

RESEARCH

Open Access



An efficient numerical approach for solving two-point fractional order nonlinear boundary value problems with Robin boundary conditions

Hyunju Kim¹, Junseo Lee² and Bongsoo Jang^{2*} 

*Correspondence:
bsjang@unist.ac.kr

²Department of Mathematical Sciences, Ulsan National Institute of Science and Technology (UNIST), 44919, Ulsan, Republic of Korea
Full list of author information is available at the end of the article

Abstract

This article proposes new strategies for solving two-point Fractional order Nonlinear Boundary Value Problems (FNBVPs) with Robin Boundary Conditions (RBCs). In the new numerical schemes, a two-point FNBVP is transformed into a system of Fractional order Initial Value Problems (FIVPs) with unknown Initial Conditions (ICs). To approximate ICs in the system of FIVPs, we develop nonlinear shooting methods based on Newton's method and Halley's method using the RBC at the right end point. To deal with FIVPs in a system, we mainly employ High-order Predictor–Corrector Methods (HPCMs) with linear interpolation and quadratic interpolation (Nguyen and Jang in *Fract. Calc. Appl. Anal.* 20(2):447–476, 2017) into Volterra integral equations which are equivalent to FIVPs. The advantage of the proposed schemes with HPCMs is that even though they are designed for solving two-point Fractional order Boundary Value Problems (FBVPs) with RBCs and have uniform convergence rates of HPCMs, $\mathcal{O}(h^2)$ and $\mathcal{O}(h^3)$ for shooting techniques with Newton's method and Halley's method, respectively. A variety of numerical examples are demonstrated to confirm the effectiveness and performance of the proposed schemes. Also we compare the accuracy and performance of our schemes with another method.

Keywords: Caputo fractional derivative; Nonlinear shooting method; Predictor–corrector scheme; Robin boundary condition

1 Introduction

Fractional calculus has proven to describe many phenomena in science and engineering more accurately than integer-order calculus because of the nonlocal property of the fractional derivative [2–7]. Many authors have introduced numerical methods for solving fractional differential equations arising in science and engineering. The authors in Refs. [8, 9] proposed a computational algorithm based on reproducing kernel Hilbert space for solving time-fractional partial differential equations in porous media and nonlinear homogeneous and nonhomogeneous time-fractional equations. In Ref. [10], a numerical method based on multiple fractional power series solution was introduced to deal with the

© The Author(s) 2021. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Schrödinger equation. The authors in Refs. [11–14] proposed several numerical methods based on collocation method, finite difference method, and L1 approximation for solving time-fractional diffusion equations. Regarding Fractional order Boundary Value Problems (FBVPs), Ref. [15] (and the references therein) investigated a Caputo fractional hybrid two-point boundary value problem describing the thermostat models. In Ref. [16], the authors studied a fractional-order nonlocal continuum model of a Euler–Bernoulli beam whose governing equation is described as a FBVP, using the fractional finite element model.

Recently, the authors in Ref. [17] developed a spectral collocation method to deal with two-point linear multi-term FBVPs with Caputo fractional operator. In Ref. [18], the authors reformulated two-point FBVPs with a Riemann–Liouville fractional operator to a Volterra integral equation of the second kind and then developed an integral discrete scheme based on finite difference method.

However, numerical methods for solving FNBVPs with Robin Boundary Conditions (RBCs) have been paid less attention to and, in this paper, we consider the two-point Fractional order Nonlinear Boundary Value Problem (FNBVP) with RBCs:

$$\begin{cases} D_a^{\alpha_2} y(t) = f(t, y, D_a^{\alpha_1} y(t)), & t \in [a, b], \\ a_1 y(a) + b_1 y'(a) = \gamma_1, & a_2 y(b) + b_2 y'(b) = \gamma_2, \end{cases} \tag{1}$$

where $0 < \alpha_1 \leq 1, 1 < \alpha_2 < 2, \alpha_1, \alpha_2, \gamma_1, \gamma_2 \in \mathbb{R}$. $D_a^{\alpha_1}$ and $D_a^{\alpha_2}$ are Caputo fractional differentiations defined as follows.

Definition 1.1 Let $\alpha \in \mathbb{R}^+$. The operator $J_{a,t}^\alpha$, defined on $L_1[a, b]$ by

$$J_a^\alpha y(t) := \frac{1}{\Gamma(\alpha)} \int_a^t (t - \tau)^{\alpha-1} y(\tau) d\tau,$$

for $a \leq t \leq b$, is called the Riemann–Liouville fractional integral operator of order α .

We set $\alpha = 0, J_a^0 = I$, the identity operator.

Definition 1.2 let $\alpha \in \mathbb{R}^+$. The operator D_a^α is defined by

$$D_a^\alpha y(t) := J_a^{[\alpha]-\alpha} D^{[\alpha]} y(t) = \frac{1}{\Gamma([\alpha] - \alpha)} \int_a^t (t - \tau)^{[\alpha]-\alpha} y^{([\alpha])}(\tau) d\tau,$$

where $\lceil \cdot \rceil$ is the ceiling function and $\lfloor \cdot \rfloor$ is the floor function.

The multi-term Caputo sense FIVP can be transformed into the system of FIVPs by Theorem 1.1 [19].

Theorem 1.1 *Let us consider the following multi-term Caputo sense fractional differential equation with initial conditions:*

$$\begin{cases} D_a^{\alpha_n} y(t) = f(t, y(t), D_a^{\alpha_1} y(t), D_a^{\alpha_2} y(t), \dots, D_a^{\alpha_{n-1}} y(t)), \\ y^{(j)}(a) = y_a^{(j)}, \quad j = 0, 1, \dots, \lfloor \alpha_n \rfloor, \end{cases} \tag{2}$$

where $y^{(j)}(a)$ is the j th derivative at $t = a$, $\alpha_n > \alpha_{n-1} > \dots > \alpha_1 > 0$, $\alpha_i - \alpha_{i-1} \leq 1$ for all $i = 2, 3, \dots, n$ and $0 < \alpha_1 < 1$. Then we can define β_i ,

$$\begin{cases} \beta_1 := \alpha_1, \\ \beta_i := \alpha_i - \alpha_{i-1}, \quad i = 2, 3, \dots, n. \end{cases}$$

Then the multi-term fractional differential equation with initial conditions (2) is equivalent to the following system of fractional differential equations:

$$\begin{aligned} D_a^{\beta_1} y_1(t) &= y_2(t), \\ D_a^{\beta_2} y_2(t) &= y_3(t), \\ &\vdots \\ D_a^{\beta_{n-1}} y_{n-1}(t) &= y_n(t), \\ D_a^{\beta_n} y_n &= f(t, y_1, y_2, \dots, y_{n-2}, y_{n-1}), \end{aligned} \tag{3a}$$

together with the initial conditions:

$$y_i(t_0) = \begin{cases} y^{(0)} & \text{if } i = 1, \\ y^{(l)} & \text{if } \alpha_{i-1} = l \in \mathbb{N}, \\ 0 & \text{else,} \end{cases} \tag{3b}$$

in the following sense:

1. Whenever the function $y \in C^{[\alpha_n]}[a, b]$ is a solution of the multi-term equation with initial conditions (2), the vector-valued function $Y := (y_1, \dots, y_n)^T$ with

$$y_i(t) := \begin{cases} y(t) & \text{if } i = 1, \\ D_a^{\alpha_{i-1}} y(t) & \text{if } i \neq 1, \end{cases}$$

is a solution of the system of fractional differential equations (3a) with initial conditions (3b)

2. Whenever the vector-valued function $Y := (y_1, \dots, y_n)^T$ is a solution of the system of multi-order fractional differential equations (3a) with initial conditions (3b), the function $y := y_1$ is a solution of the multi-term equation with initial conditions (2).

In this paper, we propose new schemes to deal with FNBVPs and the algorithms are summarized as follows:

1 In the case that $0 < \alpha_1 < 1$, we transform the FNBVP (1) with $a = 0$ into a system of FIVPs using Theorem 1.1.

In the case of $\alpha_1 = 1$, i.e. the FNBVP (1) has a single term of fractional order α_2 , we substitute the integer order $\alpha_1 = 1$ with the fractional order $\alpha_1 = 1 - \varepsilon$, $\varepsilon \rightarrow 0+$ so that the FNBVP satisfies the assumption, $0 < \alpha_1 < 1$ in Theorem 1.1. First, the Gronwall inequality for two-term equations in [19] guarantees that the difference between the solution of FNBVP with $\alpha_1 = 1$ and with $\alpha_1 = 1 - \varepsilon$ approaches 0 as $\varepsilon \rightarrow 0+$. The

FNBVP with $\alpha_1 = 1 - \varepsilon$ is transformed into a system of FIVPs and then we reduce the number of equations in the system. We prove the reduced system is equivalent to the original system as $\varepsilon \rightarrow 0+$ in Sect. 2.

- 2 To deal with FIVPs, we adopt high-order predictor–corrector methods (HPCMs) with linear interpolation and quadratic interpolation [1] into Volterra integral equations which are equivalent to FIVPs.
- 3 ICs of the FIVPs in the system equivalent to (1) are obtained by RBC at $t = 0$. But ICs include $s := y'(0)$ and since s is unknown, we approximate s by means of nonlinear shooting techniques based on Newton’s method and Halley’s method. The error function $|a_2y(b, s) + b_2y'(b, s) - \gamma_2|$ is used to construct the root-finding problem in order to make the approximate solution to $y(t)$ satisfy the RBC at $t = b$.
- 4 The algorithm of the proposed shooting technique is as follows: The system of FIVPs is solved with an initial approximation to s , s_k at the k th iteration. Using the approximate solution to the system obtained by HPCMs with s_0 , we find s_1 by solving Newton’s (Halley’s) formula. We update the approximate solution to the system with s_1 and measure the norm of the error function. We repeat this process until the norm of the error function is within the tolerance.

Similar to our proposed schemes, the authors in Refs. [20, 21] introduced numerical methods for solving FBVPs with RBCs. In Refs. [20, 21], the FBVP with RBCs is turned into the FIVP by using a shooting method with a guess for the unknown IC $y(0)$ and then the FIVP is transformed into the Volterra integral equation. The integral–differential term in the Volterra integral equation is approximated by an integral discretization scheme with constant and first-order interpolating polynomials in paper [20] and [21], respectively. However, the integral discretization schemes can only handle linear FBVPs and the rate of convergence depends on the fractional order α . This is elaborately addressed in Sect. 4.

The main advantages of our proposed schemes are as follows:

- 1 The proposed schemes can handle both linear and nonlinear FBVPs with general RBCs.
- 2 Our proposed schemes can deal with multi-term FBVPs where $0 < \alpha_1 \leq 1$ and $1 < \alpha_2 < 2$.
- 3 Our proposed methods with HPCMs have uniform convergence rates $\mathcal{O}(h^2)$ and $\mathcal{O}(h^3)$ for shooting techniques based on Newton’s method and Halley’s method, respectively, with enough iterations, regardless of fractional orders thanks to the global error estimates of HPCMs in [1].
- 4 It is not required to solve a matrix system as Newton’s method and Halley’s method are applied into a system of FIVPs.

This article is organized as follows. In Sect. 2, we describe an idea about the transformation of FNBVP with RBCs (1) into a system of FIVPs. In Sect. 3, we describe nonlinear shooting methods based on Newton’s method and Halley’s method, to approximate unknown IC $s := y(0)$ of FIVPs in the system. Also, we briefly mention how to apply the HPCMs into a system of FIVPs in Sect. 3. In Sect. 4, we demonstrate numerical examples verifying that the proposed shooting techniques combined with HPCMs guarantee the global convergence rates of HPCMs. We also confirm the performance and effectiveness of the proposed methods by comparing with the modified integral discretization scheme in Ref. [21]. A conclusion will be given in Sect. 5. Finally tables of numerical results and

the linear explicit method which is an alternative method for solving FIVPs are described in the [Appendix](#).

2 Problem formulation

In this section, we describe how to transform the FNBVP with RBCs (1) into a system of FIVPs according to the value of α_1 . Basically, we apply Theorem 1.1 with $\beta_1 := \alpha_1, \beta_2 := 1 - \alpha_1, \beta_3 := \alpha_2 - 1$ to the FNBVP in the case of $0 < \alpha_1 < 1$. If α_1 is equal to 1, then we replace α_1 with $1 - \epsilon, \epsilon \rightarrow 0+$ and set $\beta_1 := 1 - \epsilon, \beta_2 := \epsilon, \beta_3 := \alpha_2 - 1$. We reduce the size of system using the fact $\beta_2 \rightarrow 0+$.

Case 1: $0 < \alpha_1 < 1$

First, we consider a FNBVP with Dirichlet boundary conditions as follows:

$$\begin{cases} D_0^{\alpha_2} y(t) = f(t, y(t), D_0^{\alpha_1} y(t)), & t \in [0, b] \\ y(0) = y_0, & y(b) = y_b. \end{cases} \tag{4}$$

Applying Theorem 1.1 with $\beta_1 := \alpha_1, \beta_2 := 1 - \alpha_1, \beta_3 := \alpha_2 - 1$, the FNBVP with Dirichlet boundary conditions (4) can be transformed as follows:

$$\begin{cases} D_0^{\alpha_1} y(t) = w(t), & y(0) = y_0, y(b) = y_b, \\ D_0^{1-\alpha_1} w(t) = z(t), & w(0) = 0, \\ D_0^{\alpha_2-1} z(t) = f(t, y(t), w(t)), & z(0) = y^{(1)}(0). \end{cases} \tag{5}$$

From the system of fractional differential equations (5), we obtain the following system of FIVPs:

$$\begin{cases} D_0^{\alpha_1} y(t) = w(t), & y(0) = y_0, \\ D_0^{1-\alpha_1} w(t) = z(t), & w(0) = 0, \\ D_0^{\alpha_2-1} z(t) = f(t, y(t), w(t)), & z(0) = s, \end{cases} \tag{6}$$

where the IC s is unknown and so needs to be approximated.

Similar to the case of Dirichlet boundary conditions, the FNBVP with RBCs (1) can be written as follows:

$$\begin{cases} D_0^{\alpha_1} y(t) = w(t), & y(0) = y_0 = \frac{y_1 - b_1 s}{a_1} \\ D_0^{1-\alpha_1} w(t) = z(t), & w(0) = 0 \\ D_0^{\alpha_2-1} z(t) = f(t, y(t), w(t)), & z(0) = s. \end{cases} \tag{7}$$

Case 2: $\alpha_1 = 1$

We consider the following FBVP with Dirichlet boundary conditions:

$$\begin{cases} D_0^{\alpha_2} y(t) = f(t, y(t), y'(t)), & t \in [0, b], \\ y(0) = y_0, & y(b) = y_b, \end{cases} \tag{8}$$

where $1 < \alpha_2 < 2, \alpha_2 \in \mathbb{R}$. Since the fractional differential equation in (8) does not satisfy the assumption, $0 < \alpha_1 < 1$ in Theorem 1.1, we cannot apply the strategy used in (4) to

(8). So we modify the equation in (8) to meet the assumption, with the same boundary conditions as in (8) as follows:

$$\begin{cases} D_0^{\alpha_2} y(t) = f(t, y(t), D_0^{1-\epsilon} y(t)), & t \in [0, b], \\ y(0) = y_0, & y(b) = y_b, \end{cases} \tag{9}$$

where $\alpha_2 \in (1, 2)$, $\epsilon \rightarrow 0+$. By Lemma 2.1, solutions of the two FBVPs (8) and (9) are approximately equal and the absolute error depends on ϵ .

Lemma 2.1 (First Gronwall inequality for two-term equations in [19]) *Let $\alpha_2 > 0$ and $\alpha_1, \tilde{\alpha}_1 \in (0, \alpha_2)$ be chosen so that the equation*

$$D_0^{\alpha_2} y(t) = f(t, y(t), D_0^{\alpha_1} y(t)),$$

subject to the initial conditions

$$y(0) = y_0, \quad y^{(1)}(0) = y_0^1, \quad \dots, \quad y^{([\alpha_2]-1)}(0) = y_0^{[\alpha_2]-1}$$

and

$$D_0^{\alpha_2} z(t) = f(t, z(t), D_0^{\tilde{\alpha}_1} z(t))$$

subject to the same initial conditions

$$z(0) = y_0, \quad z^{(1)}(0) = y_0^1, \quad \dots, \quad z^{([\alpha_2]-1)}(0) = y_0^{[\alpha_2]-1}$$

(where f satisfies a Lipschitz condition in its second and third arguments on a suitable domain) has unique continuous solutions $y, z : [0, T] \rightarrow \mathbb{R}$. We assume further that $\lfloor \alpha_1 \rfloor = \lfloor \tilde{\alpha}_1 \rfloor$. Then there exist constants K_1 and K_2 such that

$$|y(t) - z(t)| \leq K_1 |\alpha_1 - \tilde{\alpha}_1| E_{\alpha_n}(K_2 T^{\alpha_2}), \quad \forall t \in [0, T],$$

where E_{α_n} denotes the Mittag-Leffler function of order α_n .

Applying Theorem 1.1 to (9), we obtain the following system of FIVPs:

$$\begin{cases} D_0^{1-\epsilon} y(t) = w(t), & y(0) = y_0, \\ D_0^\epsilon w(t) = z(t), & w(0) = 0, \\ D_0^{\alpha_2-1} z(t) = f(t, y(t), w(t)), & z(0) = s. \end{cases} \tag{10}$$

Now, we show that the system of FIVPs (10) is equivalent to the following system as $\epsilon \rightarrow 0$ using Lemmas 2.2 through 2.4 and Theorem 2.1:

$$\begin{cases} D_0^{1-\epsilon} \tilde{y}(t) = \tilde{z}(t), & \tilde{y}(0) = y_0, \\ D_0^{\alpha_2-1} \tilde{z}(t) = f(t, \tilde{y}(t), \tilde{z}(t)), & \tilde{z}(0) = s. \end{cases} \tag{11}$$

Lemma 2.2 (Theorem 2.10 in [19]) *Let $f \in C[a, b]$ and $\alpha \geq 0$. Moreover, assume that α_k is a sequence of positive numbers such that $\alpha_k \rightarrow \alpha$ as $k \rightarrow \infty$. Then, for every $\delta > 0$,*

$$\lim_{k \rightarrow \infty} \sup_{t \in [a+\delta, b]} |J_a^{\alpha_k} f(t) - J_a^\alpha f(t)| = 0.$$

Lemma 2.3 (Lemma 6.19 in [19]) *Let $\alpha, T, \phi_1, \phi_2 \in \mathbb{R}^+$. Moreover, assume that $\delta : [0, T] \rightarrow \mathbb{R}$ is a continuous function satisfying the inequality*

$$|\delta(t)| \leq \phi_1 + \frac{\phi_2}{\Gamma(\alpha)} \int_0^t (t - \tau)^{\alpha-1} |\delta(\tau)| d\tau, \quad \forall t \in [0, T].$$

Then

$$|\delta(t)| \leq \phi_1 E_\alpha(\phi_2 t^\alpha), \quad \forall t \in [0, T].$$

Lemma 2.4 *Let $0 < \gamma \leq \alpha \leq \beta$. Then, for any $t \in [a, b]$,*

$$|J_a^\alpha y(t) - J_a^\beta y(t)| \leq \left[\frac{\Gamma(\gamma)}{\Gamma(\alpha)} (b - a)^{\alpha-\gamma} + \frac{\Gamma(\gamma)}{\Gamma(\beta)} (b - a)^{\beta-\gamma} \right] J_a^\gamma |y(t)|.$$

Proof By Definition 1.1,

$$\begin{aligned} J_a^\alpha y(t) + J_a^\beta y(t) &= \frac{1}{\Gamma(\alpha)} \int_a^t (t - \tau)^{\alpha-1} y(\tau) d\tau + \frac{1}{\Gamma(\beta)} \int_a^t (t - \tau)^{\beta-1} y(\tau) d\tau \\ &= \frac{1}{\Gamma(\alpha)} \int_a^t (t - \tau)^{\gamma-1+(\alpha-\gamma)} y(\tau) d\tau \\ &\quad + \frac{1}{\Gamma(\beta)} \int_a^t (t - \tau)^{\gamma-1+(\beta-\gamma)} y(\tau) ds. \end{aligned}$$

Then

$$\begin{aligned} |J_a^\alpha y(t) - J_a^\beta y(t)| &\leq \frac{(b - a)^{\alpha-\gamma}}{\Gamma(\alpha)} \int_a^t (t - \tau)^{\gamma-1} |y(\tau)| d\tau \\ &\quad + \frac{(b - a)^{\beta-\gamma}}{\Gamma(\beta)} \int_a^t (t - \tau)^{\gamma-1} |y(\tau)| d\tau \\ &\leq \left[\frac{\Gamma(\gamma)}{\Gamma(\alpha)} (b - a)^{\alpha-\gamma} + \frac{\Gamma(\gamma)}{\Gamma(\beta)} (b - a)^{\beta-\gamma} \right] \frac{1}{\Gamma(\gamma)} \int_a^t (t - \tau)^{\gamma-1} |y(\tau)| d\tau \\ &\leq \left[\frac{\Gamma(\gamma)}{\Gamma(\alpha)} (b - a)^{\alpha-\gamma} + \frac{\Gamma(\gamma)}{\Gamma(\beta)} (b - a)^{\beta-\gamma} \right] J_a^\gamma |y(t)|. \quad \square \end{aligned}$$

Theorem 2.1 *Let $1 < \alpha_2 < 2, \forall T \in \mathbb{R}^+$ and $f : [0, T] \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ satisfies the Lipschitz condition in its second and third arguments on a suitable domain. Then we have the following inequality:*

$$|f(t, x_1, y_1) - f(t, x_2, y_2)| \leq L(|x_2 - x_1| + |y_2 - y_1|),$$

where $\forall t \in [0, T], x_1, x_2, y_1, y_2 : [0, T] \rightarrow \mathbb{R}$ and $0 < L$.

If, for any $0 < \epsilon \ll 1$, \hat{y} and \tilde{y} are solutions of the following systems, respectively:

$$\begin{cases} D_0^{1-\epsilon} \hat{y}(t) = w(t), & y(0) = y_0, \\ D_0^\epsilon w(t) = z(t), & w(0) = 0, \\ D_0^{\alpha_2-1} z(t) = f(t, \hat{y}(t), w(t)), & z(0) = s, \end{cases}$$

and

$$\begin{cases} D_0^{1-\epsilon} \tilde{y}(t) = \tilde{z}(t), & \tilde{y}(0) = y_0, \\ D_0^{\alpha_2-1} \tilde{z}(t) = f(t, \tilde{y}(t), \tilde{z}(t)), & \tilde{z}(0) = s, \end{cases}$$

then

$$|\hat{y}(t) - \tilde{y}(t)| \rightarrow 0, \quad \text{as } \epsilon \rightarrow 0.$$

Proof By Lemma 6.2 in [19], FIVPs $D_0^{1-\epsilon} \hat{y}(t) = w(t)$, $y(0) = y_0$, and $D_0^{1-\epsilon} \tilde{y}(t) = \tilde{z}(t)$, $\tilde{y}(0) = y_0$ are equivalent to Volterra integral equations of the second kind, respectively, as follows:

$$\hat{y}(t) = y_0 + J_0^{1-\epsilon} w(t), \quad \tilde{y}(t) = y_0 + J_0^{1-\epsilon} \tilde{z}(t).$$

Then $\hat{y}(t) - \tilde{y}(t)$ can be expressed as a Riemann–Liouville fractional integral of $w(t) - \tilde{z}(t)$ as follows:

$$\hat{y}(t) - \tilde{y}(t) = J_0^{1-\epsilon} (w(t) - \tilde{z}(t)). \tag{12}$$

Since $w(t) = J^\epsilon z(t)$ by Lemma 6.2 in [19] and rewriting $w(t) - \tilde{z}(t)$ as $w(t) - J_0^\epsilon \tilde{z}(t) + J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)$, we obtain the following inequality:

$$|w(t) - \tilde{z}(t)| \leq J_0^\epsilon |z(t) - \tilde{z}(t)| + |J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)|. \tag{13}$$

Since $D_0^{\alpha_2-1} z(t) = f(t, \hat{y}(t), w(t))$ and $D_0^{\alpha_2-1} \tilde{z}(t) = f(t, \tilde{y}(t), \tilde{z}(t))$ are equivalent to Volterra integral equations $z(t) = s + J_0^{\alpha_2-1} f(t, \hat{y}(t), w(t))$ and $\tilde{z}(t) = s + J_0^{\alpha_2-1} f(t, \tilde{y}(t), \tilde{z}(t))$ by Lemma 6.2 in [19] respectively, using the Lipschitz condition, (12), and (13), we obtain the following inequalities:

$$\begin{aligned} |z(t) - \tilde{z}(t)| &= |J_0^{\alpha_2-1} [f(t, \hat{y}(t), w(t)) - f(t, \tilde{y}(t), \tilde{z}(t))]| \\ &\leq J_0^{\alpha_2-1} |f(t, \hat{y}(t), w(t)) - f(t, \tilde{y}(t), \tilde{z}(t))| \\ &\leq LJ_0^{\alpha_2-1} (|\hat{y}(t) - \tilde{y}(t)| + |w(t) - \tilde{z}(t)|) \\ &\leq LJ_0^{\alpha_2-1} [J_0^{1-\epsilon} |w(t) - \tilde{z}(t)| + J_0^\epsilon |z(t) - \tilde{z}(t)|] \end{aligned} \tag{14}$$

$$+ |J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)|. \tag{15}$$

Since $1 < \alpha_2 < 2$ and $\Gamma(\alpha_2) = (\alpha_2 - 1)\Gamma(\alpha_2 - 1)$, we have the following inequalities for $J_0^{\alpha_2-1} |J^\epsilon \tilde{z}(t) - \tilde{z}(t)|$:

$$J_0^{\alpha_2-1} |J^\epsilon \tilde{z}(t) - \tilde{z}(t)| = \frac{1}{\Gamma(\alpha_2 - 1)} \int_0^t (t - \tau)^{\alpha_2-2} |J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)| d\tau$$

$$\begin{aligned} &\leq \frac{T^{\alpha_2-1}}{\Gamma(\alpha_2-1)} \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty \\ &\leq \frac{T^{\alpha_2-1}}{\Gamma(\alpha_2)} \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty. \end{aligned} \tag{16}$$

Similarly, we can obtain the following inequality:

$$J_0^{\alpha_2-\epsilon} |J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)| \leq \frac{T^{\alpha_2-\epsilon}}{\Gamma(\alpha_2-\epsilon+1)} \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty. \tag{17}$$

Applying the inequality (16) into (14), we have the following inequality:

$$\begin{aligned} |z(t) - \tilde{z}(t)| &\leq L \left[J_0^{\alpha_2-\epsilon} |w(t) - \tilde{z}(t)| + J_0^{\alpha_2-1+\epsilon} |z(t) - \tilde{z}(t)| \right. \\ &\quad \left. + \frac{T^{\alpha_2-1}}{\Gamma(\alpha_2)} \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty \right]. \end{aligned} \tag{18}$$

Using the inequalities (13) and (17), we have the following inequalities:

$$\begin{aligned} J_0^{\alpha_2-\epsilon} |w(t) - \tilde{z}(t)| &\leq J_0^{\alpha_2-\epsilon} (J_0^\epsilon |z(t) - \tilde{z}(t)| + |J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)|) \\ &\leq J_0^{\alpha_2} |z(t) - \tilde{z}(t)| \\ &\quad + \frac{T^{\alpha_2-\epsilon}}{\Gamma(\alpha_2-\epsilon+1)} \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty. \end{aligned} \tag{19}$$

Applying the inequality (19) into (18) and using Lemma 2.4 with $\gamma = \alpha_2 - 1$, $\alpha = \alpha_2 - 1 + \epsilon$, $\beta = \alpha_2$, we have the following inequalities:

$$\begin{aligned} |z(t) - \tilde{z}(t)| &\leq L \left[J_0^{\alpha_2} |z(t) - \tilde{z}(t)| + \frac{T^{\alpha_2-\epsilon}}{\Gamma(\alpha_2-\epsilon+1)} \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty \right. \\ &\quad \left. + J_0^{\alpha_2-1+\epsilon} |z(t) - \tilde{z}(t)| + \frac{T^{\alpha_2-1}}{\Gamma(\alpha_2)} \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty \right] \\ &= L \left[J_0^{\alpha_2-\epsilon} |z(t) - \tilde{z}(t)| + J_0^{\alpha_2-1+\epsilon} |z(t) - \tilde{z}(t)| \right. \\ &\quad \left. + \left\{ \frac{T^{\alpha_2-\epsilon}}{\Gamma(\alpha_2-\epsilon+1)} + \frac{T^{\alpha_2-1}}{\Gamma(\alpha_2)} \right\} \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty \right] \\ &\leq L \left[\left\{ \frac{\Gamma(\alpha_2-1)}{\Gamma(\alpha_2-1+\epsilon)} T^\epsilon + \frac{\Gamma(\alpha_2-1)}{\Gamma(\alpha_2)} T \right\} J_0^{\alpha_2-1} |z(t) - \tilde{z}(t)| \right. \\ &\quad \left. + \left\{ \frac{T^{\alpha_2-\epsilon}}{\Gamma(\alpha_2-\epsilon+1)} + \frac{T^{\alpha_2-1}}{\Gamma(\alpha_2)} \right\} \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty \right], \end{aligned}$$

so that

$$|z(t) - \tilde{z}(t)| \leq LC_\epsilon^1 J_0^{\alpha_2-1} |z(t) - \tilde{z}(t)| + LC_\epsilon^2 \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty, \tag{20}$$

where

$$C_\epsilon^1 \equiv \frac{\Gamma(\alpha_2-1)}{\Gamma(\alpha_2-1+\epsilon)} T^\epsilon + \frac{\Gamma(\alpha_2-1)}{\Gamma(\alpha_2)} T, \quad C_\epsilon^2 \equiv \frac{T^{\alpha_2-\epsilon}}{\Gamma(\alpha_2-\epsilon+1)} + \frac{T^{\alpha_2-1}}{\Gamma(\alpha_2)}.$$

Applying Lemma 2.3 to inequality (20), we obtain the following inequality:

$$|z(t) - \tilde{z}(t)| \leq LC_\epsilon^2 \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty E_{\alpha_2-1} [LC_\epsilon^1 T^{\alpha_2-1}].$$

Therefore,

$$\begin{aligned} |\tilde{y}(t) - \hat{y}(t)| &\leq J_0^{1-\epsilon} |w(t) - \tilde{z}(t)| \\ &\leq J_0^1 |z(t) - \tilde{z}(t)| + J_0^{1-\epsilon} |J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)| \\ &\leq T |z(t) - \tilde{z}(t)| + \frac{T^{1-\epsilon}}{\Gamma(2-\epsilon)} \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty \\ &\leq TLC_\epsilon^2 \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty E_{\alpha_2-1} [LC_\epsilon^1 T^{\alpha_2-1}] \\ &\quad + \frac{T^{1-\epsilon}}{\Gamma(2-\epsilon)} \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty \\ &= \left[TLC_\epsilon^2 E_{\alpha_2-1} [LC_\epsilon^1 T^{\alpha_2-1}] + \frac{T^{1-\epsilon}}{\Gamma(2-\epsilon)} \right] \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty. \end{aligned}$$

We have $C_\epsilon \equiv TLC_\epsilon^2 E_{\alpha_2-1} [LC_\epsilon^1 T^{\alpha_2-1}] + \frac{T^{1-\epsilon}}{\Gamma(2-\epsilon)}$. Then

$$|\tilde{y}(t) - \hat{y}(t)| \leq C_\epsilon \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty.$$

Thus, by Lemma 2.2,

$$|\tilde{y}(t) - \hat{y}(t)| \leq C_\epsilon \|J_0^\epsilon \tilde{z}(t) - \tilde{z}(t)\|_\infty \rightarrow 0, \quad \text{as } \epsilon \rightarrow 0. \quad \square$$

For the FNBVP with RBCs (1), similarly, by Lemma 2.1 and using Theorem 1.1, we obtain the following FIVPs:

$$\begin{cases} D_0^{1-\epsilon} y(t) = w(t), & y(0) = \frac{\gamma_1 - b_1 s}{a_1}, \\ D_0^\epsilon w(t) = z(t), & w(0) = 0, \\ D_0^{\alpha_2-1} z(t) = f(t, y(t), w(t)), & z(0) = s. \end{cases} \tag{21}$$

The system (21) can be reduced as follows:

$$\begin{cases} D_0^{1-\epsilon} \tilde{y}(t) = \tilde{z}(t), & \tilde{y}(0) = \frac{\gamma_1 - b_1 s}{a_1}, \\ D_0^{\alpha_2-1} \tilde{z}(t) = f(t, \tilde{y}(t), \tilde{z}(t)), & \tilde{z}(0) = s. \end{cases} \tag{22}$$

3 Nonlinear shooting methods and high-order predictor–corrector methods

FBVPs have been transformed to systems of FIVPs in Sect. 2. Before we address how to deal with systems of FIVPs (6), (7), (11), and (22) using High-order Predictor–Corrector Methods (HPCMs), the unknown IC $z(0) = s$ should be handled first. In this section, we describe two nonlinear shooting techniques based on Newton’s method and Halley’s method to approximate s . Both Newton’s formula and Halley’s formula are designed to determine the solution of a system of FIVPs satisfying the RBC at the right end point of an interval. Without loss of generality, we consider the system of FIVPs (7) that is equivalent to the FNBVP with RBCs (1).

In order that the RBC at the right end point $a_2y(b) + b_2y'(b) = \gamma_2$ is involved in approximating s , we define $y(s) := y(s, t)|_{t=b}$ and let the error function be $F(s) := a_2y(s) + b_2\frac{\partial}{\partial t}y(s) - \gamma_2$. We approximate the solution of the root-finding problem $F(s) = 0$ by using Newton's method and Halley's method, respectively. For convenience, we denote

$$g_s(t) = \frac{\partial g(s, t)}{\partial s}, \quad g_{ss} = \frac{\partial^2 g(s, t)}{\partial s^2}$$

throughout this section.

3.1 Shooting with Newton's method

The conventional Newton formula for $F(s) = 0$ can be expressed as follows:

$$s_{k+1} = s_k - \frac{F(s_k)}{F_s(s_k)}, \quad k = 0, 1, 2, \dots, m, \tag{23}$$

where m is the maximum number of iterations and

$$\begin{aligned} F_s(s_k) &= \left. \frac{\partial F}{\partial s}(s) \right|_{s=s_k, t=b} \\ &= a_2 \left. \frac{\partial y}{\partial s}(s, t) \right|_{s=s_k, t=b} + b_2 \left. \frac{\partial}{\partial s} \left[\frac{\partial y}{\partial t}(s, t) \right] \right|_{s=s_k, t=b}. \end{aligned} \tag{24}$$

Observing $y_s(s_k)$ and $y_{ts}(s_k)$, it turns out that they are equal to $\frac{\partial}{\partial s}y(t)|_{s=s_k, t=b}$ and $\frac{\partial}{\partial s}z(t)|_{s=s_k, t=b}$, respectively, in the system of FIVPs (7). Thus we solve the following system obtained from the system of FIVPs (7) by applying the operator $\frac{\partial}{\partial s}$ using HPCMs for each k :

$$\begin{cases} D_0^{\alpha_1} y_s(t) = w_s(t), & y_s(0) = -b_1/a_1, \\ D_0^{1-\alpha_1} w_s(t) = z_s(t), & w_s(0) = 0, \\ D_0^{\alpha_2-1} z_s(t) = f_s(t, y(t), w(t)), & z_s(0) = 1. \end{cases} \tag{25}$$

Since both t and s are independent variables, $f_s(t, y(t), w(t))$ can be written as

$$f_s(t, y(t), w(t)) = f_y \cdot y_s(t) + f_w \cdot w_s(t). \tag{26}$$

The detailed description of HPCMs dealing with a system of FIVPs is in Sect. 3.3. By solving the system (25), s_{k+1} in Newton's formula (23) is computed. Using the updated approximate value of IC s , s_{k+1} , we update approximate solutions of systems of FIVPs (6), (7), (11), and (22). Repeating this process, we obtain an s_k having an acceptable error of the root-finding problem $F(s) = 0$ at an appropriate number of iterations k .

3.2 Shooting with Halley's method

The conventional Halley formula for $F(s) = 0$ is as follows:

$$s_{k+1} = s_k - \frac{2F(s_k)F_s(s_k)}{2F_s^2(s_k) - F(s_k)F_{ss}(s_k)}, \quad k = 0, 1, 2, \dots, m, \tag{27}$$

where $F_s(s_k)$ is described in (24) and

$$F_{ss}(s_k) = \frac{\partial^2 F}{\partial s^2}(s) \Big|_{s=s_k, t=b} = a_2 \frac{\partial^2 y}{\partial s^2}(s, t) \Big|_{s=s_k, t=b} + b_2 \frac{\partial^2}{\partial s^2} \left[\frac{\partial y}{\partial t}(s, t) \right] \Big|_{s=s_k, t=b}.$$

Similar to the way we found $y_s(s_k)$ and $y_{ts}(s_k)$ in the shooting with Newton’s method, we find $y_{ss}(s_k)$ and $z_{ss}(s_k)$ by solving the following system of FIVPs obtained by applying the operator $\frac{\partial^2}{\partial s^2}$ using HPCMs for each k :

$$\begin{cases} D_a^{\alpha_1} y_{ss}(t) = w_{ss}(t), & y_{ss}(0) = 0, \\ D_a^{1-\alpha_1} w_{ss}(t) = z_{ss}(t), & w_{ss}(0) = 0, \\ D_a^{\alpha_2-1} z_{ss}(t) = f_{ss}(t, y(t), w(t)), & z_{ss}(0) = 0. \end{cases} \tag{28}$$

Since t and s are independent variables, $f_{ss}(t, y(t), w(t))$ can be written as

$$f_y \cdot y_{ss}(t) + f_w \cdot w_{ss}(t) + f_{yy} \cdot y_s(t)^2 + f_{ww} \cdot w_s(t)^2 + f_{wy} \cdot w_s(t)y_s(t). \tag{29}$$

3.3 High-order predictor–corrector methods for system of FIVPs

In order to find a s_k with an acceptable accuracy, we iteratively solve systems of FIVPs (25) or (28). Once we find the s_k , we solve systems of FIVPs (6), (7), (11), or (22). In this subsection, we describe how to deal with those systems of FIVPs using High-order Predictor–Corrector Methods (HPCMs) introduced in Ref. [1]. Without loss of generality, we consider the following FIVP:

$$\begin{cases} D_0^\alpha y(t) = f(t, y(t)), & t \in [0, b], \\ D^{(i)} y(0) = c_i, & i = 0, \dots, [\alpha]. \end{cases} \tag{30}$$

For convenience, let us denote y_j as approximated value of $y(t_j)$ except for $y_0 = c_0$ and let $f_j \equiv f(t_j, y_j)$, y_j^c be a corrector of y_j , y_j^p be a predictor of y_j , and $f_j^p \equiv f(t_j, y_j^p)$, $j = 1, \dots, N$. If $j = 0$ then $f_0 = f(0, c_0)$. We divide the domain Ω as follows:

$$\Phi_N := \{t_j \mid a = t_0 < \dots < t_j < \dots < t_n < t_{n+1} < \dots < t_N = b\}.$$

For simplicity, let the step size be uniform, which means $t_{j+1} - t_j = h, j = 0, 1, \dots, N - 1$. Then (30) can be rewritten at time t_{n+1} as follows:

$$y(t_{n+1}) = g(t_{n+1}) + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^n \int_{t_j}^{t_{j+1}} (t_{n+1} - \tau)^{\alpha-1} f(\tau, y(\tau)) d\tau, \tag{31}$$

where $g(t_{n+1}) = \sum_{i=0}^{[\alpha]} \frac{(t_{n+1})^i}{i!} c_i$. We interpolate $f(\tau, y(\tau))$ using linear or quadratic Lagrange polynomials over each interval $I_j = [t_j, t_{j+1}], j = 0, 1, \dots, N - 1$. Then we obtain the following predictor–corrector schemes.

1 *HPCM with linear Lagrange polynomial:*

$$\begin{aligned}
 y_{n+1}^c = g(t_{n+1}) + \frac{1}{\Gamma(\alpha)} & \left[\sum_{j=0}^{n-1} (B_{n+1}^{1j} f_j + B_{n+1}^{2j} f_{j+1}) \right. \\
 & \left. + B_{n+1}^{1,n} f_n + B_{n+1}^{2,n} f_{n+1}^P \right], \tag{32}
 \end{aligned}$$

where

$$\begin{aligned}
 B_{n+1}^{1j} &= \frac{1}{h} \int_{t_j}^{t_{j+1}} (t_{n+1} - \tau)^{\alpha-1} (t_{j+1} - \tau) d\tau, \\
 B_{n+1}^{2j} &= -\frac{1}{h} \int_{t_j}^{t_{j+1}} (t_{n+1} - \tau)^{\alpha-1} (t_j - \tau) d\tau, \\
 y_{n+1}^P &= g(t_{n+1}) + G_{\alpha,f}(t_{n+1}) + b_{n+1}^1 f_{n-1} + b_{n+1}^2 f_n, \tag{33} \\
 b_{n+1}^1 &= \frac{1}{h} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} (t_n - \tau) d\tau, \\
 b_{n+1}^2 &= -\frac{1}{h} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} (t_{n-1} - \tau) d\tau, \\
 G_{\alpha,f}(t_{n+1}) &= \frac{1}{\Gamma(\alpha)} \sum_{j=0}^{n-1} (B_{n+1}^{1j} f_j + B_{n+1}^{2j} f_{j+1}).
 \end{aligned}$$

2 *HPCM with quadratic Lagrange polynomial:*

$$\begin{aligned}
 y_{n+1}^c = g(t_{n+1}) + \frac{1}{\Gamma(\alpha)} & \left[A_{n+1}^{1,0} f_0 + A_{n+1}^{2,0} f_{1/2} + A_{n+1}^{3,0} f_1 \right. \\
 & + \sum_{j=1}^{n-1} (A_{n+1}^{1j} f_{j-1} + A_{n+1}^{2j} f_j + A_{n+1}^{3j} f_{j+1}) \\
 & \left. + A_{n+1}^{1,n} f_{n-1} + A_{n+1}^{2,n} f_n + A_{n+1}^{3,n} f_{n+1}^P \right], \tag{34}
 \end{aligned}$$

where

$$\begin{aligned}
 A_{n+1}^{1,0} &= \frac{2}{h^2} \int_{t_j}^{t_{j+1}} (t_{n+1} - \tau)^{\alpha-1} (t_{1/2} - \tau)(t_1 - \tau) d\tau, \\
 A_{n+1}^{2,0} &= -\frac{4}{h^2} \int_{t_j}^{t_{j+1}} (t_{n+1} - \tau)^{\alpha-1} (t_0 - \tau)(t_1 - \tau) d\tau, \\
 A_{n+1}^{3,0} &= \frac{2}{h^2} \int_{t_j}^{t_{j+1}} (t_{n+1} - \tau)^{\alpha-1} (t_0 - \tau)(t_{1/2} - \tau) d\tau, \\
 A_{n+1}^{1j} &= \frac{1}{2h^2} \int_{t_j}^{t_{j+1}} (t_{n+1} - \tau)^{\alpha-1} (t_j - \tau)(t_{j+1} - \tau) d\tau, \\
 A_{n+1}^{2j} &= -\frac{1}{h^2} \int_{t_j}^{t_{j+1}} (t_{n+1} - \tau)^{\alpha-1} (t_{j-1} - \tau)(t_{j+1} - \tau) d\tau,
 \end{aligned}$$

$$A_{n+1}^{3j} = \frac{1}{2h^2} \int_{t_j}^{t_{j+1}} (t_{n+1} - \tau)^{\alpha-1} (t_{j-1} - \tau)(t_j - \tau) d\tau,$$

and the predictor f_{n+1}^p is found as follows:

$$y_{n+1}^p = g(t_{n+1}) + G_{\alpha,f}(t_{n+1}) + a_{n+1}^1 f_{n-2} + a_{n+1}^2 f_{n-1} + a_{n+1}^3 f_n, \tag{35}$$

where

$$G_{\alpha,f}(t_{n+1}) = \frac{1}{\Gamma(\alpha)} \left[A_{n+1}^{1,0} f_0 + A_{n+1}^{2,0} f_{1/2} + A_{n+1}^{3,0} y(t_1) + \sum_{j=0}^{n-1} (A_{n+1}^{1,j} f_{j-1} + A_{n+1}^{2,j} f_j + A_{n+1}^{3,j} f_{j+1}) \right],$$

$$2a_{n+1}^1 = \frac{1}{2h^2} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} (t_{n-1} - \tau)(t_n - \tau) d\tau,$$

$$a_{n+1}^2 = -\frac{1}{h^2} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} (t_{n-2} - \tau)(t_n - \tau) d\tau,$$

$$a_{n+1}^3 = \frac{1}{2h^2} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} (t_{n-2} - \tau)(t_{n-1} - \tau) d\tau.$$

Remark 3.1 Using the HPCM with linear and quadratic interpolations, we find predictors $y_{n+1}^p, w_{n+1}^p, z_{n+1}^p$ individually and then find correctors $y_{n+1}^c, w_{n+1}^c, z_{n+1}^c$. The entire steps of the proposed schemes are summarized in Appendix B.

The following theorems [1] bound the global error E_{n+1} of the HPCM with linear and quadratic interpolations, respectively.

Theorem 3.1 (Theorem 2.4 in [1]: Global error of HPCM with linear interpolation) *Define E_{n+1} to be global error. Suppose $f(\cdot, y(\cdot)) \in C^2[a, b]$ and furthermore is Lipschitz continuous in the second argument, then we have*

$$E_{n+1} = |y(t_{n+1}) - \tilde{y}_{n+1}| \leq \mathcal{O}(h^2),$$

given $E_1 \leq Ch^2$.

Theorem 3.2 (Theorem 3.4 in [1]: Global error of HPCM with quadratic interpolation) *Suppose $f(\cdot, y(\cdot)) \in C^3[a, b]$ and is Lipschitz continuous in the second argument, then we have*

$$E_{n+1} \leq \mathcal{O}(h^3),$$

given $E_1, E_2 \leq \mathcal{O}(h^3)$ and $E_{1/2} \leq \mathcal{O}(h^{3-\alpha}), 0 < \alpha < 1$.

4 Numerical examples

In this section, we experimentally illustrate the performance of the proposed schemes. Numerically, we verify that our proposed schemes can deal with more complex FBVPs than

the integral discretization schemes in [20, 21]. For that purpose, the proposed schemes are implemented in FNBVPs with $0 < \alpha_1 < 1$ whose exact solutions are polynomial, exponential, and sine functions in Examples 4.1 through 4.3. We investigate absolute errors in maximum norm, convergence rates, and absolute values of the approximated error function $|\tilde{F}(s_k)|$ with various values of parameters. We discuss linear FBVPs with $\alpha_1 = 1$ whose exact solutions have low regularity and high regularity in Examples 4.4 and 4.5, respectively. We compare numerical results obtained by our proposed schemes with the integral discretization schemes. But we emphasize that our proposed methods can deal with many different FBVPs unlike the another method in Examples 4.4 and 4.5. Regarding the numerical results shown in the Appendix, let us summarize the parameters used:

- h denotes the size of time sub-interval.
- s_0 denotes the initial approximation of the sequence $\{s_k\}$ in proposed shooting methods.
- k denotes the number of iterations needed to meet a tolerance in the sequence $\{s_k\}$ generated by the proposed Newton’s method or Halley’s method.
- m denotes the maximum number of iterations in Newton’s and Halley’s methods.
- Tol denotes the tolerance used to measure the error of the approximated error function $|\tilde{F}(s_k)|$ in Newton’s method and Halley’s method.
- N denotes the number of time sub-intervals.
- y_j^c denotes the approximate solution at t_j obtained by the proposed shooting technique based on either Newton’s method or Halley’s method, with HPCM. $y(t_j)$ denotes the exact solution at t_j .
- Max. error denotes the pointwise absolute errors in the maximum norm. (i.e. $\max_{1 \leq j \leq N} |y_j^c - y(t_j)|$)
- $E_{\alpha,\beta}(t)$ denotes the two-parameter function of Mittag-Leffler type [7].

In Examples 4.1 through 4.3, we transform the FNBVP into the system of FIVPs (7) and s_0 means an initial approximation to $y'(0)$. In Examples 4.4 and 4.5, the linear FBVP is transformed into the system of FIVPs (22) with $\epsilon = 10^{-10}$ and s_0 means an initial approximation to $y(0)$. For all examples except for Example 4.4, we implement the shooting technique based on Newton’s method (Halley’s method) combined with HPCM with linear (quadratic) interpolation to verify the order of convergence $\mathcal{O}(h^2)$ ($\mathcal{O}(h^3)$), respectively.

Example 4.1 Consider the following double-term FNBVP with RBCs:

$$\begin{cases} D_0^{\alpha_2} y(t) = \frac{\Gamma(5)}{\Gamma(5-\alpha_2)} t^{4-\alpha_2} - \frac{\Gamma(5)}{\Gamma(5-\alpha_1)} t^{4-\alpha_1} - t^8 + y^2 + D_0^{\alpha_1} y(t), \\ y(0) + y'(0) = 0, \quad y(1) + y'(1) = 5, \end{cases}$$

where the exact solution is $y(t) = t^4$.

Example 4.2 Consider the following double-term FNBVP with RBCs:

$$\begin{cases} D_0^{\alpha_2} y(t) = \lambda^2 t^{2-\alpha_2} E_{1,3-\alpha_2}(\lambda t) - (\lambda^2 \frac{\Gamma(3)}{2\Gamma(3-\alpha_2)} t^{2-\alpha_2} + \lambda^3 \frac{\Gamma(4)}{6\Gamma(4-\alpha_2)} t^{3-\alpha_2}) \\ \quad - A^2 + y^2 - tB + tD_0^{\alpha_1} y(t), \\ y(0) + y'(0) = 0, \quad y(1) + y'(1) \approx 0.2699, \end{cases}$$

where

$$\begin{aligned} \lambda &= 1, \\ A &= e^{\lambda t} - \left(1 + \lambda t + \frac{\lambda^2}{2} t^2 + \frac{\lambda^3}{3!} t^3 \right), \\ B &= \lambda t^{1-\alpha_1} E_{1,2-\alpha_1}(\lambda t) \\ &\quad - \left(\lambda \frac{\Gamma(2)}{\Gamma(2-\alpha_1)} t^{1-\alpha_1} + \lambda^2 \frac{\Gamma(3)}{2\Gamma(3-\alpha_1)} t^{2-\alpha_1} + \lambda^3 \frac{\Gamma(4)}{6\Gamma(4-\alpha_1)} t^{3-\alpha_1} \right), \end{aligned}$$

and the exact solution is $y(t) = e^{\lambda t} - (1 + \lambda t + \frac{\lambda^2}{2} t^2 + \frac{\lambda^3}{3!} t^3)$.

Example 4.3 Consider the following double-term FNBVP with RBCs:

$$\begin{cases} D_0^{\alpha_2} y(t) = F_\lambda^{(\alpha_2)}(t) + \frac{\Gamma(4)}{6\Gamma(4-\alpha_2)} t^{3-\alpha_2} + (\sin(t) - t + \frac{t^3}{6})^2 - y^2 \\ \quad - F_\lambda^{(\alpha_1)}(t) + \frac{\Gamma(2)}{\Gamma(2-\alpha_1)} t^{1-\alpha_1} - \frac{\Gamma(4)}{6\Gamma(4-\alpha_1)} t^{3-\alpha_1} + D_0^{\alpha_1} y(t), \\ y(0) + y'(0) = 0, \quad y(1) + y'(1) \approx 4.84399, \end{cases}$$

where

$$\begin{aligned} F_\lambda^{(\alpha)}(t) &= -\frac{1}{2} i(i\lambda)^{([\alpha])} t^{([\alpha]-\alpha)} \\ &\quad \cdot [E_{[\alpha]-\alpha+1}(i\lambda t) - (-1)^{([\alpha])} E_{1,[\alpha]-\alpha+1}(-i\lambda t)], \end{aligned}$$

$$\lambda = 1,$$

and the exact solution is $y(t) = \sin(\lambda t) - t + \frac{t^3}{6}$.

In Examples 4.1, 4.2, and 4.3, we observe the following:

- 1 For all three examples, $w(t), z(t), f(t, y, w)$ (e.g. $D_0^{\alpha_1} y(t), y'(t), f(t, y, D_0^{\alpha_2} y(t))$) belong to $C^3[0, 1]$. By Theorems 3.1 and 3.2, thus, computed convergence profiles are estimated $\mathcal{O}(h^2)$ and $\mathcal{O}(h^3)$ for the HPCM with linear (quadratic) interpolation combined with shooting technique based on Newton's (Halley's) method, respectively.

Table 1 Errors of shooting techniques $|\tilde{F}(s_m)|$ in (41) versus the maximum number of iterations with various values of s_0 in Example 4.1. We set $h = 0.01, \alpha_1 = 0.4, \alpha_2 = 1.7$

m	Newton's method					Halley's method				
	$s_0 = 0.2$	$s_0 = 0.4$	$s_0 = 0.6$	$s_0 = 0.8$	$s_0 = 1.0$	$s_0 = 0.2$	$s_0 = 0.4$	$s_0 = 0.6$	$s_0 = 0.8$	$s_0 = 1.0$
1	0.604011	1.305541	2.105771	3.00776	4.017051	0.603191	1.304704	2.104838	3.006644	4.01565
2	0.042234	1.48E-01	0.303846	5.05E-01	0.753495	0.054329	1.90E-01	0.392297	6.59E-01	0.995211
3	0.000274	3.25E-03	0.01249	3.10E-02	0.061341	0.000604	6.84E-03	0.025715	6.31E-02	0.124161
4	7.31E-08	7.33E-07	2.17E-05	1.46E-04	0.000578	9.11E-08	8.24E-06	1.34E-04	0.000813	0.003036
5	2.28E-11	2.29E-10	6.71E-09	4.22E-08	1.25E-07	2.58E-11	2.32E-09	3.41E-08	8.56E-08	1.15E-06
6	7.11E-15	7.19E-14	2.10E-12	1.32E-11	3.91E-11	6.22E-15	6.55E-13	9.64E-12	2.42E-11	3.26E-10
7	0	0	8.88E-16	4.44E-15	1.07E-14	8.88E-16	0	2.66E-15	6.22E-15	9.15E-14
8	0	0	0	0	1.78E-15	8.88E-16	0	8.88E-16	0	8.88E-16
9	0	0	0	0	0	8.88E-16	0	8.88E-16	0	8.88E-16
10	0	0	0	0	0	8.88E-16	0	8.88E-16	0	8.88E-16

Table 2 Pointwise absolute errors in the maximum norm and computed rates of convergence versus the number of sub-intervals N with $s_0 = 0.2, 1.0$ for each in Example 4.1. We set $\alpha_1 = 0.4$, $\alpha_2 = 1.7$, and m is fixed at 10

N	Newton's method				Halley's method			
	$s_0 = 0.2$		$s_0 = 1.0$		$s_0 = 0.2$		$s_0 = 1.0$	
	Max. error	Rate	Max. error	Rate	Max. error	Rate	Max. error	Rate
10	1.17E-02	–	1.17E-02	–	1.02E-03	–	1.02E-03	–
20	5.63E-03	1.059	5.63E-03	1.059	1.83E-04	2.486	1.83E-04	2.486
40	1.78E-03	1.665	1.78E-03	1.665	3.63E-05	2.333	3.63E-05	2.332
80	4.92E-04	1.855	4.92E-04	1.854	5.47E-06	2.730	5.47E-06	2.730
160	1.29E-04	1.931	1.29E-04	1.931	7.51E-07	2.866	7.51E-07	2.866
320	3.30E-05	1.966	3.30E-05	1.966	9.88E-08	2.925	9.88E-08	2.925

- 2 Tables 1, 4, 7 show the absolute values of approximated error function (41) at s_m (i.e. $|a_2 y_N^c(s_m) + b_2 z_N^c(s_m) - \gamma_2|$) versus the maximum number of iterations m with various initial values s_0 . $y_N^c(s_m)$ and $z_N^c(s_m)$ are computed by using proposed schemes. We set $\alpha_1 = 0.4$, $\alpha_2 = 1.7$, $h = 0.01$ in all tables. From numerical results in those tables, we can verify that the sequence $\{s_k\}$ obtained by the proposed shooting algorithms approaches to the IC s within the error at least 10^{-16} when m is at most 10 with $s_0 = 0.2, 0.4, 0.6, 0.8, 1.0$ for each. This leads us to the conclusion that the proposed shooting techniques show a good performance with remarkable accuracy regarding to approximation of the IC s .
- 3 Tables 2, 5, 8 show pointwise absolute errors in the maximum norm and convergence rates computed versus the number of sub-intervals N in the cases of $s_0 = 0.2, 1.0$ for each of Newton's and Halley's method. We set $\alpha_1 = 0.4$, $\alpha_2 = 1.7$. The sequence $\{s_k\}$ was computed up to s_{10} so, based on the observation of Tables 1, 4, 7, we see that the error of the approximated error function $|\tilde{F}(s_k)|$ does not have an effect on the convergence rate of y_j^c obtained by HPCMs. In Tables 2, 5, 8, we can see that computed convergence profiles obtained by the proposed schemes approach 2 for Newton's method and 3 for Halley's method as N is increased. Thus numerical results shown in those tables support that the proposed methods follow global error estimates of HPCMs.
- 4 Proposed methods are tested for a variety of values of α_1, α_2 and numerical results for each pair of (α_1, α_2) are shown in Tables 3, 6, 9. For each pair of fractional orders (α_1, α_2) pointwise absolute errors in the maximum norm, computed convergence rates, CPU time executed in seconds, and number of iterations k such that $|\tilde{F}(s_k)| < Tol$ versus the number of sub-intervals N are listed in the tables. The initial approximation to s was set $s_0 = 0.2$ in all three tables. In order to minimize the number of iterations k , the tolerance was set $Tol = 10^{-5}$ for Newton's method and $Tol = 10^{-10}$ for Halley's method in Table 3, $Tol = 10^{-10}$ for both shooting techniques in Table 6, and $Tol = 10^{-15}$ for Newton's method and $Tol = 10^{-16}$ for Halley's method in Table 9. Numerical results shown in the tables demonstrate that, for all suggested pairs of fractional orders, rates of convergence approach 2 for Newton's method, 3 for Halley's method that are theoretical convergence rates of HPCMs. In Tables 3 and 9, we observe that the number of iterations k required to meet the tolerance at $(0.9, 1.1)$ is relatively greater than other pairs of fractional orders for both Newton's and Halley's method.

Table 4 Errors of shooting techniques $|\tilde{F}(s_m)|$ in (41) versus the maximum number of iterations m for each s_0 in Example 4.2 as we set $\alpha_1 = 0.4, \alpha_2 = 1.7, h = 0.01$

m	Newton's method					Halley's method				
	$s_0 = 0.2$	$s_0 = 0.4$	$s_0 = 0.6$	$s_0 = 0.8$	$s_0 = 1.0$	$s_0 = 0.2$	$s_0 = 0.4$	$s_0 = 0.6$	$s_0 = 0.8$	$s_0 = 1.0$
1	0.414113	0.890547	1.42639	2.02066	2.674035	0.414046	0.890413	1.426145	2.020256	2.673415
2	0.027748	9.51E-02	0.190648	3.11E-01	0.456212	0.03142	1.08E-01	0.219523	3.66E-01	0.549419
3	0.000177	1.93E-03	0.007068	1.70E-02	0.032652	0.000258	2.80E-03	0.010391	2.54E-02	0.050225
4	1.51E-08	9.56E-07	1.20E-05	6.71E-05	0.000243	1.61E-08	2.10E-06	2.89E-05	1.70E-04	0.000646
5	6.59E-13	4.20E-11	5.57E-10	3.99E-09	2.45E-08	1.25E-13	1.52E-11	1.33E-12	6.56E-09	1.08E-07
6	0	1.89E-15	2.44E-14	1.75E-13	1.07E-12	5.55E-17	1.67E-16	1.11E-16	5.11E-14	8.40E-13
7	0	0	0	0	1.67E-16	5.55E-17	5.55E-17	5.55E-17	5.55E-17	5.55E-17
8	0	0	0	0	5.55E-17	5.55E-17	5.55E-17	5.55E-17	5.55E-17	0
9	0	0	0	0	0	5.55E-17	5.55E-17	5.55E-17	5.55E-17	0
10	0	0	0	0	0	5.55E-17	5.55E-17	5.55E-17	5.55E-17	0

Table 5 Pointwise absolute errors in the maximum norm and computed rates of convergence versus the number of sub-intervals N in Example 4.2 as we set $\alpha_1 = 0.4, \alpha_2 = 1.7, h = 0.01, m = 10$ and no Tol

N	Newton's method				Halley's method			
	$s_0 = 0.2$		$s_0 = 1.0$		$s_0 = 0.2$		$s_0 = 1.0$	
	Max. error	Rate	Max. error	Rate	Max. error	Rate	Max. error	Rate
10	1.10E-03	-	1.10E-03	-	5.65E-05	-	5.65E-05	-
20	4.05E-04	1.434	4.05E-04	1.434	1.71E-05	1.725	1.71E-05	1.725
40	1.19E-04	1.765	1.19E-04	1.765	2.82E-06	2.599	2.82E-06	2.599
80	3.21E-05	1.893	3.21E-05	1.893	3.99E-07	2.824	3.99E-07	2.824
160	8.32E-06	1.949	8.32E-06	1.949	5.29E-08	2.913	5.29E-08	2.913
320	2.12E-06	1.974	2.12E-06	1.974	6.84E-09	2.952	6.84E-09	2.952

5 Figures 1, 2, and 3 illustrate convergence profiles obtained by the proposed methods with the variety of fractional orders. From the figures, we can see that computed rates of convergence are nearly $\mathcal{O}(h^2)$ and $\mathcal{O}(h^3)$ for the proposed shooting method with Newton's and Halley's schemes, respectively. Figures 4 and 5 display graphs of pointwise absolute errors versus the time step h . The approximate solution is computed by the proposed method with Newton's method (in Fig. 4) and Halley's method (in Fig. 5), respectively.

Example 4.4 Consider the following single-term linear FBVP with RBCs:

$$\begin{cases} D_0^{\alpha_2} y(t) = \varphi(t) - (2t + 6)y'(t) \\ y(0) - \frac{1}{1-\alpha_2} y'(0) = \gamma_1, \quad y(1) + y'(1) = \gamma_2, \end{cases}$$

where $1 < \alpha_2 < 2$,

$$\begin{aligned} \varphi(t) = & \frac{\Gamma(\alpha_2 + 1)}{\Gamma(1)} + \frac{\Gamma(2\alpha_2)}{\Gamma(\alpha_2)} t^{(\alpha_2-1)} + 4 \frac{\Gamma(4)}{\Gamma(4-\alpha_2)} t^{(3-\alpha_2)} + \frac{\Gamma(5)}{\Gamma(5-\alpha_2)} t^{(4-\alpha_2)} \\ & + (2t + 6)(\alpha_2 t^{\alpha_2-1} + (2\alpha_2 - 1)t^{2\alpha_2-2} + 3 + 12t^2 + 4t^3), \end{aligned}$$

and the exact solution is $y(t) = t^{\alpha_2} + t^{2\alpha_2-1} + 1 + 3t + 4t^3 + t^4$ [21].

Since $D_0^{\alpha_1} y(t), y'(t), D_0^{\alpha_2} y(t)$ do not belong to $C^3[0, 1]$, global error estimates of HPCMs in Theorems 3.1 and 3.2 cannot be applied to Example 4.4. Alternatively, we adopt the linear explicit method described in Appendix A with proposed shooting techniques. In this

Table 6 Pointwise absolute errors in the maximum norm, computed rates of convergence obtained by the proposed schemes, number of iterations k , and CPU time in second executed with various fractional orders versus the number of sub-intervals N in Example 4.2 as we set $s_0 = 0.2$, $Tol = 10^{-10}$ for both shooting techniques

N	Newton's method (α_1, α_2)					Halley's method (α_1, α_2)					
	(0.9, 1.1)	(0.7, 1.3)	(0.5, 1.5)	(0.3, 1.7)	(0.1, 1.9)	(0.9, 1.1)	(0.7, 1.3)	(0.5, 1.5)	(0.3, 1.7)	(0.1, 1.9)	
64	Max. error	1.25E-04	1.10E-05	4.15E-05	4.97E-05	5.10E-05	3.63E-06	9.39E-08	6.56E-07	7.63E-07	6.59E-07
	Rate	-	-	-	-	-	-	-	-	-	-
	k	5	5	5	5	5	6	5	5	5	5
128	CPU time (sec)	0.930	0.871	0.879	0.888	0.892	2.233	1.320	4.985	1.342	1.331
	Max. error	2.71E-05	5.55E-06	1.14E-05	1.29E-05	1.32E-05	4.00E-07	3.39E-08	9.36E-08	1.02E-07	8.86E-08
	Rate	2.201	9.891	1.861	1.942	1.955	3.181	1.471	2.808	2.899	2.896
256	k	5	5	5	5	5	6	5	5	5	5
	CPU time (sec)	3.347	3.335	3.332	3.406	3.411	6.076	5.011	19.757	5.046	5.075
	Max. error	5.80E-06	1.89E-06	3.00E-06	3.30E-06	3.35E-06	4.36E-08	6.32E-09	1.25E-08	1.33E-08	1.16E-08
512	Rate	2.226	1.555	1.928	1.970	1.974	3.197	2.421	2.905	2.944	2.935
	k	5	5	5	5	5	6	5	5	5	5
	CPU time (sec)	13.195	13.212	13.037	13.417	13.467	23.836	19.863	77.831	19.576	19.782
1024	Max. error	1.22E-06	5.61E-07	7.71E-07	8.34E-07	8.47E-07	4.72E-09	1.04E-09	1.62E-09	1.70E-09	1.49E-09
	Rate	2.249	1.752	1.962	1.984	1.983	3.209	2.604	2.950	2.966	2.955
	k	5	5	5	5	5	6	5	5	5	5
2048	CPU time (sec)	51.750	52.685	51.734	53.075	55.703	94.335	78.871	77.349	78.020	78.308
	Max. error	2.52E-07	1.56E-07	1.96E-07	2.10E-07	2.14E-07	5.06E-10	1.52E-10	2.06E-10	2.16E-10	1.91E-10
	Rate	2.273	1.848	1.979	1.991	1.988	3.221	2.779	2.973	2.978	2.965
2048	k	5	5	5	5	5	6	5	5	5	5
	CPU time (sec)	204.322	203.996	206.051	209.157	210.729	372.048	310.732	306.249	305.918	309.241
	Max. error	5.12E-08	4.17E-08	4.93E-08	5.27E-08	5.37E-08	5.39E-11	2.12E-11	2.58E-11	2.75E-11	2.44E-11
2048	Rate	2.301	1.902	1.988	1.994	1.991	3.232	2.839	2.999	2.973	2.972
	k	5	5	5	5	5	6	5	5	5	5
	CPU time (sec)	818.055	806.967	823.096	830.832	836.200	1452.813	1231.030	1223.399	1224.377	1227.764

Table 7 Errors of shooting techniques $|\tilde{F}(s_m)|$ in (41) versus the maximum number of iterations m with various initial approximations s_0 in Example 4.3 as we set $\alpha_1 = 0.4, \alpha_2 = 1.7, h = 0.01$

m	Newton's method					Halley's method				
	$s_0 = 0.2$	$s_0 = 0.4$	$s_0 = 0.6$	$s_0 = 0.8$	$s_0 = 1.0$	$s_0 = 0.2$	$s_0 = 0.4$	$s_0 = 0.6$	$s_0 = 0.8$	$s_0 = 1.0$
1	0.487617	0.891569	1.202993	1.407814	1.485109	0.487722	0.891834	1.203474	1.408578	1.486244
2	0.101684	3.44E-01	0.653431	9.74E-01	1.258505	0.106265	3.56E-01	0.668984	9.90E-01	1.270771
3	0.004491	5.07E-02	0.182979	4.14E-01	0.72559	0.005185	5.68E-02	0.199315	4.40E-01	0.757881
4	1.26E-05	1.14E-03	0.014409	0.073158	0.226095	1.69E-05	1.51E-03	1.80E-02	8.68E-02	0.255888
5	1.13E-08	1.57E-06	0.000101	0.002343	0.021949	1.54E-08	2.39E-06	1.63E-04	3.48E-03	2.96E-02
6	1.01E-11	1.41E-09	9.49E-08	4.43E-06	0.000225	1.39E-11	2.16E-09	1.59E-07	8.62E-06	4.22E-04
7	9.06E-15	1.26E-12	8.48E-11	3.97E-09	2.22E-07	1.25E-14	1.95E-12	1.44E-10	7.82E-09	4.62E-07
8	6.94E-18	1.12E-15	7.58E-14	3.55E-12	1.99E-10	6.94E-18	1.76E-15	1.30E-13	7.06E-12	4.17E-10
9	0	1.39E-17	6.94E-17	3.16E-15	1.78E-13	6.94E-18	6.94E-18	1.25E-16	6.37E-15	3.77E-13
10	0	6.94E-18	6.94E-18	6.94E-18	1.67E-16	6.94E-18	0	6.94E-18	6.94E-18	3.33E-16

Table 8 Pointwise absolute errors in the maximum norm and computed rates of convergence versus the number of sub-intervals N in Example 4.3 as we set $\alpha_1 = 0.4, \alpha_2 = 1.7, m = 10$

N	Newton's method				Halley's method			
	$s_0 = 0.2$		$s_0 = 1.0$		$s_0 = 0.2$		$s_0 = 1.0$	
	Max. error	Rate	Max. error	Rate	Max. error	Rate	Max. error	Rate
10	1.73E-04	-	1.73E-04	-	1.19E-05	-	1.19E-05	-
20	6.85E-05	1.338	6.85E-05	1.338	3.97E-06	1.578	3.97E-06	1.578
40	2.07E-05	1.727	2.07E-05	1.727	6.62E-07	2.586	6.62E-07	2.586
80	5.63E-06	1.877	5.63E-06	1.877	9.28E-08	2.834	9.28E-08	2.834
160	1.47E-06	1.942	1.47E-06	1.942	1.22E-08	2.926	1.22E-08	2.926
320	3.74E-07	1.971	3.74E-07	1.971	1.56E-09	2.966	1.56E-09	2.966

example, we compare the accuracy and convergence rate of the approximate solution obtained by the proposed shooting technique based on Newton's method with the modified integral discretization scheme [21] for each $\alpha_2 = 1.1, 1.3, 1.5, 1.7, 1.9$. In Table 10, we can observe that our proposed method consumes less CPU time than the modified integral discretization scheme [21] even though both methods shows the equal performance.

Example 4.5 Consider the following single-term linear FBVP with RBCs:

$$\begin{cases} D_0^{\alpha_2} y(t) = F(t) - \cos(t)y(t) - \sin(t)y'(t) \\ y(0) - \frac{1}{1-\alpha_2} y'(0) = \gamma_1, \quad y(1) + y'(1) = \gamma_2 \end{cases}$$

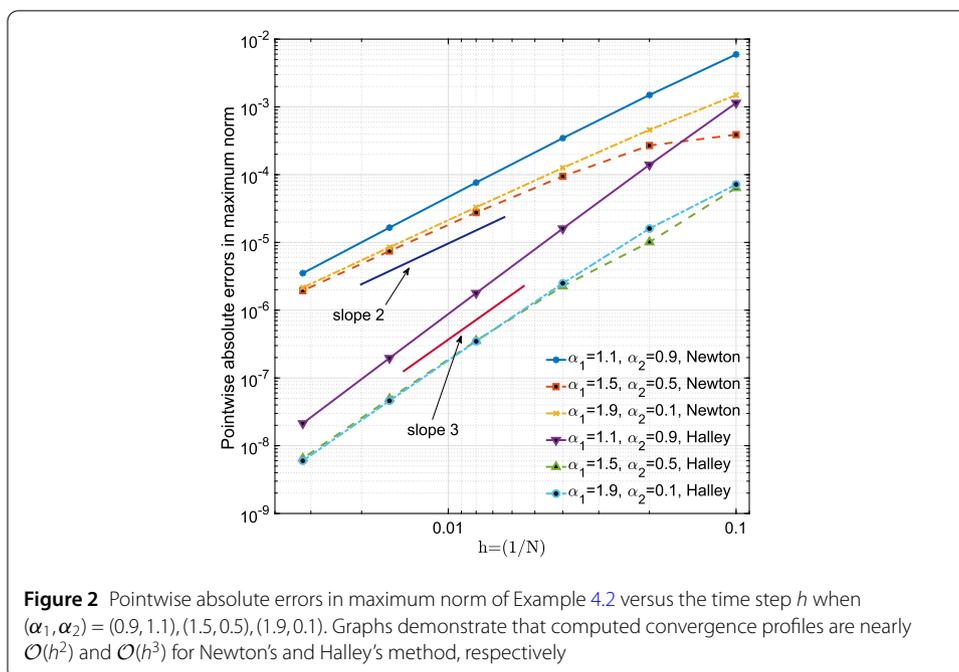
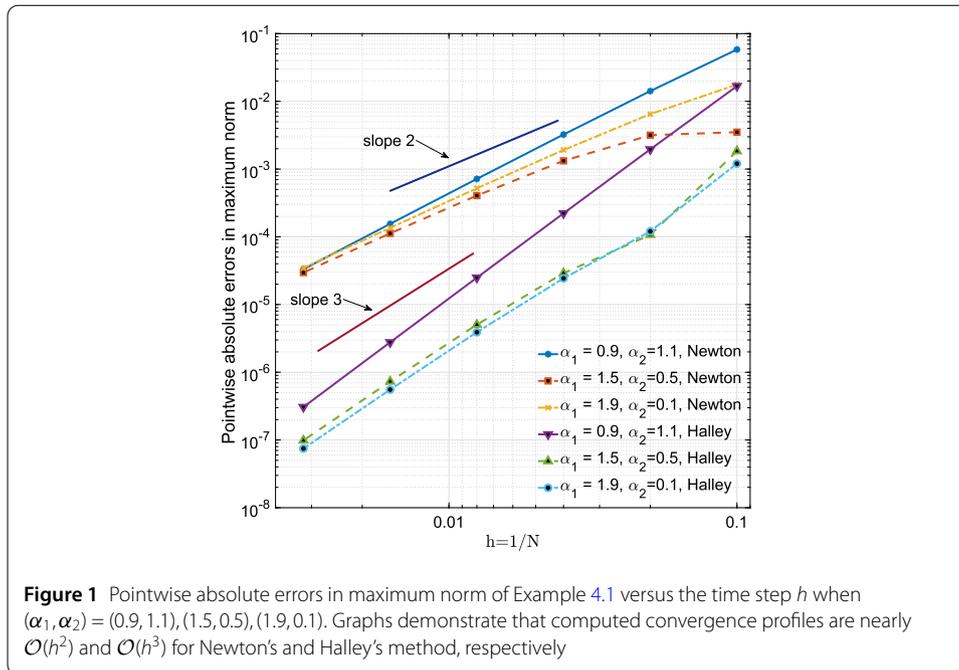
whose $1 < \alpha < 2$ and the exact solution is

$$y(t) = \sin(\lambda t) - t + \frac{t^3}{6}.$$

In Example 4.5, we compare the performance of our proposed methods with the modified integral discretization scheme [21]. Table 11 shows pointwise absolute errors and computed convergence profiles versus the number of sub-intervals for each $\alpha_2 = 1.1, 1.3, 1.5, 1.7, 1.9$. In Table 11, we can see that, for all values of α_2 , the computed rates of convergence obtained by the proposed shooting technique based on Halley's method combined with third-order HPCM are around 3.0 while the computed rates of convergence obtained by the modified integral discretization scheme [21] are around 2.0.

Table 9 Pointwise absolute errors in the maximum norm, computed rates of convergence obtained by the proposed schemes, number of iterations k , and CPU time in second executed with various fractional orders versus the number of sub-intervals in Example 4.3. We set $s_0 = 0.2$ for both shooting techniques but $Tol = 10^{-15} (10^{-16})$ for Newton's method (Halley's method), respectively

N	Newton's method										Halley's method									
	(α_1, α_2)					(α_1, α_2)					(α_1, α_2)					(α_1, α_2)				
	(0.9, 1.1)	(0.7, 1.3)	(0.5, 1.5)	(0.3, 1.7)	(0.1, 1.9)	(0.9, 1.1)	(0.7, 1.3)	(0.5, 1.5)	(0.3, 1.7)	(0.1, 1.9)	(0.9, 1.1)	(0.7, 1.3)	(0.5, 1.5)	(0.3, 1.7)	(0.1, 1.9)	(0.9, 1.1)	(0.7, 1.3)	(0.5, 1.5)	(0.3, 1.7)	(0.1, 1.9)
64	Max. error	2.26E-05	6.92E-07	6.70E-06	8.99E-06	9.95E-06	9.95E-06	8.99E-06	8.99E-06	9.95E-06	7.65E-07	5.55E-08	1.33E-07	1.84E-07	1.95E-07	7.65E-07	5.55E-08	1.33E-07	1.84E-07	1.95E-07
	Rate	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	k	20	10	9	8	6	6	8	8	6	22	10	9	8	22	10	9	8	7	7
	CPU time (sec)	3.969	1.227	1.427	1.230	0.941	0.941	1.230	1.230	0.941	5.149	2.296	2.055	1.896	5.149	2.296	2.055	1.896	1.680	1.680
128	Max. error	5.04E-06	6.51E-07	1.88E-06	2.35E-06	2.57E-06	2.57E-06	2.35E-06	2.35E-06	2.57E-06	8.54E-08	2.93E-09	1.95E-08	2.43E-08	2.55E-08	8.54E-08	2.93E-09	1.95E-08	2.43E-08	2.55E-08
	Rate	2.166	8.900	1.833	1.937	1.954	1.954	1.937	1.937	1.954	3.164	4.244	2.767	2.920	3.164	4.244	2.767	2.920	2.933	2.933
	k	20	10	9	8	7	7	8	8	7	22	11	9	8	22	11	9	8	7	7
	CPU time (sec)	3.969	4.636	5.262	4.543	4.066	4.066	4.543	4.543	4.066	18.971	9.424	7.705	6.906	18.971	9.424	7.705	6.906	6.112	6.112
256	Max. error	1.10E-06	2.50E-07	4.99E-07	6.00E-07	6.54E-07	6.54E-07	6.00E-07	6.00E-07	6.54E-07	9.41E-09	8.67E-10	2.64E-09	3.11E-09	3.27E-09	9.41E-09	8.67E-10	2.64E-09	3.11E-09	3.27E-09
	Rate	2.192	1.381	1.914	1.968	1.974	1.974	1.968	1.968	1.974	3.181	1.754	2.888	2.962	3.181	1.754	2.888	2.962	2.964	2.964
	k	20	10	9	8	7	7	8	8	7	22	11	9	8	22	11	9	8	7	7
	CPU time (sec)	15.179	17.853	20.053	17.621	15.541	15.541	17.621	17.621	15.541	73.376	33.136	29.642	26.633	73.376	33.136	29.642	26.633	23.345	23.345
512	Max. error	2.38E-07	7.81E-08	1.29E-07	1.52E-07	1.65E-07	1.65E-07	1.52E-07	1.52E-07	1.65E-07	1.03E-09	1.72E-10	3.43E-10	3.94E-10	4.14E-10	1.03E-09	1.72E-10	3.43E-10	3.94E-10	4.14E-10
	Rate	2.211	1.679	1.955	1.983	1.984	1.984	1.983	1.983	1.984	3.194	2.331	2.944	2.981	3.194	2.331	2.944	2.981	2.979	2.979
	k	20	10	9	8	7	7	8	8	7	22	10	9	8	22	10	9	8	7	7
	CPU time (sec)	60.494	69.994	80.021	69.867	62.629	62.629	69.867	69.867	62.629	289.130	131.007	116.497	106.946	289.130	131.007	116.497	106.946	92.665	92.665
1024	Max. error	5.08E-08	2.23E-08	3.27E-08	3.82E-08	4.17E-08	4.17E-08	3.82E-08	3.82E-08	4.17E-08	1.12E-10	2.71E-11	4.37E-11	4.96E-11	5.23E-11	1.12E-10	2.71E-11	4.37E-11	4.96E-11	5.23E-11
	Rate	2.228	1.810	1.975	1.990	1.989	1.989	1.990	1.990	1.989	3.205	2.669	2.971	2.990	3.205	2.669	2.971	2.990	2.986	2.986
	k	20	10	9	8	7	7	8	8	7	22	10	9	8	22	10	9	8	7	7
	CPU time (sec)	239.810	275.430	313.605	276.502	244.386	244.386	276.502	276.502	244.386	1132.549	515.751	461.774	414.636	1132.549	515.751	461.774	414.636	363.701	363.701
2048	Max. error	1.07E-08	6.05E-09	8.26E-09	9.59E-09	1.05E-08	1.05E-08	9.59E-09	9.59E-09	1.05E-08	1.20E-11	3.90E-12	5.50E-12	6.25E-12	6.58E-12	1.20E-11	3.90E-12	5.50E-12	6.25E-12	6.58E-12
	Rate	2.246	1.880	1.986	1.994	1.991	1.991	1.994	1.994	1.991	3.215	2.798	2.990	2.990	3.215	2.798	2.990	2.990	2.991	2.991
	k	20	10	9	8	7	7	8	8	7	22	11	9	8	22	11	9	8	7	7
	CPU time (sec)	961.302	1099.115	1235.384	1096.546	963.071	963.071	1096.546	1096.546	963.071	4517.411	2247.280	1837.934	1634.163	4517.411	2247.280	1837.934	1634.163	1436.284	1436.284



The algorithm of the proposed shooting techniques with second-order HPCM requires less than the number of arithmetic operations needed by the modified integral discretization scheme to solve a FBVP with RBCs than the modified integral discretization scheme [21] because the predictor and corrector in HPCMs share the computation of the memory effect. As a result the proposed shooting technique based on Newton's method consumes less CPU than the modified discretization scheme [21] and the CPU time executed by the proposed shooting technique based on Halley's method is approximately equal to the CPU time executed by the modified integral discretization scheme [21], as shown in Table 11.

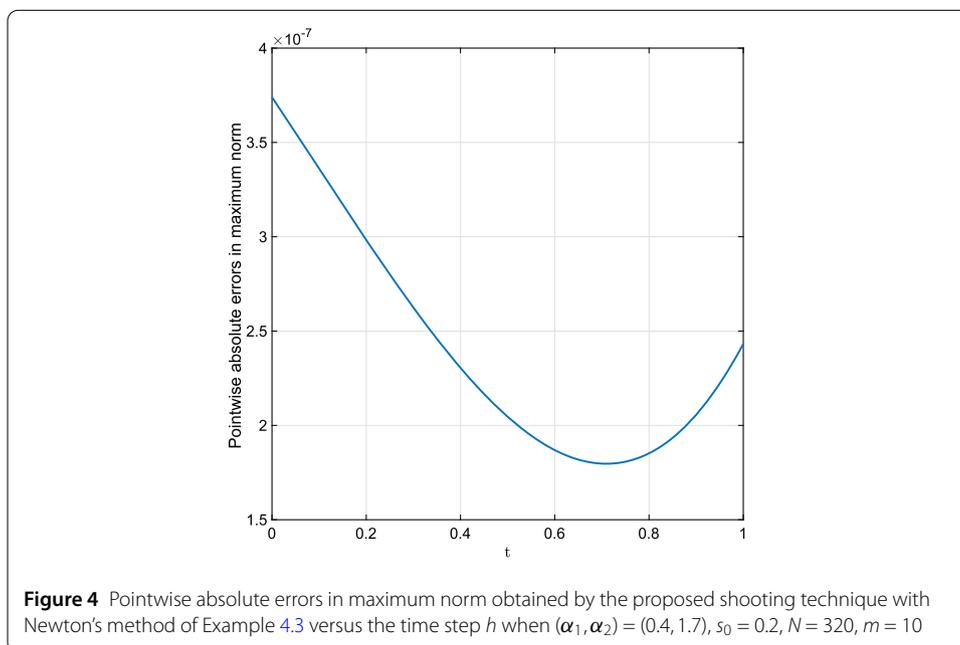
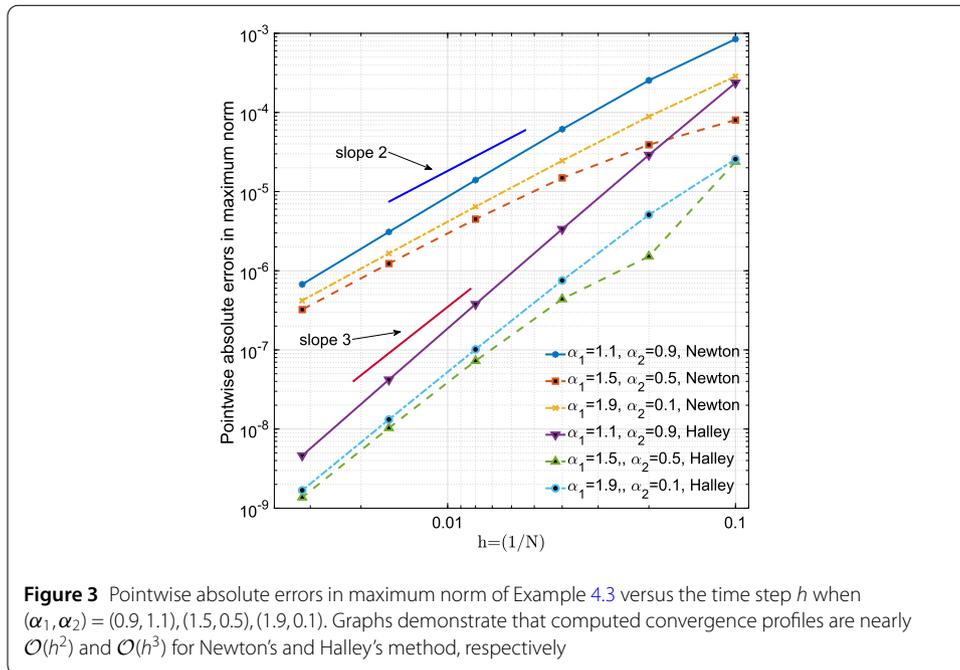
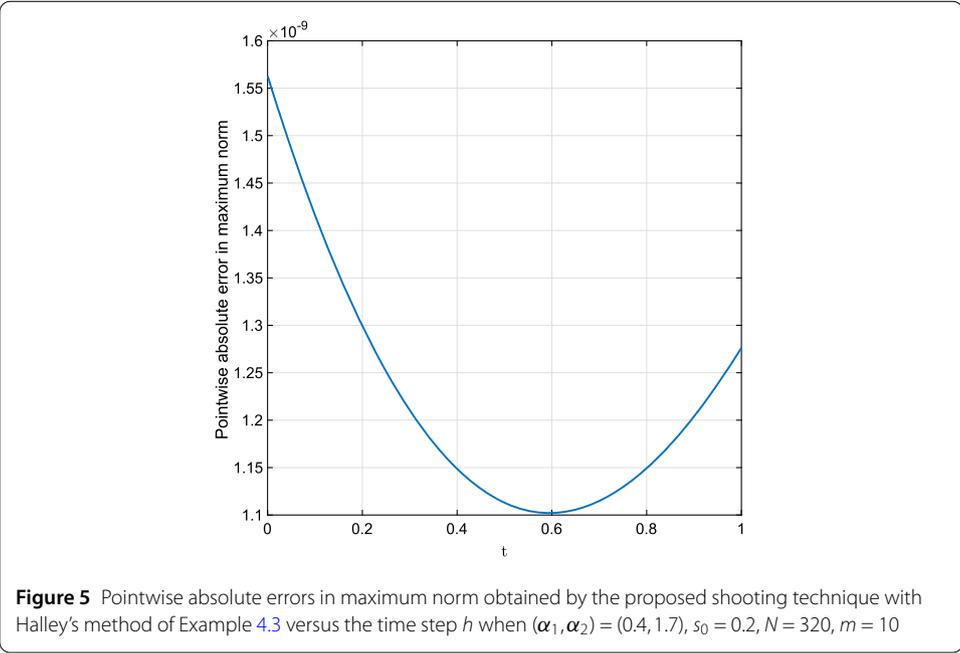


Figure 6 illustrates convergence profiles obtained by the proposed methods and the modified integral discretization scheme [21] with the variety of fractional orders. From the graphs, we can see that computed rates of convergence are nearly $\mathcal{O}(h^2)$ for the proposed shooting method with Newton's technique and the modified integral discretization scheme [21], $\mathcal{O}(h^3)$ for the proposed shooting method with Halley's scheme, respectively. Plots exhibited in Figs. 7, 8, 9 display pointwise absolute errors in maximum norm versus the time step h and the approximate solution is computed by the modified integral discretization scheme [21] (in Fig. 7), the proposed shooting technique with Newton's



method (in Fig. 8), the proposed shooting technique with Halley’s method (in Fig. 9), respectively.

Example 4.6 Consider the following single-term linear FBVP with RBCs:

$$\begin{cases} D_0^{\alpha_2} y(t) + 2y'(t) + 3y(t) = 1.25, \\ y(0) - \frac{1}{\alpha_2 - 1} y'(0) = 0.4, \quad y(1) = 1.7, \end{cases}$$

where $1 < \alpha_2 < 2$.

The exact solution of Example 4.6 is unknown so we alter the measure to estimate the errors employing the uniform two-mesh differences and orders of convergence introduced in [22]:

$$e_N = \max_{1 \leq j \leq N} |y_j^c - y_{2j}^c|,$$

where y_{2j}^c denotes the approximate solution at t_{2j} by the proposed methods with $h = 1/2N$. Using the errors e_N , we estimate the convergence profiles as follows:

$$r_N = \log_2 \left(\frac{e_N}{e_{2N}} \right).$$

In this example, we compare performances of our proposed shooting methods with the finite difference method proposed in [22] and the numerical results are shown in Table 12.

We observe the following from the results:

- 1 In Table 12, uniform two-mesh difference errors and convergence rates computed by
 - (a) the finite difference method [22],

Table 10 Pointwise absolute errors, computed rates of convergence, CPU time in second executed by the modified integral discretization scheme [21] and Newton's method. We set $s_0 = 0.2, Tol = 10^{-6}$

N	$\alpha_2 = 1.1$			$\alpha_2 = 1.3$			$\alpha_2 = 1.5$			$\alpha_2 = 1.7$			$\alpha_2 = 1.9$		
	Max. error	Rate	Time												
Modified integral discretization scheme [21]															
64	9.56E-03	-	0.028	5.14E-03	-	0.023	3.91E-03	-	0.102	3.76E-03	-	0.024	4.07E-03	-	0.023
128	3.89E-03	1.298	0.048	1.60E-03	1.686	0.048	1.05E-03	1.895	0.038	9.44E-04	1.995	0.049	1.02E-03	1.996	0.049
256	1.65E-03	1.236	0.162	5.20E-04	1.618	0.163	2.87E-04	1.872	0.100	2.37E-04	1.997	0.163	2.55E-04	1.998	0.163
512	7.22E-04	1.193	0.566	1.78E-04	1.548	0.578	8.01E-05	1.840	0.328	5.92E-05	1.998	0.583	6.39E-05	1.999	0.586
1024	3.22E-04	1.165	1.220	6.36E-05	1.484	1.221	2.30E-05	1.803	0.738	1.51E-05	1.971	1.239	1.60E-05	1.999	1.228
2048	1.45E-04	1.148	3.304	2.36E-05	1.432	3.338	6.77E-06	1.762	2.735	3.90E-06	1.955	3.363	4.00E-06	2.000	3.364
Newton's method															
64	4.42E-03	-	0.026	1.31E-03	-	0.012	1.14E-03	-	0.152	1.27E-03	-	0.012	1.42E-03	-	0.012
128	1.89E-03	1.225	0.012	3.83E-04	1.775	0.012	2.95E-04	1.955	0.013	3.17E-04	1.999	0.011	3.55E-04	2.000	0.011
256	8.10E-04	1.224	0.036	1.27E-04	1.588	0.023	7.62E-05	1.955	0.031	7.92E-05	2.002	0.023	8.87E-05	2.000	0.022
512	3.48E-04	1.219	0.047	6.31E-05	1.012	0.045	1.97E-05	1.949	0.049	1.98E-05	2.004	0.045	2.22E-05	2.000	0.045
1024	1.50E-04	1.213	0.109	2.88E-05	1.130	0.103	5.15E-06	1.938	0.113	4.92E-06	2.006	0.150	5.54E-06	2.000	0.103
2048	6.91E-05	1.120	0.268	1.26E-05	1.191	0.263	1.36E-06	1.922	0.287	1.22E-06	2.008	0.307	1.38E-06	2.001	0.281

Table 11 Pointwise absolute errors and computed rates of convergence obtained by the modified integral discretization scheme [21] and proposed methods; Newton's method, Halley's method. We set $s_0 = 0.2, Tol = 10^{-5}$

N	$\alpha_2 = 1.1$			$\alpha_2 = 1.3$			$\alpha_2 = 1.5$			$\alpha_2 = 1.7$			$\alpha_2 = 1.9$		
	Max. error	Rate	Time												
Modified integral discretization scheme [21]															
10	1.93E-03	-	0.02	1.67E-03	-	0.02	1.43E-03	-	0.02	1.29E-03	-	0.02	1.21E-03	-	0.02
20	5.48E-04	1.816	0.02	4.68E-04	1.831	0.02	3.97E-04	1.847	0.02	3.54E-04	1.862	0.02	3.32E-04	1.870	0.02
40	1.46E-04	1.907	0.09	1.24E-04	1.915	0.09	1.04E-04	1.925	0.09	9.28E-05	1.932	0.09	8.68E-05	1.937	0.1
80	3.78E-05	1.948	0.34	3.21E-05	1.954	0.34	2.68E-05	1.962	0.35	2.38E-05	1.966	0.34	2.22E-05	1.969	0.34
160	9.67E-06	1.968	1.35	8.16E-06	1.974	1.45	6.80E-06	1.980	1.34	6.01E-06	1.983	1.36	5.60E-06	1.984	1.35
320	2.45E-06	1.979	5.43	2.06E-06	1.984	5.44	1.71E-06	1.990	5.33	1.51E-06	1.991	5.32	1.41E-06	1.992	5.39
Newton's method															
10	7.27E-03	-	0.01	1.97E-03	-	0.02	9.45E-04	-	0.01	6.80E-04	-	0.02	5.80E-04	-	0.19
20	1.78E-03	2.033	0.02	3.92E-04	2.329	0.02	1.97E-04	2.263	0.02	1.56E-04	2.124	0.02	1.40E-04	2.047	0.02
40	3.92E-04	2.180	0.07	7.80E-05	2.328	0.07	4.38E-05	2.170	0.06	3.75E-05	2.055	0.07	3.47E-05	2.016	0.08
80	8.38E-05	2.226	0.24	1.62E-05	2.272	0.24	1.03E-05	2.093	0.23	9.25E-06	2.021	0.24	8.64E-06	2.005	0.24
160	1.78E-05	2.236	0.93	3.50E-06	2.205	0.93	2.49E-06	2.046	0.89	2.30E-06	2.007	0.94	2.16E-06	2.001	0.93
320	3.79E-06	2.232	3.58	7.93E-07	2.144	3.57	6.12E-07	2.021	3.55	5.74E-07	2.002	3.62	5.39E-07	2.000	3.56
Halley's method															
10	1.40E-03	-	0.02	2.79E-04	-	0.02	1.08E-04	-	0.02	6.95E-05	-	0.02	5.51E-05	-	0.02
20	1.55E-04	3.169	0.03	2.57E-05	3.441	0.03	1.06E-05	3.339	0.03	7.70E-06	3.174	0.03	6.56E-06	3.070	0.03
40	1.65E-05	3.237	0.11	2.44E-06	3.397	0.11	1.15E-06	3.211	0.11	9.19E-07	3.066	0.11	8.12E-07	3.014	0.11
80	1.73E-06	3.253	0.37	2.43E-07	3.325	0.40	1.33E-07	3.111	0.37	1.13E-07	3.023	0.37	1.02E-07	3.000	0.37
160	1.82E-07	3.252	1.40	2.56E-08	3.248	1.42	1.60E-08	3.055	1.38	1.41E-08	3.007	1.41	1.27E-08	2.998	1.40
320	1.92E-08	3.244	5.47	2.82E-09	3.180	5.44	1.96E-09	3.026	5.71	1.76E-09	3.002	5.44	1.59E-09	2.999	5.45

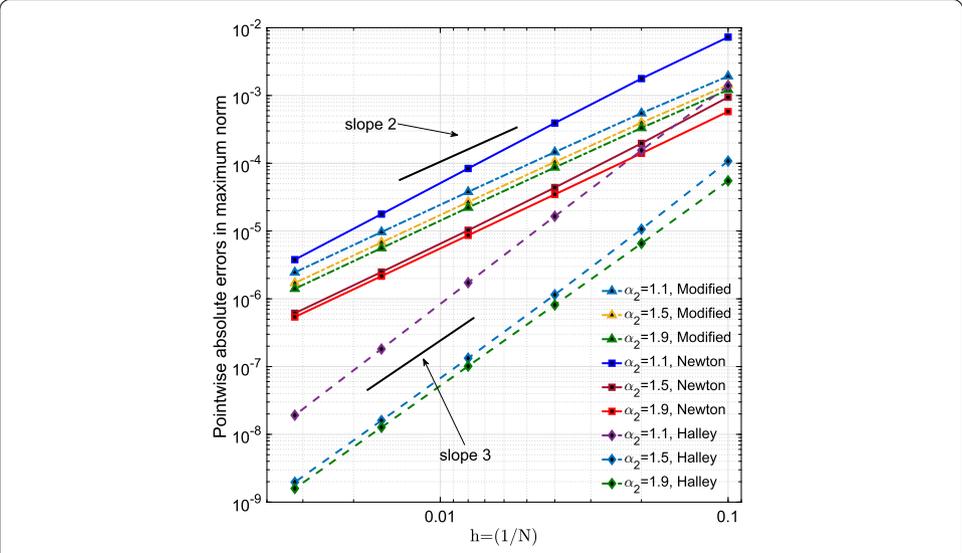


Figure 6 Pointwise absolute errors in maximum norm of Example 4.5 versus the time step h when $\alpha_2 = 1.1, 1.5, 1.9$. Graphs demonstrate that computed convergence profiles are nearly $\mathcal{O}(h^2)$ for Newton's and modified integral discretization scheme [21], $\mathcal{O}(h^3)$ for Halley's method, respectively

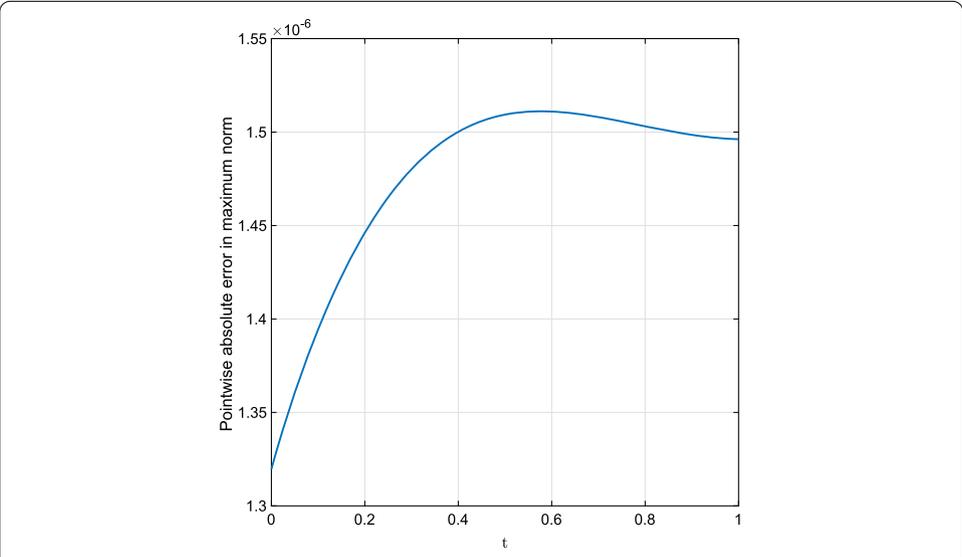
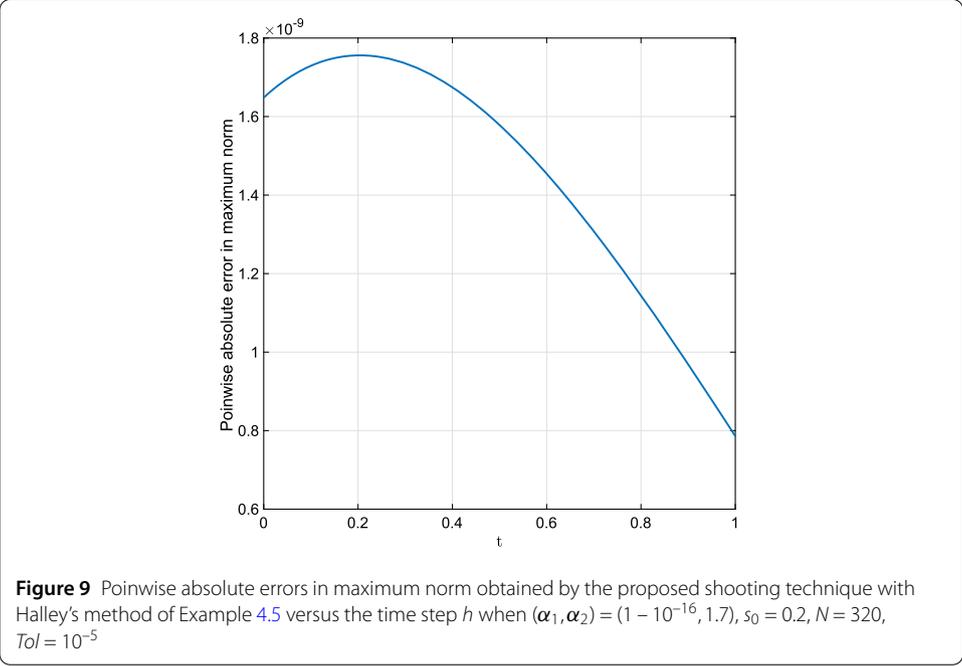
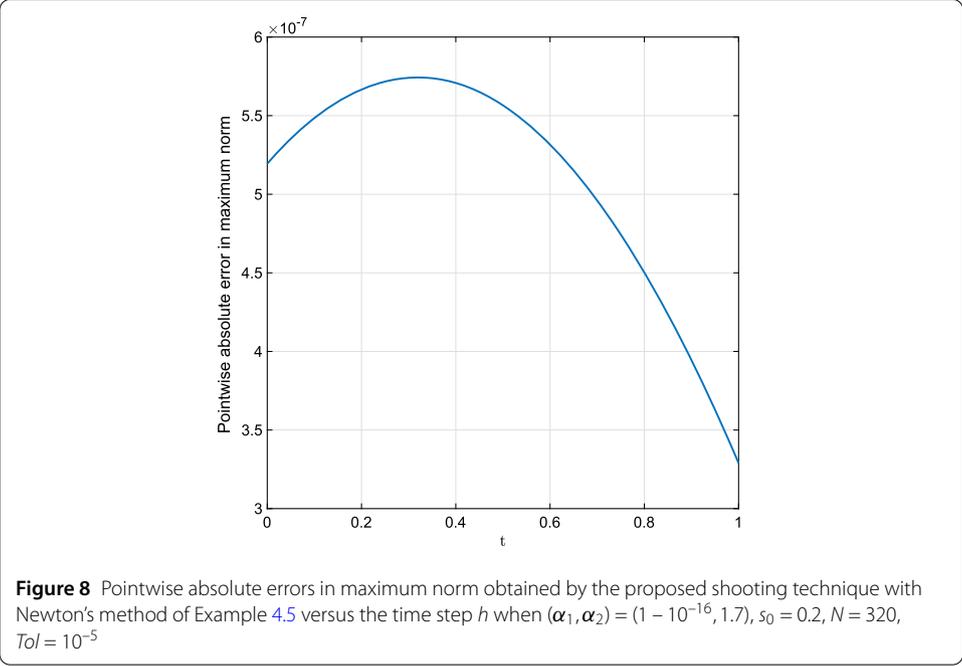


Figure 7 Pointwise absolute errors in maximum norm obtained by the modified integral discretization scheme [21] of Example 4.5 versus the time step h when $(\alpha_1, \alpha_2) = (1 - 10^{-16}, 1.7)$, $s_0 = 0.2$, $N = 320$, $Tol = 10^{-5}$

- (b) the proposed shooting technique based on Newton's method with linear explicit method described in Appendix A,
 - (c) the proposed shooting technique based on Newton's method with HPCM,
 - (d) the proposed shooting technique based on Halley's method with HPCM
- versus the number of time step N as $\alpha_2 = 1.3, 1.5, 1.7, 1.9$ are demonstrated. We set $s_0 = 0.2$, $Tol = 10^{-10}$ as our proposed methods are implemented.
- 2 It is evident that the uniform two-mesh difference errors computed by our proposed methods are less than the finite difference method [22] for all N and α_2 .



3 The convergence rates obtained by the proposed method with Newton's method are nearly 2 for all α_2 but the rates of convergence computed by Halley's method are nearly 3 when $\alpha_2 = 1.3$ only. Nevertheless we can see that the convergence profiles obtained by our proposed methods are higher than the finite difference method [22]. Figure 10 shows two-point difference errors in maximum norms versus the number of step N .

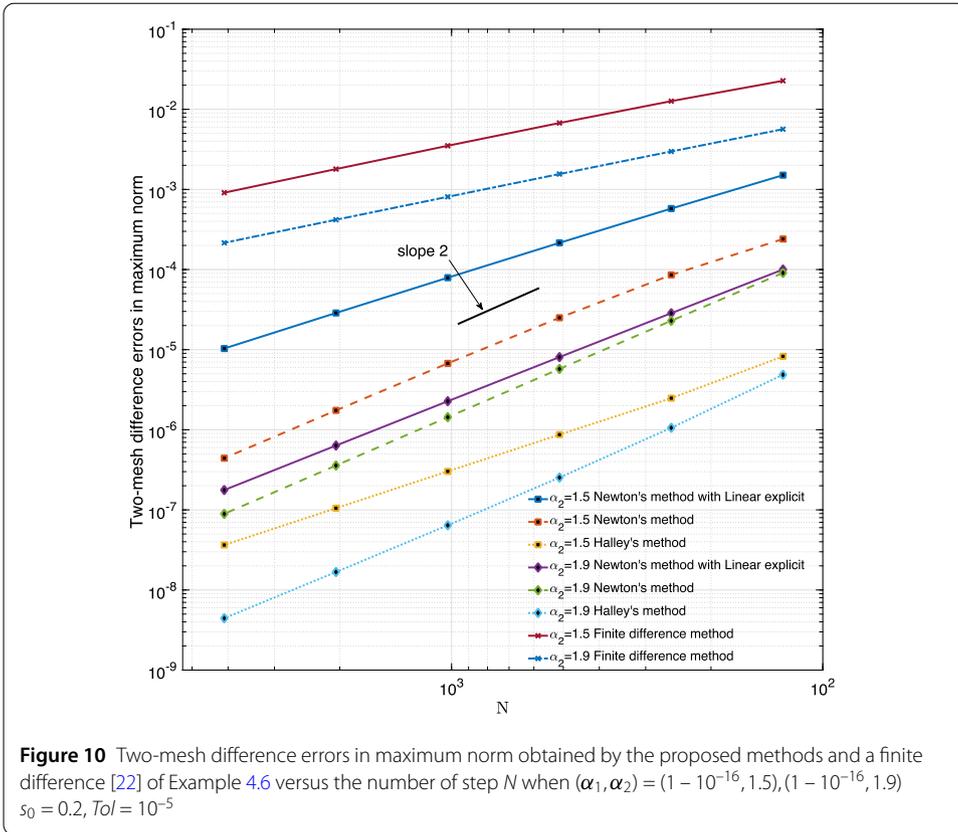
Table 12 The uniform two-mesh difference error and computed rates of convergence obtained by the proposed methods, Newton’s method, and Halley’s method. We set $s_0 = 0.2, Tol = 10^{-10}$

α_2		$N = 64$	$N = 128$	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$
Finite difference method [22]							
1.3	error	6.557E-002	4.240E-002	2.509E-002	1.387E-002	7.337E-003	3.782E-003
	rate	0.629	0.757	0.855	0.919	0.956	0.977
1.5	error	2.271E-002	1.265E-002	6.749E-003	3.509E-003	1.795E-003	9.104E-004
	rate	0.844	0.906	0.944	0.967	0.980	0.988
1.7	error	1.110E-002	5.968E-003	3.140E-003	1.628E-003	8.356E-004	4.262E-004
	rate	0.895	0.927	0.948	0.962	0.971	0.978
1.9	error	5.650E-003	2.978E-003	1.557E-003	8.090E-004	4.184E-004	2.156E-004
	rate	0.017	0.050	0.084	0.117	0.168	0.255
Newton’s method with linear explicit							
1.3	error	5.35E-03	2.55E-03	1.14E-03	4.95E-04	2.09E-04	8.68E-05
	rate	0.935	1.068	1.157	1.211	1.244	1.265
1.5	error	5.35E-03	2.55E-03	1.14E-03	4.95E-04	2.09E-04	8.68E-05
	rate	0.935	1.068	1.157	1.211	1.244	1.265
1.7	error	4.19E-04	1.39E-04	4.51E-05	1.45E-05	4.62E-06	1.46E-06
	rate	1.562	1.594	1.618	1.636	1.650	1.660
1.9	error	1.00E-04	2.85E-05	8.07E-06	2.27E-06	6.35E-07	1.77E-07
	rate	1.798	1.810	1.821	1.830	1.837	1.844
Newton’s method							
1.3	error	4.04E-03	6.03E-04	4.95E-05	8.12E-06	5.88E-06	2.17E-06
	rate	2.238	2.745	3.607	2.606	0.465	1.438
1.5	error	2.41E-04	8.57E-05	2.50E-05	6.74E-06	1.74E-06	4.42E-07
	rate	0.329	1.489	1.775	1.893	1.949	1.980
1.7	error	1.53E-04	4.05E-05	1.03E-05	2.58E-06	6.40E-07	1.58E-07
	rate	1.779	1.919	1.975	1.998	2.010	2.017
1.9	error	9.08E-05	2.29E-05	5.74E-06	1.43E-06	3.58E-07	8.92E-08
	rate	1.941	1.984	1.997	2.001	2.003	2.004
Halley’s method							
1.3	error	1.00E-03	9.19E-05	6.67E-06	3.25E-07	1.60E-08	8.44E-09
	rate	3.056	3.456	3.782	4.358	4.348	0.920
1.5	error	8.24E-06	2.48E-06	8.70E-07	3.02E-07	1.05E-07	3.64E-08
	rate	2.111	1.732	1.511	1.524	1.526	1.525
1.7	error	1.26E-05	3.60E-06	1.07E-06	3.26E-07	9.97E-08	3.06E-08
	rate	1.862	1.802	1.746	1.719	1.707	1.703
1.9	error	4.87E-06	1.06E-06	2.53E-07	6.42E-08	1.68E-08	4.44E-09
	rate	2.365	2.200	2.063	1.980	1.937	1.916

5 Conclusion

We introduced new numerical schemes for solving FNBVPs with any RBCs. The idea was to transform a FNBVP into a system of FIVPs. By doing that we could adopt a pre-existing numerical method for solving the system of FIVPs and we mainly employed HPCMs. The unknown IC s in the system was approximated by the proposed shooting methods based on Newton’s and Halley’s method and this is the main algorithm of the proposed schemes. Under the assumption that m is large enough so that $|\tilde{F}(s_m)|$ is small enough, theoretical convergence rates of proposed methods were $\mathcal{O}(h^2)$ for shooting with Newton’s method and $\mathcal{O}(h^3)$ for shooting with Halley’s method on account of global error estimates in HPCMs.

In Examples 4.1 through 4.3, we verified that the proposed schemes can handle double-term FNBVPs with RBCs whose exact solutions include polynomial, exponential, and sine



function. Convergence profiles obtained by the proposed schemes were computed as expected by the global error estimates. However, Tables 3, 6, and 9 suggest that the convergence rate of the sequence $\{s_k\}$ depends on fractional orders. We still need to address an error analysis of shooting techniques based on Newton’s and Halley’s methods for solving a system of FIVPs. This will be considered in a subsequent paper. Examples 4.4 and 4.5 demonstrated the performance of proposed methods for solving single-term linear FBVPs with exact solutions having low regularity and high regularity, respectively. Tables 10 and 11 showed that the proposed methods can deal with not only nonlinear FBVPs but also linear FBVPs. In Example 4.4, we adopted the linear explicit method described in Appendix A and this shows that the proposed shooting techniques can be assembled with not only HPCMs but also other pre-existing numerical schemes for solving a system of FIVPs. In Example 4.5, we observed that computed convergence rates obtained by our proposed shooting technique based on Halley’s method with third-order HPCM are higher than the modified integral discretization scheme [21].

Appendix A: Linear explicit method

Let us consider the following linear single-term FBVP with RBCs:

$$\begin{cases} D_0^{\alpha_2} y(t) = f(t) + c(t)y(t) + b(t)y'(t), \\ a_1 y(0) + b_1 y'(0) = \gamma_1, \quad a_2 y(1) + b_2 y'(1) = \gamma_2, \end{cases} \tag{36}$$

where $1 < \alpha_2 < 2$. By Lemmas 2.1 through 2.4 and Theorem 2.1, the FBVP (36) is equivalent to the following system:

$$\begin{cases} D_0^{1-\epsilon} y(t) = z(t) & y(0) = s, \\ D_0^\alpha z(t) = f(t) + c(t)y(t) + b(t)z(t) & z(0) = (\gamma_1 - a_1 s)/b_1. \end{cases} \tag{37}$$

Expressing the solution of (37) as the discretized form of the Volterra integral equation, which is equivalent to (37), we obtain

$$\begin{aligned} y(t_{n+1}) &= y(0) + \frac{1}{\Gamma(1-\epsilon)} \int_{t_0}^{t_{n+1}} (t_{n+1} - \tau)^{-\epsilon} z(\tau) d\tau, \\ z(t_{n+1}) &= z(0) + J_0^{\alpha_2-1} f(t_{n+1}) \\ &\quad + \frac{1}{\Gamma(\alpha_2)} \int_{t_0}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha_2-1} (c(\tau)y(\tau) + b(\tau)z(\tau)) d\tau. \end{aligned} \tag{38}$$

The approximation solutions to $y(t_{n+1})$ and $z(t_{n+1})$ in (38) with s_k can be explicitly described as follows:

$$\begin{aligned} y_{n+1} &= s_k + \frac{1}{\Gamma(1-\epsilon)} \sum_{j=0}^n \int_{t_j}^{t_{j+1}} (t_{n+1} - \tau)^{-\epsilon} z(\tau) d\tau, \\ z_{n+1} &= \frac{\gamma_1 - a_1 s_k}{b_1} + J_0^{\alpha_2-1} f(t_{n+1}) \\ &\quad + \frac{1}{\Gamma(\alpha_2)} \sum_{j=0}^n \int_{t_j}^{t_{j+1}} (t_{n+1} - \tau)^{\alpha_2-1} (c(\tau)y(\tau) + b(\tau)z(\tau)) d\tau. \end{aligned}$$

Replacing $y(\tau)$ and $z(\tau)$ with linear interpolation, we have

$$\begin{aligned} y_{n+1} &= s_k + \frac{1}{\Gamma(1-\epsilon)} \sum_{j=0}^n \int_{t_j}^{t_{j+1}} (t_{n+1} - \tau)^{-\epsilon} \left(\frac{t_{j+1} - \tau}{h} z_j + \frac{t_j - \tau}{-h} z_{j+1} \right) d\tau, \\ z_{n+1} &= \frac{\gamma_1 - a_1 s_k}{b_1} + J_0^{\alpha_2-1} f(t_{n+1}) \\ &\quad + \frac{1}{\Gamma(\alpha_2)} \sum_{j=0}^n \int_{t_j}^{t_{j+1}} (t_{n+1} - \tau)^{\alpha_2-1} \left\{ \frac{t_{j+1} - \tau}{h} (c(t_j)y_j + b(t_j)z_j) \right. \\ &\quad \left. + \frac{t_j - \tau}{-h} (c(t_{j+1})y_{j+1} + b(t_{j+1})z_{j+1}) \right\} d\tau. \end{aligned} \tag{39}$$

Let us shorten the expression of y_{n+1}, z_{n+1} in (39) as follows:

$$y_{n+1} = s_k + \sum_{j=0}^n [A_j^1 z_j + A_j^2 z_{j+1}],$$

$$\begin{aligned}
 z_{n+1} &= \frac{\gamma_1 - a_1 s_k}{b_1} + J_0^{\alpha_2 - 1} f(t_{n+1}) \\
 &\quad + \sum_{j=0}^{n-1} [B_j^1 (c(t_j)y_j + b(t_j)z_j) + B_j^2 (c(t_{j+1})y_{j+1} + b(t_{j+1})z_{j+1})] \\
 &\quad + B_n^1 (c(t_n)y_n + b(t_n)z_n) + B_n^2 (c(t_{n+1})y_{n+1} + b(t_{n+1})z_{n+1}).
 \end{aligned}$$

We omit describing the definitions of $A_j^i, B_j^i, i = 1, 2, j = 0, \dots, n$, because that is straightforward. Substituting the explicit form of y_{n+1} into the right-hand side of z_{n+1} , we have

$$\begin{aligned}
 z_{n+1} &= \frac{\gamma_1 - a_1 s_k}{b_1} + J_0^{\alpha_2 - 1} f(t_{n+1}) \\
 &\quad + \sum_{j=0}^{n-1} [B_j^1 (c(t_j)y_j + b(t_j)z_j) + B_j^2 (c(t_{j+1})y_{j+1} + b(t_{j+1})z_{j+1})] \\
 &\quad + B_n^1 (c(t_n)y_n + b(t_n)z_n) \\
 &\quad + B_n^2 \left[c(t_{n+1}) \left\{ s_k + \sum_{j=0}^n (A_j^1 z_j + A_j^2 z_{j+1}) \right\} + b(t_{n+1})z_{n+1} \right] \\
 &= \frac{\gamma_1 - a_1 s_k}{b_1} + J_0^{\alpha_2 - 1} f(t_{n+1}) \tag{40} \\
 &\quad + \sum_{j=0}^{n-1} [B_j^1 (c(t_j)y_j + b(t_j)z_j) + B_j^2 (c(t_{j+1})y_{j+1} + b(t_{j+1})z_{j+1})] \\
 &\quad + B_n^1 (c(t_n)y_n + b(t_n)z_n) \\
 &\quad + B_n^2 c(t_{n+1}) \left\{ s_k + \sum_{j=0}^{n-1} (A_j^1 z_j + A_j^2 z_{j+1}) + A_n^1 z_n \right\} \\
 &\quad + (B_n^2 A_n^2 c(t_{n+1}) + b(t_{n+1}))z_{n+1}.
 \end{aligned}$$

Since the right-hand side of (40) is linear in $z_j, j = 0, \dots, n + 1, z_{n+1}$ can be explicitly expressed as follows:

$$\begin{aligned}
 z_{n+1} &= \left[\frac{\gamma_1 - a_1 s_k}{b_1} + J_0^{\alpha_2 - 1} f(t_{n+1}) + \sum_{j=0}^{n-1} [B_j^1 (c(t_j)y_j + b(t_j)z_j) \right. \\
 &\quad \left. + B_j^2 (c(t_{j+1})y_{j+1} + b(t_{j+1})z_{j+1})] + B_n^1 (c(t_n)y_n + b(t_n)z_n) \right. \\
 &\quad \left. + B_n^2 c(t_{n+1}) \left\{ s_k + \sum_{j=0}^{n-1} (A_j^1 z_j + A_j^2 z_{j+1}) + A_n^1 z_n \right\} \right] \\
 &\quad / [1 - \{B_n^2 A_n^2 c(t_{n+1}) + b(t_{n+1})\}].
 \end{aligned}$$

Appendix B: Step of algorithm

Steps of the proposed algorithms for solving the FNVBP (7)

Input parameters: fractional orders α_1, α_2 ; endpoints $0, b$; RBCs $a_1, b_1, \gamma_1, a_2, b_2, \gamma_2$; number of time sub-intervals N ; maximum number of iterations in Newton’s and Halley’s methods m ; initial approximation of s s_0 ; tolerance Tol .

Step 1) Set $h = b/N, t_n = n \cdot h, k = 0$.

Step 2) While $(k \leq m)$ do Steps 3 \dots 6.

Step 3) For $n = 0, \dots, N - 1$ do Steps 3 - 1 \dots 3 - 3.

3 - 1) Compute predictors $y_{n+1}^p, w_{n+1}^p, z_{n+1}^p$ using the formulas (33) (Newton’s method) or (35) (Halley’s method) with s_k individually.

3 - 2) Compute $f_{n+1}^p \equiv f(t_{n+1}, y_{n+1}^p, w_{n+1}^p)$.

3 - 3) Compute correctors $y_{n+1}^c, w_{n+1}^c, z_{n+1}^c$ using the formulas (32) (Newton’s method) or (34) (Halley’s method) individually.

Step 4) Compute the approximated error function with s_k :

$$|\tilde{F}(s_k)| = |a_2 y_N^c(s_k) + b_2 z_N^c(s_k) - \gamma_2|. \tag{41}$$

If $|\tilde{F}(s_k)| \leq Tol$ then STOP.

Do Step 5 to compute $F_s(s_k)$ (Newton’s or Halley’s method) and $F_{ss}(s_k)$ (Halley’s method).

Step 5) For $n = 0, \dots, N - 1$ do Steps 5 - 1 \dots 5 - 5.

5 - 1) Compute predictors $y_{s,n+1}^p, w_{s,n+1}^p, z_{s,n+1}^p$ in (25) using the formulas (33) (Newton’s method) or (35) (Halley’s method) individually.

5 - 2) Compute predictors $y_{ss,n+1}^p, w_{ss,n+1}^p, z_{ss,n+1}^p$ in (28) using the formulas (35) (Halley’s method) individually. (Skip step for Newton’s method.)

5 - 3) Compute $f_{s,n+1}^p$ defined by (26) (Newton’s and Halley’s methods). Compute $f_{ss,n+1}^p$ defined by (29) (Halley’s method).

5 - 4) Compute correctors $y_{s,n+1}^c, w_{s,n+1}^c, z_{s,n+1}^c$ in (25) using the formulas (32) (Newton’s method) or (35) (Halley’s method) individually.

5 - 5) Compute predictors $y_{ss,n+1}^c, w_{ss,n+1}^c, z_{ss,n+1}^c$ in (28) using the formulas (34) (Halley’s method) individually. (Skip step for Newton’s method.)

Step 6) Compute s_{k+1} using the formulas (23) (Newton’s method) or (27) (Halley’s method).

$k = k + 1$.

Acknowledgements

We would like to thank the editors of the journal as well as the anonymous reviewers for their valuable suggestions that make the manuscript more consistent.

Funding

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (NRF-2016R1D1A1B03935514, NRF-2019R1A2C1005142).

Availability of data and materials

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Consent for publication

Not applicable.

Authors' contributions

HK: Methodology, Writing—Original and Revision draft, JL: Data curation, Simulation, BJ: Conceptualization of this study, Analysis, Writing—Original and Revision draft. All authors read and approved the final manuscript.

Author details

¹Department of Mathematics, North Greenville University, 29688 Tigerville, SC, USA. ²Department of Mathematical Sciences, Ulsan National Institute of Science and Technology (UNIST), 44919, Ulsan, Republic of Korea.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 29 October 2020 Accepted: 24 March 2021 Published online: 01 April 2021

References

1. Nguyen, T.B., Jang, B.: A high-order predictor–corrector method for solving nonlinear differential equations of fractional order. *Fract. Calc. Appl. Anal.* **20**(2), 447–476 (2017)
2. Caputo, M.: Linear model of dissipation whose Q is almost frequency independent—II. *Geophys. J. R. Astron. Soc.* **13**, 529–539 (1967)
3. Dalir, M., Bashour, M.: Applications of fractional calculus. *Appl. Math. Sci.* **4**, 1021–1032 (2010)
4. Das, S.: *Functional Fractional Calculus for System Identification and Controls*. Springer, Berlin (2008)
5. Hilfer, R.: *Applications of Fractional Calculus in Physics*. World Scientific, Singapore (2000)
6. Magin, R.L.: *Fractional Calculus in Bioengineering*. Begell House Publishers (2006)
7. Podlubny, I.: *Fractional Differential Equations*. Academic Press, San Diego (1999)
8. Arqub, O.A., Shaqagfeh, N.: Application of reproducing kernel algorithm for solving Dirichlet time-fractional diffusion-Gordon types equations in porous media. *J. Porous Media* **22**(4), 411–434 (2019)
9. Arqub, O.A.: Numerical algorithm for the solutions of fractional order systems of Dirichlet function types with comparative analysis. *Fundam. Inform.* **166**(2), 111–137 (2019)
10. Arqub, O.A.: Application of residual power series method for the solution of time-fractional Schrödinger equations in one-dimensional space. *Fundam. Inform.* **166**(2), 87–110 (2019)
11. Roul, P., Goura, V.M.K.P.: A high order numerical scheme for solving a class of non-homogeneous time-fractional reaction diffusion equation. *Numer. Methods Partial Differ. Equ.* <https://doi.org/10.1002/num.22594>
12. Roul, P., Rohil, V., Espinosa-Paredes, G., Goura, V.M.K.P., Gedam, R.S., Obaidurrahman, K.: Design and analysis of a numerical method for fractional neutron diffusion equation with delayed neutrons. *Appl. Numer. Math.* **157**, 634–653 (2020)
13. Roul, P., Goura, V.M.K.P.: A high-order b-spline collocation scheme for solving a nonhomogeneous time-fractional diffusion equation. *Math. Methods Appl. Sci.* **44**(1), 546–567 (2020)
14. Roul, P., Rohil, V., Espinosa-Paredes, G., Obaidurrahman, K.: An efficient numerical method for fractional neutron diffusion equation in the presence of different types of reactivities. *Ann. Nucl. Energy* **152**, 108038 (2021)
15. Baleanu, D., Etemad, S., Rezapour, S.: A hybrid Caputo fractional modeling for thermostat with hybrid boundary value conditions. *Bound. Value Probl.* **2020**(1), 64 (2020). <https://doi.org/10.1186/s13661-020-01361-0>
16. Patnaik, S., Sidhardh, S., Semperlotti, F.: A Ritz-based finite element method for a fractional-order boundary value problem of nonlocal elasticity. *Int. J. Solids Struct.* **202**, 398–417 (2020)
17. Cai, H., An, Q.: A fractional spectral collocation method for general Caputo two-point boundary value problems. *Appl. Numer. Math.* (2021). <https://doi.org/10.1016/j.apnum.2021.01.006>
18. Liu, L.-B., Liang, Z., Long, G., Liang, Y.: Convergence analysis of a finite difference scheme for a Riemann–Liouville fractional derivative two-point boundary value problem on an adaptive grid. *J. Comput. Appl. Math.* **375**, 112809 (2020). <https://doi.org/10.1016/j.cam.2020.112809>
19. Diethelm, K.: *The Analysis of Fractional Differential Equations: An Application-Oriented Exposition Using Differential Operators of Caputo Type*. Springer, Berlin (2010)
20. Cen, Z., Huang, J., Le, A.: A modified integral discretization scheme for a two-point boundary value problem with a Caputo fractional derivative. *J. Comput. Appl. Math.* **367**(15), 112465 (2020)
21. Cen, Z., Huang, J., Xu, A.: An efficient numerical method for a two-point boundary value problem with a Caputo fractional derivative an efficient numerical method for a two-point boundary value problem with a Caputo fractional derivative. *J. Comput. Appl. Math.* **336**, 1–7 (2018)
22. Stynes, M., Gracia, J.L.: A finite difference method for a two-point boundary value problem with a Caputo fractional derivative. *IMA J. Numer. Anal.* **35**, 698–721 (2015)