CrossMark

Efficient high order algorithms for fractional integrals and fractional differential equations

L. Banjai[1] · M. López-Fernández[2]

Abstract

We propose an efficient algorithm for the approximation of fractional integrals by using Runge–Kutta based convolution quadrature. The algorithm is based on a novel integral representation of the convolution weights and a special quadrature for it. The resulting method is easy to implement, allows for high order, relies on rigorous error estimates and its performance in terms of memory and computational cost is among the best to date. Several numerical results illustrate the method and we describe how to apply the new algorithm to solve fractional diffusion equations. For a class of fractional diffusion equations we give the error analysis of the full space-time discretization obtained by coupling the FEM method in space with Runge–Kutta based convolution quadrature in time.

Mathematics Subject Classification 65R20 · 65L06 · 65M15 · 26A33 · 35R11

1 Introduction

Fractional Differential Equations (FDEs) have nowadays become very popular for modeling different physical processes, such as anomalous diffusion [26] or viscoelasticity [1,25]. In the present paper we develop a fast and memory efficient method to

✉ L. Banjai
  l.banjai@hw.ac.uk

  M. López-Fernández
  lopez@mat.uniroma1.it

1 The Maxwell Institute for Mathematical Sciences, School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh EH14 4AS, UK

2 Dipartimento di Matematica Guido Castelnuovo, Sapienza Università di Roma, Piazzale Aldo Moro 5, 00185 Rome, Italy

compute the fractional integral

$$\mathcal{I}^{\alpha}[f](t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} f(s)\, ds, \tag{1}$$

for a given $\alpha \in (0,1)$. A standard discretization of (1) is obtained by convolution quadrature (CQ) based on a Runge–Kutta scheme [5,18]

$$\mathcal{I}^{\alpha}[f](t_n) \approx \sum_{j=0}^n \boldsymbol{\omega}_{n-j} \mathbf{f}_j, \tag{2}$$

where the convolution weights $\boldsymbol{\omega}_n$ can be expressed as, see Lemma 9,

$$\boldsymbol{\omega}_n = \frac{h \sin(\pi\alpha)}{\pi} \int_0^\infty x^{-\alpha} \mathbf{e}_n(-hx)\, dx, \tag{3}$$

with $\mathbf{e}_n(\cdot)$ a function that depends on the Runge–Kutta scheme. For discretizations based on linear multistep methods, see [14].

To compute up to time $T = Nh$ using formula (2) requires $O(N)$ memory and $O(N^2)$ arithmetic operations. Algorithms based on FFT can reduce the computational complexity to $O(N \log N)$ [16] or $O(N \log^2 N)$ [9], but not the memory requirements; for an overview of FFT algorithms see [7]. Here we develop algorithms that reduce the memory requirement to $O(|\log\varepsilon| \log N)$ and the computational cost to $O(|\log\varepsilon| N \log N)$, with ε the accuracy in the computation of the convolution weights. Hence, our algorithm has the same complexity as the fast and oblivious quadratures of [19] and [23], but as we will see, a simpler construction.

The algorithms will depend on an efficient quadrature of (3) for $n \geq n_0$, with a very moderate threshold value for n_0, say $n_0 = 5$. As $\mathbf{e}_n(z) = r(z)^n \mathbf{q}(z)$ and $r(z) = e^z + O(z^{p+1})$, where p is the order of the underlying RK method, this is intimately related to the construction of an efficient quadrature for the integral representation of the convolution kernel

$$t^{\alpha-1} = \frac{1}{\Gamma(1-\alpha)} \int_0^\infty x^{-\alpha} e^{-tx} dx, \tag{4}$$

with $t \in [n_0 h, T]$. Note that as $\Gamma(1-\alpha)\Gamma(\alpha) = \pi/\sin(\pi\alpha)$, $h^{-1}\boldsymbol{\omega}_n$ is an approximation of $\frac{1}{\Gamma(\alpha)} t^{\alpha-1}$, i.e., the kernel of (1).

Even though we eventually only require the quadrature for (3), we begin with developing a quadrature formula for (4) for a number of reasons: the calculation for (4) is cleaner and easier to follow, such a quadrature allows for efficient algorithms that are not based on CQ, and finally once this is available the analysis for (3) is much shorter. The quadrature we develop for (4) is closely related to the one developed in [13], the main difference being our treatment of the singularity at $x = 0$ by Gauss–Jacobi quadrature and the restriction of t to the finite interval rather than

semi-infinite as used in [13]. Both these decisions allow us to substantially reduce constants in the above asymptotic estimates of memory and computational costs. Recent references [2,12,27] also consider fast computation of (1), but do not address the approximation of the convolution quadrature approximation exploiting (3). Our main contribution here is the development of an efficient quadrature to approximate (3) and its use in a fast and memory efficient scheme for computing the discrete convolution (2).

The stability and convergence properties of RK convolution quadrature are well understood, see [6,18]. This allows us to apply convolution quadrature not only to the evaluation of fractional integrals, but also to the solution of fractional subdiffusion or diffusion-wave equations of the form

$$\partial_t^\beta u - \Delta u = f, \qquad u^{(j)}(0) = 0, \ j = 0, \ldots, m - 1,$$

with $\beta \in (0, 2)$. Here, $\partial_t^\beta = \mathcal{I}^{m-\beta} \partial_t^m$, with $m = \lceil \beta \rceil$, denotes the Caputo fractional derivative. Solutions of such equations typically have low regularity at $t = 0$, but a discussion of adaptive or modified quadratures for this case is beyond the scope of the current paper. For a careful analysis of BDF2 based convolution quadrature of fractional differential equations see [8].

To our knowledge, underlying high order solvers for ODEs have been considered for the approximation of (1) only at experimental level in [2,3] and in [23], where a fast and oblivious implementation of RK based CQ is considered for more general applications than (1). The fast and oblivious quadratures of [19] and [23] have the same asymptotic complexity as our algorithm, but have a more complicated memory management structure and require the optimization of the shape of the integration contour. Our new algorithm has the advantage of being much easier to implement, as it does not require sophisticated memory management and the optimization of quadrature parameters is much simpler, and furthermore only real arithmetic is required. The new method is also much better suited for the extension to variable steps — this will be investigated in a follow up work. On the other hand, the present algorithm is specially tailored to the application to (1) and related FDEs, whereas the algorithms in [19,23] allow for a wider range of applications.

The paper is organized as follows. In Sect. 2 we develop and fully analyze a special quadrature for (1), which uses the same nodes and weights for every $t \in [n_0 h, T]$. In Sect. 3, we recall Convolution Quadrature based on Runge–Kutta methods and derive the special representation of the associated weights already stated in (3). In Sect. 4 we derive a special quadrature for (3), which uses the same nodes and weights for every $n \in [n_0, N]$, with $T = hN$. In Sect. 5 we explain how to turn our quadrature for the CQ weights into a fast and memory saving algorithm. In Sect. 6 we test our algorithm with a scalar problem and in Sect. 7 we consider the application to a fractional diffusion equation. We provide a complete error analysis of the discretization in space and time of a class of fractional diffusion equations.

2 Efficient quadrature for $t^{\alpha-1}$

In the following we fix an integer $n_0 > 0$, time step $h > 0$, and the final computational time $T > 0$. Throughout, the parameter α is restricted to the interval $(0, 1)$. We develop an efficient quadrature for (4) accurate for $t \in [n_0 h, T]$.

2.1 Truncation

First of all we truncate the integral

$$t^{\alpha-1} = \frac{1}{\Gamma(1-\alpha)} \int_0^L x^{-\alpha} e^{-tx} dx + \tau(L),$$

where $\tau(L)$ denotes the truncation error.

Lemma 1 *For $t \geq n_0 h$ and $L = A/h$ we have that*

$$|\tau(L)| \leq \frac{h^{\alpha-1}}{\Gamma(1-\alpha)} \int_A^\infty x^{-\alpha} e^{-n_0 x} dx. \tag{5}$$

Proof

$$
\begin{aligned}
|\tau(L)| &= \frac{h^{\alpha-1}}{\Gamma(1-\alpha)} \int_A^\infty x^{-\alpha} e^{-\frac{t}{h}x} dx \\
&\leq \frac{h^{\alpha-1} A^{-\alpha}}{\Gamma(1-\alpha)} \int_A^\infty e^{-n_0 x} dx \\
&= \frac{A^{-\alpha} h^{\alpha-1} e^{-n_0 A}}{n_0 \Gamma(1-\alpha)}.
\end{aligned}
$$

$\square$

Remark 2 Given a tolerance tol > 0, $|\tau(L)| \leq$ tol if

$$A + \frac{\alpha}{n_0} \log(A) \geq \frac{1}{n_0} \log\left(\frac{h^{\alpha-1}}{n_0 \Gamma(1-\alpha)\text{tol}}\right). \tag{6}$$

Assuming $A \geq 1$ we can choose

$$A = \log\left(\frac{1}{n_0 \Gamma(1-\alpha)\text{tol}}\right) + (1-\alpha) \log\left(\frac{1}{h}\right).$$

However, in practice it is advantageous to use the bound (5) to numerically find the optimal A.

2.2 Gauss–Jacobi quadrature for the initial interval

We choose an initial integration interval

$$I_0 = \frac{1}{\Gamma(1-\alpha)} \int_0^{L_0} x^{-\alpha} e^{-tx} dx,$$

along which we will perform Gauss–Jacobi integration.

Recall the Bernstein ellipse $\mathcal{E}_\varrho$, which is given as the image of the circle of radius $\varrho > 1$ under the map $z \mapsto (z+z^{-1})/2$. The largest imaginary part on $\mathcal{E}_\rho$ is $(\varrho - \varrho^{-1})/2$ and the largest real part is $(\varrho + \varrho^{-1})/2$.

Theorem 3 *Let f be analytic inside the Bernstein ellipse $\mathcal{E}_\varrho$ with $\varrho > 1$ and bounded there by M. Then the error of Gauss quadrature with weight $w(x)$ is bounded by*

$$|If - I_Q f| \le 4M \frac{\varrho^{-2Q+1}}{\varrho - 1} \int_{-1}^1 w(x) dx,$$

where $If = \int_{-1}^1 w(x) f(x) dx$ and $I_Q f = \sum_{j=1}^Q w_j f(x_j)$ is the corresponding Gauss formula, with weights $w_j > 0$.

Proof A proof of this result for $w(x) \equiv 1$ can be found in [24, Chapter 19]. The same proof works for the weighted Gauss quadrature as well. We give the details next.

First of all note that we can expand f in Chebyshev series

$$f(x) = \sum_{k=0}^\infty a_k T_k(x)$$

with $|a_k| \le 2M\varrho^{-k}$ [24, Theorem 8.1]. If we denote by $f_K(x) = \sum_{k=0}^K a_k T_k(x)$ the truncated series then

$$|f - f_K| \le \frac{2M\varrho^{-K}}{\varrho - 1}.$$

As I_Q is exact for polynomials of degree $2Q - 1$, we have that

$$|If - I_Q f| = \left| I(f - f_{2Q-1}) - I_Q(f - f_{2Q-1}) \right|$$
$$\le \frac{2M\varrho^{-2Q+1}}{\varrho - 1} \left(\int_{-1}^1 w(x) dx + \sum_{j=1}^{2Q-1} w_j \right)$$
$$= \frac{4M\varrho^{-2Q+1}}{\varrho - 1} \int_{-1}^1 w(x) dx,$$

where we have used the fact the weights are positive and integrate constants exactly.
□

Changing variables to the reference interval $[-1, 1]$ we obtain

$$I_0 = \frac{1}{\Gamma(1-\alpha)} \left(\frac{L_0}{2}\right)^{1-\alpha} \int_{-1}^{1} e^{-t(y+1)L_0/2}(y+1)^{-\alpha}dy.$$

We apply Theorem 3 to the case

$$f_0(x) = \frac{1}{\Gamma(1-\alpha)} \left(\frac{L_0}{2}\right)^{1-\alpha} e^{-t(x+1)L_0/2}, \qquad w(x) = (1+x)^{-\alpha} \qquad (7)$$

and denote

$$\tau_{\mathrm{GJ}}(Q) := If_0 - I_Q f_0 = \int_{-1}^{1} f_0(x)w(x)\, dx - \sum_{j=1}^{Q} w_j\, f_0(x_j).$$

Theorem 4 *For $t \in [0, T]$ and any $Q \geq 1$ we have the bound*

$$|\tau_{\mathrm{GJ}}(Q)| \leq \frac{4L_0^{1-\alpha}}{\Gamma(2-\alpha)} \left(1 + \frac{TL_0}{4Q}\right) \left(\frac{eTL_0}{8Q}\right)^{2Q}.$$

Proof Since f_0 in (7) is an entire function, by Theorem 3 we can estimate

$$|\tau_{\mathrm{GJ}}(Q)| \leq \frac{4}{\Gamma(1-\alpha)} \left(\frac{L_0}{2}\right)^{1-\alpha} \left(\int_{-1}^{1} (1+x)^{-\alpha}\, dx\right) \min_{\varrho>1}\left(\frac{\varrho^{-2Q+1}}{\rho-1} \max_{\xi\in\mathcal{E}_\rho} \left|e^{-t(\xi+1)L_0/2}\right|\right)$$

$$= \frac{4}{\Gamma(1-\alpha)} \frac{L_0^{1-\alpha}}{1-\alpha} \min_{\varrho>1}\left(\frac{\varrho^{-2Q+1}}{\rho-1} \max_{\xi\in\mathcal{E}_\rho} e^{-t(\operatorname{Re}\xi+1)L_0/2}\right)$$

$$= \frac{4L_0^{1-\alpha}}{\Gamma(2-\alpha)} \min_{\varrho>1}\left(\frac{\varrho^{-2Q+1}}{\rho-1} e^{t(\varrho+\varrho^{-1}-2)L_0/4}\right)$$

$$\leq \frac{4L_0^{1-\alpha}}{\Gamma(2-\alpha)} \min_{\varrho>1}\left(\frac{\varrho^{-2Q+1}}{\rho-1} e^{T(\varrho+\varrho^{-1}-2)L_0/4}\right).$$

Let $\varrho = e^\delta$ with $\delta > 0$. Then the error bound can be written as

$$|\tau_{\mathrm{GJ}}(Q)| \leq \frac{4L_0^{1-\alpha}}{\Gamma(2-\alpha)} \min_{\delta>0} \frac{e^\delta}{e^\delta-1} e^{-2Q\delta+L_0T(\cosh\delta-1)/2}.$$

We now choose δ so that it maximises the function

$$g(\delta) = 2Q\delta - L_0T\,(\cosh\delta - 1)\,/2.$$

As

$$g'(\delta) = 2Q - L_0T \sinh\delta/2, \quad g''(\delta) = -L_0T \cosh\delta/2 < 0,$$

we have a maximum at

$$2Q - L_0 T \sinh \delta/2 = 0 \implies \delta = \sinh^{-1}\left(\frac{4Q}{TL_0}\right).$$

Using the identities

$$\sinh^{-1} y = \log\left(y + \sqrt{1 + y^2}\right), \quad \cosh x = \sqrt{1 + \sinh^2 x},$$

we derive an error estimate with the above choice of δ:

$$|\tau_{\mathrm{GJ}}(Q)| \le \frac{4L_0^{1-\alpha}}{\Gamma(2-\alpha)}\left(1 + \frac{TL_0}{4Q}\right) e^{-2Q\delta + L_0 T(\cosh \delta - 1)/2}$$

$$\le \frac{4L_0^{1-\alpha}}{\Gamma(2-\alpha)}\left(1 + \frac{TL_0}{4Q}\right)\left(\frac{TL_0}{8Q}\right)^{2Q} e^{L_0 T\left(-1 + \sqrt{1 + (4Q/(TL_0))^2}\right)/2}$$

$$\le \frac{4L_0^{1-\alpha}}{\Gamma(2-\alpha)}\left(1 + \frac{TL_0}{4Q}\right)\left(\frac{TL_0}{8Q}\right)^{2Q} e^{2Q},$$

where in the last step above we have used that $-1 + \sqrt{1 + x^2} \le x$ for $x > 0$. This gives the stated result. □

2.3 Gauss quadrature on increasing intervals

We next split the remaining integral as

$$\frac{1}{\Gamma(1-\alpha)}\int_{L_0}^L x^{-\alpha} e^{-xt}\, dx = \sum_{j=1}^J I_j,$$

where

$$I_j = \frac{1}{\Gamma(1-\alpha)}\int_{L_{j-1}}^{L_j} x^{-\alpha} e^{-xt}\, dx$$

$$= \frac{\Delta L_j}{2\Gamma(1-\alpha)} e^{-L_{j-1}t}\int_{-1}^1\left(L_{j-1} + \frac{\Delta L_j}{2}(y+1)\right)^{-\alpha} e^{-t(y+1)\Delta L_j/2}\, dy,$$

where $\Delta L_j = L_j - L_{j-1}$, $j = 1, \ldots, J$, with $L_J = L$. The intervals are chosen so that for some $B \ge 1$, $\Delta L_j = BL_{j-1}$, i.e., $L_j = (B+1)L_{j-1}$ and $J = \lceil \log_{B+1} L/L_0 \rceil$. To each integral we apply standard, i.e., $w(x) \equiv 1$ in Theorem 3, Gauss quadrature with Q nodes and denote the corresponding error by

$$\tau_j(Q) := If_j - I_Q f_j$$

with

$$f_j(x) = \frac{4\Delta L_j}{\Gamma(1-\alpha)} e^{-L_{j-1}t} \left(L_{j-1} + \frac{\Delta L_j}{2}(x+1) \right)^{-\alpha} e^{-t(x+1)\Delta L_j/2}. \qquad (8)$$

Theorem 5 *For any $Q \geq 1$ and $t \geq 0$*

$$|\tau_j(Q)| \leq \frac{4BL_{j-1}^{1-\alpha}}{\Gamma(1-\alpha)} \min_{0<\varepsilon<1} \frac{g(\varepsilon, B)^{-2Q+1}}{g(\varepsilon, B)-1} \varepsilon^{-\alpha} e^{-tL_{j-1}\varepsilon},$$

with

$$g(\varepsilon, B) = 1 + \frac{2}{B}(1-\varepsilon) + \sqrt{\left(1 + \frac{2}{B}(1-\varepsilon) \right)^2 - 1}.$$

Proof Note that the integrand f_j in (8) is now not entire and there will be a restriction $\varrho < \varrho_{max}$ on the choice of the Bernstein ellipse $\mathcal{E}_\varrho$ in order to avoid the singularity of the fractional power. In particular we require

$$L_{j-1} - \frac{\Delta L_j}{4}(\varrho + \varrho^{-1} - 2) = L_{j-1} \left(1 - \frac{B}{4}(\varrho + \varrho^{-1} - 2) \right) > 0,$$

which is satisfied for $1 < \varrho < \varrho_{max}$ and

$$\varrho_{max} = 1 + \frac{2}{B}(1 + \sqrt{1+B}).$$

Setting

$$\varepsilon(\varrho) = 1 - \frac{B}{4}(\varrho + \varrho^{-1} - 2)$$

we see that $\varepsilon \in (0, 1)$ for $\varrho \in (1, \varrho_{max})$ and that

$$L_{j-1} - \frac{\Delta L_j}{4}(\varrho + \varrho^{-1} - 2) = L_{j-1}\varepsilon.$$

Hence

$$|\tau_j(Q)| \leq \frac{4\Delta L_j}{\Gamma(1-\alpha)} L_{j-1}^{-\alpha} \min_{1<\varrho<\varrho_{max}} \frac{\varrho^{-2Q+1}}{\varrho - 1} \varepsilon^{-\alpha} e^{-tL_{j-1}\varepsilon}.$$

The result is now obtained by using

$$\cosh^{-1} y = \log \left(y + \sqrt{y^2 - 1} \right), \qquad y \geq 1,$$

to show that $\varrho = g(\varepsilon, B)$. $\qquad \square$

Remark 6 Choosing for instance $\varepsilon = 0.1$ and $B = 1$ in the above estimate, we obtain

$$\varrho_{\max} = g(0, B) = 3 + 2\sqrt{2} = 5.83$$

and

$$|\tau_j(Q)| \leq \frac{10^\alpha L_{j-1}^{1-\alpha} \exp(-0.1 t L_{j-1})}{1.1\Gamma(1-\alpha)} (5.41)^{-2Q+1}.$$

As we will require a uniform bound for $t \in [t_{n_0}, T]$, we can substitute $t = t_{n_0}$ in this estimate.

3 Runge–Kutta convolution quadrature

Let us consider an s-stage Runge–Kutta method described by the coefficient matrix $Q = (a_{ij})_{i,j=1}^s \in \mathbb{R}^{s \times s}$, the vectors of weights $\mathbf{b} = (b_1, \ldots, b_s)^T \in \mathbb{R}^s$ and the vector of abcissae $\mathbf{c} = (c_1, \ldots, c_s)^T \in [0, 1]^s$. We assume that the method is A-stable, has classical order $p \geq 1$, stage order q and satisfies $a_{s,j} = b_j$, $j = 1, \ldots, s$, [10]. The corresponding stability function is given by

$$r(z) = 1 + z\mathbf{b}^T (\mathbf{I} - z Q)^{-1} \mathbb{1}, \tag{9}$$

where

$$\mathbb{1} = (1, 1, \ldots, 1)^T.$$

Our assumptions imply the following properties:

1. $c_s = 1$.
2. $r(\infty) = \mathbf{b}^T Q^{-1} \mathbb{1} - 1 = 0$.
3. $r(z) = e^z + O(z^{p+1})$
4. $|r(z)| \leq 1$ for $\operatorname{Re} z \leq 0$.

Important examples of RK methods satisfying our assumptions are Radau IIA and Lobatto IIIC methods.

Let us consider the convolution

$$K(\partial_t) f := \int_0^t k(t - \tau) f(\tau) d\tau, \tag{10}$$

where $K(z)$ denotes the Laplace transform of the convolution kernel $k(t)$. K is assumed to be analytic for $\operatorname{Re} z > 0$ and bounded there as $|K(z)| \leq |z|^{-\mu}$ for some $\mu > 0$. The operational notation $K(\partial_t) f$ introduced in [15], is useful in emphasising certain properties of convolutions. Of particular importance is the composition rule, namely, if $K(s) = K_1(s) K_s(s)$ then $K(\partial_t) f = K_1(\partial_t) K_2(\partial_t) f$. This will be used when solving fractional differential equations in Sect. 7.2.

If $\mu < 0$, the convolution is defined by

$$K(\partial_t) f = \left(\frac{d}{dt}\right)^m K_m(\partial_t) f,$$

where $K_m(z) = z^{-m} K(z)$ and m smallest integer such that $m > -\mu$.

For $K(z) = z^{-\alpha}$ the convolution coincides with the fractional integral of order α, i.e., according to the operational notation, we can write

$$\mathcal{I}^\alpha[f](t) = \partial_t^{-\alpha} f(t), \qquad t > 0, \ \alpha \in (0, 1).$$

For $\beta > 0$, ∂_t^β is equivalent to the Riemann–Liouville fractional derivative, see definition (28).

Runge–Kutta convolution quadrature has been derived in [18] and applied to (10) provides approximations at time-vectors $\mathbf{t}_n = (t_{n,j})_{j=1}^s$, with $t_{n,j} = t_n + c_j h$ and $t_n = nh$, defined by

$$K(\partial_t) f(\mathbf{t}_n) \approx K(\partial_t^h) f(\mathbf{t}_n) := \sum_{j=0}^n \mathbf{W}_{n-j}(K)\mathbf{f}_j, \tag{11}$$

where $(K(\partial_t) f(\mathbf{t}_n))_\ell = K(\partial_t) f(t_{n,\ell})$, $(\mathbf{f}_j)_\ell = f(t_{j,\ell})$ and the weight matrices $\mathbf{W}_j$ are the coefficients of the power series

$$K\left(\frac{\mathbf{\Delta}(\zeta)}{h}\right) = \sum_{j=0}^\infty \mathbf{W}_j(K)\zeta^j \tag{12}$$

with

$$\mathbf{\Delta}(\zeta) = \left(\mathcal{Q} + \frac{\zeta}{1-\zeta}\mathbb{1}\mathbf{b}^T\right)^{-1} = \mathcal{Q}^{-1} - \zeta \mathcal{Q}^{-1}\mathbb{1}\mathbf{b}^T\mathcal{Q}^{-1}. \tag{13}$$

The notation in (11) again emphasises that the composition rule holds also after discretization: if $K(s) = K_1(s) K_s(s)$ then $K(\partial_t^h) f = K_1(\partial_t^h) K_2(\partial_t^h) f$.

The last row in (11) defines the approximation at the time grid t_{n+1}, since $c_s = 1$. Denoting $\boldsymbol{\omega}_j(K)$ the last row of $\mathbf{W}_j(K)$, the approximation reads

$$K(\partial_t) f(t_{n+1}) \approx K(\partial_t^h) f(t_{n+1}) := \sum_{j=0}^n \boldsymbol{\omega}_{n-j}(K)\mathbf{f}_j, \quad (\mathbf{f}_j)_\ell = f(t_{j,\ell}). \tag{14}$$

For the rest of the paper we will denote by $\mathbf{W}_j = \mathbf{W}_j(K)$ and $\boldsymbol{\omega}_j = \boldsymbol{\omega}_j(K)$ the weights for the fractional integral case, i.e., for $K(z) = z^{-\alpha}$.

Remark 7 (Notation). We have defined the discrete convolution $K(\partial_t^h) f$ for functions f. For a sequence $\mathbf{f}_0, \ldots, \mathbf{f}_N \in \mathbb{R}^s$, we use the same notation $K(\partial_t^h)\mathbf{f}$ to denote

$$K(\partial_t^h)\mathbf{f}(t_{n+1}) = \sum_{j=0}^{n} \boldsymbol{\omega}_{n-j}(K)\mathbf{f}_j, \qquad n = 0, \ldots, N,$$

and similarly for $K(\partial_t^h)\mathbf{f}(\mathbf{t}_n)$ with the meaning

$$K(\partial_t^h)\mathbf{f}(\mathbf{t}_n) = \sum_{j=0}^{n} \mathbf{W}_{n-j}(K)\mathbf{f}_j$$

and

$$K(\partial_t^h)\mathbf{f}(t_{n,\ell}) = \left(\sum_{j=0}^{n} \mathbf{W}_{n-j}(K)\mathbf{f}_j \right)_\ell.$$

Note also that

$$K(\partial_t^h)\mathbf{f}(t_{n,s}) = K(\partial_t^h)\mathbf{f}(t_{n+1}).$$

FFT techniques based on (12) can be applied to compute at once all the required $\mathbf{W}_j, \, j = 0, \ldots, N$, with $N = \lceil T/h \rceil$, [16]. The computational cost associated to this method is $O(N \log(N))$. It implies precomputing and keeping in memory all weight matrices for the approximation of every $\mathcal{I}^\alpha[f](t_n)$, $n = 1, \ldots, N$, see [4] for details and many experiments.

The following error estimate for the approximation of (1) by (14) is given by [18, Theorem 2.2]. Notice that we allow $K(z)$ to be a map between two Banach spaces with appropriate norms denoted by $\|\cdot\|$ in the following. This will be needed in Sect. 7.

Theorem 8 *Assume that there exist $c \in \mathbb{R}$, $0 < \delta < \frac{\pi}{2}$ and $M > 0$ such that $K(z)$ is analytic in a sector $|\arg(z - c)| < \pi - \delta$ and satisfies there the bound $\|K(z)\| \leq M|z|^{-\alpha}$. Then if $f \in C^p[0, T]$, there exists $h_0 > 0$ and $C > 0$ such that for $h \leq h_0$ it holds*

$$\left\| K(\partial_t) f(t_n) - K(\partial_t^h) f(t_n) \right\| \leq Ch^p \sum_{\ell=0}^{q} \left(1 + t_n^{\alpha+\ell-p} \right) \|f^{(\ell)}(0)\|$$

$$+ C \left(h^p + h^{q+1+\alpha} |\log(h)| \right) \left(\sum_{\ell=q+1}^{p-1} \|f^{(\ell)}(0)\| + \max_{0 \leq \tau \leq t_n} \|f^{(p)}(\tau)\| \right).$$

3.1 Real integral representation of the CQ weights

The convolution quadrature weights ω_j can also be expressed as [23]

$$\omega_n = \frac{h}{2\pi i} \int_\Gamma z^{-\alpha} \mathbf{e}_n(hz) \, dz, \tag{15}$$

for $\mathbf{e}_n(\lambda)$ a function which depends on the ODE method underlying the CQ formula and an integration contour Γ which can be chosen as a Hankel contour beginning and ending in the left half of the complex plane.

Lemma 9 *The weights are given by*

$$\mathbf{W}_n = \frac{h \sin(\pi\alpha)}{\pi} \int_0^\infty x^{-\alpha} \mathbf{E}_n(-hx) \, dx, \tag{16}$$

and

$$\omega_n = \frac{h \sin(\pi\alpha)}{\pi} \int_0^\infty x^{-\alpha} \mathbf{e}_n(-hx) \, dx, \tag{17}$$

where

$$(\mathbf{\Delta}(\zeta) - zI)^{-1} = \sum_{n=0}^\infty \mathbf{E}_n(z)\zeta^n \tag{18}$$

and $\mathbf{e}_n(z)$ is the last row of $\mathbf{E}_n(z)$.
   Explicit formulas for $\mathbf{E}_n$ and $\mathbf{e}_n$ are given by

$$\mathbf{E}_0 = \mathcal{Q}(I - z\mathcal{Q})^{-1}, \qquad \mathbf{E}_n(z) = r(z)^{n-1}(I - z\mathcal{Q})^{-1}\mathbb{1}\mathbf{q}(z) \tag{19}$$

and

$$\mathbf{e}_n(z) = r(z)^n \mathbf{q}(z), \tag{20}$$

where r is the stability function of the method and $\mathbf{q}(z) = \mathbf{b}^T(I - z\mathcal{Q})^{-1}$.

Proof Since $z^{-\alpha}$ is analytic in the whole complex plane but for the branch cut on the negative real axis, the Hankel contour Γ can be degenerated into negative real axis as in the derivation of the real inversion formula for the Laplace transform [11, Sect. 10.7] to obtain

$$\begin{aligned}
\omega_n &= \frac{h}{2\pi i} \int_0^\infty (e^{i\pi\alpha} - e^{-i\pi\alpha})x^{-\alpha}\mathbf{e}_n(-hx) \, dx \\
&= \frac{h \sin(\pi\alpha)}{\pi} \int_0^\infty x^{-\alpha}\mathbf{e}_n(-hx) \, dx.
\end{aligned}$$

The expression for $\mathbf{W}_n$ is obtained in the same way and the explicit formulas for $\mathbf{E}_n$ and $\mathbf{e}_n$ can be found in [23]. □

The next properties will be used later in Sect. 4

Lemma 10 *There exist constants $\gamma > 1$, $b > 0$ and $C_{\mathbf{q}} > 0$ such that*

$$|r(z)| \le e^{\gamma \operatorname{Re} z}, \qquad \text{for } 0 \le \operatorname{Re} z \le b,$$

and

$$\|\mathbf{q}(z)\| \le C_{\mathbf{q}}, \qquad \text{for } \operatorname{Re} z \le b,$$

where $C_{\mathbf{q}}$ depends on the choice of the norm $\| \cdot \|$.

Proof Fix a $b > 0$ such that all the poles of $r(z)$ (and hence $\mathbf{q}(z)$) belong to $\operatorname{Re} z > b$. Define now

$$\gamma = \sup_{0 \le \operatorname{Re} z \le b} \frac{1}{\operatorname{Re} z} \log |r(z)| = \max \left\{ 1, \frac{1}{b} \sup_{\operatorname{Re} z = b} \log |r(z)| \right\}, \tag{21}$$

where we have used the properties of $r(z)$ to see that $\sup_{\operatorname{Re} z = 0} \frac{1}{\operatorname{Re} z} \log |r(z)| = 1$.

Recall that $\mathbf{q}(z) = \mathbf{b}^T (I - z\mathcal{Q})^{-1}$. As all the singularities of $\mathbf{q}$ are in the half-plane $\operatorname{Re} z > b$ and $\|\mathbf{q}(z)\| \to 0$ as $|z| \to \infty$, we have that $\|\mathbf{q}(z)\|$ is bounded in the region $\operatorname{Re} z \le b$. $\qquad \square$

Remark 11 (a) Note that for BDF1 we can choose $b \in (0, 1)$. Hence, $\gamma = b^{-1} \log \frac{1}{1-b}$ and since $\mathbf{q}(z) = r(z)$ for BDF1, we can set $C_{\mathbf{q}} = e^{\gamma b}$.
(b) For the 2-stage Radau IIA method we have

$$r(z) = \frac{2z + 6}{z^2 - 4z + 6}, \quad \mathbf{q}(z) = \frac{1}{2(z^2 - 4z + 6)} \begin{bmatrix} 9 & 3 - 2z \end{bmatrix}.$$

As the poles of r and $\mathbf{q}$ are at $z = 2 \pm \sqrt{2}\mathrm{i}$, we can choose any $b \in (0, 2)$ and obtain the optimal γ numerically using (21). For example for $b = 1$, we can choose $\gamma \approx 1.0735$. Similarly we can compute $C_{\mathbf{q}}$ by computing

$$C_{\mathbf{q}} = \sup_{\operatorname{Re} z = 0 \text{ or } \operatorname{Re} z = b} \|\mathbf{q}(z)\|.$$

For $b = 1$ and the Euclidian norm we have $C_{\mathbf{q}} \approx 1.6429$. Using the same procedure, for $b = 3/2$, we have $\gamma \approx 1.2617$ and $C_{\mathbf{q}} \approx 3.3183$.
(c) For the 3-stage Radau IIA method the poles of $r(z)$ and $\mathbf{q}(z)$ belong to $\operatorname{Re} z \ge \frac{9^{2/3}}{6} - \frac{9^{1/3}}{2} + 3 \approx 2.681$. Choosing $b = 1$ gives $\gamma \approx 1.0117$ and $C_{\mathbf{q}} \approx 1.1803$, whereas for $b = 1.5$ we obtain $\gamma \approx 1.0521$ and $C_{\mathbf{q}} \approx 1.7954$.

Lemma 12 *There exist constants $c > 0$ and $x_0 > 0$ such that*

$$\max\{\|\mathbf{e}_n(z)\|, \|\mathbf{E}_n(z)\|\} \le |x_0 - c \operatorname{Re} z|^{-n-1}, \qquad \text{for } \operatorname{Re} z < 0.$$

Proof Using that $r(\infty) = \mathbf{b}^T \mathcal{Q}^{-1} \mathbb{1} - 1 = 0$ it can be shown that $r(z) = \mathbf{b}^T \mathcal{Q}^{-1}(I - z\mathcal{Q})^{-1}\mathbb{1}$. Let all eigenvalues of $\mathcal{Q}^{-1}$ and hence all poles of $r(z)$, $\mathbf{q}(z)$, and $(I - z\mathcal{Q})^{-1}$ lie in Re $z \geq \tilde{x}_0 > 0$. There exists a constant C such that for all Re $z < 0$

$$\max\{|r(z)|, \|\mathbf{q}(z)\|, \|(I - z\mathcal{Q})^{-1}\|\} \leq C|\operatorname{Re} z - \tilde{x}_0|^{-1} \leq |x_0 - c\operatorname{Re} z|^{-1},$$

where we can set $x_0 = \frac{1}{C}\tilde{x}_0$ and $c = \frac{1}{C}$. □

Remark 13 1. For BDF1, $c = 1$ and $x_0 = 1$.

2. For 2-stage Radau IIA the constant can be obtained following the proof. Namely we choose $\tilde{x}_0 = 2$ and find numerically that

$$\max\{|r(z)|, \|\mathbf{q}(z)\|, \|(I - z\mathcal{Q})^{-1}\|\}|\operatorname{Re} z - \tilde{x}_0| \leq 2, \qquad \operatorname{Re} z \leq 0.$$

Hence we can choose $C = 2$ and $c = 1/2$ and $x_0 = 2/C = 1$.

3. Similarly, for 3-stage Radau IIA we choose $\tilde{x}_0 = 2.6811$ and find that

$$\max\{|r(z)|, \|\mathbf{q}(z)\|, \|(I - z\mathcal{Q})^{-1}\|\}|\operatorname{Re} z - \tilde{x}_0| \leq 3.0821, \qquad \operatorname{Re} z \leq 0.$$

Hence we can choose $C = 3.0821$ and $c = 1/C = 0.3245$ and $x_0 = \tilde{x}_0/C = 0.8699$.

In the rest of the section our goal is to derive a good quadrature for the approximation of $\boldsymbol{\omega}_n$ and $\mathbf{W}_n$. We will perform the same steps as in Sect. 2 for the $\boldsymbol{\omega}_n$. The same quadrature rules will give essentially the same error estimates for the $\mathbf{W}_n$; see Remark 21.

4 Efficient quadrature for the CQ weights

Analogously to the the continuous case (1), we fix n_0, h, and T and develop an efficient quadrature for the CQ weights representation (17), for $nh \in [(n_0 + 1)h, T]$ and $\alpha \in (0, 1)$.

4.1 Truncation of the CQ weights integral representation

Again we truncate the integral

$$\boldsymbol{\omega}_n = \frac{h \sin(\pi\alpha)}{\pi} \int_0^L x^{-\alpha} \mathbf{e}_n(-hx)\, dx + \boldsymbol{\tau}(L)$$

and give a bound on the truncation error $\boldsymbol{\tau}(L)$.

Lemma 14 *With the choice* $L = Ah^{-1}$, *the truncation error is bounded as*

$$\|\boldsymbol{\tau}(L)\| \leq \frac{h^\alpha \sin(\pi\alpha)}{\pi} \int_A^\infty \|\mathbf{e}_n(-x)\| x^{-\alpha} dx \tag{22}$$

or more explicitly

$$\|\boldsymbol{\tau}(L)\| \leq \frac{h^\alpha \sin(\pi\alpha)}{cn\pi} A^{-\alpha}(x_0 + cA)^{-n}.$$

Proof From Lemma 12 we have that

$$
\begin{aligned}
\|\boldsymbol{\tau}(L)\| &\leq \frac{h^\alpha \sin(\pi\alpha)}{\pi} \int_A^\infty \|\mathbf{e}_n(-x)\| x^{-\alpha} dx \\
&\leq \frac{L^{-\alpha} \sin(\pi\alpha)}{\pi} \int_A^\infty (x_0 + cx)^{-n-1} dx \\
&= \frac{h^\alpha \sin(\pi\alpha)}{cn\pi} A^{-\alpha}(x_0 + cA)^{-n}.
\end{aligned}
$$

$\square$

Corollary 15 *Let $L = A/h$. Given* tol > 0, *choosing*

$$A > \left(\frac{h^\alpha \sin(\pi\alpha)}{\text{tol}\, n\pi c^{n+1}}\right)^{\frac{1}{n+\alpha}}$$

ensures $\|\boldsymbol{\tau}(L)\| \leq$ tol. The estimate becomes uniform in $n > n_0$ by setting $n = n_0 + 1$ in the above error bound.

Proof We have from above

$$\|\boldsymbol{\tau}(L)\| \leq \frac{h^\alpha \sin(\pi\alpha)}{cn\pi} A^{-\alpha}|cA + z_0|^{-n} \leq \frac{h^\alpha \sin(\pi\alpha)}{n\pi} A^{-\alpha-n} c^{-n-1},$$

from which the result follows.

$\square$

Remark 16 In practice we find that instead of using Corollary 15, better results are obtained if a simple numerical search is done to find the optimal A such that the right-hand side in (22) with $n = n_0 + 1$ is less than tol. To do this, we start from $A = 0$ and iteratively approximate the integral in (22) for increased values of A ($A \leftarrow A + 0.125$ in our code) until the resulting quantity is below our error tolerance. The approximation of the integrals is done by the MATLAB built-in routine `integral`. Notice that this has to be done only once for each RK-CQ formula and value of $\alpha \in (0, 1)$.

4.2 Gauss–Jacobi quadrature for the CQ weights

In a similar way as in Sect. 2.2, we consider the approximation of the integral

$$\boldsymbol{\omega}_n = \frac{h \sin(\pi\alpha)}{\pi} \int_0^\infty x^{-\alpha} \mathbf{e}_n(-xh)\, dx$$

and investigate in the first place the approximation of

$$\mathbf{I}_{0,n} := \frac{h\sin(\pi\alpha)}{\pi} \int_0^{L_0} x^{-\alpha} \mathbf{e}_n(-xh)\, dx,$$

for some suitable $L_0 > 0$ by using Gauss–Jacobi quadrature. Changing variables as in Sect. 2.2 we obtain

$$\mathbf{I}_{0,n} = \frac{h\sin(\pi\alpha)}{\pi} \left(\frac{L_0}{2}\right)^{1-\alpha} \int_{-1}^1 (y+1)^{-\alpha} \mathbf{e}_n(-h(y+1)L_0/2)\,dy$$

and apply Theorem 3 to estimate the error

$$\boldsymbol{\tau}_{\mathrm{GJ},n}(Q) = \mathbf{I}_Q \mathbf{f}_0 - \mathbf{I}\mathbf{f}_0$$

with the weight $w(x) = (x+1)^{-\alpha}$ and integrand

$$\mathbf{f}_0(x) = \frac{h\sin(\pi\alpha)}{\pi} \left(\frac{L_0}{2}\right)^{1-\alpha} \mathbf{e}_n(-h(x+1)L_0/2).$$

Theorem 17 *Let*

$$\varrho_{\max} = 1 + \frac{2b}{L_0 h} + \sqrt{\left(\frac{2b}{L_0 h}\right)^2 + \frac{4b}{L_0 h}},$$

with b and γ from Lemma 10, and

$$\varrho_{opt} = \frac{4Q}{\gamma T L_0} + \sqrt{1 + \left(\frac{4Q}{\gamma T L_0}\right)^2}.$$

If $\varrho_{opt} \in (1, \varrho_{\max})$, we have the bound

$$\|\boldsymbol{\tau}_{\mathrm{GJ},n}(Q)\| \le C_{\mathbf{q}} \frac{h L_0^{1-\alpha}\sin(\pi\alpha)}{\pi(1-\alpha)} \left(1 + \frac{\gamma T L_0}{4Q}\right) \left(\frac{e\gamma T L_0}{8Q}\right)^{2Q}.$$

Otherwise we have the bound

$$\|\boldsymbol{\tau}_{\mathrm{GJ},n}(Q)\| \le C_{\mathbf{q}} \frac{h L_0^{1-\alpha}\sin(\pi\alpha)}{\pi(1-\alpha)} \left(\frac{\varrho_{\max}^{-2Q+1}}{\varrho_{\max}-1} e^{\gamma Tb/h}\right).$$

Proof We again consider the Bernstein ellipse $\mathcal{E}_\varrho$ around $[-1, 1]$, but now in order to be able to use Lemma 10 and avoid the singularities of $\mathbf{e}_n(z)$ in the right-half plane we have a restriction on ϱ. Namely, the maximal value of ϱ is given by

$$h\left(\varrho_{\max} + \varrho_{\max}^{-1} - 2\right) L_0/4 = b,$$

which implies, writing $\varrho_{\max} = e^{\delta_{\max}}$,

$$\cosh(\delta_{\max}) - 1 = \frac{2b}{L_0 h},$$

and thus

$$\delta_{\max} = \cosh^{-1}\left(1 + \frac{2b}{L_0 h}\right)$$

giving the expression for $\varrho_{\max}$ from the statement of the theorem. The error estimate for Gauss–Jacobi quadrature then reads, by using Lemma 10,

$$
\begin{aligned}
\|\boldsymbol{\tau}_{\mathrm{GJ},n}\| &\leq \frac{hL_0^{1-\alpha}\sin(\pi\alpha)}{\pi(1-\alpha)} \min_{1<\varrho\leq\varrho_{\max}}\left(\frac{\varrho^{-2Q+1}}{\varrho-1}\max_{\zeta\in\mathcal{E}_\rho}\|\mathbf{e}_n(-h(\zeta+1)L_0/2)\|\right) \\
&\leq C_{\mathbf{q}}\frac{hL_0^{1-\alpha}\sin(\pi\alpha)}{\pi(1-\alpha)} \min_{1<\varrho\leq\varrho_{\max}}\left(\frac{\varrho^{-2Q+1}}{\varrho-1}e^{\gamma t_n L_0(\varrho+\varrho^{-1}-2)/4}\right) \\
&\leq C_{\mathbf{q}}\frac{hL_0^{1-\alpha}\sin(\pi\alpha)}{\pi(1-\alpha)} \min_{1<\varrho<\varrho_{\max}}\left(\frac{\varrho^{-2Q+1}}{\varrho-1}e^{\gamma T L_0(\varrho+\varrho^{-1}-2)/4}\right).
\end{aligned}
$$

Proceeding as in Sect. 2.2 with γT in place of T we obtain the bound

$$\|\boldsymbol{\tau}_{\mathrm{GJ},n}(Q)\| \leq C_{\mathbf{q}}\frac{hL_0^{1-\alpha}\sin(\pi\alpha)}{\pi(1-\alpha)}\left(1 + \frac{\gamma T L_0}{4Q}\right)\left(\frac{e\gamma T L_0}{8Q}\right)^{2Q},$$

provided that the optimal value for ϱ is within the accepted interval

$$\varrho_{opt} = \frac{4Q}{\gamma T L_0} + \sqrt{1 + \left(\frac{4Q}{\gamma T L_0}\right)^2} \in (1, \varrho_{\max}),$$

otherwise we make the choice $\varrho = \varrho_{\max}$. $\qquad\square$

Remark 18 In all our numerical experiments, we have found that $\varrho_{opt} < \varrho_{\max}$.

4.3 Gauss quadrature on increasing intervals for the CQ weights

We next split the remaining integral into the sum

$$\frac{h\sin(\pi\alpha)}{\pi}\int_{L_0}^{L} x^{-\alpha}\mathbf{e}_n(-xh)\,dx = \sum_{j=1}^{J} I_{n,j},$$

where

$$I_{n,j} = \frac{h\sin(\pi\alpha)}{\pi}\int_{L_{j-1}}^{L_j} x^{-\alpha}\mathbf{e}_n(-xh)\,dx.$$

The intervals are again chosen so that for some $B \geq 1$, $L_j = (B + 1)L_{j-1}$. To each integral we apply standard Gauss quadrature, i.e., $w(x) \equiv 1$ in Theorem 3, with Q nodes and denote the corresponding error by $\boldsymbol{\tau}_{n,j}(Q)$.

Theorem 19

$$\|\boldsymbol{\tau}_{n,j}(Q)\| \leq \frac{4hBL_{j-1}^{1-\alpha}\sin(\pi\alpha)}{\pi} \min_{0<\varepsilon<1} \frac{g(\varepsilon, B)^{-2Q+1}}{g(\varepsilon, B) - 1} \varepsilon^{-\alpha} \min(C_{\mathbf{q}}, |x_0 + cL_{j-1}h\varepsilon|^{-n-1}),$$

with constants $C_{\mathbf{q}}$, c, x_0 from Lemmas 10 and 12 and

$$g(\varepsilon, B) = 1 + \frac{2}{B}(1 - \varepsilon) + \sqrt{\left(1 + \frac{2}{B}(1 - \varepsilon)\right)^2 - 1}. \tag{23}$$

Proof The proof is the same as the proof of Theorem 5, we only need to combine the facts that $|r(z)| \leq 1$ for $\operatorname{Re} z \leq 0$, the bound $\|q(z)\| \leq C_{\mathbf{q}}$ from Lemma 10, and the bound from Lemma 12. □

Remark 20 To obtain a uniform bound for $t_n \in [t_{n_0+1}, T]$, we replace n by $n_0 + 1$ in the above bound.

Remark 21 We have developed the quadrature for the weights $\boldsymbol{\omega}_n$. However, up to a small difference in constants, the same error estimates hold for the matrix weights $\mathbf{W}_n$. Certainly, due to Lemma 12, the truncation estimate is the same. The main estimate used in the proof of Theorem 17 is the bound on the stability function $r(z)$ and on $q(z)$. The additional terms in $\mathbf{E}_n(z)$ would only contribute to the constant. Similar comment holds for Theorem 17.

5 Fast summation and computational cost

Now the efficient quadrature is available we explain how to use it to develop a fast algorithm for computing the corresponding discrete convolution. In order to do this, we split the convolution as

$$\sum_{j=0}^{n} \boldsymbol{\omega}_j \mathbf{f}_{n-j} = \sum_{j=0}^{n_0} \boldsymbol{\omega}_j \mathbf{f}_{n-j} + \sum_{j=n_0+1}^{n} \boldsymbol{\omega}_j \mathbf{f}_{n-j} = I_n^1 + I_n^2,$$

where as before $(\mathbf{f}_n)_\ell = f(t_{n,\ell})$; see (11). The first term is computed exactly, whereas for the second we can use the quadrature. Let N_Q be the total number of quadrature nodes and let (w_k, x_k) denote the quadrature weights and nodes with the weights including the values of $x_k^{-\alpha}$ in the region L_0 to L where Gauss quadrature is used. Then our approximation of I_n^2 has the form

$$\sum_{j=n_0+1}^{n} \boldsymbol{\omega}_j \mathbf{f}_{n-j} \approx \sum_{k=1}^{N_Q} w_k (r(-hx_k))^{n_0+1} \sum_{j=0}^{n-n_0-1} (r(-hx_k))^j \mathbf{q}(-hx_k)\mathbf{f}_{n-n_0-1-j}.$$

Defining

$$Q_{n,k} = \sum_{j=0}^{n-n_0-1} (r(-hx_k))^j \mathbf{q}(-hx_k)\mathbf{f}_{n-n_0-1-j} \qquad (24)$$

we see that

$$Q_{n,k} = r(-hx_k)Q_{n-1,k} + \mathbf{q}(-hx_k)\mathbf{f}_{n-n_0-1}, \qquad Q_{n_0,k} = 0.$$

Hence the convolution can be approximated as

$$\sum_{j=0}^{n} \boldsymbol{\omega}_j \mathbf{f}_{n-j} \approx \sum_{j=0}^{n_0} \boldsymbol{\omega}_j \mathbf{f}_{n-j} + \sum_{k=1}^{N_Q} w_k (r(-hx_k))^{n_0+1} Q_{n,k},$$

with the $Q_{n,k}$ satisfying the above recursion. Notice that for each $k = 1, \ldots, N_Q$, $Q_{n,k}$ is the RK approximation at time t_{n-n_0-1} of the ODE:

$$\dot{q} = -x_k q + f, \qquad q(0) = 0.$$

Thus, from one step to the next one we only need updating $Q_{n,k}$, for $k = 1, \ldots, N_Q$, N_Q being the total number of quadrature nodes. Set ε the target accuracy of the quadrature. Then, from the results in Sect. 4 it follows that the total computational cost is $O(NN_Q)$ with

$$N_Q = O(|\log(\varepsilon)| \log(L/L_0)). \qquad (25)$$

For $n \geq 5$, Corollary 15 implies $L \sim h^{-1}$ and from Theorem 17 a reasonable choice for L_0 is $L_0 = 4/(eT)$, which leads to

$$N_Q = O(|\log(\varepsilon)| \log(h^{-1}T)) = O(|\log(\varepsilon)| \log(NT)).$$

Therefore, the computational complexity is $O(|\log \varepsilon| N \log N)$, whereas the storage requirement scales as $O(N_Q) = O(|\log \varepsilon| \log N)$.

6 Numerical experiments

Given a tolerance tol > 0, time step $h > 0$, minimal index n_0, final time $T > 0$, and the fractional power $\alpha \in (0, 1)$ we use the above estimates to choose the parameters in the quadrature.

In particular we choose $L_0 = 4/T$ and $L = Ah^{-1}$ with A such that the upper bound for the trunction error in (22) is less than tol/3. We set $\tilde{B} = 3$ and

$$J = \left\lfloor \frac{\log(L/L_0)}{\log(1 + \tilde{B})} \right\rfloor \text{ and } B = (L/L_0)^{1/J} - 1.$$
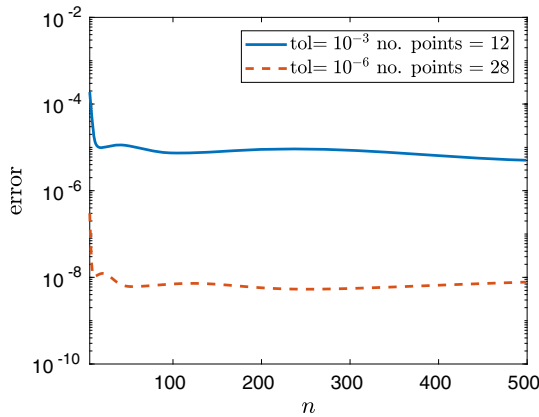
Fig. 1 We show the error in the computation of the 2-stage Radau IIA weights ω_n for $n > n_0$ with two different tolerances. The number of quadrature points is also shown. The results are for $\alpha = 0.5$, $T = 5$, $h = 10^{-2}$, $n_0 = 5$, and $B = 3$

Note that in general this choice results in fewer integration intervals than when fixing B and setting J to the smallest integer such that $L \leq L_0(1 + B)^J$. Next, we set $L_j = L_0(1 + B)^j$, for $j = 0, \ldots, J$ and let Q_0 denote the number of quadrature points in the Gauss–Jacobi quadrature on $[0, L_0]$ and Q_j, $j = 0, \ldots, J - 1$, the number of Gauss quadrature points in the interval $[L_j, L_{j+1}]$. We choose the smallest Q_0 so that the bound on $\|\tau_{\mathrm{GJ},n}(Q_0)\|$ in Theorem 17 is less than tol/3; note that in all of the experiments below we had $\varrho_{\mathrm{opt}} < \varrho_{\mathrm{max}}$. By doing a simple numerical minimization on the bound in Theorem 19, we find the optimal Q_j such that the error $\|\tau_{n,j}(Q_j)\| < $ tol $J^{-1}/3$. With this choice of parameters each weight ω_j, $j > n_0$, is computed to accuracy less than tol.

In Fig. 1 we show the error $\|\widetilde{\omega}_n - \omega_n\|$, where $\widetilde{\omega}_n$ is the nth weight computed using the new quadrature scheme and ω_n is an accurate approximation of the weight computed by standard means. We see that the error is bounded by the tolerance and that for the initial weights the error is close to this bound. The error for larger n is considerably smaller than the required tolerance. This is expected, as in Corollary 15 we need to use the worst case $n = n_0 + 1$ to determine the trunction parameter A.

We also investigate the number of quadrature points in dependence on h, T in Table 1 and on α and tol in Table 2. We observe only a moderate increase with decreasing h, tol and increasing T. The dependence on α is mild.

Table 1 Dependence of the total number of quadrature points on time step h and final time T. The other parameters are fixed at $n_0 = 5$, $B = 3$, tol $= 10^{-6}$, $\alpha = 0.5$. On the left the data is for backward Euler and on the right for the 2-stage Radau IIA CQ

$h\backslash T$	1	10	100	1000	$h\backslash T$	1	10	100	1000
10^{-1}	20	30	40	49	10^{-1}	13	25	34	44
10^{-2}	27	36	44	52	10^{-2}	21	31	39	46
10^{-3}	31	39	46	50	10^{-3}	28	35	41	46
10^{-4}	34	40	45	48	10^{-4}	31	37	43	45

Table 2 Dependence of the total number of quadrature points on the tolerance tol and the fractional power α. The other parameters are fixed at $h = 10^{-2}$, $T = 50$, $n_0 = 5$, $B = 3$. Again the data on the left is for backward Euler and on the right for 2-stage Radau IIA

tol\\α	0.1	0.3	0.5	0.7	0.9	tol\\α	0.1	0.3	0.5	0.7	0.9
10^{-2}	11	11	10	8	6	10^{-2}	9	9	8	8	6
10^{-4}	27	27	26	25	21	10^{-4}	23	25	24	23	20
10^{-6}	45	44	45	43	36	10^{-6}	39	39	39	37	35
10^{-8}	66	65	64	61	55	10^{-8}	71	68	65	53	51
10^{-10}	86	87	85	82	74	10^{-10}	96	93	90	86	77

6.1 Fractional integral

Let us now consider the evaluation of a fractional integral

$$u(t) = \mathcal{I}^\alpha[g](t), \tag{26}$$

where

$$g(t) = t^3 e^{-t}.$$

First we investigate the behaviour of the standard implementation of CQ, based on FFT. In the particular case of the two-stage Radau IIA, $p = 3$ and $q = 2$, from Theorem 8 we would expect full order convergence. We set $T = 128$, and $h = 2^{3-j}$, $j = 0, \ldots, 7$, $\alpha = 1/4$. We do not have access to the exact solution $u(t)$, so its role is taken by an accurate numerical approximation. In Fig. 2 we show the convergence of the error $\max_n |u(t_n) - u_n|$ using the standard implementation of CQ. We compare it with the theoretical reference curve $10^{-2.5}(h^3 + |\log(h)| h^{3+\alpha})$, which fits the results better in this pre-asymptotic regime than the dominant term h^3 on its own.

Next, we apply our new quadrature implementation of CQ. We set tol $= 10^{-6}$, and the rest of the parameters as in the above section. We denote by $\tilde{u}_n$ the new approximation of u_n and plot the error $|u_n - \tilde{u}_n|$ in Fig. 3. We see that the error is bounded by 10^{-6} for all n, showing that the final perturbation error introduced by our approximation of the CQ weights remains bounded with respect to the target accuracy in our quadrature, cf. [23]. We also compare computational times in Fig. 3. For the implementation of the standard CQ we have used the $O(N \log N)$ FFT based algorithm from [16]. We see that for larger time-steps the FFT method is faster due to a certain overhead in constructing the quadrature points for the new method. For smaller time steps however the new method is even marginally faster. The main advantage of the new method is the $O(\log N)$ amount of memory required compared to $O(N)$ amount of memory by the standard method. For example, in this computation with the smallest time step, there are $N = 2048$ time steps and the total number of quadrature points is 37. As each quadrature point carries approximately the same amount of memory as one directly computed time-step, we see that the memory requirement is around 50 times smaller with the new method for this example. Such a difference in memory
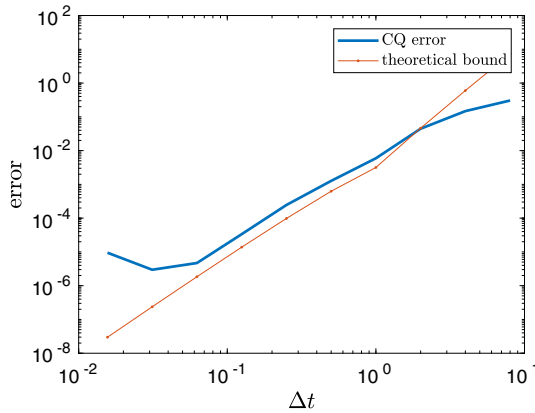
Fig. 2 Convergence of the error $\max_n |u(t_n) - u_n|$ for the 2-stage Radau IIA convolution quadrature of the fractional integral (26)
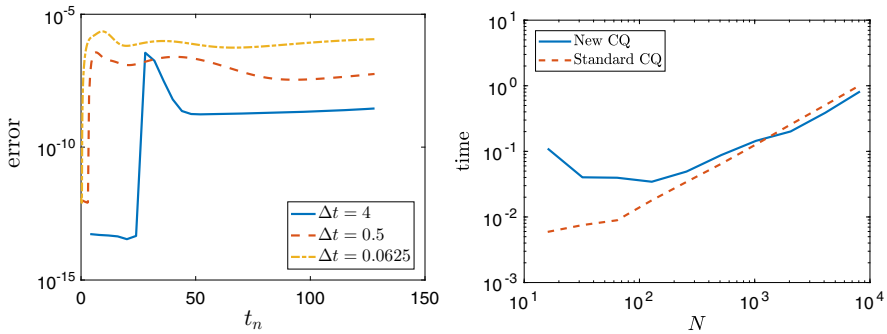

Fig. 3 We plot in the left graph the difference $|\tilde{u}_n - u_n|$ against t_n, where u_n is computed using the standard implementation of CQ and $\tilde{u}_n$ with the new method at tol $= 10^{-6}$. On the right we plot the required time for the two methods

requirements becomes of crucial importance when faced with non-scalar examples coming from discretizations of PDE. The next section considers this case.

7 Application to a fractional diffusion equation

We now consider the problem of finding $u(t) \in H_0^1(\Omega)$ such that

$$
\begin{aligned}
\partial_t^\beta u - \Delta u &= f, \quad \text{for } (x, t) \in \Omega \times [0, T], \\
u^{(k)}(x, 0) &= 0, \quad \text{for } x \in \Omega, \ k = 0, \ldots, m - 1,
\end{aligned}
\tag{27}
$$

with $\beta \in (0, 2) \setminus \{1\}$ and $m = \lceil \beta \rceil$. Here, Ω is a bounded, convex Lipschitz domain in $\mathbb{R}^d$, $d = 1, 2, 3$, $H_0^1(\Omega)$ the Sobolev space of functions with zero trace, and ∂_t^β the

fractional derivative

$$\partial_t^\beta u := \mathcal{I}^{m-\beta}[\partial_t^m u](t) = \frac{1}{\Gamma(1-m+\beta)} \int_0^t (t-s)^{m-\beta-1} \partial_t^m u(s)\, ds. \qquad (28)$$

This is the fractional derivative in the Caputo sense, which in the case $u^{(k)}(0) = 0$, $k = 0, \ldots, m-1$, is equivalent to the Riemann–Liouville derivative.

Remark 22 For simplicity we avoid the integer case $\beta = 1$ as it is just the standard heat equation and in some places this case would have to be treated slighlty differently.

The application of CQ based on BDF2 to integrate (28) in time has been analyzed in [8, Sect. 8]. A related problem with a fractional power of the Laplacian has been studied in [20], but not with a CQ time discretization. Here we apply Runge–Kutta based CQ. The analysis of the application of RK based CQ to (27) is not available in the literature, hence we give the analysis here for sufficiently smooth and compatible right-hand side f. We first analyze the error of the spatial discretization.

7.1 Space-time discretization of the FPDE: error estimates

Let $X_{\Delta x} \subset H_0^1(\Omega)$ be a finite element space of piecewise linear functions and let Δx be the meshwidth. Applying the Galerkin method in space we obtain a system of fractional differential equations: Find $\mathbf{u}(t) \in X_{\Delta x}$ such that

$$\int_\Omega \partial_t^\beta \mathbf{u}(t)\mathbf{v} + \nabla \mathbf{u}(t)\,\nabla \mathbf{v}\, dx = \int_\Omega f(t)\mathbf{v}dx, \quad \text{for } t \in [0, T], \mathbf{v} \in X_{\Delta x}$$
$$\mathbf{u}^{(k)}(0) = \mathbf{0}, \quad \text{for } x \in \Omega, \ k = 0, \ldots, m-1, \qquad (29)$$

Theorem 23 *Let $f \in C^m([0, T]; L^2(\Omega))$ with $f^{(k)}(0) = 0$, $k = 0, \ldots, m-1$ and let $\mathbf{u}(t)$ be the solution of (29) and $u(t)$ the solution of (27). Then if $m > \beta$ we have*

$$\|u(t) - \mathbf{u}(t)\|_{H^1(\Omega)} \le C\Delta x \int_0^t \|f^{(m)}(\tau)\|_{L^2(\Omega)} d\tau.$$

If further $m > 2\beta$ we have

$$\|u(t) - \mathbf{u}(t)\|_{L^2(\Omega)} \le C(\Delta x)^2 \int_0^t \|f^{(m)}(\tau)\|_{L^2(\Omega)} d\tau.$$

Proof Consider the Laplace transform of (27)

$$z^\beta \hat{u} - \Delta \hat{u} = \hat{f}, \qquad |\arg(z)| < \min(\pi, (\pi - \delta)/\beta), \qquad (30)$$

for some fixed $\delta > 0$, and the bilinear form

$$a(u, v) = \int_\Omega z^\beta u\bar{v}dx + \int_\Omega \nabla u \cdot \nabla \bar{v}dx.$$

Hence, $a(u, v) = \int_\Omega \hat{f} \overline{v} dx$ is the weak form of (30). The bilinear form is continuous

$$|a(u, v)| \leq \max(1, |z|^\beta) \|u\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)}$$

and

$$\text{Re } a(z^{-\beta} u, u) = \|u\|_{L^2(\Omega)}^2 + \text{Re } z^{-\beta} \|\nabla u\|_{L^2(\Omega)}^2 \geq \text{Re } z^{-\beta} \|\nabla u\|_{L^2(\Omega)}^2$$

and

$$|\text{Im } a(z^{-\beta} u, u)| = |\text{Im } z^{-\beta}| \|\nabla u\|_{L^2(\Omega)}^2.$$

Hence

$$|a(u, u)| = |z|^\beta |a(z^{-\beta} u, u)| \geq \begin{cases} \|\nabla u\|_{L^2(\Omega)}^2 & \text{if } \text{Re } z^{-\beta} > 0, \\ |z|^\beta |\text{Im } z^{-\beta}| \|\nabla u\|_{L^2(\Omega)}^2 & \text{otherwise.} \end{cases}$$

As $|\arg(z^\beta)| < \pi - \delta$, we have that $|a(u, u)| \geq C \|\nabla u\|_{L^2(\Omega)}^2$ and using the Poincaré inequality we obtain coercivity in $H_0^1(\Omega)$. Lax-Milgram gives us that there exists a unique $\hat{u} \in H_0^1(\Omega)$ solution of (30) and that

$$\|\hat{u}\|_{H^1(\Omega)} \leq C_\Omega \|\hat{f}\|_{H^{-1}(\Omega)}.$$

If furthermore $\hat{f} \in L^2(\Omega)$, we have that

$$|\text{Im } z^\beta| \|\hat{u}\|_{L^2(\Omega)}^2 = \left| \text{Im } \int_\Omega \hat{f} \overline{\hat{u}} dx \right| \leq \|\hat{f}\|_{L^2(\Omega)} \|\hat{u}\|_{L^2(\Omega)}$$

and

$$\text{Re } z^\beta \|\hat{u}\|_{L^2(\Omega)}^2 = -\|\nabla u\|_{L^2(\Omega)}^2 + \text{Re } \int_\Omega \hat{f} \overline{\hat{u}} dx \leq \|\hat{f}\|_{L^2(\Omega)} \|\hat{u}\|_{L^2(\Omega)}.$$

Dividing by $|z|^\beta \|\hat{u}\|_{L^2(\Omega)}$ and using the fact that $|\arg(z^\beta)| < \pi - \delta$ gives

$$\|\hat{u}\|_{L^2(\Omega)} \leq C |z|^{-\beta} \|\hat{f}\|_{L^2(\Omega)}. \tag{31}$$

For the finite element solution, Céa's lemma gives that

$$\|\hat{u} - \hat{\mathbf{u}}\|_{H^1(\Omega)} \leq \max(1, |z|^\beta) \inf_{\mathbf{v} \in X_{\Delta x}} \|\hat{u} - \mathbf{v}\|_{H^1(\Omega)},$$

where $\hat{\mathbf{u}}$ denotes the Laplace transform of $\mathbf{u}$. Using the Aubin-Nitche trick, we obtain the estimate in the weaker norm

$$\|\hat{u} - \hat{\mathbf{u}}\|_{L^2(\Omega)} \leq C \max(1, |z|^\beta) \|\hat{u} - \hat{\mathbf{u}}\|_{H^1(\Omega)} \sup_{g \in L^2(\Omega), g \neq 0} \inf_{\mathbf{v} \in X_{\Delta x}} \frac{\|\varphi_g - \mathbf{v}\|_{H^1(\Omega)}}{\|g\|_{L^2(\Omega)}},$$

where φ_g is the solution of the dual problem

$$a(\mathbf{v}, \varphi_g) = \int_\Omega g\overline{\mathbf{v}} \, dx, \qquad \text{for all } \mathbf{v} \in X_{\Delta x}.$$

Recalling that Ω is assumed to be convex, we can use standard elliptic regularity results together with $-\Delta \hat{u} = \hat{f} - z^\beta \hat{u}$ to show that

$$\|\hat{u}\|_{H^2(\Omega)} \leq C\|\hat{f} - z^\beta \hat{u}\|_{L^2(\Omega)} \leq C(\|\hat{f}\|_{L^2(\Omega)} + \|z^\beta \hat{u}\|_{L^2(\Omega)}) \leq C\|\hat{f}\|_{L^2(\Omega)},$$

where the final inequality follows from (31). Similarly

$$\|\varphi_g\|_{H^2(\Omega)} \leq C\|g\|_{L^2(\Omega)}$$

and using standard approximation results we have that

$$\|\hat{u} - \hat{\mathbf{u}}\|_{H^1(\Omega)} \leq C\Delta x \max(1, |z|^\beta)\|\hat{f}\|_{L^2(\Omega)} = C\Delta x \max(|z|^{-m}, |z|^{\beta-m})\|z^m \hat{f}\|_{L^2(\Omega)}$$

and

$$\|\hat{u} - \hat{\mathbf{u}}\|_{L^2(\Omega)} \leq C \max(|z|^{-m}, |z|^{2\beta-m})(\Delta x)^2 \|z^m \hat{f}\|_{L^2(\Omega)}.$$

The proof is completed by applying Parseval's theorem. $\qquad\square$

The fully discrete system is now obtained by simply discretizing the fractional derivative at stage level using RK-CQ:

$$\int_\Omega (\partial_t^h)^\beta \mathbf{U}(\mathbf{t}_n)\mathbf{v} + \nabla \mathbf{U}_n \, \nabla \mathbf{v} \, dx = \int_\Omega f(\mathbf{t}_n)\mathbf{v} dx, \tag{32}$$

for $n = 1, \ldots, N - 1, \mathbf{v} \in X_h$.

Theorem 24 *Let an A-stable, s-stage Runge–Kutta method of order p and stage order q be given which satisifes the assumptions of Sect. 3 and let $u(t)$ be the solution of (27) and $\mathbf{U}$ solution of (32). If $\mathbf{u}^h$ denotes the solution at full time steps, i.e., $\mathbf{u}^h_{n+1} = \mathbf{U}_{n,s}$ and if $f \in C^p([0, T]; L^2(\Omega))$ with $f^{(k)}(0) = 0$, for $k = 0, \ldots, \lceil\beta\rceil - 1$ then*

$$\|u(t_n) - \mathbf{u}^h_n\|_{L^2(\Omega)} = O(\Delta x^2) + O(h^p + h^{q+1+\beta})$$

$$+ O(h^p) \left(\sum_{\ell=0}^q \left(1 + t_n^{\beta+\ell-p}\right) \|f^{(\ell)}(0)\|_{L^2(\Omega)} \right).$$

Proof Denote by $K(z) : f \mapsto u$ the solution operator of the Laplace transformed problem (30) and note the resolvent estimate

$$\|K(z)\|_{L^2(\Omega)\leftarrow L^2(\Omega)} \leq C|z|^{-\beta}, \qquad |\arg(z)| < \min(\pi, (\pi - \delta)/\beta),$$

for any $\delta > 0$, following from (31); see also [8] and [21]. The same estimate holds for the solution operator $K_{\Delta x} : f \mapsto \mathbf{u}$ of the Galerkin discretization in the space $X_{\Delta x}$. Note also that

$$u(t) = K(\partial_t)f, \quad \mathbf{u}(t) = K_{\Delta x}(\partial_t)f, \quad \mathbf{U} = K_{\Delta x}(\partial_t^h)f, \quad \mathbf{u}^h = K_{\Delta x}(\partial_t^h)f.$$

The second to last equality above follows from standard properties of convolution quadrature, see for example [17, Sect. 4] and [22, Chap. 9], whereas the last one is simply the use of operational notation explained in Remark 7. The result now follows from Theorems 8, 23, and the triangle inequality. □

7.2 Implementation and numerical experiments

Though all the information needed for the implementation is given in the preceding pages, for the benefit of the reader we give some more detail here. Let M denote the number of degrees of freedom in space, i.e., $M = \dim X_{\Delta x}$, let $\mathbf{B}$ and $\mathbf{A}$ be the mass and stiffness matrices.

For simplicity of presentation we assume $\beta \in (0, 1)$ and let $\mathbf{U}_n \in \mathbb{R}^{sM}$ now denote the vector $\mathbf{U}_n = [\mathbf{U}_{n,1}, \ldots, \mathbf{U}_{n,s}]^T$ with $\mathbf{U}_{n,\ell} \approx \mathbf{u}(t_{n,\ell})$, $\ell = 1, \ldots, s$. Hence the fully discrete system can be written as a system of linear equations

$$\left((\partial_t^h)^\beta \otimes \mathbf{B}\right)\mathbf{U} + (\mathbf{I}_s \otimes \mathbf{A})\mathbf{U} = \mathbf{F}, \quad \text{for } \mathbf{t}_n \in [0, T], \tag{33}$$

where $\mathbf{I}_D$ denotes the identity matrix of size $D \times D$ and $\mathbf{F}_j \in \mathbb{R}^{sM}$.

Note that the composition rule allows us to write the CQ approximation to $\partial_t^\beta y$, as

$$(\partial_t^h)^\beta \mathbf{Y} = (\partial_t^h)^{\beta-m}(\partial_t^h)^m \mathbf{Y},$$

with $(\mathbf{Y}_j)_\ell \approx y(t_{j,\ell})$ and $m = \lceil \beta \rceil$. As $\beta - m < 0$, the discrete version $(\partial_t^h)^{\beta-1}$ of the fractional integral $\partial_t^{\beta-1} = \mathcal{I}^{1-\beta}$ can be evaluated by our fast algorithm, whereas $(\partial_t^h)^m$ is the standard one-step Runge–Kutta approximation of the derivative repeated m times.

For simplicity of presentation introduce new variables $\mathbf{V}_j \in \mathbb{R}^{Ms}$ with

$$\mathbf{V}_n = \left(\partial_t^h \otimes \mathbf{I}_M\right)\mathbf{U}(\mathbf{t}_n).$$

Note that

$$\mathbf{V}_n = (\mathbf{I}_M \otimes \mathbf{D}_0)\mathbf{U}_n + (\mathbf{I}_M \otimes \mathbf{D}_1)\mathbf{U}_{n-1}, \quad n = 1, \ldots, N,$$

where from (13) we have that

$$\mathbf{D}_0 = \frac{1}{h}\mathcal{Q}^{-1}, \quad \mathbf{D}_1 = \frac{1}{h}\mathcal{Q}^{-1}\mathbb{1}\mathbf{b}^T\mathcal{Q}^{-1}.$$

Then the fully discrete system (33) becomes

$$\sum_{j=0}^{n} \left(\mathbf{W}_{n-j} \otimes \mathbf{I}_M\right) \mathbf{V}_j + (\mathbf{I}_s \otimes \mathbf{B})^{-1} (\mathbf{I}_s \otimes \mathbf{A}) \mathbf{U}_n = (\mathbf{I}_s \otimes \mathbf{B})^{-1} \mathbf{F}_n,$$

where $\mathbf{W}_j$ are the weight matrices for the fractional integral $\mathcal{I}^\alpha$ with $\alpha = 1 - \beta$. Rearranging terms so that the known vectors are on the right-hand side and denoting

$$\mathcal{A} = \mathbf{I}_s \otimes \mathbf{A}, \qquad \mathcal{B} = \mathbf{I}_s \otimes \mathbf{B},$$

we obtain

$$(\mathbf{W}_0 \otimes \mathbf{I}_M) \mathbf{V}_n + \mathcal{B}^{-1} \mathcal{A} \mathbf{U}_n = \mathcal{B}^{-1} \mathbf{F}_n - \sum_{j=0}^{n-1} \left(\mathbf{W}_{n-j} \otimes \mathbf{I}_M\right) \mathbf{V}_j$$

or using the definition of $\mathbf{V}_n$

$$(\mathbf{W}_0 \mathbf{D}_0 \otimes \mathbf{I}_M) \mathbf{U}_n + \mathcal{B}^{-1} \mathcal{A} \mathbf{U}_n = - (\mathbf{W}_0 \mathbf{D}_1 \otimes \mathbf{I}_M) \mathbf{U}_{n-1} - \mathcal{B}^{-1} \mathbf{F}_n$$
$$- \sum_{j=0}^{n-1} \left(\mathbf{W}_{n-j} \otimes \mathbf{I}_M\right) \mathbf{V}_j.$$

At each time step this system needs to be solved, where the expensive part is the computation of the discrete convolution in the right-hand side and the storage of all the vectors $\mathbf{V}_j$. This problem is resolved by our fast method of evaluation of discrete convolutions with the following variation with respect to Sect. 5 in order to deal with the stages:

$$\sum_{j=n_0+1}^{n} \left(\mathbf{W}_j \otimes \mathbf{I}_M\right) \mathbf{V}_{n-j} \approx h \sum_{k=1}^{N_Q} w_k (r(-hx_k))^{n_0} \mathbf{Q}_{n-1,k} \tag{34}$$

with

$$\mathbf{Q}_{\ell,k} = \sum_{j=0}^{\ell-n_0-1} (r(-hx_k))^j \left((\mathbf{I}_s + hx_k \mathcal{Q})^{-1} \mathbb{1}\mathbf{q}(-hx_k) \otimes \mathbf{I}_M \right) \mathbf{V}_{\ell-n_0-1-j}$$

satisfying the recursion

$$\mathbf{Q}_{\ell,k} = r(-hx_k)\mathbf{Q}_{\ell-1,k} + \left((\mathbf{I}_s + hx_k \mathcal{Q})^{-1} \mathbb{1}\mathbf{q}(-hx_k) \otimes \mathbf{I}_M \right) \mathbf{f}_{\ell-n_0-1}, \quad \mathbf{Q}_{n_0,k} = 0.$$

As a final point let us note that due to (12)

$$\mathbf{W}_0 = \left(\frac{\mathbf{\Delta}(0)}{h} \right)^{-\alpha} = h^\alpha \mathcal{Q}^\alpha$$

Table 3 We show the error and the memory requirements for the new method and the standard implementation of CQ

N	Error	Memory (MB)	SE	Standard mem. (MB)
32	2.94×10^{-1}	39.1	2.94×10^{-1}	59.2
64	3.07×10^{-2}	40.3	3.07×10^{-2}	98.7
128	2.61×10^{-3}	42.8	2.61×10^{-3}	177.6
256	2.98×10^{-4}	44.0	3.01×10^{-4}	335.4

and hence

$$\mathbf{W}_0 \mathbf{D}_0 = (h)^{\alpha-1} \mathcal{Q}^{\alpha-1} = (h)^{-\beta} \mathcal{Q}^{-\beta}.$$

As the spectrum of $\mathcal{Q}^{-\beta}$ is in the right-half complex plane the problem to be solved at each time-step has a unique solution.

For the numerical experiments we let Ω be the square with corners $(-1, -1)$ and $(1, 1)$ and choose f so that the exact solution is

$$u(x, t) = \sin^3 \left(\tfrac{3}{2}\pi t\right) \cos \left(\tfrac{1}{2}\pi x_1\right) \cos \left(\tfrac{1}{2}\pi x_2\right).$$

We let the final time be $T = 7$, fix the finite element space on a triangular mesh with meshwidth $\Delta x = 5 \times 10^{-3}$ and compute the error in the $L^2(\Omega)$ norm at $t = T$. The error and memory requirements as the number of time-steps is increased are given in Table 3 for our new method and for the standard implementation of the CQ. We have used as tolerance tol $= 10^{-4}$ and the 2-stage RadauIIA based CQ, for which the theory predicts convergence of order $O(h^3)$. We see that the error is the same for the two implementations of the CQ, achieving in both cases the predicted order 3, but that the memory requirements for the new method stay almost constant whereas for the standard implementation they grow linearly.

References

1. Adolfsson, K., Enelund, M., Larsson, S.: Space-time discretization of an integro-differential equation modeling quasi-static fractional-order viscoelasticity. J. Vib. Control **14**(9–10), 1631–1649 (2008)
2. Baffet, D.: A Gauss–Jacobi kernel compression scheme for fractional differential equations. (2018). arXiv:1801.06095
3. Baffet, D., Hesthaven, J.S.: A kernel compression scheme for fractional differential equations. SIAM J. Numer. Anal. **55**(2), 496–520 (2017)
4. Banjai, L.: Multistep and multistage convolution quadrature for the wave equation: algorithms and experiments. SIAM J. Sci. Comput. **32**(5), 2964–2994 (2010)

5. Banjai, L., Lubich, C.: An error analysis of Runge–Kutta convolution quadrature. BIT **51**(3), 483–496 (2011)
6. Banjai, L., Lubich, C., Melenk, J.M.: Runge–Kutta convolution quadrature for operators arising in wave propagation. Numer. Math. **119**(1), 1–20 (2011)
7. Banjai, L., Schanz, M.: Wave propagation problems treated with convolution quadrature and BEM. In: Langer, U., Schanz, M., Steinbach, O., Wendland, W.L. (eds.) Fast Boundary Element Methods in Engineering and Industrial Applications, Lecture Notes in Applied and Computational Mechanics, vol. 63, pp. 145–184. Springer, Berlin (2012)
8. Cuesta, E., Lubich, C., Palencia, C.: Convolution quadrature time discretization of fractional diffusion-wave equations. Math. Comput. **75**(254), 673–696 (2006)
9. Hairer, E., Lubich, C., Schlichte, M.: Fast numerical solution of nonlinear Volterra convolution equations. SIAM J. Sci. Stat. Comput. **6**(3), 532–541 (1985)
10. Hairer, E., Wanner, G.: Solving ordinary differential equations. II, vol. 14 of Springer Series in Computational Mathematics, 2nd edition. Springer, Berlin (1996)
11. Henrici, P.: Applied and Computational Complex Analysis, vol. 2. Wiley, New York (1977)
12. Jiang, S., Zhang, J., Zhang, Q., Zhang, Z.: Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations. (2015). arXiv:1511.03453
13. Li, J.-R.: A fast time stepping method for evaluating fractional integrals. SIAM J. Sci. Comput. **31**(6), 4696–4714 (2009/10)
14. Lubich, C.: Discretized fractional calculus. SIAM J. Math. Anal. **17**(3), 704–719 (1986)
15. Lubich, C.: Convolution quadrature and discretized operational calculus I. Numer. Math. **52**, 129–145 (1988)
16. Lubich, C.: Convolution quadrature and discretized operational calculus II. Numer. Math. **52**, 413–425 (1988)
17. Lubich, C.: On the multistep time discretization of linear initial-boundary value problems and their boundary integral equations. Numer. Math. **67**, 365–389 (1994)
18. Lubich, C., Ostermann, A.: Runge–Kutta methods for parabolic equations and convolution quadrature. Math. Comput. **60**(201), 105–131 (1993)
19. Lubich, C., Schädle, A.: Fast convolution for nonreflecting boundary conditions. SIAM J. Sci. Comput. **24**(1), 161–182 (2002)
20. Nochetto, R.H., Otárola, E., Salgado, A.J.: A PDE approach to space-time fractional parabolic problems. SIAM J. Numer. Anal. **54**(2), 848–873 (2016)
21. Pazy, A.: Semigroups of Linear Operators and Applications to Partial Differential Equations, vol. 44. Springer, New York (1983)
22. Sayas, F.-J.: Retarded Potentials and Time Domain Boundary Integral Equations, vol. 50. Springer, Heidelberg (2016)
23. Schädle, A., López-Fernández, M., Lubich, C.: Fast and oblivious convolution quadrature. SIAM J. Sci. Comput. **28**(2), 421–438 (2006)
24. Trefethen, L.N.: Approximation Theory and Approximation Practice. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2013)
25. Yu, Y., Perdikaris, P., Karniadakis, G.E.: Fractional modeling of viscoelasticity in 3D cerebral arteries and aneurysms. J. Comput. Phys. **323**, 219–242 (2016)
26. Yuste, S.B., Acedo, L., Lindenberg, K.: Reaction front in an $A + B \rightarrow C$ reaction-subdiffusion process. Phys. Rev. E **69**, 036126 (2004)
27. Zeng, F., Turner, I., Burrage, K.: A stable fast time-stepping method for fractional integral and derivative operators. (2017). arXiv:1703.05480