\pi d\, K\, (1-\rho_{\mathrm{opt}})}{\Lambda^{\ell}\, (\kappa+1)\, h\, a(\rho_{\mathrm{opt}})}$ according to the results in section 4.3.

On discretizing the convolution quadrature weights w_n via (4.2), we obtain the approximation $U_N^{(\ell)}$ to $u_N^{(\ell)}$ that, with $m_\ell := \Lambda^\ell(\kappa + 1)$ for $\ell = 0, \ldots, L - 1$ and $m_L := N$, can be formulated as

$$\begin{aligned}
U_N^{(\ell)} &= h \sum_{n=m_{\ell-1}}^{m_\ell - 1} \sum_{k=-K}^{K} \omega_k^{(\ell)} \left(r(h\lambda_k^{(\ell)})^n\, q(h\lambda_k^{(\ell)}) \otimes R\left((\lambda_k^{(\ell)})^{\alpha},\, A\right)\right) G_{N-1-n} \\
&= h \sum_{k=-K}^{K} \omega_k^{(\ell)} \sum_{n=N-m_\ell}^{N-1-m_{\ell-1}} \left(r(h\lambda_k^{(\ell)})^{N-1-n}\, q(h\lambda_k^{(\ell)}) \otimes R\left((\lambda_k^{(\ell)})^{\alpha},\, A\right)\right) G_n \\
&= \sum_{k=-K}^{K} \omega_k^{(\ell)}\, r(h\lambda_k^{(\ell)})^{m_{\ell-1}}\, R\left((\lambda_k^{(\ell)})^{\alpha},\, A\right) y_k^{(\ell)},
\end{aligned}$$

where

$$y_k^{(\ell)} = h \sum_{n=N-m_\ell}^{N-1-m_{\ell-1}} \left(r(h\lambda_k^{(\ell)})^{N-1-m_{\ell-1}-n}\, q(h\lambda_k^{(\ell)}) \otimes \mathrm{Id}\right) G_n$$

is the Runge–Kutta approximation to the solution at time $t = (N - m_{\ell-1})h$ of the linear initial-value problem

$$y'(t) = \lambda_k^{(\ell)}\, y(t) + g(t), \quad y((N - m_\ell)h) = 0. \tag{5.5}$$

Thus, in order to compute $U_N^{(1)}, \dots, U_N^{(L)}$, we require $N - \kappa - 1$ Runge–Kutta steps for a total of $2K+1$ differential equations (5.5). Bringing to mind that $K = \mathcal{O}\left(\log(1/\varepsilon)\right)$, this means the computation of $\mathcal{O}\left(N \log(1/\varepsilon)\right)$ Runge–Kutta steps.

With the solution $x_k^{(\ell)}$ of the linear system of equations

$$\left((\lambda_k^{(\ell)})^\alpha \,\mathrm{Id} - A\right) x_k^{(\ell)} = y_k^{(\ell)}, \tag{5.6}$$

the approximation $U_N^{(\ell)}$ is obtained as the linear combination

$$U_N^{(\ell)} = \sum_{k=-K}^{K} c_k^{(\ell)}\, x_k^{(\ell)}, \quad \text{where} \quad c_k^{(\ell)} = \omega_k^{(\ell)}\, r(h\lambda_k^{(\ell)})^{m_{\ell-1}}. \tag{5.7}$$

Therefore, in general, it is required to solve $2K+1$ linear systems (5.6) when computing $U_N^{(\ell)}$. If the inhomogeneity (2.7) though is real-valued, the following consideration shows that this number reduces to $K+1$: Since the quadrature points $\lambda_k^{(\ell)}$ lie symmetrically with respect to the real axis, this kind of symmetry is inherited by $y_k^{(\ell)}$ and $x_k^{(\ell)}$. Together with $\omega_{-k}^{(\ell)} = \omega_k^{(\ell)}$ this leads to the cancellation of the imaginary parts in (5.7), and therefore only the sum of the real parts of half the terms needs to be computed.

In addition to the previously stated number of Runge–Kutta steps, we thus only require $\mathcal{O}\left(\log(N)\log(1/\varepsilon)\right)$ solutions to the linear system (5.6) to obtain an approximation with an accuracy up to ε, which can be done in parallel.

In a nutshell, the fast algorithm consists of the following steps:

1. Compute (5.4) using (5.2) and solving sJ systems in parallel.
2. Compute in parallel for $\ell = 1, \dots, L$ (5.7) with (5.5) and (5.6). Here (5.5) can be computed in parallel for k and for the entries of the inhomogeneity g, whereas (5.6) can be parallelized over k.
3. Obtain the approximation U_N to u_N as the sum

$$U_N = U_N^{(0)} + U_N^{(1)} + \cdots + U_N^{(L)}.$$

Remark 5.1. If we consider an additional operator M and assume that $(\lambda M - A)^{-1}$ is analytic in Σ_{θ_0} and bounded there by $\left\|(\lambda M - A)^{-1}\right\| \leq C/|\lambda|$ for some $C > 0$, then we can extend the algorithm to the fractional evolution equation

$$M \mathrm{D}_t^\alpha u(t) = Au(t) + g(t) \quad \text{for} \quad t \in I \quad \text{with} \quad u(0) = u_0,$$

which can be found in the following numerical experiments. In (5.6) now $x_k^{(\ell)}$ is the solution to the linear system $\left((\lambda_k^{(\ell)})^\alpha\, M - A\right) x_k^{(\ell)} = y_k^{(\ell)}$, and the left side of (5.3) becomes $\left((\mathcal{D}_j)^\alpha \otimes M - \mathrm{Id} \otimes A\right) x_j$.

6. Numerical experiments. We give the following examples in order to illustrate the application and behavior of the fast and parallel algorithm and to show the paper's main results.

6.1. Fractional evolution equation with 2×2 matrix A.

As a first example we consider the inhomogeneous evolution equation

$$\mathrm{D}_t^{1/2} u(t) = Au(t) + g(t) \quad \text{for} \quad t > 0 \quad \text{with} \quad A = \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix} \tag{6.1}$$

under the initial data $u(0) = 0$. By using the function `fracdiff` in Maple 18, we compute the inhomogeneity $g(t)$ such that the exact solution of (6.1) is given by

$$u(t) = \begin{bmatrix} \sin(2t)^6 \\ \left(\frac{1}{2} - \frac{1}{2}\cos\left(\sqrt{5}t\right)\right)^6 \end{bmatrix}.$$

Thus, we obtain an inhomogeneity that is five times continuously differentiable in I, and each of these derivations vanishes at $t = 0$. Just like the inhomogeneity in the next example, $g(t)$ includes Fresnel integrals (see [15]), which we compute using the implementation provided in [9].

To have a look at the convergence, we use $\Lambda = 5$ in the numerical experiment and test two different numbers of quadrature points ($K = 10$ and $K = 25$). The parameters for the hyperbolas Γ_ℓ are chosen as described in section 4.3, where we set $\theta_1 = \pi/2$ according to (2.1). To avoid errors arising from the poor approximation of the first few quadrature weights, we compute $u_N^{(0)}$ with $\kappa = 20$ as described in section 5.1, where we set $J = 160$.

Figure 6.1 shows the absolute errors $\|U_N - u(Nh)\|_\infty$ at time $t = Nh = 10$ for $K = 10$ and $K = 25$ and for the Radau IIA methods of orders 1, 3, and 5 versus the step size h. Here the dashed lines represent the theoretical orders of convergence for the Runge–Kutta based convolution quadrature which are 1, 3, and 4.5 respectively; cf. Theorem 3.3. For the choice of $K = 10$ we notice an error saturation which is due to the insufficient approximation of the contour integrals. This saturation is prevented if we increase the number of quadrature points to, e.g., $K = 25$ as in the right figure.

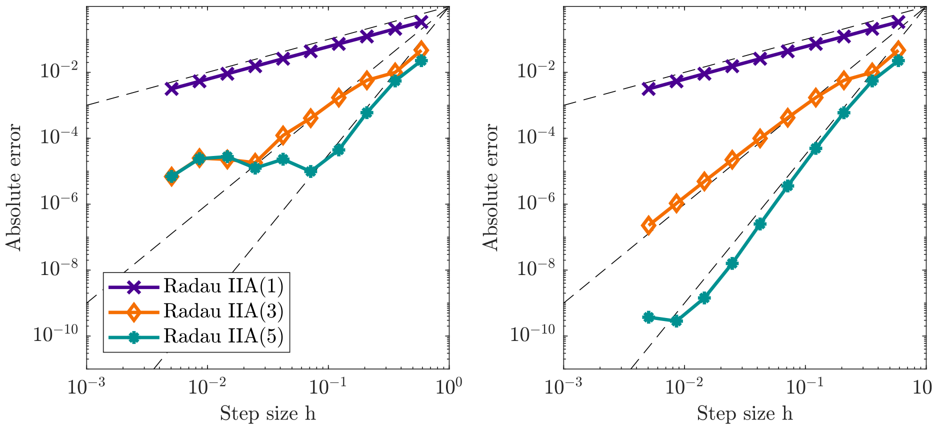

FIG. 6.1. *Absolute error versus time step size h for three different Runge–Kutta methods, with $K = 10$ (left) and $K = 25$ (right).*

6.2. Subdiffusion in three space dimensions with periodic boundary conditions.

In order to illustrate the findings in section 5, we now study a subdiffusion equation in three space dimensions,

$$\mathrm{D}_t^{1/2} u(x,y,z,t) = \Delta_{x,y,z} u(x,y,z,t) + g(x,y,z,t) \quad \text{for} \quad x,y,z \in (0,2\pi)\,,\ t>0, \tag{6.2}$$

subject to the initial data $u(x,y,z,0)=0$ and with 2π-periodic boundary conditions. We consider the inhomogeneity

$$g(x,y,z,t) = h_-(x,y,z) f_1(t) + h_+(x,y,z) f_2(t),$$

where f_1, f_2 are functions that solely depend on time and

$$h_\mp(x,y,z) := \cos(x) + \cos(y) + \cos(z) \mp (\sin(x) + \sin(y) + \sin(z)),$$

such that the exact solution to (6.2) is given by

$$u(x,y,z,t) = h_-(x,y,z)\,\sin(\pi t) - h_+(x,y,z)\,(\cos(\pi t) - 1).$$

For the spatial discretization we make use of the following compact finite difference scheme, which we, for reasons of clarity, state here for a one-dimensional problem.

Consider the differential equation $\Delta_x w(x) = f(x)$ with $x \in (0,2\pi)$ under 2π-periodic boundary conditions and an equidistant grid $x_j = j\eta$ in $[0,2\pi]$ for a step size η. Then the scheme

$$\tfrac{1}{\eta^2}(w(x_{j-1}) - 2w(x_j) + w(x_{j+1})) \approx \tfrac{1}{12} f(x_{j-1}) + \tfrac{5}{6} f(x_j) + \tfrac{1}{12} f(x_{j+1}) \tag{6.3}$$

gives a fourth-order approximation in space [21].

The execution time in seconds which is required to approximate the solution at time $t = 123.45$ with varying numbers of time steps N is shown in Figure 6.2. Here, we distinguish between the time we need to compute the summand $u_N^{(0)}$ via (5.2), to perform the necessary Runge–Kutta steps (5.5) using Radau IIA(5), and to solve the linear systems (5.6). Since the evaluation of the inhomogeneity in (5.5) is very expensive, we precompute the functions f_1 and f_2 for all required values of t.

As in the previous example, we set $\Lambda = 5$; however, as we are interested in time rather than accuracy, we choose $\kappa = 12$, $J = \kappa + 2$, and $K = 20$. The computations are distributed to a pool of eight workers using the MATLAB Parallel Computing Toolbox. We clearly see the linear growth of the time used for the Runge–Kutta steps and the logarithmic growth related to the linear systems, whereas the computational work of the first summands of the approximation stays constant.

Figure 6.3 illustrates the strong scaling of the algorithm. Here, we consider a fixed problem size but increase the number of parallel workers from 3 to a total of 32 workers. As in the previous figure, we compute the solution at $t = 123.45$, this time with a fixed number of steps $N = 10^5$, and distinguish again between the time we need for the approximation of $u_N^{(0)}$, for the Runge–Kutta steps, and for the solution to the linear systems. In the ideal case the time for the computation with j workers would be j times faster than the computation with one worker as it is shown in the dashed line. Whereas the Runge–Kutta steps and the solutions of the linear systems show a good parallelization, the computation of the summand $u_N^{(0)}$ indicates an insufficient load balancing.

6.3. Time-fractional Schrödinger equation with transparent boundary conditions. The last example illustrates the algorithm's extensibility to more complicated problems. We consider the homogeneous fractional Schrödinger equation on the real line

$$\mathrm{D}_t^{\alpha} v(x,t) = i\Delta_x v(x,t) \quad \text{for} \quad x \in \mathbb{R}, \quad t > 0, \tag{6.4}$$

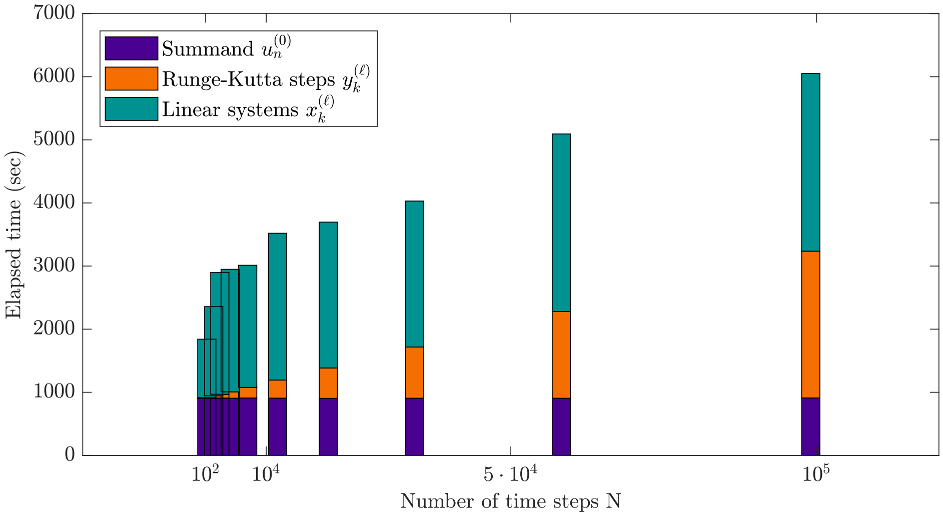

Fig. 6.2. *Elapsed time in seconds versus the number of time steps N.*

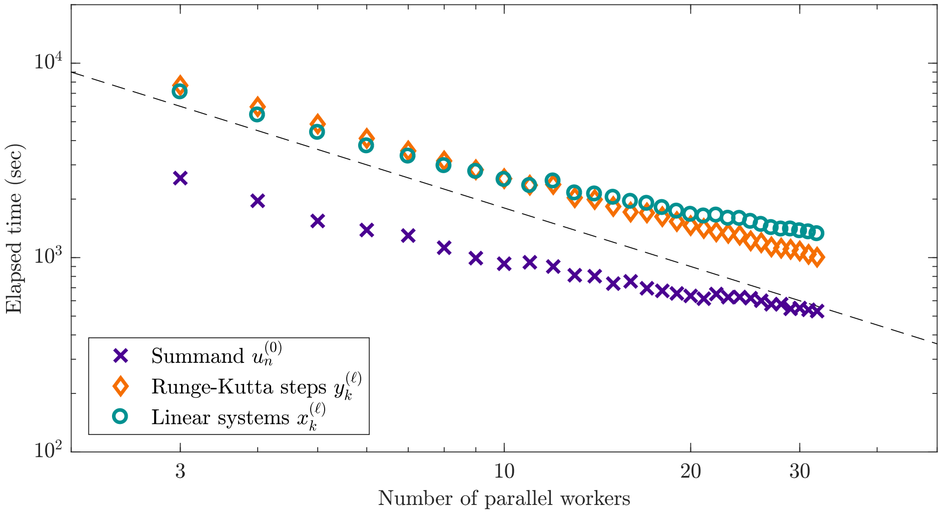

Fig. 6.3. *Elapsed time in seconds versus the number of workers we distribute the computations to. The dashed line is the ideal line for strong scaling.*

with an arbitrary α in $(0, 1)$ and the asymptotic condition $v(x,t) \to 0$ for $x \to \pm\infty$. The fractional equation is complemented with the initial condition $v(x, 0) = u_0(x)$ for $x \in \mathbb{R}$. As is known, the eigenvalues of the operator $i\Delta_x$ lie on the negative imaginary axis [17]; hence, strictly speaking, the operator $i\Delta_x$ is not sectorial. However, the bound (2.2) is the one that is decisive for the application of the algorithm. Thus, to make sure that $R\,(\lambda^\alpha, i\Delta_x)$ for $\lambda \in \Sigma_{\theta_1}$ is analytic in $\Sigma_{\pi/2}$, we have to choose θ_1 according to (2.1).

As mentioned in the beginning of section 3.2, we first transform (6.4) into an equivalent differential equation with zero initial data by substituting $v = u + u_0$.

Thus, we get the inhomogeneous fractional Schrödinger equation

$$\text{(6.5)}\qquad \mathrm{D}_t^\alpha u(x,t) = i\Delta_x u(x,t) + i\Delta_x u_0(x) \quad \text{for} \quad x \in \mathbb{R},\ t>0 \quad \text{with} \quad u(x,0)=0.$$

In order to reduce the computation to a finite domain $[-a,a]$, we assume $\operatorname{supp}(u_0(x)) \subseteq [-a,a]$, so that we can neglect the inhomogeneity $i\Delta_x u_0(x)$ in the following derivation of the transparent boundary conditions. The method used for this purpose is well known; see, e.g., [30] for the Schrödinger equation in the case of a discrete spatial Laplacian and [13, 36] for subdiffusion equations in the continuous case. It is based on the Laplace transform of (6.5) which reads

$$\text{(6.6)}\qquad \lambda^\alpha U(x,\lambda) = i\Delta_x U(x,\lambda) \quad \text{for} \quad \operatorname{Re}(\lambda)>0 \quad \text{and} \quad |x|>a.$$

We only show the derivation of the right boundary condition at point a in detail since the left condition at $-a$ is found analogously. This means we consider (6.6) with $x>a$, which, spatially discretized with (6.3) on a grid $x_j = j\eta$ with $j \geq N+1$ and $x_N = a$ for a fixed $N \in \mathbb{N}$ and step size η, becomes

$$\varphi U^R(x_{j-1}, s\lambda) + \psi U^R(x_j,\lambda) + \varphi U^R(x_{j+1},\lambda) = 0 \quad \text{for } j \geq N+1,$$

with $\varphi = \frac{1}{12}\lambda^\alpha - \frac{i}{\eta^2}$ and $\psi = \frac{5}{6}\lambda^\alpha + \frac{2i}{\eta^2}$. The solution to the characteristic equation $\varphi z^2 + \psi z + \varphi = 0$ yields

$$z_{1,2} = \frac{-\psi \pm \sqrt{\psi^2 - 4\varphi^2}}{2\varphi},$$

where $|z_2|>1$ and $|z_1|<1$. Therefore, the decaying solution of the above three-term recursion is given by $U^R(x_j,\lambda) = U^I(a,\lambda)\, z_1^{N-j}$ for $j \geq N+1$. Here U^I belongs to the spatial discretization of the Laplace transform of (6.5) in the inner domain $[-a,a]$.

To obtain a transparent boundary, we set the conditions

$$U^I(a,\lambda) = U^R(a,\lambda) \quad \text{and} \quad \delta_\nu U^I(a,\lambda) = \delta_\nu U^R(a,\lambda)$$

with

$$\delta_\nu U(a,\lambda) = \delta_\nu U(x_N,\lambda) = \tfrac{1}{h}\left(U(x_{N+1},\lambda) - U(x_N,\lambda)\right).$$

This leads to the equation

$$\frac{h}{z_1(\lambda)-1}\delta_\nu U^I(a,\lambda) = U^I(a,\lambda).$$

Transforming back gives the transparent boundary condition at $x=a$, which can be formulated as the "discrete Neumann-to-Dirichlet operator"

$$\text{(6.7)}\qquad u(a,t) = \int_0^t f(t-\tau)\,\delta_\nu u(a,\tau)\,\mathrm{d}\tau,$$

where $f(t)$ is now the function with Laplace transform $F(\lambda) = \frac{h}{z_1(\lambda)-1}$.

In our numerical experiment we consider the fractional Schrödinger equation (6.5) with initial value $u_0(x) = 10\exp(-(4x)^2 + 10ix)$ and with a fractional derivative of order $\alpha = 3/4$. Hence, according to our previous considerations, we set $\theta_1 = \pi/6$ to make sure that (2.2) is satisfied for $\lambda \in \Sigma_{\theta_1}$.

In order to check the correctness of the transparent boundary conditions (6.7), we compute the solution of (6.4) on the finite domain $[-2, 2]$ with 801 spatial grid points and a fixed number of quadrature points $K = 50$. As in the first example, we set $\Lambda = 5$ and $\kappa = 20$, where this time we choose $J = 4\kappa$. To obtain a reference solution, we consider the finite domain $[-8, 8]$, a spatial grid with 1601 grid points, and set $\kappa = 60$ and $J = 4\kappa$ in the computation of the first convolution weights and $K = 110$ for the nodes of the integration contour. In both cases, we make use of the convolution quadrature based on the Radau IIA(5) method and choose the parameters for the hyperbolas according to section 4.3 with θ_1, as discussed above. Figure 6.4 shows the modulus of the computed solution and the absolute error at times $t = 0.05, 0.1, \ldots, 1$ with time step size $h = 0.00025$ at each grid point in the domain $[-2, 2]$.

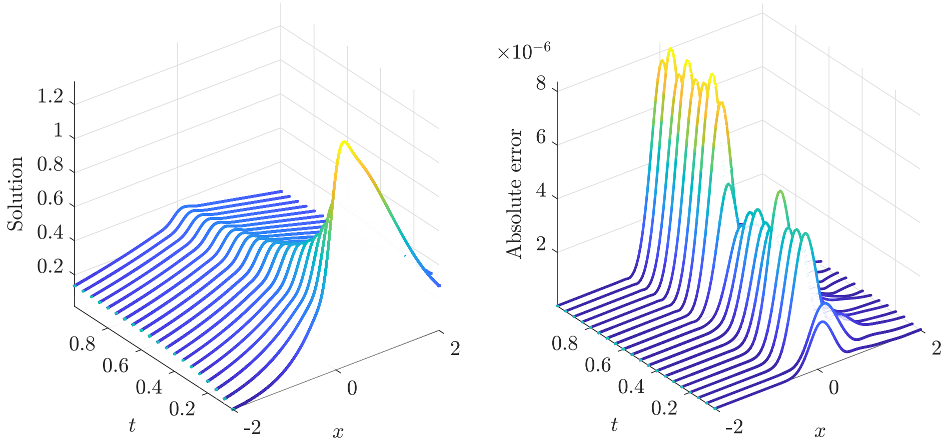

Fig. 6.4. *Modulus of the solution (left) and absolute error (right) at time* $t = 0.05, 0.1, \ldots, 1$ *on the domain* $[-2, 2]$ *with* $K = 50$.

Whereas in Figure 6.4 we considered the fixed parameter $K = 50$, Figure 6.5 plots the absolute error of the computed solution versus the number of quadrature points. Here, the left side of the figure shows the solution at time $t = 0.5$ with 6000 time steps and different numbers of spatial grid points (201, 401, 801). For the right side we compute the solution at time $t = 0.5$ with 401 spatial grid points and take a look at different numbers of time steps N. For the reference solutions we again use $\Lambda = 5$, $\kappa = 110$, and $J = 4\kappa$ and compute them such that the step size of the spatial discretization and the time step size are identical to those of the computed solution. Thus, the error that we see in Figure 6.5 is dominated by the discretization of the contour integral (4.2). The exponential convergence that we expected due to Theorem 4.1 can be clearly seen. Here the error saturation that can be observed is related to the chosen value of κ.

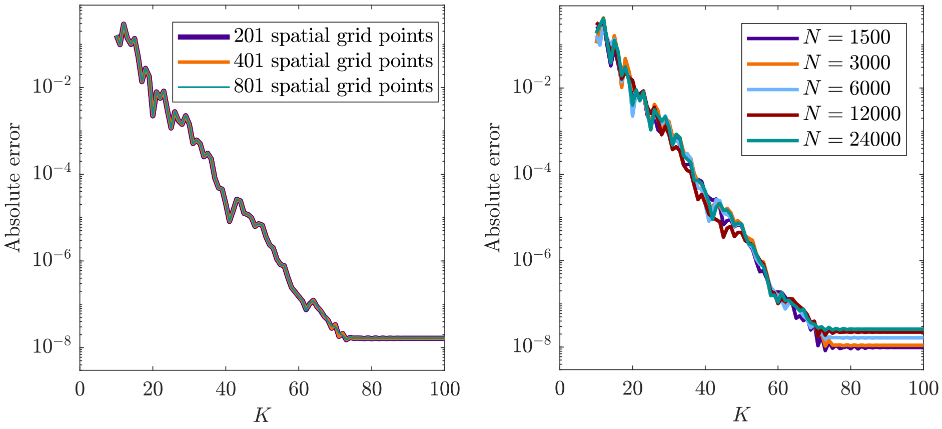

FIG. 6.5. *Absolute error at time* $t = 0.5$ *versus* K *for different pairs of time steps and numbers of spatial grid points. In the left figure we use* $N = 6000$ *and in the right figure* 401 *grid points for the space discretization.*

Acknowledgment. I thank Achim Schädle for the suggestion to write this paper, the helpful discussions during its preparation, and for kindly providing his algorithm for the fast Runge–Kutta approximation of inhomogeneous parabolic equations.

REFERENCES

[1] W. Arendt, C. Batty, M. Hieber, and F. Neubrandner, *Vector-Valued Laplace Transforms and Cauchy Problems*, Birkhäuser, Basel, 2001, https://doi.org/10.1007/978-3-0348-5075-9.

[2] A. Ashyralyev and P. Sobolevskii, *Well-Posedness of Parabolic Difference Equations*, Birkhäuser, Basel, 1994, https://doi.org/10.1007/978-3-0348-8518-8.

[3] D. Baffet and J. S. Hesthaven, *A kernel compression scheme for fractional differential equations*, SIAM J. Numer. Anal., 55 (2017), pp. 496–520, https://doi.org/10.1137/15M1043960.

[4] N. Bakaev, V. Thomée, and L. Wahlbin, *Maximum-norm estimates for resolvents of elliptic finite element operators*, Math. Comp., 72 (2002), pp. 1597–1610, https://doi.org/10.1090/S0025-5718-02-01488-6.

[5] L. Banjai, C. Lubich, and J. M. Melenk, *Runge-Kutta convolution quadrature for operators arising in wave propagation*, Numer. Math., 119 (2011), pp. 1–20, https://doi.org/10.1007/s00211-011-0378-z.

[6] E. G. Bazhlekova, *Fractional Evolution Equations in Banach Spaces*, Ph.D. thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2001, https://doi.org/10.6100/IR549476.

[7] C. Chen and M. Li, *On fractional resolvent operator functions*, Semigroup Forum, 80 (2010), pp. 121–142, https://doi.org/10.1007/s00233-009-9184-7.

[8] E. Cuesta, C. Lubich, and C. Palencia, *Convolution quadrature time discretization of fractional diffusion-wave equations*, Math. Comp., 75 (2006), pp. 673–696, https://doi.org/10.1090/S0025-5718-06-01788-1.

[9] J. D'Errico, *FresnelS and FresnelC*, https://de.mathworks.com/matlabcentral/fileexchange/28765-fresnels-and-fresnelc, 2012 (accessed 2017/08/21).

[10] K. Diethelm, J. M. Ford, N. J. Ford, and M. Weilbeer, *Pitfalls in fast numerical solvers for fractional differential equations*, J. Comput. Appl. Math., 186 (2006), pp. 482–503, https://doi.org/10.1016/j.cam.2005.03.023.

[11] G. Doetsch, *Einführung in Theorie und Anwendung der Laplace-Transformation*, Birkhäuser, Basel, 1970, https://doi.org/10.1007/978-3-0348-4141-2.

[12] K.-J. Engel and R. Nagel, *One-Parameter Semigroups for Linear Evolution Equations*, Springer-Verlag, New York, 2000, https://doi.org/10.1007/b97696.

[13] G. H. Gao, Z. Z. Sun, and Y. N. Zhang, *A finite difference scheme for fractional sub-diffusion equations on an unbounded domain using artificial boundary conditions*, J. Comput. Phys., 231 (2012), pp. 2865–2879, https://doi.org/10.1016/j.jcp.2011.12.028.

[14] R. Garrappa, *Trapezoidal methods for fractional differential equations: Theoretical and computational aspects*, Math. Comput. Simul., 110 (2015), pp. 96–112, https://doi.org/10.1016/j.matcom.2013.09.012.

[15] W. Gautschi, *Error function and Fresnel integrals*, in Handbook of Mathematical Functions, M. Abramowitz and I. Stegun, eds., Dover, New York, 1972, Ch. 7, https://doi.org/10.1115/1.3625776.

[16] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations* II, 2nd ed., Springer-Verlag, Berlin, Heidelberg, 2002, https://doi.org/10.1007/978-3-642-05221-7.

[17] D. Henry, *Geometric Theory of Semilinear Parabolic Equations*, Springer-Verlag, Berlin, Heidelberg, 1981, https://doi.org/10.1007/BFb0089647.

[18] N. Heymans and I. Podlubny, *Physical interpretation of initial conditions for fractional differential equations with Riemann-Liouville fractional derivatives*, Rheol. Acta, 45 (2006), pp. 765–771, https://doi.org/10.1007/s00397-005-0043-5.

[19] B. A. Jacobs, *High-order compact finite difference and Laplace transform method for the solution of time-fractional heat equations with Dirichlet and Neumann boundary conditions*, Numer. Methods Partial Differential Equations, 32 (2016), pp. 1184–1199, https://doi.org/10.1002/num.22046.

[20] S. Jiang, J. Zhang, Q. Zhang, and Z. Zhang, *Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations*, Commun. Comput. Phys., 21 (2017), pp. 650–678, https://doi.org/10.4208/cicp.OA-2016-0136.

[21] S. K. Lele, *Compact finite difference schemes with spectral-like resolution*, J. Comput. Phys., 103 (1992), pp. 16–42, https://doi.org/10.1016/0021-9991(92)90324-R.

[22] K. Li, J. Peng, and J. Jia, *Cauchy problems for fractional differential equations with Riemann-Liouville fractional derivatives*, J. Funct. Anal., 263 (2012), pp. 476–510, https://doi.org/10.1016/j.jfa.2012.04.011.

[23] M. López-Fernández, C. Lubich, C. Palencia, and A. Schädle, *Fast Runge-Kutta approximation of inhomogeneous parabolic equations*, Numer. Math., 102 (2005), pp. 277–291, https://doi.org/10.1007/s00211-005-0624-3.

[24] M. López-Fernández, C. Lubich, and A. Schädle, *Adaptive, fast, and oblivious convolution in evolution equations with memory*, SIAM J. Sci. Comput., 30 (2008), pp. 1015–1037, https://doi.org/10.1137/060674168.

[25] M. López-Fernández and C. Palencia, *On the numerical inversion of the Laplace transform of certain holomorphic mappings*, Appl. Numer. Math., 51 (2004), pp. 289–303, https://doi.org/10.1016/j.apnum.2004.06.015.

[26] M. López-Fernández, C. Palencia, and A. Schädle, *A spectral order method for inverting sectorial Laplace transforms*, SIAM J. Numer. Anal., 44 (2006), pp. 1332–1350, https://doi.org/10.1137/050629653.

[27] C. Lubich, *Discretized fractional calculus*, SIAM J. Math. Anal., 17 (1986), pp. 704–719, https://doi.org/10.1137/0517050.

[28] C. Lubich, *Convolution quadrature and discretized operational calculus.* II, Numer. Math., 52 (1988), pp. 413–425, https://doi.org/10.1007/BF01462237.

[29] C. Lubich and A. Ostermann, *Runge-Kutta methods for parabolic equations and convolution quadrature*, Math. Comp., 60 (1993), pp. 105–131, https://doi.org/10.2307/2153158.

[30] C. Lubich and A. Schädle, *Fast convolution for nonreflecting boundary conditions*, SIAM J. Sci. Comput., 24 (2002), pp. 161–182, https://doi.org/10.1137/S1064827501388741.

[31] F. Mainardi, *Fractional Calculus and Waves in Linear Viscoelasticity*, Imperial College Press, London, 2012, https://doi.org/10.1142/p614.

[32] W. McLean and V. Thomée, *Numerical solution via Laplace transforms of a fractional order evolution equation*, J. Integral Equations Appl., 22 (2010), pp. 57–94, https://doi.org/10.1216/JIE-2010-22-1-57.

[33] A. Pazy, *Semigroups of Linear Operators and Applications to Partial Differential Equations*, Springer-Verlag, New York, 1983, https://doi.org/10.1007/978-1-4612-5561-1.

[34] I. Podlubny, *Fractional Differential Equations*, Academic Press, San Diego, CA, 1999.

[35] J. Prüss, *Evolutionary Integral Equations and Applications*, Birkhäuser, Basel, 1993, https://doi.org/10.1007/978-3-0348-8570-6.

[36] A. Schädle, M. López-Fernández, and C. Lubich, *Fast and oblivious convolution quadrature*, SIAM J. Sci. Comput., 28 (2006), pp. 421–438, https://doi.org/10.1137/050623139.

[37] M. R. Spiegel, *Laplace-Transformationen*, McGraw-Hill, Düsseldorf, Germany, 1977.

[38] M. Stynes, E. O'Riordan, and J. L. Gracia, *Error analysis of a finite difference method on graded meshes for a time-fractional diffusion equation*, SIAM J. Numer. Anal., 55 (2017), pp. 1057–1079, https://doi.org/10.1137/16M1082329.

[39] J. A. C. Weideman and L. N. Trefethen, *Parabolic and hyperbolic contours for computing the Bromwich integral*, Math. Comp., 76 (2007), pp. 1341–1356, https://doi.org/10.1090/S0025-5718-07-01945-X.