SURVEY PAPER

AN APPROACH TO CONSTRUCT HIGHER ORDER
TIME DISCRETISATION SCHEMES FOR TIME
FRACTIONAL PARTIAL DIFFERENTIAL EQUATIONS
WITH NONSMOOTH DATA

Neville J. Ford 1, Yubin Yan 2

Dedicated to the 20th anniversary of FCAA
and 65th anniversary of Professor Virginia Kiryakova

Also, to the Memory of Christopher TH Baker (1939-2017)

Abstract

In this paper, we shall review an approach by which we can seek higher
order time discretisation schemes for solving time fractional partial differ-
cential equations with nonsmooth data. The low regularity of the solutions
of time fractional partial differential equations implies standard time dis-
cretisation schemes only yield first order accuracy. To obtain higher order
time discretisation schemes when the solutions of time fractional partial
differential equations have low regularities, one may correct the starting
steps of the standard time discretisation schemes to capture the singulari-
ties of the solutions. We will consider these corrections of some higher order
time discretisation schemes obtained by using Lubich’s fractional multistep
methods, L1 scheme and its modification, discontinuous Galerkin methods,
etc. Numerical examples are given to show that the theoretical results are
consistent with the numerical results.

MSC 2010: Primary 65M06; Secondary 65M12, 65M15, 26A33, 35R11
Key Words and Phrases: time fractional partial differential equations; finite difference method; convolution quadrature; error estimates

1. Introduction

In this paper, we review an approach to looking for higher order time discretisation methods for solving the following time fractional partial differential equations, with $0 < \alpha < 1$,

$$\frac{\mathcal{C}_0}{\mathcal{D}^\alpha_t} u(t) + Au(t) = f(t), \quad \text{for } 0 < t \leq T,$$

with $u(0) = u_0$, \hspace{1cm} (1.1)

where $\frac{\mathcal{C}_0}{\mathcal{D}^\alpha_t} u(t)$ denotes the Caputo fractional derivative defined by

$$\frac{\mathcal{C}_0}{\mathcal{D}^\alpha_t} u(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-s)^{-\alpha} u'(s) \, ds,$$

and $u'(s) = \partial u/\partial s$ and $A$ is a selfadjoint positive definite second order elliptic partial differential operator in $\Omega \subset \mathbb{R}^d, d = 1, 2, 3$, with $\mathcal{D}(A) = H^1_0(\Omega) \cap H^2(\Omega)$, where $H^1_0(\Omega), H^2(\Omega)$ denote the standard Sobolev spaces.

For example, we may choose $A = -\Delta$ subject to the Dirichlet boundary condition, where $\Delta$ denotes the Laplacian.

The assumption that $A$ is positive definite implies that $A$ generates an analytic semigroup, so that for some $\pi/2 < \theta_0 < \pi$ and with $C = C_{\theta_0}$ we have the resolvent estimate, see Lubich et al. \cite{33}, Thomée \cite{48},

$$\| (zI + A)^{-1} \| \leq C |z|^{-1} \quad \text{for } z \in \Sigma_{\theta_0} = \{ z \neq 0 : |\arg z| < \theta_0 \}. \hspace{1cm} (1.2)$$

In our analysis, we will choose $\theta > \pi/2$ close to $\pi/2$ such that $\theta < \theta_0$ which implies that $z^\alpha \in \Sigma_{\theta_0}$ for any $z \in \Sigma_{\theta}$ since $\arg(z^\alpha) = \alpha \theta < \theta < \theta_0$ for $0 < \alpha < 1$. Hence there exists a constant $C$ which depends only on $\theta$ and $\alpha$ such that, see Jin et al. \cite{21} (2.3)],

$$\| (z^\alpha I + A)^{-1} \| \leq C |z|^{-\alpha}, \quad \forall z \in \Sigma_\theta = \{ z \neq 0 : |\arg z| < \theta \}. \hspace{1cm} (1.3)$$

Under the assumptions that the solutions of (1.1) are sufficiently smooth, for example $u \in C^2[0, T]$, there are many time discretisation schemes in the literature, for example, Lubich’s fractional multistep methods \cite{7}, \cite{30}, \cite{55}, \cite{65}, \cite{61}, \cite{62}, \cite{64}, the L1 scheme and its modification \cite{28}, \cite{26}, \cite{18}, \cite{29}, \cite{47}, the spectral method \cite{55}, \cite{63}, \cite{4}, \cite{68}, \cite{57}, nonpolynomial collocation method \cite{14}, discontinuous Galerkin method \cite{39}, etc.

However, the solutions of (1.1) are not smooth in general. The regularity of the solution of (1.1) is restrictive. For example, for the homogeneous equation with initial data $u_0 \in L_2(\Omega)$, we have the following stability estimate \cite{41}

$$\| \frac{\mathcal{C}_0}{\mathcal{D}^\alpha_t} u \|_{L_2} \leq C t^{-\alpha} \| u_0 \|_{L_2}.$$

That is, the $\alpha$th order Caputo derivative is already unbounded when $t \to 0$. Hence, the $C^2$-regularity assumption generally does not hold for (1.1).
Numerical experiments indicate that the convergence rate actually does not hold uniformly in $t$ even for smooth data $u_0$. Stynes et al. [46] showed that the solution $u$ of (1.1) behaves as $t^\alpha$, $0 < \alpha < 1$ which is not in $C^2[0, T]$, see also Stynes [45]. Hence error estimates of the time discretisation schemes of (1.1) obtained under the assumptions that the solutions of (1.1) are sufficiently smooth do not hold when the solutions of (1.1) are not smooth, for example $u \notin C^2[0, T]$. In other words, the optimal convergence rates of the time discretisation schemes cannot be achieved when the solutions are not smooth. Therefore it is interesting to consider error estimates of time discretisation schemes for solving (1.1) when the solutions of (1.1) are not smooth. There are several alternative approaches to construct higher order time discretisation schemes for solving (1.1) when the solutions of (1.1) are not smooth.

1. Use non-uniform meshes and approximate the solution near the singular point $t = 0$ by using finer meshes, see [36], [39], [43], [56], [66].

2. Use some nonpolynomial (or singular) basis functions or collocation spectral methods to capture the singularity of the solutions of (1.1), see [1], [5], [13], [14], [34], [27], [59], [63], [67].

3. Separate the solution into two parts: smooth and nonsmooth parts. The nonsmooth part of the solution has the form $u(t) = t^{\lambda_l}, l = 0, 1, 2, \ldots, s$ with $0 \leq \lambda_0 < \lambda_1 < \cdots < \lambda_s$. Approximate the nonsmooth part by introducing some starting weights such that the time discretisation scheme holds exactly for the nonsmooth part $u(t) = t^{\lambda_l}, l = 0, 1, 2, \ldots, s$ with $0 \leq \lambda_0 < \lambda_1 < \cdots < \lambda_s$, see [30], [60], [62].

4. Introduce a Correction to the starting steps of the time discretisation schemes for solving (1.1) to capture the singularity of the solution of (1.1), see [33], [22], [23], [50], [52], [53].

Approach 4 is the main focus of this paper. Approach 3 is most closely related to it and we make the following remarks: Let $0 < t_0 < t_1 < \cdots < t_N = T$ be a partition of $[0, T]$ and $\tau$ the time step size. To construct a time discretisation scheme for solving (1.1), we first need to approximate the Caputo fractional derivative $C_0^\alpha D_t^\tau u(t)$ at $t = t_n, n = 1, 2, \ldots, N$. Note that the Caputo fractional derivative has the following relation with the Riemann-Liouville fractional derivative, $C_0^\alpha D_t^\tau u(t) = R_0^\alpha D_t^\tau (u(t) - u(0)), 0 < \alpha < 1$, where $R_0^\alpha D_t^\tau u(t)$ denotes the Riemann-Liouville fractional derivative defined by

$$R_0^\alpha D_t^\tau u(t) = \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_0^t (t-s)^{-\alpha} u(s) \, ds.$$
We may consider the approximation of the Riemann-Liouville fractional derivative instead. Based on Lubich’s convolution quadrature [30], [31], [32], the Riemann-Liouville fractional derivative $R_0^t D_\alpha^t u(t)\text{ can be approximated by, with } 0 < \alpha < 1,$

$$R_0^t D_\alpha^t u(t) \approx \tau^{-\alpha} \sum_{j=0}^{n} w_{n-j} u(t_j). \quad (1.4)$$

Here $w_j, j = 0, 1, 2, \ldots$ are the weights generated by the characteristic polynomial $\delta(\zeta)$ where, [23, (2.2)]

$$\delta(\zeta) = \sum_{j=0}^{\infty} w_j \zeta^j \quad \text{with} \quad \delta(\zeta) = \sum_{j=1}^{p} \frac{1}{j} (1 - \zeta)^j, \quad p = 1, 2, \ldots, 6. \quad (1.5)$$

The approximation method (1.4) generated by (1.5) is also called the backward difference method with order $p$ (BDF$^p$) where $p = 1, 2, \ldots, 6,$ [30], [31], [32]. When $p = 1,$ it is also called the Grünwald-Letnikov approximation method [42].

If the solution $u$ is sufficiently smooth and has many vanishing derivatives at 0, then we have, [30], [31], [32],

$$R_0^t D_\alpha^t u(t_n) = \tau^{-\alpha} \sum_{j=0}^{n} w_{n-j} u(t_j) + O(\tau^p), \quad p = 1, 2, \ldots, 6. \quad (1.6)$$

However the solutions of (1.1) has low regularity [44]. We only obtain first-order accuracy when solving (1.1) by using the approximation scheme (1.6) with $p \geq 2.$ This has been observed numerically [7], [20].

To obtain the convergence rate $O(\tau^p), p = 2, \ldots, 6,$ for the approximation scheme (1.6) when $u(t)$ has low regularity, one may use the starting weights to correct the approximate scheme (1.6). More precisely, we may have, for $n \geq 1,$

$$R_0^t D_\alpha^t u(t_n) = \tau^{-\alpha} \sum_{j=0}^{n} w_{n-j} u(t_j) + \sum_{j=0}^{s} w_{n,j} u(t_j) + O(\tau^p), \quad p = 2, \ldots, 6, \quad (1.7)$$

for some suitable starting weights $w_{n,j}, j = 0, 1, \ldots, s$ with some $s \in \mathbb{Z}^+.$

To determine the starting weights, we assume that (1.7) holds exactly for $u(t) = t^{\lambda_l}, l = 0, 1, \ldots, s,$ where $0 \leq \lambda_0 < \lambda_1 < \cdots < \lambda_s$ are some numbers depending on $0 < \alpha < 1$ and $p.$ Hence we may obtain the starting weights by solving the following linear system, with $u(t) = t^{\lambda_l}, l = 0, 1, \ldots, s,$

$$R_0^t D_\alpha^t u(t_n) = \tau^{-\alpha} \sum_{j=0}^{n} w_{n-j} u(t_j) + \sum_{j=0}^{s} w_{n,j} u(t_j), \quad p = 2, \ldots, 6. \quad (1.8)$$
For each fixed $n \geq 1, n \geq s$, we need to solve the linear system (1.8) to determine the starting weights $w_{n,j}, j = 0, 1, \ldots, s$ in order to calculate the approximate solutions of the considered fractional differential equations. This idea works well for solving the following nonlinear fractional differential equation, with $0 < \alpha < 1$,

$$\frac{D^\alpha_0}{D_t^\alpha} y(t) = f(t, y(t)), \quad \text{with } y(0) = y_0,$$

(1.9)

since the solution of (1.9) is well understood and it can be expressed as the following form, for sufficiently smooth $f$,

$$u(t) = c_0 t^\lambda_0 + c_1 t^\lambda_1 + \cdots + c_s t^\lambda_s + \psi(t),$$

for some $0 = \lambda_0 < \lambda_1 < \cdots < \lambda_s$. Here $\psi(t)$ is some smooth function and $c_i \in \mathbb{R}, i = 0, 1, \ldots, s, s \in \mathbb{Z}^+$. Diethelm et al. [12] and Weilbeer [49] discussed the implementation of this idea in detail for solving (1.9), see also [3], [54], [6], [58], [61].

More recently, Zeng et al. [65] extended this idea to construct the higher order time discretisation scheme for solving time fractional partial differential equation (1.1). Under the assumption that the solution $u(t, x)$ of (1.1) satisfies

$$u(t, x) - u(0, x) = \sum_{j=0}^{s} c_j(x) t^{\gamma_j} + c_{s+1}(x) t^{\gamma_{s+1}} + \ldots,$$

(1.10)

for some $0 < \gamma_0 < \gamma_1 < \cdots < \gamma_s, s \in \mathbb{Z}^+$, Zeng et al. [65] constructed a second order numerical method for solving (1.1) by using starting weights and the weighted shifted Grünwald-Letnikov formula for both smooth and nonsmooth solutions. However Jin et al. [23] mentioned that the assumption (1.10) requires imposing certain compatibility conditions on the source term $f$ which may not be satisfied for the general $f$ in (1.1).

Without using the assumption (1.10) and the starting weights, we may also obtain higher order time discretisation schemes by using Approach 4, i.e., by correcting the starting steps of the standard time discretisation schemes when the solutions of (1.1) are not smooth. This idea was first used in Lubich et al. [33] to construct a second order time discretisation scheme for solving diffusion wave equation, see (3.14) and (3.15) in Section 3 below. They proved the error estimates by using the Laplace transform method. Recently Jin et al. [22], [23] and Yan et al. [50], [52], [53] used the similar ideas as in Lubich et al. [33] to construct higher order time discretisation methods for solving (1.1). In this paper, we will review some recent results for constructing the time discretisation scheme for solving (1.1) by using Approach 4 when the solutions of (1.1) are not smooth.

The paper is organized as follows. In Section 2, we review the corrected Lubich’s fractional multistep methods for solving time fractional partial differential equations. In Section 3, we review the corrected L1 scheme (or...
Diethelm’s finite difference method) and its modifications for solving time fractional partial differential equations. In Section 4, we consider the discontinuous Galerkin method for solving time fractional partial differential equations. Finally, we propose some open problems and issues for further work.

Throughout, the notations $C$ and $c$, with or without a subscript, denote generic constants, which may differ at different occurrences, but are always independent of the step size $k$.

2. Correction of the Lubich fractional multistep methods

To obtain a higher order time discretisation method for solving (1.1), Jin et al. [23] introduced some corrected backward difference formula (BDF) for solving (1.1).

To see this, we first note that (1.1) is equivalent to the following form

$$ R_0^\alpha D_t^\alpha (u(t) - u(0)) + Au(t) = f(t), \quad \text{for } 0 < t \leq T, \quad \text{with } u(0) = u_0. \quad (2.1) $$

Let $U^n \approx u(t_n)$ denote the approximate solution of $u(t_n)$. We define the following time discretisation scheme for solving (2.1), with $n \geq 1$,

$$ \tau^{-\alpha} \sum_{j=0}^{n} w_{n-j}(U^j - U^0) + AU^n = f(t_n), \quad \text{with } U^0 = u_0. \quad (2.2) $$

Jin et al. [23] showed that $\|U^n - u(t_n)\|$ has only first order accuracy even if the initial data $u_0$ and the source term $f$ are sufficiently smooth.

To restore the $p$th accuracy with $p = 2, 3, \ldots, 6$, Jin et al. [23] corrected the BDFp at the starting $p-1$ steps by (as usual, the summation disappears if the upper index is smaller than the lower one)

$$ \tau^{-\alpha} \sum_{j=0}^{n} w_{n-j}(U^j - U^0) + AU^n $$

$$ = f(t_n) + a_n^{(p)}(Au_0 + f(0)) + \tau^l \sum_{l=1}^{p-2} b_{l,n}^{(p)} f(l)(0), 1 \leq n \leq p - 1, \quad (2.3) $$

$$ \tau^{-\alpha} \sum_{j=0}^{n} w_{n-j}(U^j - U^0) + AU^n = f(t_n), \quad p \leq n \leq N, \quad (2.4) $$

$$ U^0 = u_0, \quad (2.5) $$

where $a_n^{(p)}$ and $b_{l,n}^{(p)}$ are coefficients determined in Tables 1 and 2, respectively. They are constructed so as to improve the accuracy of the overall scheme to $O(\tau^p)$ for general initial data $u_0 \in D(A)$ and a possibly incompatible right-hand side $f$. 

Unauthenticated
Download Date | 5/30/19 11:39 PM
Theorem 2.1. (23, Theorem 2.2) Let \( u(t_n) \) and \( U^n \) be the solutions of (2.1) and (2.3)-(2.5), respectively. Then we have, with \( p = 1, 2, \ldots, 6, \)

\[
\| U^n - u(t_n) \| \leq C \tau^p \left( t_n^{\alpha-p} \| f(0) \| + A u_0 \| + \sum_{l=1}^{p-1} t_n^{\alpha+l-p} \| f^{(l)}(0) \| \right.
\]

\[
+ \int_0^{t_n} (t_n - s)^{\alpha-1} \| f^{(p)}(s) \| \, ds \right).
\]
Remark 2.1. ([23, Remark 2.3]) For any fixed $t_n > 0$, by Theorem 2.1, the corrected BDFp method (2.3)-(2.5) has convergence rate $O(\tau^p), p = 1, 2, \ldots, 6$. To get a uniform rate $O(\tau^p)$, one may need the following compatibility conditions [32]

$$f(0) + Au(0) = 0, \quad \text{and} \quad f^{(l)}(0) = 0, \quad l = 1, 2, \ldots, p - 1. \quad (2.6)$$

When these conditions do not hold, the error estimates may deteriorate as $t \to 0$ and we cannot obtain the uniform rate $O(\tau^p), p = 1, 2, \ldots, 6$.

Remark 2.2. ([23, Remark 2.4]) When the initial value $u_0 \in H = L_2(\Omega)$, we may obtain similar error estimates as in Theorem 2.1. More precisely, we have

$$\|U^n - u(t_n)\| \leq C\tau^p \left( t_n^{-p}\|f(0) + u_0\| + \sum_{l=1}^{p-1} t_n^{\alpha+l-p}\|f^{(l)}(0)\| \right. \left. + \int_0^{t_n} (t_n - s)^{\alpha-1}\|f^{(p)}(s)\| ds \right).$$

We see that if the initial value $u_0$ is nonsmooth, i.e., $u_0 \in H$, then in the error estimates, $t_n^{\alpha-p}$ reduces to $t_n^p$, in other words, $t_n$ loses $\alpha$th power if $u_0$ is nonsmooth.

3. Correction of L1 scheme and its modification

3.1. Correction of the L1 scheme. Another approach to approximate the fractional derivative is by using the L1 scheme (or the Diethelm finite difference method). The L1 scheme first appeared in the book [41] for the approximation of the Caputo fractional derivative. The L1 scheme may be obtained by direct approximation of the derivative in the definition of the Caputo fractional derivative, e.g., [28], [26], [18], [29], [47], [19], or by the approximation of the Hadamard finite-part integral, e.g., [9], [11], [15], [16], [17], [24], [51].

Let us first consider how to obtain the L1 scheme by the direct approximation of the derivative in the definition of the Caputo fractional derivative. Note that, with $0 < \alpha < 1$,

$$C_0D_t^\alpha u(t_n) = \frac{1}{\Gamma(1-\alpha)} \int_0^{t_n} (t_n - s)^{-\alpha} u'(s) \, ds = \frac{1}{\Gamma(1-\alpha)} \sum_{k=0}^{n-1} \int_{t_k}^{t_{k+1}} (t_n - s)^{-\alpha} u'(s) \, ds,$$
where \( u'(s) = \frac{du}{ds} \). Approximating \( u'(s) \) on \([t_k, t_{k+1}]\) by using the finite difference quotient \((u(t_{k+1}) - u(t_k))/\tau\), we obtain the following so-called L1 scheme

\[
\frac{C}{0} D_t^\alpha u(t_n) \approx \frac{1}{\Gamma(1-\alpha)} \int_0^{t_n} (t_n - s)^{-\alpha} \frac{u(t_{k+1}) - u(t_k)}{\tau} ds - \sum_{j=0}^{n} w_{n-j,n} u(t_j),
\]

(3.1)

where \( w_{j,n}, j = 0, 1, 2, \ldots, n \) are given by

\[
\Gamma(2-\alpha)w_{j,n} = \begin{cases} 
1, & \text{for } j = 0, \\
-2j^{1-\alpha} + (j-1)^{1-\alpha} + (j+1)^{1-\alpha}, & \text{for } j = 1, 2, \ldots, n-1, \\
(j-1)^{1-\alpha} - j^{1-\alpha}, & \text{for } j = n.
\end{cases}
\]

Lin et al. [28] and Sun et al. [47] independently proved that the truncation error of the L1 scheme is \( O(\tau^{2-\alpha}) \) for a sufficiently smooth function \( u \).

We next consider how to obtain the L1 scheme by approximating the Hadamard finite-part integral. Noting that Riemann-Liouville fractional derivative can be expressed by the Hadamard finite-part integral, Diethelm [9], [10] approximated the Hadamard finite-part integral by using the piecewise interpolation polynomials and obtained the approximate schemes to the Riemann-Liouville fractional derivatives. More precisely, we have, with \( 0 < \alpha < 1 \),

\[
R D_t^\alpha u(t) = \frac{1}{\Gamma(-\alpha)} \int_0^t (t - s)^{-\alpha-1} u(s) ds,
\]

(3.2)

where the integral \( \oint \) is in the Hadamard sense [9], [10]. At \( t = t_n \), we have

\[
R D_t^\alpha u(t_n) = \frac{1}{\Gamma(-\alpha)} \sum_{k=0}^{n-1} \oint_{t_k}^{t_{k+1}} (t_n - s)^{-\alpha-1} u(s) ds.
\]

Approximating \( u(s) \) on \([t_k, t_{k+1}]\) by using the linear interpolation polynomial

\[
P_1(s) = \frac{s - t_{k+1}}{t_k - t_{k+1}} u(t_k) + \frac{s - t_k}{t_{k+1} - t_k} u(t_{k+1}),
\]

we get

\[
\frac{C}{0} D_t^\alpha u(t_n) \approx \frac{R}{0} D_t^\alpha (u(t_n) - u(0))
\]

\[
\approx \frac{1}{\Gamma(-\alpha)} \sum_{k=0}^{n-1} \oint_{t_k}^{t_{k+1}} (t_n - s)^{-\alpha-1} P_1(s) ds - \frac{R}{0} D_t^\alpha u(0) = \tau^{-\alpha} \sum_{j=0}^{n} w_{n-j,n} u(t_j),
\]

(3.3)

where the weights \( w_{j,n}, j = 0, 1, \ldots, n \), are exactly the same as the weights defined in (3.1). Hence we conclude that, when using the piecewise linear interpolation polynomial to approximate the Hadamard finite-part integral,
the scheme (3.3) for approximating the Riemann-Liouville fractional derivative is the same as the L1 scheme (3.1) for approximating the Caputo fractional derivative.

We now define the L1 scheme for solving (2.1). From now on, we only consider the homogeneous case with $f = 0$. The inhomogeneous case can be discussed similarly by using the argument in Jin et al. [20], [22], [23].

Let $u(t) - u_0 = V(t)$. Then solving (1.1) with $f = 0$ is equivalent to find $V(t)$ such that, noting that $C_0^\alpha D_t^\alpha u_0 = 0$,

$$C_0^\alpha D_t^\alpha V(t) + AV(t) = -Au_0, \quad \text{with } V(0) = 0. \quad (3.4)$$

It proves more convenient to consider the error estimates of the time discretisation scheme of (3.4), see [33].

Let $V_j \approx V(t_j)$ be the approximate solution of $V(t_j)$ in (3.4). We define the following L1 scheme for solving (3.4),

$$\tau^{-\alpha} \sum_{j=0}^{n} w_{n-j,n} V_j + AV^n = -Au_0, \quad n \geq 1, \quad \text{with } V^0 = 0, \quad (3.5)$$

or

$$\tau^{-\alpha} \sum_{j=1}^{n} w_{n-j,n} V_j + AV^n = -Au_0, \quad n \geq 1. \quad (3.6)$$

For any fixed $n \geq 1$, we observe that $w_{j,n}, j = 0, 1, \ldots, n-1$ only depend on $j = 0, 1, 2, \ldots, n-1$. For example, we have $w_{0,n} = 1/\Gamma(2 - \alpha)$ for any $n \geq 1$, $w_{1,n} = 1/\Gamma(2 - \alpha)\{(2)1^{1-\alpha} = (1-1)^{1-\alpha} + (1+1)^{1-\alpha}\}$ for any $n \geq 2, \ldots$. Therefore, it is reasonable to write $w_{j,n} = 0, 1, 2, \ldots, n-1$ with $w_j$ since $w_{j,n}, j = 0, 1, 2, \ldots, n-1$ are independent of $n$. In other words, we denote $w_0 = w_{0,n}, w_1 = w_{1,n}, w_2 = w_{2,n}, \ldots, w_{n-1} = w_{n-1,n}$ for any fixed $n \geq 1$. Further we define $w_j$ for all $j = 0, 1, 2, \ldots$ as follows:

$$\Gamma(2 - \alpha)w_j = \begin{cases} 1, & \text{for } j = 0, \\ -2j^{1-\alpha} + (j-1)^{1-\alpha} + (j+1)^{1-\alpha}, & \text{for } j = 1, 2, \ldots. \end{cases} \quad (3.7)$$

Using the notations $w_j, j = 0, 1, 2, \ldots$ defined in (3.7), the time discretisation scheme (3.6) can now be written as

$$\tau^{-\alpha} \sum_{j=1}^{n} w_{n-j} V_j + AV^n = -Au_0, \quad n \geq 1. \quad (3.8)$$

**Theorem 3.1.** ([21 Theorem 3.16]) Let $V(t_n)$ and $V^n$ be the solutions of (3.4) and (3.8), respectively. Let $u_0 \in H = L_2(\Omega)$. Then we have, with $0 < \alpha < 1$,

$$\|V(t_n) - V^n\| \leq C \tau t_n^{-1} \|u_0\|. \quad (3.9)$$
To obtain a higher order method for solving (1.4), we correct the starting steps of the L1 scheme as used in Jin et al. [23]. Let $c_0 = 1/2$, we correct the L1 scheme for solving (3.4) as follows

$$
\tau^{-\alpha} \sum_{j=1}^{n} w_{n-j} V^j + AV^n = (-Au_0)(1 + c_0), \text{ for } n = 1,
$$

(3.10)

$$
\tau^{-\alpha} \sum_{j=1}^{n} w_{n-j} V^j + AV^n = -Au_0, \text{ for } n \geq 2,
$$

(3.11)

$$
V^0 = 0,
$$

(3.12)

where the weights $w_{n-j}, j = 1, 2, \ldots, n$ are given by (3.7).

The idea of introducing the correction term in the first step $n = 1$ in (3.10) comes from the idea in Lubich et al. [33] where the authors introduced a modified scheme to construct a higher order time discretisation scheme for solving an evolution equation with a positive-type memory term. To see this, let us first note that (3.4) is equivalent to the following form, with $0 < \alpha < 1$,

$$
V(t) + R_0 D_t^{-\alpha} (AV(t)) = -R_0 D_t^{-\alpha} (Au_0), \text{ with } V(0) = 0,
$$

(3.13)

where $R_0 D_t^{-\alpha} V(t)$ denotes the Riemann-Liouville fractional integral. To obtain a higher order time discretisation scheme for solving (3.13), following the idea in Lubich et al. [33], we may introduce the following modified time discretisation scheme to approximate (3.13),

$$
V^n + q^c_n(AV) = -q^c_n(Au_0), \text{ with } V^0 = 0,
$$

(3.14)

where $q^c_n(\varphi)$ is the modification of the quadrature formula approximating the Riemann-Liouville fractional integral $R_0 D_t^{-\alpha} \varphi$, defined by

$$
q^c_n(\varphi) = k^{-\alpha} \sum_{j=1}^{n} \beta_{n-j} \varphi^j + c_0 \beta_{n-1} \varphi^0, \text{ with } c_0 = 1/2.
$$

(3.15)

Here $\beta_0, \beta_1, \ldots$ are generated by some function $\tilde{\beta}(\zeta) = \sum_{j=0}^{\infty} \beta_j \zeta^j$.

**Lemma 3.1.** ([52]) Assume that $\tilde{\beta}(\zeta) = (\sum_{j=0}^{\infty} w_j \zeta^j)^{-1}$, where $w_j, j = 0, 1, 2, \ldots$ are defined in (3.7). Then the modified L1 scheme (3.10)-(3.12) is equivalent to (3.14).

**Remark 3.1.** From Lemma 3.1 we note that the correction on the first step $n = 1$ in (3.10)-(3.12) is equivalent to the correction in (3.14). Therefore we see that the corrected L1 scheme (3.10)-(3.12) is actually equivalent to the modified scheme (3.14) which has been used to improve the
convergence rate of the time discretisation scheme for solving an evolution equation with a positive-type memory term in Lubich et al. [33].

**Theorem 3.2.** (52) Let \( V(t_n) \) and \( V^n \) be the solutions of (3.4) and (3.10)-(3.12), respectively. Let \( u_0 \in H = L_2(\Omega) \). We have
\[
\|V(t_n) - V^n\| \leq C\tau^{2-\alpha}t_n^{\alpha-2}\|u_0\|.
\]

3.2. **Numerical examples.** In this subsection, we will consider the experimentally determined convergence rates of the L1 and the corrected L1 schemes for smooth and nonsmooth data. We only show the numerical results for the homogeneous problem here. Similarly we may consider the numerical results for the inhomogeneous problem with \( f \neq 0 \) as discussed in Jin et al. [20], [23]. Let us consider
\[
\frac{C_0}{\tau}D^\alpha_t u(x,t) - u_{xx} = 0, \quad 0 < x < 1, \quad t > 0,
\]
\[
u(0,t) = u(1,t) = 0,
\]
\[
u(x,0) = u_0(x),
\]
where \( u_0(x) = x(1 - x) \) or \( u_0(x) = \chi_{(0,1/2)} \).

Let \( 0 < t_0 < t_1 < \ldots t_N = T \) be the time partition and \( \tau \) the time step size. Let \( N_h \) be a positive integer. Let \( 0 = x_0 < x_1 < x_2 < \ldots x_{N_h} = 1 \) be the space partitions and \( h \) the space step size. We will use finite element methods to consider the spatial discretisation.

We first consider the scheme (3.8) and the convergence rate was proved to be \( O(\tau) \) for both smooth and nonsmooth data in [21].

To observe this convergence order, we first calculate the reference exact solution \( u_{ref}(t) \) at \( T = 1 \) with \( h_{ref} = 2^{-6} \) and \( \tau_{ref} = 2^{-10} \). We then use \( h = 2^{-6} \) and \( \tau = \kappa \tau_{ref} \) with \( \kappa = [2^2, 2^3, 2^4, 2^5, 2^6] \) to obtain the approximate solutions \( u(t) \) at \( t = 1 \). We choose the smooth and nonsmooth initial data (a) \( u_0 = x(1-x) \) and the nonsmooth data (b) \( u_0 = \chi_{(0,1/2)} \) we obtain the following results which are consistent with the Table 1 in [21]. In Table 3 we observe that the convergence rate indeed is almost \( O(\tau) \) for the different \( \alpha \in (0,1) \) in both smooth and nonsmooth data cases.

We next consider the corrected L1 scheme (3.10)-(3.12). By Theorem 3.2, the convergence rate of the corrected L1 scheme (3.10)-(3.12) is \( O(\tau^{2-\alpha}) \) for smooth and nonsmooth data. To observe this convergence rate, we first calculate the reference exact solution \( u_{ref}(t) \) at \( T = 1 \) with \( h_{ref} = 2^{-6} \) and \( \tau_{ref} = 2^{-10} \). We then use \( h = 2^{-6} \) and \( \tau = \kappa \tau_{ref} \) with \( \kappa = [2^2, 2^3, 2^4, 2^5, 2^6] \) to obtain the approximate solutions \( u(t) \) at \( t = 1 \). We choose the smooth and nonsmooth initial data (a) \( u_0 = \sin(2\pi x) \) and (b) \( u_0 = \chi_{(0,1/2)} \) we obtain the following results.
We found that in Table 4 the corrected L1 scheme has the better accuracy than the L1 scheme and the errors are about $1e-05$ or $1e-04$ for all $\alpha \in (0, 1)$. The errors of the L1 scheme are only $1e-03$. For the convergence rates, when $\alpha < 1/2$, we observe that, in Table 4, the convergence rates are almost 2 which is better than the theoretical results $2 - \alpha$. However when $\alpha > 1/2$, the convergence rates are almost $2 - \alpha$ as we expected.

### Table 3. Time convergence rates with the different $\alpha$ for the L1 scheme (3.3)

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\tau = 2^{-5}$</th>
<th>$\tau = 2^{-4}$</th>
<th>$\tau = 2^{-3}$</th>
<th>$\tau = 2^{-2}$</th>
<th>$\tau = 2^{-1}$</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.0007e-5</td>
<td>0.0030e-5</td>
<td>0.0125e-5</td>
<td>0.0517e-5</td>
<td>0.2197e-5</td>
<td>2.067</td>
</tr>
<tr>
<td>0.3</td>
<td>0.0013e-5</td>
<td>0.0064e-5</td>
<td>0.0291e-5</td>
<td>0.1302e-5</td>
<td>0.5891e-5</td>
<td>2.191</td>
</tr>
<tr>
<td>0.8</td>
<td>0.0079e-4</td>
<td>0.0201e-4</td>
<td>0.0462e-4</td>
<td>0.1782e-4</td>
<td>0.7485e-4</td>
<td>1.123</td>
</tr>
<tr>
<td>0.9</td>
<td>0.0141e-4</td>
<td>0.0345e-4</td>
<td>0.0778e-4</td>
<td>0.3484e-4</td>
<td>1.1573</td>
<td></td>
</tr>
</tbody>
</table>

### Table 4. Time convergence rates with the different $\alpha$ for the modified L1 scheme (3.10)-(3.12)

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\tau = 2^{-5}$</th>
<th>$\tau = 2^{-4}$</th>
<th>$\tau = 2^{-3}$</th>
<th>$\tau = 2^{-2}$</th>
<th>$\tau = 2^{-1}$</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.0001e-5</td>
<td>0.0003e-5</td>
<td>0.00125e-5</td>
<td>0.00517e-5</td>
<td>0.02197e-5</td>
<td>2.067</td>
</tr>
<tr>
<td>0.3</td>
<td>0.00013e-5</td>
<td>0.00064e-5</td>
<td>0.00291e-5</td>
<td>0.01302e-5</td>
<td>0.05891e-5</td>
<td>2.191</td>
</tr>
<tr>
<td>0.8</td>
<td>0.00079e-4</td>
<td>0.00201e-4</td>
<td>0.00462e-4</td>
<td>0.01782e-4</td>
<td>0.07485e-4</td>
<td>1.123</td>
</tr>
<tr>
<td>0.9</td>
<td>0.00141e-4</td>
<td>0.00345e-4</td>
<td>0.00778e-4</td>
<td>0.03484e-4</td>
<td>0.11573</td>
<td></td>
</tr>
</tbody>
</table>

### 3.3. Correction of the modified L1 scheme

In this subsection we will introduce a corrected higher order time discretisation scheme (which we call the modified L1 scheme) for solving (3.4) and show the error estimates.
for the nonsmooth data [50]. Let us first introduce a higher order scheme (the modified L1 scheme) to approximate the Riemann-Liouville derivative by using Diethelm’s approach [9], [10].

**Lemma 3.2.** Let \(0 < \alpha < 1\) and assume that \(u \in C^3[0, T]\). Let \(n \geq 5\), then we have

\[
R_0^\alpha D_t^n u(t_n) = \tau^{-\alpha} \sum_{j=0}^{n} w_{j,n} u(t_{n-j}) + O(\tau^{3-\alpha}),
\]

where

\[
\Gamma(3-\alpha)w_{j,n} = \begin{cases} 
(1 + \frac{\alpha}{2})2^{1-\alpha}, & j = 0, \\
(2 + \frac{3\alpha}{2})3^{1-\alpha} - (3 + \frac{3\alpha}{2})2^{1-\alpha}, & j = 1, \\
(3 + \frac{1}{2})4^{1-\alpha} - (6 + \frac{3\alpha}{2})3^{1-\alpha} + (3 + \frac{3\alpha}{2})2^{1-\alpha}, & j = 2,
\end{cases}
\]

and, with \(j = 3, 4, \ldots, n - 2\),

\[
\Gamma(3-\alpha)w_{j,n} = (j + 2)^{2-\alpha} - (1 - \frac{\alpha}{2})(j + 2)^{1-\alpha} - 3(j + 1)^{2-\alpha} \\
+ (3 - \frac{3\alpha}{2})(j + 1)^{1-\alpha} + 3j^{2-\alpha} - (3 - \frac{3\alpha}{2})j^{1-\alpha} - (j - 1)^{2-\alpha} \\
+ (1 - \frac{\alpha}{2})(j - 1)^{1-\alpha}, & j = 3, 4, \ldots n - 2,
\]

and

\[
\Gamma(3-\alpha)w_{n-1,n} = (4 - 2n - 2\alpha)n^{1-\alpha} + (3n - 6 + \frac{3\alpha}{2})(n - 1)^{1-\alpha} \\
- (n + \frac{\alpha}{2} - 3)(n - 2)^{1-\alpha},
\]

\[
\Gamma(3-\alpha)w_{n,n} = (n - 3 + \frac{3\alpha}{2})n^{1-\alpha} + (1 - \alpha)(2 - \alpha)n^{-\alpha} - (n - 2 + \frac{\alpha}{2})(n - 1)^{1-\alpha}.
\]

**Proof.** Note that the Riemann-Liouville fractional derivative \(R_0^\alpha D_t^n u(t)\), \(0 < \alpha < 1\) may be written as

\[
R_0^\alpha D_t^n u(t) = \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_0^t (t-s)^{-\alpha} u(s) \, ds = \frac{1}{\Gamma(-\alpha)} \int_0^t (t-s)^{-\alpha-1} u(s) \, ds,
\]

where the integral \(\int\) must be interpreted as a Hadamard finite-part integral, see e.g., [9], [10].

At \(t = t_n\), \(n = 5, 6, \ldots, N\), we have

\[
R_0^\alpha D_t^n u(t_n) = \frac{1}{\Gamma(-\alpha)} \int_0^{t_n} (t_n-s)^{-\alpha-1} u(s) \, ds = \frac{t_n^{-\alpha}}{\Gamma(-\alpha)} \int_0^1 w^{-\alpha-1} u(t_n - t_n w) \, dw
\]

\[
= \frac{t_n^{-\alpha}}{\Gamma(-\alpha)} \sum_{j=1}^{n} \int_{w_{j-1}}^{w_j} w^{-\alpha-1} u(t_n - t_n w) \, dw.
\]
Denote \( g(w) = u(t_n - t_n w) \) and approximate \( g(w) \) by the piecewise quadratic interpolation polynomial \( g_2(w) \) defined on the nodes \( w_i = \frac{n}{n}, l = 0, 1, 2, \ldots, n \) by

\[
g_2(w) = \frac{(w - w_1)(w - w_2)}{(w_0 - w_1)(w_0 - w_2)}g(w_0) + \frac{(w - w_0)(w - w_2)}{(w_1 - w_0)(w_1 - w_2)}g(w_1)
+ \frac{(w - w_0)(w - w_1)}{(w_2 - w_0)(w_2 - w_1)}g(w_2), \quad \text{for } w \in [w_0, w_1],
\]

and

\[
g_2(w) = \frac{(w-w_{j-1})(w-w_j)}{(w_{j-2}-w_{j-1})(w_{j-2}-w_j)}g(w_{j-2}) + \frac{(w-w_{j-2})(w-w_j)}{(w_{j-1}-w_{j-2})(w_{j-1}-w_j)}g(w_{j-1})
+ \frac{(w-w_{j-2})(w-w_{j-1})}{(w_j-w_{j-2})(w_j-w_{j-1})}g(w_j), \quad \text{for } w \in [w_{j-1}, w_j], j = 2, 3, \ldots, n.
\]

More precisely, on \([w_{j-1}, w_j], j = 2, 3, \ldots, n\), \( g(w) \) is approximated by the quadratic interpolation polynomial \( g_2(w) \) determined by \( g(w_{j-2}), g(w_{j-1}), g(w_j) \), and on \([w_0, w_1], g(w) \) is approximated by the quadratic interpolation polynomial \( g_2(w) \) determined by \( g(w_0), g(w_1), g(w_2) \).

By \([10, Theorem 2.4]\), we have the following error estimates

\[
R_n^\alpha D_t^\alpha u(t_n) = \frac{t_n^\alpha}{\Gamma(-\alpha)} \int_0^1 w^{-\alpha-1} g(w) \, dw
= \frac{t_n^\alpha}{\Gamma(-\alpha)} \int_0^1 w^{-\alpha-1} g_2(w) \, dw + O(k^{3-\alpha}). \tag{3.19}
\]

We next calculate the integral on the right side of \((3.19)\). Note that

\[
\int_0^1 w^{-\alpha-1} g_2(w) \, dw = \int_0^{w_1} w^{-\alpha-1} g_2(w) \, dw + \sum_{j=2}^n \int_{w_{j-1}}^{w_j} w^{-\alpha-1} g_2(w) \, dw, \tag{3.20}
\]

where only the first integral on the right side of \((3.20)\) is in the sense of the Hadamard finite-part integral and the other integrals on the right side of \((3.20)\) are in the usual sense. By using the definition of the Hadamard finite-part integral \([9, 10]\), we have

\[
\int_0^{w_1} w^{-\alpha-1} g_2(w) \, dw = \frac{w_1^{-\alpha}}{\Gamma(1-\alpha)} g_2(0) + \int_0^{w_1} g_2'(y) \, dy \, dw
= \frac{w_1^{-\alpha}}{\Gamma(1-\alpha)} g_2(0) + \int_0^{w_1} w^{-\alpha-1} (\int_0^y g_2''(y) \, dy) \, dw,
\]

where

\[
g_2'(y) = (n^2/2)(2y - (w_1 + w_2))g(0) - n^2(2y - w_2)g(w_1)
+ (n^2/2)(2y - w_1)g(w_2).
\]
AN APPROACH TO CONSTRUCT HIGHER ORDER ... 1091

Hence,

\[
\int_0^{w_1} w^{-\alpha-1} g_2(w) \, dw = \frac{1}{(-\alpha)(-\alpha+1)(-\alpha+2)\alpha} \times \left((2 - \alpha/2)g(0) + (-\alpha)(3 - \alpha)g(w_1) + (\alpha/2)g(w_2)\right).
\]

For the other integrals in \((3.20)\), we have, with \(j = 2, 3, \ldots, n\),

\[
\int_{w_{j-1}}^{w_j} w^{-\alpha-1} g_2(w) \, dw = \left(n^2/2\right)g(w_{j-2}) \int_{w_{j-1}}^{w_j} w^{-\alpha-1}(w - w_{j-1})(w - w_j) \, dw
\]

\[
- n^2 g(w_{j-1}) \int_{w_{j-1}}^{w_j} w^{-\alpha-1}(w - w_{j-2})(w - w_j) \, dw
\]

\[
+ \left(n^2/2\right)g(w_j) \int_{w_{j-1}}^{w_j} w^{-\alpha-1}(w - w_{j-2})(w - w_{j-1}) \, dw
\]

\[
= \frac{1}{(-\alpha)(-\alpha+1)(-\alpha+2)\alpha} \left(\frac{1}{2}E_j g(w_{j-2}) - F_j g(w_{j-1}) + \frac{1}{2}G_j g(w_j)\right),
\]

where

\[
E_j = 2j^{-\alpha+2} - (-\alpha + 2)j^{-\alpha+1} - 2(j - 1)^{-\alpha+2} - (-\alpha + 2)(j - 1)^{-\alpha+1},
\]

\[
F_j = 2j^{-\alpha+2} - 2(-\alpha + 2)j^{-\alpha+1} - 2(j - 1)^{-\alpha+2} + (-\alpha + 1)(-\alpha + 2)(j - 1)^{-\alpha},
\]

\[
G_j = 2j^{-\alpha+2} - 3(-\alpha + 2)j^{-\alpha+1} - 2(j - 1)^{-\alpha+2} + (-\alpha + 2)(j - 1)^{-\alpha+1} + 2(-\alpha + 1)(-\alpha + 2)j^{-\alpha}.
\]

Thus we get

\[
\int_0^1 w^{-\alpha-1} g_2(w) \, dw = \sum_{j=0}^{n} \alpha_j g(w_j),
\]

where

\[
(-\alpha)(-\alpha+1)(-\alpha+2)n^{-\alpha} \alpha_j = \begin{cases} 
2 - \frac{1}{2} \alpha + \frac{1}{2}E_2, & \text{for } j = 0, \\
(-\alpha)(3 - \alpha) + \frac{1}{2}E_3 - F_2, & \text{for } j = 1, \\
\frac{1}{2} \alpha + \frac{1}{2}E_4 - F_3 + \frac{1}{2}G_2, & \text{for } j = 2, \\
\frac{1}{2}E_{j+2} - F_{j+1} + \frac{1}{2}G_j, & \text{for } j = 3, \ldots, n-2, \\
-F_n + \frac{1}{2}G_{n-1}, & \text{for } j = n-1, \\
\frac{1}{2}G_n, & \text{for } j = n.
\end{cases}
\]
Therefore, we have
\[
R D_t^\alpha u(t_n) = \frac{t_n^{-\alpha}}{\Gamma(-\alpha)} \int_0^1 w^{-\alpha-1} g_2(w) dw + O(k^{3-\alpha})
\]
\[
= \frac{t_n^{-\alpha}}{\Gamma(-\alpha)} \sum_{j=0}^n \alpha_j g(w_j) + O(k^{3-\alpha})
\]
\[
= k^{-\alpha} \sum_{j=0}^n (-\alpha)(-\alpha + 1)(-\alpha + 2)n^{-\alpha} \alpha_j g(w_j) + O(k^{3-\alpha})
\]
\[
= k^{-\alpha} \sum_{j=0}^n w_{j,n} u(t_{n-j}) + O(k^{3-\alpha}),
\]
where, with \( j = 0, 1, 2, \ldots, n, n = 5, 6, \ldots, N, \)
\[
\Gamma(3-\alpha) w_{j,n} = (-\alpha)(-\alpha + 1)(-\alpha + 2)n^{-\alpha} \alpha_j n.
\]
More precisely, for \( j = 0, \) we have
\[
\Gamma(3-\alpha) w_{0,n} = 2 - \frac{\alpha}{2} + \frac{1}{2} E_2 = (1 + \frac{\alpha}{2})2^{1-\alpha}.
\]
For \( j = 1, \) we have
\[
\Gamma(3-\alpha) w_{1,n} = (-\alpha)(3-\alpha) + \frac{1}{2} E_3 - F_2 = (2 + \frac{\alpha}{2})3^{1-\alpha} - (3 + \frac{3}{2})2^{1-\alpha}.
\]
For \( j = 2, \) we have
\[
\Gamma(3-\alpha) w_{2,n} = \frac{1}{2} \alpha + \frac{1}{2} E_4 - F_3 + \frac{1}{2} G_2 = (3 + \frac{\alpha}{2})4^{1-\alpha} - (6 + \frac{3}{2})3^{1-\alpha} + (3 + \frac{3}{2})2^{1-\alpha}.
\]
For \( j = 3, 4, \ldots, n - 2, \)
\[
\Gamma(3-\alpha) w_{j,n} = \frac{1}{2} E_{j+2} - F_{j+1} + \frac{1}{2} G_j = (j + 2)^{2-\alpha} - (1 - \frac{\alpha}{2})(j + 2)^{1-\alpha}
\]
\[
- 3(j+1)^{2-\alpha} + (3 - \frac{3}{2}\alpha)(j+1)^{1-\alpha} + 3j^{2-\alpha} - (3 - \frac{3\alpha}{2})j^{1-\alpha}
\]
\[
- (j-1)^{2-\alpha} + (1 - \frac{\alpha}{2})(j-1)^{1-\alpha}.
\]
For \( j = n - 1, \) we have
\[
\Gamma(3-\alpha) w_{j,n} = (4 - 2n - 2\alpha)n^{1-\alpha} + (3n - 6 + \frac{3\alpha}{2})(n-1)^{1-\alpha} - (n + \frac{\alpha}{2} - 3)(n-2)^{1-\alpha},
\]
For \( j = n, \) we have
\[
\Gamma(3-\alpha) w_{n,n} = (n - 3 + \frac{3\alpha}{2})n^{1-\alpha} + (1 - \alpha)(2 - \alpha)n^{-\alpha} - (n - 2 + \frac{\alpha}{2})(n-1)^{1-\alpha}.
\]
Together, these estimates complete the proof of Lemma 3.2. \( \square \)
For any fixed $n \geq 5$, $w_{j,n}, j = 0, 1, \ldots, n - 2$ are independent on $n$ and only $w_{n-1,n}$ and $w_{n,n}$ depend on $n$. For example, we have $w_{0,n} = (1 + \frac{\alpha}{2})2^{1-\alpha}$ for any $n \geq 5$, $w_{1,n} = (2 + \frac{\alpha}{2})3^{1-\alpha} - (3 + \frac{3}{2}\alpha)2^{1-\alpha}$ for any $n \geq 5$, \ldots. Based on this observation, we may define $w_{j,n} = 0, 1, 2, \ldots, n - 1$ which we need in the numerical scheme (3.22)-(3.25) as follows. For $j = 0, 1, 2$, we define $w_{0,n} = w_{0,n}$, $w_{1,n} = w_{1,n}$, $w_{2,n} = w_{2,n}$ with $n \geq 5$. For $w_{j,n}, j = 3, 4, \ldots, n - 2$, we define $w_{j,n} = w_{j,n}, j = 3, 4, \ldots, n - 2$, where

\[
\Gamma(3 - \alpha)w_{j,n} = \frac{1}{2}E_{j+2} - F_{j+1} + \frac{1}{2}G_{j} = (j + 2)^{2-\alpha} - (1 - \frac{\alpha}{2})(j + 2)^{1-\alpha} - 3(j + 1)^{2-\alpha} + (3 - \frac{3}{2}\alpha)(j + 1)^{1-\alpha} + 3j^{2-\alpha} - (3 - \frac{3\alpha}{2})j^{1-\alpha} - (j - 1)^{2-\alpha} + (1 - \frac{\alpha}{2})(j - 1)^{1-\alpha}.
\]

For $w_{j,n} \geq n - 1$, we also define $w_{j}$ by

\[
\Gamma(3 - \alpha)w_{j} = \frac{1}{2}E_{j+2} - F_{j+1} + \frac{1}{2}G_{j} = (j + 2)^{2-\alpha} - (1 - \frac{\alpha}{2})(j + 2)^{1-\alpha} - 3(j + 1)^{2-\alpha} + (3 - \frac{3}{2}\alpha)(j + 1)^{1-\alpha} + 3j^{2-\alpha} - (3 - \frac{3\alpha}{2})j^{1-\alpha} - (j - 1)^{2-\alpha} + (1 - \frac{\alpha}{2})(j - 1)^{1-\alpha}.
\]

Now for any fixed $n \geq 5$, we have defined all $w_{j,n}, j = 0, 1, 2, \ldots, n - 2, n - 1, n$, \ldots. We remark that in our numerical method (3.22)-(3.25), we only need the weights $w_{j,n} = 0, 1, 2, \ldots, n - 1$, where $w_{j} = w_{j,n}, j = 0, 1, 2, \ldots, n - 2$, but $w_{n-1} \neq w_{n-1,n}$. We shall see that in the proof of Theorem 3.3, we need to define all $w_{j,n}, j = 0, 1, 2, \ldots, n - 2, n - 1, n$, \ldots in order to use the discrete Laplace transform. For $w_{j,n} = 0, 1, 2$, \ldots defined above, we may introduce the characteristic function $\delta(\zeta)$

\[
\delta(\zeta)^{\alpha} = \sum_{j=0}^{\infty} w_{j} \zeta^{j}. \quad (3.21)
\]

We now introduce the following corrected higher order time discretisation scheme for solving (2.1), with $c_{0} = 11/12$, $c_{1} = -5/12$.
\[ \tau^{-\alpha} \sum_{j=1}^{n} w_{n-j} V^j + AV^n = (-Au_0)(1 + c_0), \quad n = 1, \quad (3.22) \]
\[ \tau^{-\alpha} \sum_{j=1}^{n} w_{n-j} V^j + AV^n = (-Au_0)(1 + c_1), \quad n = 2, \quad (3.23) \]
\[ \tau^{-\alpha} \sum_{j=1}^{n} w_{n-j} V^j + AV^n = -Au_0, \quad n = 3, 4, \ldots, N, \quad (3.24) \]
\[ V^0 = 0, \quad (3.25) \]

where \( w_j, j = 0, 1, 2, \ldots \) are determined by (3.21) below.

**Remark 3.2.** Note that the weights \( w_j, j = 0, 1, 2, \ldots \) in (3.22)-(3.25) are deduced based on Lemma 3.2 for the approximation of Riemann-Liouville fractional derivative \( R_0 D_t^\alpha u(t_n) \) with \( n \geq 5 \). Therefore one needs to choose at least \( N \geq 5 \) for the numerical method (3.22)-(3.25).

**Theorem 3.3.** ([50]) Let \( c_0 = 11/12 \), \( c_1 = -5/12 \) and let \( V(t_n) \) and \( V^n \) be the solutions of (3.1) and (3.22)-(3.25), respectively. Let \( u_0 \in L_2(\Omega) \). Then there exists a positive constant \( C \) such that
\[ \| V^n - V(t_n) \| \leq C k^{3-\alpha} t_n^{\alpha-3} \| u_0 \|, \quad n \geq 5. \]

4. Correction of the discontinuous Galerkin method in McLean and Mustapha [35]

In this section, we will consider the discontinuous Galerkin method for solving (1.1) introduced by McLean and Mustapha [35]. We first obtain the same time discretisation scheme as in McLean and Mustapha [35] by approximating the Riemann-Liouville fractional derivative with the L1 method (or Diethelm’s finite difference method). We then introduce a higher order numerical method for solving (1.1) based on this scheme.

4.1. The discontinuous Galerkin scheme in McLean and Mustapha [35]. In this subsection, we obtain the same time discretisation scheme as in McLean and Mustapha [35] by approximating the Riemann-Liouville fractional derivative by using the L1 method. We first write (1.1) as the following equivalent form
\[ u_t + \frac{R}{0} D_t^{1-\alpha} Au = 0, \quad 0 < t \leq T, \quad \text{with} \quad u(0) = u_0, \quad (4.1) \]
where \( u_t \) denotes the time derivative and \( \frac{R}{0} D_t^\alpha u(t) \) denotes the Riemann-Liouville fractional derivative.
AN APPROACH TO CONSTRUCT HIGHER ORDER ...

At \( t = t_n \), we approximate the time derivative by using the backward Euler method
\[
    u_t(t_n) = \frac{(u(t_n) - u(t_{n-1}))}{\tau} + O(\tau), \quad \text{as } \tau \to 0.
\]

To approximate the Riemann-Liouville fractional derivative \( \frac{R}{0}D_{1-\alpha}^{1-\alpha}Au(t_n) \), we shall use the following Diethelm finite difference method \([9]\), with \( u \in C^2[0,T;D(A)] \)
\[
    \frac{R}{0}D_{1-\alpha}^{1-\alpha}Au(t_n) = \tau^{\alpha-1} \sum_{j=0}^{n} w_{n-j}Au(t_j) + O(\tau^{1+\alpha}), \quad \text{as } \tau \to 0, \quad (4.2)
\]
where \( w_j, j = 0, 1, 2, \ldots, n - 1 \) are given again by \((3.7)\) and \( w_n \) satisfies
\[
    \Gamma(1 + \alpha)w_n = (n - 1)^\alpha - n^\alpha + \alpha n^{\alpha-1}. \quad (4.3)
\]

With \( U^n \approx u(t_n) \), we define the following time discretisation problem for solving \((4.1)\)
\[
    U^n - U^{n-1} + \tau^\alpha \sum_{j=0}^{n} w_{n-j}AU^j = 0, \quad n \geq 1, \quad \text{with } U^0 = u_0, \quad (4.4)
\]
where \( w_j, j = 0, 1, 2, \ldots, n - 1 \) are given by \((3.7)\) and \( w_n \) is corrected as
\[
    \Gamma(1 + \alpha)w_n = -2n^\alpha + (n - 1)^\alpha + (n + 1)^\alpha. \quad (4.5)
\]

Now all \( w_j, j = 0, 1, 2, \ldots, n \) are defined by \((3.7)\). The reason for correcting \( w_n \) is that we shall use the discrete Laplace transform for \((w_0, w_1, w_2, \ldots, w_n, \ldots)\) in the proof of the error estimates.

**Theorem 4.1.** \((53)\) Let \( u(t_n) \) and \( U^n \) be the solutions of \((4.1)\) and \((4.4)\), respectively. Let \( u_0 \in L_2(\Omega) \). Then we have, with \( 0 < \alpha < 1 \),
\[
    ||u(t_n) - U^n|| \leq C(\tau^\alpha t_n^{-\alpha} + \tau t_n^{-1})||u_0||. \quad (4.6)
\]

We note that the convergence rate of the time discretisation scheme \((4.4)\) is only \( O(\tau^\alpha) \), \( 0 < \alpha < 1 \) for smooth and nonsmooth initial data. To derive a time discretisation method for solving \((4.1)\) with the convergence rate \( O(\tau) \) for smooth and nonsmooth initial data \( u_0 \), we will approximate \( \frac{R}{0}D_{1-\alpha}^{1-\alpha}Au(t_n) \) by
\[
    \frac{R}{0}D_{1-\alpha}^{1-\alpha}Au(t_n) \approx \tau^{\alpha-1} \sum_{j=1}^{n} w_{n-j}Au(t_j),
\]
where we ignore the term \( Au(t_0) \) in \((4.2)\). More precisely, we choose \( w_n = 0 \) in the summation \( \sum_{j=0}^{n} w_{n-j}Au(t_j) \) in \((4.2)\). It is easy to show that
\[
    \frac{R}{0}D_{1-\alpha}^{1-\alpha}Au(t_n) = \tau^{\alpha-1} \sum_{j=1}^{n} w_{n-j}Au(t_j) + O(\tau), \quad \text{as } \tau \to 0. \quad (4.7)
\]
To see this, by (4.2), it suffices to show that, for the fixed \( t_n = n\tau \),
\[
\tau^{\alpha-1}w_n = O(\tau), \quad \text{as} \quad \tau \to 0.
\]
(4.8)

In fact, let \( t_n \) be fixed, for example, assume that \( t_n = 1, n = 1/\tau \), we have
\[
\Gamma(1+\alpha)\tau^{\alpha-1}w_n = \tau^{\alpha-1}(\alpha n^{\alpha-1} + (n-1)^{\alpha} - n^{\alpha})
\]
\[
= \alpha t_n^{\alpha-1} + (n-1)^{\alpha}\tau^{\alpha-1} - n^{\alpha}\tau^{\alpha-1} = t_n^{\alpha-1}\left(\alpha + \frac{(n-1)^{\alpha} - n^{\alpha}}{n^{\alpha-1}}\right)
\]
\[
= t_n^{\alpha-1}\left(\alpha + \frac{(1-\tau\alpha + O(\tau^2) - 1)}{\tau}\right) = t_n^{\alpha-1}O(\tau) = O(\tau), \quad \text{as} \quad \tau \to 0,
\]
which implies (4.8) and therefore (4.7) follows.

Now we may define the time discretisation scheme for solving (4.1), with
\[
U^0 = u_0,
\]
(4.9)

where \( w_j, j = 0, 1, 2, \ldots, n-1, n \geq 1 \) are given by
\[
\Gamma(1+\alpha)w_j = \begin{cases} 1, & \text{for } j = 0, \\ -2j^\alpha + (j-1)^\alpha + (j+1)^\alpha, & \text{for } j = 1, 2, \ldots, n-1. \end{cases}
\]
(4.10)

The time stepping method (4.9) was first introduced in McLean and Mustapha [35] for solving (4.1) by using the piecewise discontinuous Galerkin method, see other related works for the discontinuous Galerkin method [8], [37], [38], [40].

**Theorem 4.2.** ([35]) Let \( u(t_n) \) and \( U^n \) be the solutions of (4.1) and (4.9), respectively. Let \( u_0 \in L_2(\Omega) \). Then we have, with \( 0 < \alpha < 1 \),
\[
\|u(t_n) - U^n\| \leq C\tau t_n^{-1}\|u_0\|. \quad (4.11)
\]

4.2. **Correction of the discontinuous Galerkin method of McLean and Mustapha [35].** In this subsection, we shall introduce a modified piecewise constant discontinuous Galerkin method for solving (4.1). We show that this method has the convergence rate \( O(\tau^{1+\alpha}) \) for smooth and nonsmooth initial data \( u_0 \).

Following the idea in Lubich et al. [33], we approximate the time derivative \( u(t_n) \) by using a second order backward difference method
\[
u(t_n) = \frac{3}{2}u(t_n) - 2u(t_{n-1}) + \frac{1}{2}u(t_{n-2}) + O(\tau^2), \quad \text{as} \quad \tau \to 0.
\]
We define the following finite difference method for solving (4.1), with $U^n \approx u(t_n)$ and $c_0 = 1/2$, 

\[ \bar{D}U^n + \tau^{-1} \left( \sum_{j=1}^{n} w_{n-j} AU^j + c_0 w_{n-1} Au_0 \right) = 0, \quad n \geq 2, \quad (4.12) \]

\[ \bar{D}U^n + \tau^{-1} \left( \sum_{j=1}^{n} w_{n-j} AU^j + c_0 w_{n-1} Au_0 \right) = 0, \quad n = 1, \quad (4.13) \]

\[ U^0 = u_0, \quad U^{-1} = u_0, \quad (4.14) \]

where

\[ \bar{D}U^n = \frac{3}{2}U^n - 2U^{n-1} + \frac{1}{2}U^{n-2}, \quad n \geq 1, \]

and $w_j, j = 0, 1, 2, \ldots, n-1$ are defined by (4.10). Here we use a modification term $c_0 w_{n-1} Au_0$ as in Lubich et al. [33, (1.18)].

**Theorem 4.3.** ([53]) Let $u(t_n)$ and $U^n$ be the solutions of (4.1) and (4.12)-(4.14), respectively. Let $u_0 \in L^2(\Omega)$. Then we have, with $0 < \alpha < 1$,

\[ \|u(t_n) - U^n\| \leq C \tau^{1+\alpha} t_n^{-1-\alpha}\|u_0\|, \quad n \geq 1. \quad (4.15) \]

**4.3. Numerical examples.** In this subsection, we will consider the experimentally determined convergence rates of the different numerical methods discussed in Section 4 for solving (4.1).

Let us consider the following time fractional partial differential equation in the one dimensional case:

\[ C_0 D_t^\alpha u(x,t) - u_{xx} = 0, \quad 0 < x < 1, \quad 0 < t \leq T, \]

\[ u(0,t) = u(1,t) = 0, \]

\[ u(x,0) = u_0(x). \]

Let $0 < t_0 < t_1 < \cdots < t_N = T$ be the time partition of $[0,T]$ with $T = 1$ and $\tau$ the time step size. Let $N_h$ be a positive integer. Let $0 = x_0 < x_1 < x_2 < \cdots < x_{N_h} = 1$ be the space partition and $h$ the space step size. The space is discretised by using the standard linear finite element method.

We first consider the scheme (4.4): the convergence rate was proved to be $O(\tau^\alpha)$ for both smooth and nonsmooth data in Theorem 4.1. To observe this convergence rate, we first calculate the reference solution $u_{ref}(t)$ at $T = 1$ with $h_{ref} = 2^{-6}$ and $\tau_{ref} = 2^{-10}$. We then use $h = 2^{-6}$ and $\tau = \kappa \tau_{ref}$ with $\kappa = [2^2, 2^3, 2^4, 2^5, 2^6]$ to obtain the approximate solution at $u(T)$ with $T = 1$. Let $e_\tau$ denote the error of $u(T)$ at $T = 1$.
with the time step size $\tau$ and the fixed space step size $h = 2^{-6}$. By Theorem 4.1, we have $\|e_{\tau}\| \leq C\tau^{\alpha}$. Thus the convergence rate $\alpha$ is determined experimentally by $\alpha \approx \log_2 \left( \frac{\|e_{2\tau}\|}{\|e_{\tau}\|} \right)$.

Choosing the smooth and nonsmooth initial data (a) $u_0 = \sin(\pi x)$ and (b) $u_0 = \chi_{[0,1/2]}$, we observe, in Table 5, that the experimentally determined convergence rate is indeed almost $O(k^{\alpha})$ for the different $\alpha \in (0,1)$ in both smooth and nonsmooth data cases.

We next consider the numerical method (4.9) proposed by McLean and Mustapha [35] which has the convergence rate $O(\tau)$ for both smooth and nonsmooth data. Using the same notations and the same initial data as in Table 5, we found, in Table 6, that the experimentally determined convergence rate of this method is indeed approximately 1.

Finally we consider the improved numerical method (3.22)-(3.24) which has the convergence rate $O(\tau^{1+\alpha})$ for both smooth and nonsmooth data. Using the same notations and the same initial data as in Tables 5 and 6, we found, in Table 7, that the experimentally determined convergence rate is
approximately $\tau^{1+\alpha}$ (actually the experimentally determined convergence rate is better than $1 + \alpha$) as we expected.

5. **Open problems and issues for further work**

In this paper, we have reviewed an approach for constructing the higher order time discretisation schemes for solving (1.1). We propose some open problems and issues for further works.

- To use the starting weights to construct higher order time discretisation schemes for solving (1.1), one needs to assume that the solution $u(t, x)$ has the series expansion form (1.10) which implies that the source term $f$ should satisfy some compatibility conditions. We may need to ask: which types of compatibility conditions should $f$ satisfy? For which types of $f$ in (1.1) may we use the starting weights techniques to improve the convergence rates of the numerical methods for solving (1.1)?

- To approximate the Hadamard finite-part integral with piecewise $p$-th order polynomials with $p \geq 3$ to obtain the approximate scheme to the Riemann-Liouville fractional derivative with the convergence
order $p+1-\alpha$. Can one use such a scheme to construct the corrected higher order time discretisation method for solving (1.1)?

- To construct the corrected higher order numerical methods for solving (1.1) by using the higher order approximate schemes to the Riemann-Liouville fractional derivatives developed in Li et al. [2], [25].
- To construct the corrected higher order time discretisation method for solving
  \[
  u(t) - u(0) + \frac{R}{0} D_t^{-\alpha} (Au(t)) = 0, \tag{5.1}
  \]
  by approximating the Riemann-Liouville integral $\frac{R}{0} D_t^{-\alpha} (Au(t))$ with the Lubich convolution quadratures discussed in Jin et al. [23].

References


AN APPROACH TO CONSTRUCT HIGHER ORDER ... 1105


---

Department of Mathematics, University of Chester
Parkgate Road, Chester – CH1 4BJ, UK

1 e-mail: njford@chester.ac.uk
Received: May 24, 2017

2 e-mail: y.yan@chester.ac.uk
Revised: August 31, 2017

Please cite to this paper as published in:

DOI: 10.1515/fca-2017-0058