Applied Mathematics and Computation 489 (2025) 129164

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Full Length Article

A shooting-Newton procedure for solving fractional terminal value problems

Luigi Brugnano [a,*], Gianmarco Gurioli [a], Felice Iavernaro [b]

[a] Dipartimento di Matematica e Informatica "U. Dini", Università di Firenze, Italy
[b] Dipartimento di Matematica, Università di Bari Aldo Moro, Italy

ARTICLE INFO

MSC:
34A08
65R20

Keywords:
Fractional differential equations
Fractional integrals
Terminal value problems
Jacobi polynomials
Fractional Hamiltonian Boundary Value Methods
FHBVMs

ABSTRACT

In this paper we consider the numerical solution of *fractional terminal value problems*: namely, *terminal value problems for fractional differential equations*. In particular, the proposed method uses a Newton-type iteration which is particularly efficient when coupled with a recently-introduced step-by-step procedure for solving *fractional initial value problems*, i.e., *initial value problems for fractional differential equations*. As a result, the method is able to produce spectrally accurate solutions of fractional terminal value problems. Some numerical tests are reported to make evidence of its effectiveness.

1. Introduction

Fractional differential equations have gained more and more importance in many applications: we refer, e.g., to the classical references [14,28] for an introduction.

The present contribution is addressed for solving *fractional terminal value problems* namely, *terminal value problems for fractional differential equations* (in short, *FDE-TVPs*) in the form

$$y^{(\alpha)}(t) = f(y(t)), \qquad t \in [0,T], \qquad y(T) = \eta \in \mathbb{R}^m, \tag{1}$$

where, for the sake of brevity, we have omitted the argument t for f. Here, for $\alpha \in (0,1)$, $y^{(\alpha)}(t) \equiv D^\alpha y(t)$ is the Caputo fractional derivative:

$$D^\alpha g(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-x)^{-\alpha} \left[\frac{\mathrm{d}}{\mathrm{d}x} g(x)\right] \mathrm{d}x. \tag{2}$$

The Riemann-Liouville integral associated to (2) is given by:

* Corresponding author.
E-mail addresses: luigi.brugnano@unifi.it (L. Brugnano), gianmarco.gurioli@unifi.it (G. Gurioli), felice.iavernaro@uniba.it (F. Iavernaro).

https://doi.org/10.1016/j.amc.2024.129164
Received 5 September 2024; Received in revised form 24 October 2024; Accepted 27 October 2024

Available online 4 November 2024

$$I^{\alpha} g(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-x)^{\alpha-1} g(x) \mathrm{d}x. \tag{3}$$

Usually, one solves *fractional initial value problems*, that is, *initial value problems for fractional differential equations* (in short, *FDE-IVPs*) (see, e.g. [16,21,22,26,27,30]):

$$y^{(\alpha)}(t) = f(y(t)), \qquad t \in [0,T], \qquad y(0) = \rho_0 \in \mathbb{R}^m, \tag{4}$$

whose solution is, under suitable assumptions on f,

$$y(t) = \rho_0 + I^{\alpha} f(y(t)) \equiv \rho_0 + \frac{1}{\Gamma(\alpha)} \int_0^t (t-x)^{\alpha-1} f(y(x)) \mathrm{d}x, \qquad t \in [0,T]. \tag{5}$$

However, under suitable hypothesis on f and T, also the FDE-TVP is well-posed (see, e.g., [18] for the scalar case, and [29]). Consequently, its numerical solution has been considered by many authors (see, e.g., [17,18,23–25,29]). In particular, the scalar case of (1) ($m = 1$) allows using a shooting procedure coupled with the bisection method [15,19] or, more recently, with the secant method [17].

However, as is clear by their very definition, both the above procedures *cannot* be applied for solving vector problems. Motivated by this drawback, in this paper, we propose an alternative approach, based on a straight Newton procedure, able to handle vector problems as well.

The procedure takes advantage of a recently introduced method for solving FDE-IVPs, able to obtain *spectrally accurate* approximations [6,8]. This latter approach has been derived as an extension of Hamiltonian Boundary Value Methods (HBVMs), special low-rank Runge-Kutta methods originally devised for Hamiltonian problems (see, e.g., [9,10]), and later extended along several directions (see, e.g., [1,3,5–7,11]), including the numerical solution of FDEs. A main feature of HBVMs is the fact that they can gain spectrally accuracy, when approximating ODE-IVPs [2,12,13], and such a feature has been recently extended to the FDE case [6,8].

With this premise, the structure of the paper is as follows: in Section 2 we sketch the shooting-Newton procedure for solving (1), along with a corresponding simplified variant; in Section 3 we recall the main facts about the (possibly spectrally accurate) numerical solution of FDE-IVPs recently proposed in [6], with the extension for the shooting-Newton procedure; in Section 4 we report a few numerical tests, including the case of vector problems; at last, a few conclusions are given in Section 5.

2. The shooting-Newton procedure

To begin with, let us introduce a perturbation result concerning the solution of the FDE-IVP (4). In particular, let us denote by $y(t,\rho_0)$ the solution of this problem, in order to emphasize its dependence from the initial condition. The following result holds true.

Theorem 1. *For $t \in [0,T]$, one has:*

$$\frac{\partial}{\partial \rho_0} y(t,\rho_0) = \Phi(t,\rho_0), \tag{6}$$

which is the solution of the fractional variational problem[1]

$$\Phi^{(\alpha)}(t,\rho_0) = f'(y(t,\rho_0))\Phi(t,\rho_0), \qquad t \in [0,T], \qquad \Phi(0,\rho_0) = I, \tag{7}$$

explicitly given by:

$$\Phi(t,\rho_0) = I + \frac{1}{\Gamma(\alpha)} \int_0^t (t-x)^{\alpha-1} f'(y(x,\rho_0))\Phi(x,\rho_0) \mathrm{d}x. \tag{8}$$

Proof. In fact, from (2) and (4), one has:

$$D^{\alpha} \frac{\partial}{\partial \rho_0} y(t,\rho_0) = \frac{\partial}{\partial \rho_0} D^{\alpha} y(t) = \frac{\partial}{\partial \rho_0} f(y(t,\rho_0)) = f'(y(t,\rho_0)) \frac{\partial}{\partial \rho_0} y(t,\rho_0),$$

and

$$\frac{\partial}{\partial \rho_0} y(t,\rho_0)\Big|_{t=0} = \frac{\partial}{\partial \rho_0} \rho_0 = I.$$

Consequently, (6)-(7) follows and, therefore, also (8) follows from (3) and (5). □

[1] As is usual, $f'(y)$ denotes the Jacobian matrix of $f(y)$.

Table 1
Algorithm 1 – the shooting-Newton procedure.

fix ρ_0
for $\ell = 0, 1, \ldots$
solve: $y^{(\alpha)}(t, \rho_\ell) = f(y(t, \rho_\ell))$, $t \in [0, T]$, $y(0, \rho_\ell) = \rho_\ell$
and $\Phi^{(\alpha)}(t, \rho_\ell) = f'(y(t, \rho_\ell))\Phi(t, \rho_\ell)$, $t \in [0, T]$, $\Phi(0, \rho_\ell) = I$
set: $\rho_{\ell+1} = \rho_\ell - \Phi(T, \rho_\ell)^{-1}\left[y(T, \rho_\ell) - \eta\right]$
end

Remark 1. Hereafter, in order to guarantee the well-posedness of problem (1), if $y(0) = \rho^*$ is the initial value of (4) fulfilling the FDE-TVP, i.e.,

$$y(T, \rho^*) = \eta, \tag{9}$$

we shall assume that (see (6))

$$\det\left(\Phi(T, \rho^*)\right) \neq 0. \tag{10}$$

Assuming that f is continuously differentiable in a neighborhood of the solution, in turn (10) implies that

$$\exists\, \delta > 0 \ s.t. \ \|\rho^* - \rho\| \leq \delta \ \Rightarrow \ \det\left(\Phi(T, \rho)\right) \neq 0. \tag{11}$$

The previous results allow us stating the shooting-Newton procedure for solving (1) sketched in Table 1, where a suitable stopping criterion has to be adopted. Moreover, the starting approximation ρ_0 for the shooting-Newton iteration has to be chosen in some way, possibly exploiting any additional information; conversely, the choice $\rho_0 = \eta$ (i.e., the final value in (1)) can be considered, as proposed in [17].

Remark 2. Though the procedure described in Algorithm 1 appears to be easily derived, at the best of our knowledge, it has not yet been considered for solving FDE-TVPs, so far. Moreover, the use of the variational problem, involved in its implementation and described in the next section, is novel as well.

The following straightforward convergence result holds true.

Theorem 2. *Assume that, in a neighborhood of the solution $\xi = \rho^*$:*

(i) *f is continuously differentiable,*
(ii) *$\Phi(T, \xi)^{-1}$ is differentiable.*

Then, the shooting-Newton procedure given in Table 1 converges in a suitable neighborhood of ρ^*.

Proof. In fact, from (9) it follows that ρ^* is a fixed-point of the corresponding iteration function,

$$\Psi(\xi) := \xi - \Phi(T, \xi)^{-1}\left[y(T, \xi) - \eta\right],$$

whose Jacobian (recall (6)) vanishes at $\xi = \rho^*$. Consequently, from the Perron Theorem [31, Corollary 4.7.2], exponential convergence is granted, in a suitable neighborhood of ρ^*. □

Further, if convergent, the procedure converges quadratically. However, to prove this statement, we need to recall some well-known results about the Taylor theorem. In more detail, with reference to (6), assume that $\Phi(T, \xi)$ is continuously differentiable in a suitable neighborhood of the solution. Then, by setting y_i the i-th entry of y, for a given ρ suitably close to ρ_ℓ there exists $\theta_i \in [0, 1]$ such that:

$$\begin{aligned} y_i(T, \rho) = y_i(T, \rho_\ell) &+ \frac{\partial}{\partial \xi} y_i(T, \xi)\Big|_{\xi = \rho_\ell} (\rho - \rho_\ell) \\ &+ \frac{1}{2}(\rho - \rho_\ell)^\top \frac{\partial^2}{\partial \xi^2} y_i(T, \xi)\Big|_{\xi = \rho_\ell + \theta_i(\rho - \rho_\ell)} (\rho - \rho_\ell), \qquad i = 1, \ldots, m, \end{aligned}$$

with $\frac{\partial^2}{\partial \xi^2} y_i(T, \xi)$ the Hessian matrix of $y_i(T, \xi)$. The previous relations can be written in vector form as follows:

$$y(T, \rho) = y(T, \rho_\ell) + \Phi(T, \rho_\ell)(\rho - \rho_\ell) + \frac{1}{2}\Phi'(T, \Sigma_\ell(\rho))\left((\rho - \rho_\ell), (\rho - \rho_\ell)\right),$$

with

$$\Sigma_\ell(\rho) = \left(\rho_\ell + \theta_1(\rho - \rho_\ell), \quad \ldots, \quad \rho_\ell + \theta_m(\rho - \rho_\ell) \right) \in \mathbb{R}^{m\times m},$$

and $\Phi'(T, \Sigma_\ell(\rho))$ denoting the derivative of Φ, whose i-th "slice" is evaluated in the i-th column of $\Sigma_\ell(\rho)$. With this premise, we can now state the following result.

Theorem 3. *Assume that, in a neighborhood of the solution $\xi = \rho^*$, $\Phi(T, \xi)$ is continuously differentiable. Then, if convergent, the shooting-Newton procedure given in Table 1 converges quadratically.*

Proof. By using the notation about the Taylor theorem exposed before, one has:

$$\begin{aligned} 0 &= y(T, \rho^*) - \eta \\ &= y(T, \rho_\ell) - \eta + \Phi(T, \rho_\ell)\left(\rho^* - \rho_\ell\right) + \frac{1}{2}\Phi'(T, \Sigma_\ell(\rho^*))\left((\rho^* - \rho_\ell), (\rho^* - \rho_\ell)\right). \end{aligned}$$

Consequently, considering that[2]

$$\rho_{\ell+1} = \rho_\ell - \Phi(T, \rho_\ell)^{-1}[y(T, \rho_\ell) - \eta],$$

and setting $e_\ell = \rho^* - \rho_\ell$ the error at step ℓ, one derives:

$$e_{\ell+1} = -\frac{1}{2}\Phi(T, \rho_\ell)^{-1}\Phi'(T, \Sigma_\ell(\rho^*))\left(e_\ell, e_\ell\right).$$

Passing to norms, one eventually obtains

$$\frac{\|e_{\ell+1}\|}{\|e_\ell\|^2} \le \frac{1}{2}\|\Phi(T, \rho_\ell)^{-1}\|\,\|\Phi'(T, \Sigma_\ell(\rho^*))\|.$$

Consequently,

$$\lim_{\ell\to\infty} \frac{\|e_{\ell+1}\|}{\|e_\ell\|^2} \le \frac{1}{2}\|\Phi(T, \rho^*)^{-1}\|\,\|\Phi'(T, \Sigma^*)\|,$$

where Σ^* now denotes the matrix with all the columns equal to ρ^*. □

An interesting additional feature is given by the following result.

Theorem 4. *For problems in the form*

$$y^{(\alpha)} = A(t)y + b(t), \qquad t \in [0, T], \qquad y(T) = \eta \in \mathbb{R}^m, \tag{12}$$

with $A(t)$ and $b(t)$ continuous functions, the algorithm described in Table 1 converges in exactly one iteration.

Proof. In fact, in such a case, the variational problem (7) simplifies to

$$\Phi^{(\alpha)}(t) = A(t)\Phi(t), \qquad t \in [0, T], \qquad \Phi(0) = I,$$

i.e., $\Phi(t)$ does not depend on the initial condition. Further, by using the same notation as above,

$$\left[y(t, \rho_0) - y(t, \rho^*)\right]^{(\alpha)} = A(t)\left[y(t, \rho_0) - y(t, \rho^*)\right], \qquad t \in [0, T],$$

whose solution is given by

$$\left[y(t, \rho_0) - y(t, \rho^*)\right] = \Phi(t)\left[\rho_0 - \rho^*\right], \qquad t \in [0, T].$$

Consequently, at $t = T$ one has:

$$\left[y(T, \rho_0) - \eta\right] = \Phi(T)\left[\rho_0 - \rho^*\right].$$

That is,[3]

$$\rho^* = \rho_0 - \Phi(T)^{-1}\left[y(T, \rho_0) - \eta\right],$$

so that convergence is gained in exactly one iteration, since the r.h.s. amounts to the very first iteration of Algorithm 1 used for solving (12). □

[2] We recall that (11) holds true.

[3] We recall that the assumption $\det(\Phi(T)) \neq 0$ must clearly hold.

Table 2
Algorithm 2 – the simplified shooting-Newton procedure.

fix ρ_0
for $\ell = 0, 1, \dots$
solve: $y^{(\alpha)}(t, \rho_\ell) = f(y(t, \rho_\ell))$, $t \in [0, T]$, $y(0, \rho_\ell) = \rho_\ell$
and compute $\hat{\Phi}(T, \rho_\ell) \approx \Phi(T, \rho_\ell)$
set: $\rho_{\ell+1} = \rho_\ell - \hat{\Phi}(T, \rho_\ell)^{-1} \left[y(T, \rho_\ell) - \eta \right]$
end

2.1. A simplified Newton-iteration

As is well-known, sometimes it can be computationally convenient to implement a *simplified Newton iteration*, instead of the basic one. This involves using a simplified version of the algorithm shown in Table 1: it is sketched in Table 2.

For this simplified shooting-Newton procedure, the following result holds true, the proof being similar to that of Theorem 2.

Theorem 5. *Assume the hypotheses of Theorem 2 hold true and that $\hat{\Phi}(T, \xi)$ is continuously invertible in a neighborhood of $\xi = \rho^*$. Further, assume that the spectral radius of the matrix*

$$I - \hat{\Phi}(T, \rho^*)^{-1} \Phi(T, \rho^*)$$

is less than 1. Then, the algorithm described in Table 2 converges in a suitable neighborhood of the solution ρ^*.

Remark 3. However, as one may expect, in this case the results of Theorems 3 and 4 does not hold, in general. As matter of fact, only a linear convergence can be granted and, for linear problems, convergence in one iteration cannot be expected, in general.

3. Implementing the algorithm

Following the approach in [6], let us now explain the way we compute $y(T, \rho_\ell)$ in Algorithms 1 and 2, and $\Phi(T, \rho_\ell)$ in Algorithm 1.[4] From (5) and (8), we have to compute:

$$y(T, \rho_\ell) \;=\; \rho_\ell + \frac{1}{\Gamma(\alpha)} \int_0^T (T - x)^{\alpha - 1} f(y(x, \rho_\ell)) \mathrm{d}x, \tag{13}$$

and

$$\Phi(T, \rho_\ell) \;=\; I + \frac{1}{\Gamma(\alpha)} \int_0^T (T - x)^{\alpha - 1} f'(y(x, \rho_\ell)) \Phi(x, \rho_\ell) \mathrm{d}x. \tag{14}$$

To begin with, in order to obtain a piecewise approximation to the solution of the two problems, we consider a partition of the integration interval in the form:

$$t_n = t_{n-1} + h_n, \qquad n = 1, \dots, N, \tag{15}$$

where

$$t_0 = 0, \qquad t_N = T \equiv \sum_{n=1}^{N} h_n. \tag{16}$$

In general [6], for coping with possible singularities in the derivative of the vector field at the origin, we shall consider the following *graded mesh*,

$$h_n = r^{n-1} h_1, \qquad n = 1 \dots, N, \tag{17}$$

where $r > 1$ and $h_1 > 0$ satisfy, by virtue of (16)-(17),

$$h_1 \frac{r^N - 1}{r - 1} = T. \tag{18}$$

Clearly, when a uniform mesh is considered then, in (17), $r = 1$ and $h_1 = h := T/N$, so that $h_n = h$, $n = 1, \dots, N$.

[4] On the contrary, $\hat{\Phi}(T, \rho_\ell)$ in Algorithm 2 is strictly problem dependent and its computation cannot be stated in general: however, a relevant specific case will be considered in Section 3.3.

By setting

$$y_n(ch_n, \rho_\ell) := y(t_{n-1} + ch_n, \rho_\ell), \qquad c \in [0,1], \qquad n = 1, \dots, N, \tag{19}$$

the restriction of the solution of (13) on the interval $[t_{n-1}, t_n]$, and taking into account (15)–(17), one then obtains:

$$\begin{aligned} y(T, \rho_\ell) \equiv y_N(h_N, \rho_\ell) &= \rho_\ell + \frac{1}{\Gamma(\alpha)} \int_0^T (T-x)^{\alpha-1} f(y(x, \rho_\ell))\mathrm{d}x \\ &= \rho_\ell + \frac{1}{\Gamma(\alpha)} \sum_{n=1}^N \int_{t_{n-1}}^{t_n} (t_N - x)^{\alpha-1} f(y(x, \rho_\ell))\mathrm{d}x \\ &= \rho_\ell + \frac{1}{\Gamma(\alpha)} \sum_{n=1}^N \int_0^{h_n} (t_N - t_{n-1} - x)^{\alpha-1} f(y_n(x, \rho_\ell))\mathrm{d}x \\ &= \rho_\ell + \frac{1}{\Gamma(\alpha)} \sum_{n=1}^N h_n^\alpha \int_0^1 \left(\frac{r^{N-n+1} - 1}{r-1} - \tau \right)^{\alpha-1} f(y_n(\tau h_n, \rho_\ell))\mathrm{d}\tau. \end{aligned} \tag{20}$$

In case of a constant stepsize $h = T/N$ is used, the previous equation becomes:

$$\begin{aligned} y(T, \rho_\ell) &\equiv y_N(h, \rho_\ell) \\ &= \rho_\ell + \frac{h^\alpha}{\Gamma(\alpha)} \sum_{n=1}^N \int_0^1 (N - n + 1 - \tau)^{\alpha-1} f(y_n(\tau h, \rho_\ell))\mathrm{d}\tau. \end{aligned} \tag{21}$$

Similarly, for (14), by setting

$$\Phi_n(ch_n, \rho_\ell) := \Phi(t_{n-1} + ch_n, \rho_\ell), \qquad c \in [0,1], \qquad n = 1, \dots, N, \tag{22}$$

the restriction of the solution on the interval $[t_{n-1}, t_n]$, again by virtue of (15)–(17), one obtains:

$$\begin{aligned} \Phi(T, \rho_\ell) &\equiv \Phi_N(h_N, \rho_\ell) \\ &= I + \frac{1}{\Gamma(\alpha)} \int_0^T (T-x)^{\alpha-1} f'(y(x, \rho_\ell))\Phi(x, \rho_\ell)\mathrm{d}x \\ &= I + \frac{1}{\Gamma(\alpha)} \sum_{n=1}^N \int_{t_{n-1}}^{t_n} (t_N - x)^{\alpha-1} f'(y(x, \rho_\ell))\Phi(x, \rho_\ell)\mathrm{d}x \\ &= I + \frac{1}{\Gamma(\alpha)} \sum_{n=1}^N \int_0^{h_n} (t_N - t_{n-1} - x)^{\alpha-1} f'(y_n(x, \rho_\ell))\Phi_n(x, \rho_\ell)\mathrm{d}x \\ &= I + \frac{1}{\Gamma(\alpha)} \sum_{n=1}^N h_n^\alpha \int_0^1 \left(\frac{r^{N-n+1} - 1}{r-1} - \tau \right)^{\alpha-1} f'(y_n(\tau h_n, \rho_\ell))\Phi_n(\tau h_n, \rho_\ell)\mathrm{d}\tau, \end{aligned} \tag{23}$$

and, in case of a constant stepsize $h = T/N$,

$$\begin{aligned} \Phi(T, \rho_\ell) &\equiv \Phi_N(h, \rho_\ell) \\ &= I + \frac{h^\alpha}{\Gamma(\alpha)} \sum_{n=1}^N \int_0^1 (N - n + 1 - \tau)^{\alpha-1} f'(y_n(\tau h, \rho_\ell))\Phi_n(\tau h, \rho_\ell)\mathrm{d}\tau. \end{aligned} \tag{24}$$

3.1. Piecewise quasi-polynomial approximation

The previous functions are then approximated via a piecewise quasi-polynomial approximation, as described in [6], which we here briefly recall, and generalize to the approximation of the fundamental matrix solution, too. In more detail, with reference to (19) and (22), we shall look for approximations:

$$\sigma_n(ch_n,\rho_\ell) \simeq y_n(ch_n,\rho_\ell), \tag{25}$$

$$\Psi_n(ch_n,\rho_\ell) \simeq \Phi_n(ch_n,\rho_\ell), \qquad c\in[0,1], \qquad n=1,\dots,N,$$

and, consequently,

$$y(T,\rho_\ell) \simeq \sigma_N(h_N,\rho_\ell), \qquad \Phi(T,\rho_\ell) \simeq \Psi_N(h_N,\rho_\ell). \tag{26}$$

Following steps similar to those in [6, Section 2], we consider the expansion of the vector field along the orthonormal polynomial basis, w.r.t. the weight function

$$\omega(x)=\alpha(1-x)^{\alpha-1}, \qquad s.t. \qquad \int_0^1 \omega(x)\,\mathrm{d}x = 1,$$

resulting into a scaled and shifted family of Jacobi polynomials[5]:

$$P_j(x) := \sqrt{\frac{2j+\alpha}{\alpha}}\,\bar P_j^{(\alpha-1,0)}(2x-1), \qquad x\in[0,1], \qquad j=0,1,\dots.$$

In so doing, for $n=1,\dots,N$, one obtains:

$$f(y_n(ch_n,\rho_\ell)) = \sum_{j\ge 0} P_j(c)\gamma_j(y_n,\rho_\ell), \qquad c\in[0,1],$$

with

$$\gamma_j(y_n,\rho_\ell) = \alpha\int_0^1 (1-\tau)^{\alpha-1}P_j(\tau)f(y_n(\tau h_n,\rho_\ell))\mathrm{d}\tau, \qquad j=0,1,\dots.$$

The approximations are derived by truncating the infinite series to a finite sum with s terms. Consequently, for $n=1,\dots,N$, one obtains[6]:

$$\sigma_n(ch_n,\rho_\ell) = \phi^\alpha_{n-1}(c,\rho_\ell) + h_n^\alpha\sum_{j=0}^{s-1}\gamma_j(\sigma_n,\rho_\ell)I^\alpha P_j(c), \qquad c\in[0,1], \tag{27}$$

with

$$\gamma_j(\sigma_n,\rho_\ell) = \alpha\int_0^1 (1-\tau)^{\alpha-1}P_j(\tau)f(\sigma_n(\tau h_n,\rho_\ell))\mathrm{d}\tau, \qquad j=0,\dots,s-1, \tag{28}$$

and

$$\phi^\alpha_{n-1}(c,\rho_\ell) = \rho_\ell + \sum_{\nu=1}^{n-1}h_\nu^\alpha\sum_{j=0}^{s-1}J_j^\alpha\left(\frac{r^{n-\nu}-1}{r-1}+cr^{n-\nu}\right)\gamma_j(\sigma_\nu,\rho_\ell), \tag{29}$$

having set, for $x>1$,[7]

$$J_j^\alpha(x) := \frac{1}{\Gamma(\alpha)}\int_0^1 (x-\tau)^{\alpha-1}P_j(\tau)\mathrm{d}\tau, \qquad j=0,\dots,s-1. \tag{30}$$

If a constant stepsize $h=T/N$ is used, then (29) reads:

$$\phi^\alpha_{n-1}(c,\rho_\ell) = \rho_\ell + h^\alpha\sum_{\nu=1}^{n-1}\sum_{j=0}^{s-1}J_j^\alpha(n-\nu+c)\gamma_j(\sigma_\nu,\rho_\ell), \tag{31}$$

and similarly one modifies (27) and (28).

[5] Here, $\bar P_j^{(a,b)}(x)$ denotes the jth Jacobi polynomial with parameters a and b, in $[-1,1]$.

[6] We refer to [4,8] for efficient procedures for computing the fractional integrals $I_j^\alpha P_j(c)$, $j=0,\dots,s-1$.

[7] We refer to [6,8] for the efficient computation of such integrals.

It can be shown (see [6]) that $\phi^{\alpha}_{n-1}(c,\rho_\ell)$ is nothing but the approximation of the *memory term*

$$G^{\alpha}_{n-1}(c,\rho_\ell)=\rho_\ell+\frac{1}{\Gamma(\alpha)}\sum_{\nu=1}^{n-1}h_\nu^{\alpha}\int_0^1\left(\frac{r^{n-\nu}-1}{r-1}+cr^{n-\nu}-\tau\right)^{\alpha-1}f(y_n(\tau h_n,\rho_\ell))\mathrm{d}\tau,$$

such that, for all $c\in[0,1]$, and $n=1,\dots,N$:

$$y_n(ch_n,\rho_\ell)=G^{\alpha}_{n-1}(c,\rho_\ell)+\frac{h_n^{\alpha}}{\Gamma(\alpha)}\int_0^c(c-\tau)^{\alpha-1}f(y_n(\tau h_n,\rho_\ell))\mathrm{d}\tau. \tag{32}$$

As matter of fact, (20) corresponds to set $n=N$ and $c=1$ in (32).

Similarly, when a constant stepsize $h=T/N$ is used, then

$$G^{\alpha}_{n-1}(c,\rho_\ell)=\rho_\ell+\frac{h^{\alpha}}{\Gamma(\alpha)}\sum_{\nu=1}^{n-1}\int_0^1(n-\nu+c-\tau)^{\alpha-1}f(y_n(\tau h,\rho_\ell))\mathrm{d}\tau,$$

and (32) still formally holds, upon replacing h_n with h. Consequently, (21) corresponds again to set $n=N$ and $c=1$ in (32).

The Fourier coefficients (28) can be approximated up to machine precision by using the Gauss-Jacobi formula of order $2k$ based at the zeros of $P_k(x)$, $c_1,\dots,c_k$, with corresponding weights $b_1,\dots,b_k$, by choosing k large enough. As is explained in [6, Section 3], this allows formulating the discrete problem for computing them as[8]:

$$\boldsymbol{\gamma}^n=\mathcal{P}_s^{\top}\Omega\otimes I_m f\left(\boldsymbol{\phi}^{\alpha}_{n-1}+h_n^{\alpha}\mathcal{I}_s^{\alpha}\otimes I_m\boldsymbol{\gamma}^n\right), \tag{33}$$

with, by setting $\gamma_j^n(\rho_\ell)$, $j=0,\dots,s-1$, the approximation to $\gamma_j(\sigma_n,\rho_\ell)$ obtained by using the Gauss-Jacobi quadrature formula,

$$\boldsymbol{\gamma}^n=\begin{pmatrix}\gamma_0^n(\rho_\ell)\\ \vdots\\ \gamma_{s-1}^n(\rho_\ell)\end{pmatrix}\in\mathbb{R}^{sm},\qquad \boldsymbol{\phi}^{\alpha}_{n-1}=\begin{pmatrix}\phi_{n-1}(c_1,\rho_\ell)\\ \vdots\\ \phi_{n-1}(c_k,\rho_\ell)\end{pmatrix}\in\mathbb{R}^{km},$$

and

$$\mathcal{P}_s=\begin{pmatrix}P_0(c_1) & \dots & P_{s-1}(c_1)\\ \vdots & & \vdots\\ P_0(c_k) & \dots & P_{s-1}(c_k)\end{pmatrix},\quad \mathcal{I}_s^{\alpha}=\begin{pmatrix}I^{\alpha}P_0(c_1) & \dots & I^{\alpha}P_{s-1}(c_1)\\ \vdots & & \vdots\\ I^{\alpha}P_0(c_k) & \dots & I^{\alpha}P_{s-1}(c_k)\end{pmatrix}\in\mathbb{R}^{k\times s},$$

$$\Omega=\begin{pmatrix}b_1 & & \\ & \ddots & \\ & & b_k\end{pmatrix}\in\mathbb{R}^{k\times k}.$$

Remark 4. It is worth noticing that the discrete problem (33) has (block) dimension s, *independently of* k. This, in turn, allows using relatively large values of k, in order to have an accurate approximation of the Fourier coefficients, without increasing too much the computational cost.

Moreover, the vector $\boldsymbol{\phi}^{\alpha}_{n-1}$ in (33) only depends on known quantities, computed at the previous timesteps.

Further, we observe that also the matrices $\mathcal{P}_s$, $\mathcal{I}_s^{\alpha}$, as well as all the required integrals (30), can be computed in advance, once for all, and they can be used for each new approximation ρ_ℓ in both Algorithms 1 and 2. Additionally, it is worth mentioning that, since they only depend on s,k,α,r, in principle they could be tabulated, without needing to be evaluated.

Considering that

$$\mathcal{I}^{\alpha}P_j(1)=\frac{1}{\Gamma(\alpha)}\int_0^1(1-x)^{\alpha-1}P_j(x)\mathrm{d}x=\frac{\delta_{j0}}{\Gamma(\alpha+1)},\qquad j=0,\dots,s-1,$$

the approximations of the solution at t_n is given by:

$$y(t_n,\rho_\ell)\simeq\sigma_n(h_n,\rho_\ell)\equiv\phi^{\alpha}_{n-1}(1,\rho_\ell)+\frac{h_n^{\alpha}}{\Gamma(\alpha+1)}\gamma_0^n(\rho_\ell),\qquad n=1,\dots,N. \tag{34}$$

According to [6] (see also [8]), we give the following definition.

[8] As is usual, the function f, here evaluated in a (block) vector of dimension k, denotes the (block) vector made up by f evaluated in each (block) entry of the input argument.

Definition 1. We shall refer to the method defined by (33)-(34), as a *Fractional HBVM with parameters k and s*, in short *FHBVM*(k,s).

In particular, from (26) and (34) one obtains, considering that $t_N = T$:

$$y(T,\rho_\ell) \simeq \sigma_N(h_N,\rho_\ell) \equiv \phi^\alpha_{N-1}(1,\rho_\ell) + \frac{h_N^\alpha}{\Gamma(\alpha+1)}\gamma_0^N(\rho_\ell). \tag{35}$$

Remark 5. When $\alpha = 1$, the polynomials $\{P_j\}_{\geq 0}$, become the usual Legendre polynomials orthonormal in $[0,1]$. Consequently, a FHBVM(k,s) method reduces to a standard HBVM(k,s) method, when $\alpha = 1$.

In a similar way, for $n = 1,\dots,N$:

$$\Psi_n(ch_n,\rho_\ell) = \Theta^\alpha_{n-1}(c,\rho_\ell) + h_n^\alpha \sum_{j=0}^{s-1} \Gamma_j(\sigma_n,\rho_\ell) I^\alpha P_j(c), \qquad c\in[0,1], \tag{36}$$

with

$$\Gamma_j(\sigma_n,\rho_\ell) = \alpha \int_0^1 (1-\tau)^{\alpha-1} P_j(\tau) f'(\sigma_n(\tau h_n,\rho_\ell))\Psi_n(\tau h_n,\rho_\ell)\mathrm{d}\tau, \qquad j = 0,\dots,s-1, \tag{37}$$

and (see (30))

$$\Theta^\alpha_{n-1}(c,\rho_\ell) = I + \sum_{\nu=1}^{n-1} h_\nu^\alpha \sum_{j=0}^{s-1} J_j^\alpha \left(\frac{r^{n-\nu}-1}{r-1} + c r^{n-\nu}\right)\Gamma_j(\sigma_\nu,\rho_\ell). \tag{38}$$

Similarly as in (31), when a constant stepsize $h = T/N$ is used, then (38) becomes:

$$\Theta^\alpha_{n-1}(c,\rho_\ell) = I + h^\alpha \sum_{\nu=1}^{n-1}\sum_{j=0}^{s-1} J_j^\alpha (n-\nu+c)\Gamma_j(\sigma_\nu,\rho_\ell). \tag{39}$$

As done for (28), by approximating the integrals in (37) by using the same Gauss-Jacobi formula as before, from (36) and (37) one derives a discrete problem in the form

$$\boldsymbol{\Gamma}^n = \mathcal{P}_s^\top \Omega \otimes I_m f'\left(\boldsymbol{\phi}^\alpha_{n-1} + h_n^\alpha \mathcal{I}_s^\alpha \otimes I_m \boldsymbol{\gamma}^n\right)\left[\boldsymbol{\Theta}^\alpha_{n-1} + h_n^\alpha \mathcal{I}_s^\alpha \otimes I_m \boldsymbol{\Gamma}^n\right], \tag{40}$$

where $\boldsymbol{\phi}^\alpha_{n-1}$ and $\boldsymbol{\gamma}^n$ have been already computed in (33),

$$f'\left(\boldsymbol{\phi}^\alpha_{n-1} + h_n^\alpha \mathcal{I}_s^\alpha \otimes I_m \boldsymbol{\gamma}^n\right) \in \mathbb{R}^{km\times km}$$

is the block diagonal matrix whose diagonal blocks are given by the corresponding evaluations of the Jacobian of f,

$$\boldsymbol{\Theta}^\alpha_{n-1} = \begin{pmatrix} \Theta^\alpha_{n-1}(c_1,\rho_\ell) \\ \vdots \\ \Theta^\alpha_{n-1}(c_k,\rho_\ell) \end{pmatrix} \in \mathbb{R}^{km\times m},$$

and, by setting $\Gamma_j^n(\rho_\ell)$, $j = 0,\dots,s-1$, the approximation to $\Gamma_j(\sigma_n,\rho_\ell)$ obtained through the Gauss-Jacobi formula,

$$\boldsymbol{\Gamma}^n = \begin{pmatrix} \Gamma_0^n(\rho_\ell) \\ \vdots \\ \Gamma_{s-1}^n(\rho_\ell) \end{pmatrix} \in \mathbb{R}^{sm\times m},$$

with the approximation of the solution at t_n given by:

$$\Phi(t_n,\rho_\ell) \simeq \Psi_n(h_n,\rho_\ell) \equiv \Theta^\alpha_{n-1}(1,\rho_\ell) + \frac{h_n^\alpha}{\Gamma(\alpha+1)}\Gamma_0^n(\rho_\ell), \qquad n = 1,\dots,N. \tag{41}$$

As is clear, (40)-(41) define the application of the FHBVM(k,s) method to the variational problem.

We observe that considerations similar to those made in Remark 4 for (33) can be now repeated for (40), with the additional fact that (40) amounts to just solving a *linear system of equations*.

At last, from (26) and (41) one eventually obtains:

$$\Phi(T,\rho_\ell) \simeq \Psi_N(h_N,\rho_\ell) \equiv \Theta^\alpha_{N-1}(1,\rho_\ell) + \frac{h_N^\alpha}{\Gamma(\alpha+1)}\Gamma_0^N(\rho_\ell). \tag{42}$$

Remark 6. By choosing values of s, and $k \geq s$, large enough, it can be seen that the approximations (35) and (42) provided by a FHBVM(k,s) method can be accurate up to machine precision. In fact, from the analysis carried out in [6], the error in approximating (13) and (14) is proved to be bounded by

$$O(h_1^{2\alpha} + h_N^{s+\alpha}),$$

if a graded mesh (15)–(17) is used, or by

$$O(h^{s+\alpha-1}), \qquad h = T/N,$$

if a uniform mesh can be considered. This latter case is appropriate when the vector field is everywhere smooth, in a neighborhood of the solution.

Actually, this amounts to using the method as a *spectrally accurate method in time*, as is the case for HBVMs [2,5,12,13]. This kind of approximations will be considered in the implementation of the algorithm listed in Table 1 (and for the simplified version of it, listed in Table 2), which we shall use for the numerical tests reported in Section 4.

3.2. Error estimation

It is worth mentioning that the procedure explained in the previous section allows to derive, as a by-product, an estimate for the error in the computed solution, due to the fact that, in Algorithm 1, the iteration is stopped when, for a suitably small tolerance tol,

$$|\rho_{\ell+1} - \rho_\ell| \leq tol. \tag{43}$$

In fact, in such a case, one expects that $|\rho_{\ell+1} - \rho^*| \approx tol$ as well. Consequently, by considering that at the mesh points, for $n = 1, \dots, N$:

$$y(t_n, \rho_\ell) \simeq \sigma_n(h_n, \rho_\ell) \equiv \phi_{n-1}^\alpha(1, \rho_\ell) + \frac{h_n^\alpha}{\Gamma(\alpha+1)}\gamma_0^n(\rho_\ell), \tag{44}$$

and, similarly,

$$\Phi(t_n, \rho_\ell) \simeq \Psi_n(h_n, \rho_\ell) \equiv \Theta_{n-1}^\alpha(1, \rho_\ell) + \frac{h_n^\alpha}{\Gamma(\alpha+1)}\Gamma_0^n(\rho_\ell),$$

by virtue of the perturbation result of Theorem 1, one derives the estimates

$$\|y(t_n, \rho^*) - y(t_n, \rho_\ell)\| \approx 2 \cdot tol \cdot \|\Psi_n(h_n, \rho_\ell)\|, \qquad n = 1, \dots, N. \tag{45}$$

3.3. The simplified shooting-Newton algorithm for semi-linear problems

For the simplified algorithm in Table 2, the approximation to $\hat{\Phi}(T, \rho_\ell)$ is in general problem dependent. However, there is a specific case where an efficient approximation can be readily obtained, i.e., when problem (1) is semi-linear:

$$y^{(\alpha)}(t) = Ly(t) + g(y(t)), \qquad t \in [0,T], \qquad y(T) = \eta \in \mathbb{R}^m, \tag{46}$$

with $L \in \mathbb{R}^{m\times m}$ and $\|L\| \gg \|g\|$ in a suitable neighborhood of the solution. In fact, in such a case, one can approximate the variational problem (7) with the linear part only (thus, independent of ρ),

$$\hat{\Phi}^{(\alpha)}(t) = L\,\Phi(t), \qquad t \in [0,T], \qquad \Phi(0) = I.$$

In so doing, one obtains the approximation

$$\hat{\Phi}(T) = E_\alpha(LT^\alpha) := \sum_{j\geq 0} \frac{(LT^\alpha)^j}{\Gamma(\alpha j + 1)},$$

with E_α the one-parameter Mittag-Leffler function. Furthermore, since we are interested in deriving only a convenient approximation to the fundamental matrix function, we can truncate the above series to a suitable finite sum. As an example, for a given tolerance ϵ, one may consider the approximation:

$$\hat{\Phi}(T) = \sum_{j=0}^{J} \frac{(LT^\alpha)^j}{\Gamma(\alpha j + 1)}, \qquad s.t. \qquad \frac{\|(LT^\alpha)^J\|}{\Gamma(\alpha J + 1)} \leq \epsilon. \tag{47}$$

4. Numerical tests

We here report a few numerical tests aimed at illustrating the theoretical findings. For all tests, we use $k = 22$ and $s = 20$, so that we are going to use a FHBVM(22,20) method. In other words, we use a local polynomial approximation of degree $s - 1 = 19$ for the vector field, coupled with a Gauss-Jacobi quadrature formula of order $2k = 44$ for approximating the Fourier coefficients (28)

Table 3
Results for Problem (48).

ℓ	ρ_ℓ
0	2.500000000000000e-01
1	-6.974105632991501e-03
2	-6.267686473630449e-06
3	-5.040632537594832e-12
4	-2.508583045846617e-15

and (37). We have used straightforward fixed-point iterations, derived from (33) and (40), respectively, to solve the corresponding discrete problems.[9] Namely,

$$\boldsymbol{\gamma}^{n,j+1} = \mathcal{P}_s^\top \Omega \otimes I_m f\left(\boldsymbol{\phi}_{n-1}^\alpha + h_n \mathcal{I}_s^\alpha \otimes I_m \boldsymbol{\gamma}^{n,j}\right), \qquad j = 0, 1, \dots,$$

starting from $\boldsymbol{\gamma}^{n,0} = \mathbf{0}$, for (33), and similarly for (40).[10] The iterations are carried out until full machine accuracy is gained, so that we expect full machine accuracy for the computed approximation (42) to $\Phi(T, \rho_\ell)$, as well as a corresponding fully accurate discrete solution (44).

We consider 6 test problems:

- the first 3 problems are the same scalar test problems in [17, Section 5];
- the last 3 problems are vector problems.

For all problems, (see (1)) the initial guess $\rho_0 = \eta$ has been considered. All numerical tests have been performed in Matlab© (Rel. 2023b) on a Silicon M2 laptop with 16GB of shared memory. The iteration of Algorithm 1 is stopped by using a tolerance $tol = 10^{-14}$ in (43). The same tolerance and stopping criterion will be used for Algorithm 2. To be more precise, we shall consider Algorithm 1 for solving all the problems, and Algorithm 2 for solving the last problem, which is semi-linear.

4.1. Example 1

The first problem is given by:

$$y^{(0.3)} = -|y|^{1.5} + \frac{8!}{\Gamma(8.7)} t^{7.7} - 3\frac{\Gamma(5.15)}{\Gamma(4.85)} t^{3.85} + \left(\frac{3}{2} t^{0.15} - t^4\right)^3 + \frac{9}{4}\Gamma(1.3),$$
$$t \in [0, 1], \qquad y(1) = \frac{1}{4}, \tag{48}$$

whose solution is

$$y(t) = t^8 - 3\, t^{4.15} + \frac{9}{4}\, t^{0.3}.$$

In this case, we use a uniform mesh with stepsize $h = 1/10$. The method converges in 4 iterations producing the approximations in Table 3. It is possible to appreciate the quadratic convergence of the iteration in the first iterations (in the last one, roundoff errors clearly dominate). The maximum error on the final solution is $\approx 6 \cdot 10^{-15}$, whereas the estimated one, by using (45), is $2 \cdot 10^{-14}$.

4.2. Example 2

The second problem is given by:

$$y^{(0.3)} = -\frac{3}{2}\, y, \qquad t \in [0, 7],$$
$$y(7) = \frac{14}{5} E_{0.3}\left(-\frac{3}{2}\, 7^{0.3}\right) \simeq .6476128469955936, \tag{49}$$

with $E_{0.3}$ the Mittag-Leffler function of order 0.3, with solution

$$y(t) = \frac{14}{5} E_{0.3}\left(-\frac{3}{2}\, t^{0.3}\right).$$

We refer to [20] and the accompanying software `ml.m`, for an efficient Matlab© implementation of the Mittag-Leffler function.

In this case a uniform mesh is not appropriate, since the vector field is proportional to the solution, which has a singularity in the first derivative at the origin. Consequently, we use a graded mesh, according to (17), with $h_1 = 10^{-14}$ and $N = 500$. Taking into

[9] We have used a fixed point iteration also for solving (40), despite the fact that it is just a linear system of equations.
[10] More refined nonlinear iterations are described in [8].

Table 4
Results for Problem (49).

ℓ	ρ_ℓ
0	.6476128469955936
1	2.799999999999968

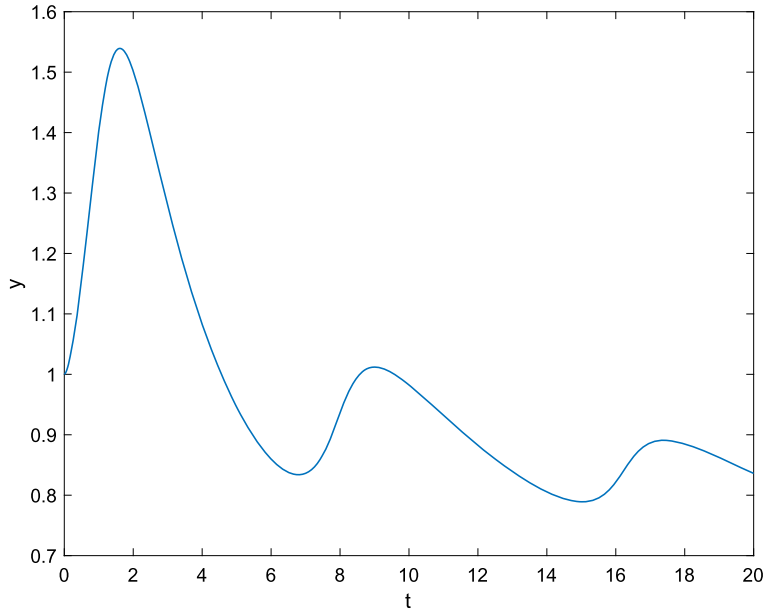

Fig. 1. Reference solution for problem (50).

Table 5
Results for Problem (50).

ℓ	ρ_ℓ
0	.8360565285776644
1	1.115178544783084
2	1.057854760373079
3	1.006528883050734
4	.9999714859685488
5	.9999999991678453
6	.9999999999999855

account (18), this implies $r \simeq 1.064914852480467$. According to the result of Theorem 4, the method converges in one iteration, as is shown in Table 4. The maximum error on the final solution is $\approx 2 \cdot 10^{-13}$, whereas the estimated one, by using is (45), is $2 \cdot 10^{-14}$ (in this case, the maximum error is essentially close to the origin, where there is the singularity of the derivative).

4.3. Example 3

The third problem is given by:

$$\begin{aligned} y^{(0.7)} &= \frac{1}{t+1}\sin(t \cdot y), \qquad t \in [0, 20], \\ y(20) &= 0.8360565285776644, \end{aligned} \tag{50}$$

which corresponds to the initial value $y(0) = 1$. In such a case, the solution is not known in closed form, and the final value has been taken from a reference solution computed by using the FHBVM(22,20) method with a constant stepsize $h = 0.02$ (i.e., by using 1000 timesteps). This solution is depicted in Fig. 1, and the estimated error (by using a doubled mesh) is $\approx 1.8 \cdot 10^{-14}$.

For solving problem (50), we use a uniform mesh with stepsize $h = 20/400 = 1/20$. The method converges in 6 iterations, producing the approximations listed in Table 5. Also in the case, it is possible to appreciate a quadratic-like convergence of the iteration. The maximum error in the final solution is $\approx 2 \cdot 10^{-14}$, whereas the estimated one, by using is (45), is $6 \cdot 10^{-14}$.

Table 6
Results for Problem (51).

ℓ	ρ_ℓ	
0	.2591172572977875	.5953212597441289
1	2.000000000000012	3.000000000000012

4.4. Example 4

We now consider the following linear (vector) FDE-TVP:

$$y^{(0.5)} = \begin{pmatrix} -3 & 0 \\ -2 & -1 \end{pmatrix} y, \qquad t \in [0,2], \tag{51}$$
$$y(2) = \begin{pmatrix} 2\,E_{0.5}\left(-3\cdot 2^{0.5}\right) \\ 2\,E_{0.5}\left(-3\cdot 2^{0.5}\right) + E_{0.5}\left(-2^{0.5}\right) \end{pmatrix} \simeq \begin{pmatrix} .2591172572977875 \\ .5953212597441289 \end{pmatrix},$$

having solution

$$y(t) = \begin{pmatrix} 2\,E_{0.5}\left(-3\cdot t^{0.5}\right) \\ 2\,E_{0.5}\left(-3\cdot t^{0.5}\right) + E_{0.5}\left(-t^{0.5}\right) \end{pmatrix},$$

corresponding to the initial value $y(0) = (2,\,3)^{\mathsf{T}}$. Since the vector field is linearly related to the solution, which has a singularity in the first derivative at the origin, we use a graded mesh with $h_1 = 10^{-14}$ and $N = 100$. According to the result of Theorem 4, convergence is gained in just one iteration, as is confirmed by Table 6. The maximum error in the final solution is $\approx 7\cdot 10^{-15}$, whereas the estimated one, by using (45), is $2\cdot 10^{-14}$.

4.5. Example 5

We now consider the following fractional Brusselator model:

$$\begin{aligned} y_1^{(0.7)} &= 1 - 4y_1 + y_1^2 y_2, \\ y_2^{(0.7)} &= 3y_1 - y_1^2 y_2, \qquad t \in [0,5], \\ y(5) &= \begin{pmatrix} .8904632063462272 \\ 3.326603532694057 \end{pmatrix}. \end{aligned} \tag{52}$$

In such a case, the solution is not explicitly known, and we have computed the final value starting from $y(0) = (1.2,\,2.8)^{\mathsf{T}}$ by using the FHBVM(22,20) method with a graded mesh with $h_1 = 10^{-14}$ and $N = 1000$: the reference solution is plotted in Fig. 2 in solid line, with the initial condition marked by the circle. We solve the problem by using the FHBVM(22,20) method on a graded mesh with $h_1 = 10^{-14}$ and $N = 200$. In so doing, the algorithm described in Table 1 converges in 5 iterations, with a quadratic-like order, obtaining the results listed in Table 7. The maximum estimated error in the final solution is $\approx 10^{-13}$, whereas that in the final point is $\approx 4\cdot 10^{-16}$.

4.6. Example 6

As a last example, we consider a family of semi-linear problems with $y \in \mathbb{R}^{2\nu}$ and

$$y^{(0.7)} = \begin{pmatrix} & I_\nu \\ -I_\nu & \end{pmatrix} y + \frac{1}{20}\cos(D_\nu y), \qquad t \in [0,5], \tag{53}$$

where $I_\nu \in \mathbb{R}^{\nu\times\nu}$ is the identity matrix, the function cos is meant to be applied in vector mode, and

$$D_\nu = \mathrm{diag}\,(1,\,2,\,\ldots,2\nu)^{-1}\,.$$

The reference solution at $t = 5$ has been computed by using the FHBVM(22,20) method on a graded mesh with $N = 300$ and $h_1 = 10^{-14}$, solving (53) starting from the initial value with entries:

$$y_i(0) = \frac{1}{i}\cos\left((i-1)\frac{\pi}{\nu}\right), \qquad i = 1,\ldots,2\nu. \tag{54}$$

We solve, at first, the FDE-TVP (53) with $y(5)$ given, by using Algorithm 1 with the FHBVM(22,20) method on a graded mesh with $N = 35$ and $h_1 = 10^{-8}$, for $\nu = 1,\ldots,35$, thus solving FDE-TVPs having dimension 2, 4, …, 70.

The algorithm in Table 1 turns out to always converge in 4–5 iterations. The error in the computed initial value is always less than $1.5\cdot 10^{-13}$. In Fig. 3 is the plot of the execution mean times (over 5 runs) of the algorithm versus the dimension of the problem. In more detail, the figure plots:

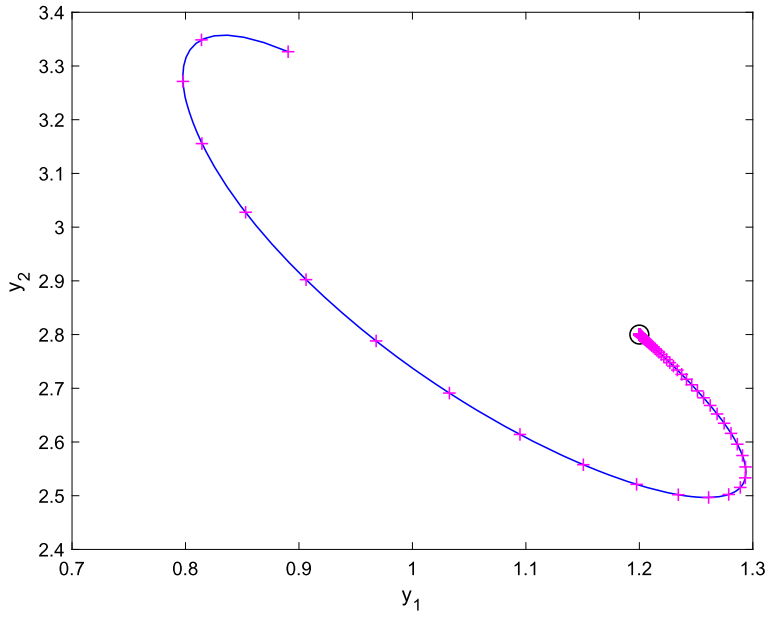

Fig. 2. Reference solution for problem (52) (solid line). The circle denotes the actual initial condition, whereas the pluses denote the final approximate solution.

Table 7
Results for Problem (52).

ℓ	ρ_ℓ	
0	.8904632063462272	3.326603532694057
1	1.195221947994766	2.798766749634182
2	1.199608077826518	2.800213499824565
3	1.199998157974212	2.800001859877902
4	1.199999999973615	2.800000000034993
5	1.199999999999924	2.800000000000298

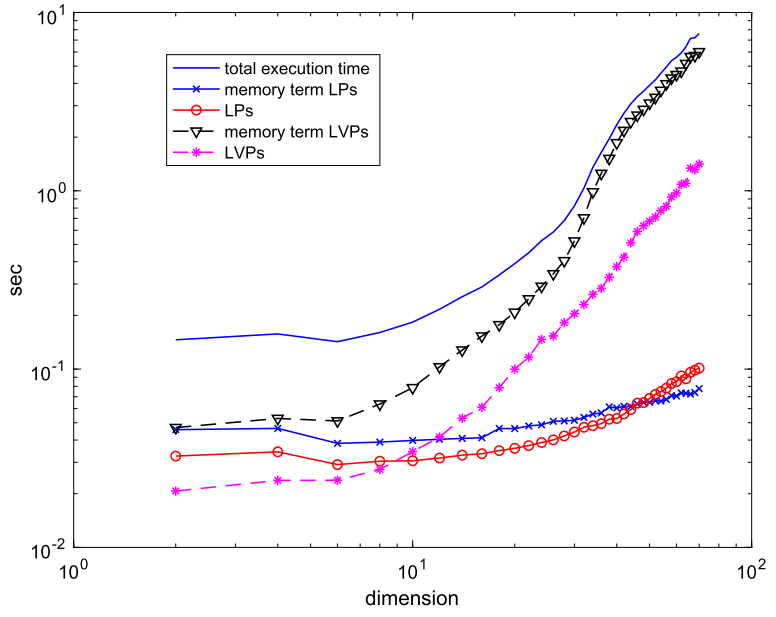

Fig. 3. Execution times of Algorithm 1 for solving problem (53) with $y(5)$ given, for dimensions ranging from 2 to 70. See the text for details.

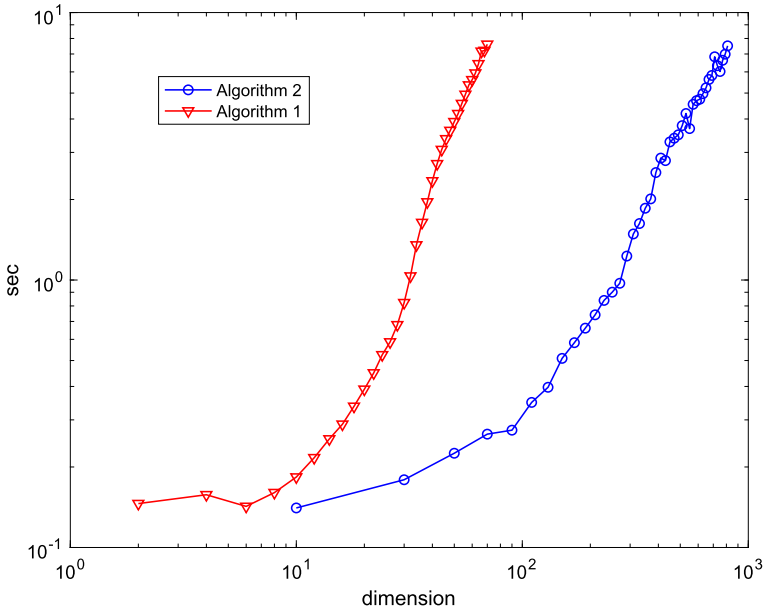

Fig. 4. Comparison of the total execution times of Algorithm 1 and Algorithm 2 for solving problem (53) with $y(5)$ given. See the text for details.

- the total execution time (all times are in `sec`);
- the time for computing the required memory terms $\phi^{\alpha}_{n-1}(c, \rho_{\ell})$ (29) in the local problems (LPs);
- the time for solving the local problems (33);
- the time for computing the memory terms (38) in the local variational problems (LVPs);
- the time for solving the local variational problems (40).

According to Remark 4, we have not considered the pre-processing time for evaluating the integrals $I^{\alpha}P_j$ in (27) and $J^{\alpha}_j(x)$ (30), also because they require an extended precision arithmetic (quadruple precision would be enough) but, at the moment, they are computed symbolically in Matlab, and not numerically, so that this part of the code is not yet optimized.

From the obtained results, one may conclude that most of the computational time of Algorithm 1 is spent in the solution of the variational problem: in particular, the evaluation of the memory terms for the local variational problems. For this reason, we now consider Algorithm 2 for solving problem (53). In fact, since the problem is in the form (46), we can use the (quite cheap) approximation (47) in place of the fundamental matrix function. Having fixed a tolerance $\varepsilon = 10^{-10}$, this results in using $J = 40$ in (47), which is quite inexpensive. In such a case, the algorithm in Table 2 converges in 9–10 iterations, instead of 4–5. Nevertheless, the overall execution time results to be relatively small, due to the fact that the solution of the variational problem is no more required. Fig. 4 contains the comparison between the total execution times of Algorithm 1, as seen in Fig. 3, and of Algorithm 2: this latter is used for solving problem (53) with $\nu = 5, 10, 15, 20, \ldots, 405$. The highest dimension ($2\nu = 810$) is chosen because the corresponding execution time is practically the same as that of Algorithm 1 when solving the problem of dimension 70 (about 7.5 sec). This clearly shows the superiority of the simplified shooting-Newton iteration over the original one, for this semi-linear problem.

5. Conclusions

In this paper we have described a novel shooting procedure which, coupled with the Newton method, proves very appealing for numerically solving terminal value problems for fractional differential equations. The implementation details of the given procedure have been thoroughly given, when the underlying numerical methods are FHBVMs. These latter methods, when used as spectrally accurate methods in time, allow deriving very accurate solutions, along with a suitable estimate of the error in the computed solution.

A corresponding cheaper procedure, relying on a simplified Newton method, has been also described. This latter procedure appears to be very promising for semi-linear problems since, in such a case, the associated variational equation is no more required. Numerical tests on both scalar and vector problems confirm the effectiveness of the presented approach.

Further directions of investigations include the extension for solving two-point boundary value problems, as well as the efficient numerical solution of the local variational problems, due to the fact that they amount to solving just linear systems of algebraic equations.

Funding

No funding was received for conducting this study.

Acknowledgements

The authors want to thank the referees for carefully reading the manuscript. All authors are members of GNCS-INdAM.

Data availability

All data reported in the manuscript have been obtained by a Matlab© implementation of the methods presented. They can be made available on request.

References

[1] P. Amodio, L. Brugnano, F. Iavernaro, Spectrally accurate solutions of nonlinear fractional initial value problems, AIP Conf. Proc. 2116 (2019) 140005, https://doi.org/10.1063/1.5114132.
[2] P. Amodio, L. Brugnano, F. Iavernaro, Analysis of spectral Hamiltonian boundary value methods (SHBVMs) for the numerical solution of ODE problems, Numer. Algorithms 83 (2020) 1489–1508, https://doi.org/10.1007/s11075-019-00733-7.
[3] P. Amodio, L. Brugnano, F. Iavernaro, Arbitrarily high-order energy-conserving methods for Poisson problems, Numer. Algorithms 91 (2022) 861–894, https://doi.org/10.1007/s11075-022-01285-z.
[4] P. Amodio, L. Brugnano, F. Iavernaro, A note on a stable algorithm for computing the fractional integrals of orthogonal polynomials, Appl. Math. Lett. 134 (2022) 108338, https://doi.org/10.1016/j.aml.2022.108338.
[5] P. Amodio, L. Brugnano, F. Iavernaro, (Spectral) Chebyshev collocation methods for solving differential equations, Numer. Algorithms 93 (2023) 1613–1638, https://doi.org/10.1007/s11075-022-01482-w.
[6] L. Brugnano, K. Burrage, P. Burrage, F. Iavernaro, A spectrally accurate step-by-step method for the numerical solution of fractional differential equations, J. Sci. Comput. 99 (2024) 48, https://doi.org/10.1007/s10915-024-02517-1.
[7] L. Brugnano, G. Frasca-Caccia, F. Iavernaro, V. Vespri, A new framework for polynomial approximation to differential equations, Adv. Comput. Math. 48 (2022) 76, https://doi.org/10.1007/s10444-022-09992-w.
[8] L. Brugnano, G. Gurioli, F. Iavernaro, Numerical solution of FDE-IVPs by using fractional HBVMs: the fhbvm code, Numer. Algorithms (2024), https://doi.org/10.1007/s11075-024-01884-y.
[9] L. Brugnano, F. Iavernaro, Line Integral Methods for Conservative Problems, Chapman et Hall/CRC, Boca Raton, FL, USA, 2016.
[10] L. Brugnano, F. Iavernaro, Line integral solution of differential problems, Axioms 7 (2) (2018) 36, https://doi.org/10.3390/axioms7020036.
[11] L. Brugnano, F. Iavernaro, A general framework for solving differential equations, Ann. Univ. Ferrara, Sez. 7: Sci. Mat. 68 (2022) 243–258, https://doi.org/10.1007/s11565-022-00409-6.
[12] L. Brugnano, J.I. Montijano, F. Iavernaro, L. Randéz, Spectrally accurate space-time solution of Hamiltonian PDEs, Numer. Algorithms 81 (2019) 1183–1202, https://doi.org/10.1007/s11075-018-0586-z.
[13] L. Brugnano, J.I. Montijano, L. Randéz, On the effectiveness of spectral methods for the numerical solution of multi-frequency highly-oscillatory Hamiltonian problems, Numer. Algorithms 81 (2019) 345–376, https://doi.org/10.1007/s11075-018-0552-9.
[14] K. Diethelm, The Analysis of Fractional Differential Equations. An Application-Oriented Exposition Using Differential Operators of Caputo Type, Lecture Notes in Math., vol. 2004, Springer-Verlag, Berlin, 2010.
[15] K. Diethelm, Increasing the efficiency of shooting methods for terminal value problems of fractional order, J. Comput. Phys. 293 (2015) 135–141, https://doi.org/10.1016/j.jcp.2014.10.054.
[16] K. Diethelm, N.J. Ford, A.D. Freed, Detailed error analysis for a fractional Adams method, Numer. Algorithms 36 (2004) 31–52, https://doi.org/10.1023/B:NUMA.0000027736.85078.be.
[17] K. Diethelm, F. Uhlig, A new approach to shooting methods for terminal value problems of fractional differential equations, J. Sci. Comput. 97 (2023) 38, https://doi.org/10.1007/s10915-023-02361-9.
[18] N.J. Ford, M.L. Morgado, Fractional boundary value problems: analysis and numerical algorithms, Fract. Calc. Appl. Anal. 14 (2011) 554–567, https://doi.org/10.2478/s13540-011-0034-4.
[19] N.J. Ford, M.L. Morgado, M. Rebelo, High order numerical methods for fractional terminal value problems, Comput. Methods Appl. Math. 14 (2014) 55–70, https://doi.org/10.1515/cmam-2013-0022.
[20] R. Garrappa, Numerical evaluation of two and three parameter Mittag-Leffler functions, SIAM J. Numer. Anal. 53 (3) (2015) 1350–1369, https://doi.org/10.1137/140971191.
[21] R. Garrappa, Trapezoidal methods for fractional differential equations: theoretical and computational aspects, Math. Comput. Simul. 110 (2015) 96–112, https://doi.org/10.1016/j.matcom.2013.09.012.
[22] R. Garrappa, Numerical solution of fractional differential equations: a survey and a software tutorial, Mathematics 6 (2) (2018) 16, https://doi.org/10.3390/math6020016.
[23] Z. Gu, Spectral collocation method for nonlinear Riemann-Liouville fractional terminal value problems, J. Comput. Appl. Math. 398 (2021) 113640, https://doi.org/10.1016/j.cam.2021.113640.
[24] Z. Gu, Y. Kong, Spectral collocation method for Caputo fractional terminal value problems, Numer. Algorithms 88 (2021) 93–111, https://doi.org/10.1007/s11075-020-01031-3.
[25] C. Li, M.-M. Li, H. Zhou, Terminal value problem for a generalized fractional ordinary differential equation, Math. Methods Appl. Sci. 44 (2021) 12963–12979, https://doi.org/10.1002/mma.7600.
[26] C. Li, Q. Yi, A. Chen, Finite difference methods with non-uniform meshes for nonlinear fractional differential equations, J. Comput. Phys. 316 (2016) 614–631, https://doi.org/10.1016/j.jcp.2016.04.039.
[27] Ch. Lubich, Fractional linear multistep methods for Abel-Volterra integral equations of the second kind, Math. Comput. 45 (172) (1985) 463–469, https://doi.org/10.1090/S0025-5718-1985-0804935-7.
[28] I. Podlubny, Fractional Differential Equations. An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications, Academic Press, Inc., San Diego, CA, 1999.
[29] B. Shiri, G.-C. Wu, D. Baleanu, Terminal value problems for the nonlinear systems of fractional differential equations, Appl. Numer. Math. 170 (2021) 162–178, https://doi.org/10.1016/j.apnum.2021.06.015.
[30] M. Stynes, E. O'Riordan, J.L. Gracia, Error analysis of a finite difference method on graded meshes for a time-fractional diffusion equation, SIAM J. Numer. Anal. 55 (2017) 1057–1079, https://doi.org/10.1137/16M1082329.
[31] V. Lakshmikantham, D. Trigiante, Theory of Difference Equations: Numerical Methods and Applications, Academic Press Inc., Boston, 1988.