Computers and Mathematics with Applications 62 (2011) 918–929

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Numerical solution of fractional differential equations with a collocation method based on Müntz polynomials

Shahrokh Esmaeili [a], M. Shamsi [a,*], Yury Luchko [b]

[a] Department of Applied Mathematics, Faculty of Mathematics and Computer Science, Amirkabir University of Technology, No. 424, Hafez Ave. Tehran, Iran

[b] Department of Mathematics II, Technical University of Applied Sciences Berlin, Berlin, Germany

ARTICLE INFO

Keywords:
Fractional derivatives
Fractional differential equations
Orthogonal polynomials
Müntz polynomials
Gaussian quadrature

ABSTRACT

This paper presents a computational technique based on the collocation method and Müntz polynomials for the solution of fractional differential equations. An appropriate representation of the solution via the Müntz polynomials reduces its numerical treatment to the solution of a system of algebraic equations. The main advantage of the present method is its superior accuracy and exponential convergence. Consequently, one can obtain good results even by using a small number of collocation points. The accuracy and performance of the proposed method are examined by means of some numerical experiments.

1. Introduction

Fractional calculus was first developed as a pure mathematical theory in the middle of the 19th century [1]. About 100 years later, engineers and physicists found applications for these concepts in their areas [2–4]. Fractional derivatives provide an excellent instrument for the description of memory and hereditary properties of various materials and processes [5]. In some cases, the fractional order models of real systems are more adequate compared to the integer order models. So, in recent decades, the field of fractional calculus has attracted the interest of researchers in several areas including physics, chemistry, engineering and even finance and social sciences [6,7,5].

This paper concerns the numerical solution of single term fractional differential equations

$$D^q_\star y(t) = f(t, y(t)), \quad 0 < q \leq 1, \tag{1}$$

with an initial condition $y(0) = y_0$. Here $D^q_\star$ denotes the fractional derivative in the Caputo sense [8,4],

$$D^q_\star y(t) = \frac{1}{\Gamma(1-q)} \int_0^t (t-\tau)^{-q} y'(\tau) \mathrm{d}\tau, \quad 0 < q < 1, \tag{2}$$

and f is a function of two variables from an appropriate functional space. Equations of this type arise in a number of applications (see, for example [3–5]). During the past decades, several numerical methods have been used to solve fractional differential equations of this type. Ford and Connolly [9] and Diethelm et al. [10] have reviewed some of the existing methods and demonstrated their respective strengths and weaknesses. For further analytical and numerical schemes to solve fractional differential equations, we refer the reader to [11–19].

In this article, a different approach for the numerical treatment of fractional differential equations is proposed. This approach is based on the collocation method. Since a fractional derivative is a global operator, it is very natural to consider

* Corresponding author. Tel.: +98 21 64542538; fax: +98 21 66497930.
E-mail addresses: esmaeili@aut.ac.ir (S. Esmaeili), m_shamsi@aut.ac.ir (M. Shamsi), Luchko@tfh-berlin.de (Y. Luchko).

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.04.023

a global method like the collocation method for its numerical solution [20]. Spectral collocation methods are efficient and highly accurate techniques for numerical solution of nonlinear differential equations. One of their prominent features is the exponential convergence of the approximations [21–24].

The basic idea of the spectral collocation method is to assume that the unknown solution $y(t)$ can be approximated by a linear combination of some basis functions, called the trial functions, such as orthogonal polynomials. The orthogonal polynomials can be chosen according to their special properties, which make them particularly suitable for a problem under consideration.

Whereas the classical orthogonal polynomials work well for numerical solution of conventional differential equations, their application for the fractional differential equations implies at least two difficulties in connection with the collocation method. First, according to Theorems 6.33 and 6.38 in [8], solutions of fractional differential equation (1) can contain some fractional-power terms that the classical orthogonal polynomials cannot match. In this case the rate of convergence of the numerical approximations is not reasonable when the classical polynomial bases are used. Second, to apply a collocation method, it is crucial that derivatives of any trial function can be expressed in terms of the same trial bases. But the fractional derivatives of a classical polynomial are not polynomials, so we cannot hope to obtain a good approximation for the fractional derivatives via the classical orthogonal polynomials.

In the present article, we use the Müntz–Legendre polynomials, which are a family of generalized orthogonal polynomials. These polynomials were introduced and investigated in [25–27]. A fractional derivative of a Müntz polynomial is again a Müntz polynomial. This is a crucial feature of these bases for using them in the collocation method for numerical solution of the fractional differential equations. We start with a construction of a stable numerical method for evaluating the Müntz–Legendre polynomials and their fractional derivatives, which, to the best of our knowledge, has not been investigated before. Then, by applying an appropriate finite or discrete representation of the solution based on the Müntz–Legendre polynomials, the problem under consideration can be reduced to solve a system of algebraic equations that can be done by a number of known methods.

The rest of the paper is organized as follows: In the next section, a brief overview of the orthogonal polynomials and Gaussian quadrature is given. In Section 3, the Müntz–Legendre polynomials are introduced. The Section 4 is devoted to presentation of a stable numerical method for evaluating the fractional derivatives of the Müntz–Legendre polynomials. In Section 5, the collocation method is applied for numerical solution of the fractional differential equations. Finally, some numerical examples are given in Section 6 to demonstrate the effectiveness of the proposed method.

2. A brief overview of orthogonal polynomials

2.1. Recurrence relations for orthogonal polynomials

Let $w = w(t)$ be a weight function on the interval $[a, b]$, i.e., a non-negative integrable function defined in $[a, b]$. The integral

$$(u, v)_w = \int_a^b u(t)v(t)w(t)\mathrm{d}t, \tag{3}$$

defines the inner or scalar product of the functions $u(t)$ and $v(t)$ over the interval $[a, b]$ with respect to the weight function $w(t)$. The numbers

$$\mu_r = \int_a^b t^r w(t)\mathrm{d}t, \quad r = 0, 1, \ldots, \tag{4}$$

are called the moments related to the weight function w. For each weighted scalar product (3), there exist uniquely determined polynomials $\pi_k \in \mathbb{P}_k$ with the leading coefficient 1, which are orthogonal each to other with respect to this scalar product. They satisfy a three-term recurrence relation

$$\pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k\pi_{k-1}(t), \quad k = 0, 1, 2, \ldots, \tag{5}$$

with starting values $\pi_{-1} := 0,\ \pi_0 := 1$, and with the coefficients

$$\alpha_k = \frac{(t\pi_k, \pi_k)_w}{(\pi_k, \pi_k)_w}, \quad k = 0, 1, 2, \ldots,$$

$$\beta_k = \frac{(\pi_k, \pi_k)_w}{(\pi_{k-1}, \pi_{k-1})_w}, \quad k = 1, 2, \ldots. \tag{6}$$

The three-term recurrence relation (5) is quite stable and can thus be conveniently employed for the numerical computation of orthogonal polynomials [28].

For the classical orthogonal polynomials, e.g. Jacobi, Laguerre, and Hermite polynomials, formulae for the coefficients α_k and β_k are known in closed form [28,29]. For the nonclassical weight functions, their recurrence coefficients are not explicitly known. In this case, numerical techniques such as *Stieltjes procedure* or *Chebyshev algorithm* are used to evaluate

the coefficients [30]. In this paper, we use the Chebyshev algorithm that derives the desired coefficients from the moments of the underlying weight function $w(t)$. In this way, all recurrence coefficients can be calculated. For the readers convenience, the Chebyshev algorithm is presented below [28].

Algorithm 1. Chebyshev Algorithm

1. *Initialization:*

$$\alpha_0 = \frac{\mu_1}{\mu_0}, \qquad \beta_0 = \mu_0,$$
$$\sigma_{-1,\ell} = 0, \quad \ell = 1, 2, \ldots, 2n-2,$$
$$\sigma_{0,\ell} = \mu_\ell, \quad \ell = 0, 1, \ldots, 2n-1.$$

2. *Construction* (if $n > 1$): for $k = 1, 2, \ldots, n-1$ do

$$\sigma_{k\ell} = \sigma_{k-1,\ell+1} - \alpha_{k-1}\sigma_{k-1,\ell} - \beta_{k-1}\sigma_{k-2,\ell}, \quad \ell = k, k+1, \ldots, 2n-k-1,$$
$$\alpha_k = \frac{\sigma_{k,k+1}}{\sigma_{k,k}} - \frac{\sigma_{k-1,k}}{\sigma_{k-1,k-1}},$$
$$\beta_k = \frac{\sigma_{k,k}}{\sigma_{k-1,k-1}}.$$

The Chebyshev algorithm requires moments $\{\mu_r\}_{r=0}^{2n-1}$ as input, and produces the recurrence coefficients $\{\alpha_k, \beta_k\}_{k=0}^{n-1}$. The complexity of the algorithm is $\mathcal{O}(n^2)$.

2.2. Jacobi polynomials

The well-known Jacobi polynomials $P_k^{(\alpha,\beta)}$ with parameters $\alpha, \beta > -1$ and their special cases are probably the most widely used classical orthogonal polynomials for numerical solution of differential equations. The explicit form of Jacobi polynomials that has been used e.g. in [29] is

$$P_k^{(\alpha,\beta)}(x) = \sum_{m=0}^{k} \frac{(-1)^{k-m}(1+\beta)_k(1+\alpha+\beta)_{k+m}}{m!(k-m)!(1+\beta)_m(1+\alpha+\beta)_k}\left(\frac{1+x}{2}\right)^m, \tag{7}$$

where

$$(d)_0 = 1, \qquad (d)_i = d(d+1)\cdots(d+i-1).$$

This expression shows that $P_k^{(\alpha,\beta)}$ are analytic functions of the parameters α and β. The Jacobi polynomials are orthogonal over the interval $(-1, 1)$ with respect to the weight function $w^{(\alpha,\beta)}(x) = (1-x)^\alpha(1+x)^\beta$. The choice $\alpha = \beta = 0$ yields the Legendre polynomials, while choosing $\alpha = \beta = -1/2$ gives Chebyshev polynomials.

In practice, one can compute the Jacobi polynomials using the following recurrence relation [23,28,29]

$$P_0^{(\alpha,\beta)}(x) = 1, \qquad P_1^{(\alpha,\beta)}(x) = \frac{1}{2}[(\alpha-\beta) + (\alpha+\beta+2)x],$$
$$a_{1,k}^{\alpha,\beta}P_{k+1}^{(\alpha,\beta)}(x) = a_{2,k}^{\alpha,\beta}(x)P_k^{(\alpha,\beta)}(x) - a_{3,k}^{\alpha,\beta}P_{k-1}^{(\alpha,\beta)}(x), \tag{8}$$

where

$$a_{1,k}^{\alpha,\beta} = 2(k+1)(k+\alpha+\beta+1)(2k+\alpha+\beta),$$
$$a_{2,k}^{\alpha,\beta}(x) = (2k+\alpha+\beta+1)[(2k+\alpha+\beta)(2k+\alpha+\beta+2)x + \alpha^2 - \beta^2],$$
$$a_{3,k}^{\alpha,\beta} = 2(k+\alpha)(k+\beta)(2k+\alpha+\beta+2). \tag{9}$$

A useful formula that relates Jacobi polynomials and their derivatives is

$$\frac{\mathrm{d}}{\mathrm{d}x}P_k^{(\alpha,\beta)}(x) = \frac{1}{2}(k+\alpha+\beta+1)P_{k-1}^{(\alpha+1,\beta+1)}(x). \tag{10}$$

2.3. Gauss-type quadrature rules

An n-point quadrature rule for the weight function w is called a formula of the type

$$\int_a^b f(t)w(t)\mathrm{d}t = \sum_{k=1}^{n}\omega_k f(\tau_k) + R_n(f), \tag{11}$$

where the sum on the right-hand side of the equation provides an approximation to the integral and R_n is the error. The numbers τ_k, $k = 1, \ldots, n$ are called nodes and ω_k are called weights of the quadrature rule. Among all quadrature rules of the form (11) those of the Gaussian type have the best performance. More precisely, if nodes τ_k and weights ω_k are chosen in

the way that quadrature rule (11) becomes exact for polynomials of degree at most $2n-1$, then this quadrature rule is called a Gauss-type quadrature rule. It can be proved that the nodes τ_k in a Gaussian quadrature are the roots of the orthogonal polynomial π_n associated with the weight function and the weights ω_k can be obtained from the following system of linear equations:

$$\sum_{k=1}^{n} \omega_k \tau_k^r = \mu_r, \quad r = 0, 1, \ldots, 2n-1. \tag{12}$$

As n increases, finding roots of π_n and solving the linear system (12) become an ill-conditioned and time consuming problem. Alternatively, we can use the Golub–Welsch algorithm [31] to determine the nodes and the weights of a Gaussian quadrature. This approach is based on determining the eigenvalues and normalized eigenvectors of the following symmetric tridiagonal matrix

$$\mathbf{J}_n = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & & \mathbf{0} \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & & \\ & \sqrt{\beta_2} & \alpha_2 & \ddots & \\ & & \ddots & \ddots & \sqrt{\beta_{n-1}} \\ \mathbf{0} & & & \sqrt{\beta_{n-1}} & \alpha_{n-1} \end{bmatrix}. \tag{13}$$

This matrix is known as the *Jacobi matrix* and its elements are obtained from the coefficients of the three-term recurrence relation (5). The nodes τ_k are the eigenvalues of the matrix $\mathbf{J}_n$ and the corresponding weights ω_k may be obtained from the first components of the normalized eigenvectors. For more details see [28,32,31].

3. Müntz polynomials

In 1885, Weierstrass proved that every continuous function on a compact interval can be uniformly approximated by algebraic polynomials. One of the first outstanding generalizations of the Weierstrass theorem is due to German mathematician Herman Müntz [33].

Let $\Lambda = \{\lambda_0, \lambda_1, \lambda_2, \ldots\}$ be a sequence of distinct positive numbers such that $0 \le \lambda_0 < \lambda_1 < \cdots \to \infty$. The classical Müntz–Szász theorem states that the Müntz polynomials of the form $\sum_{k=0}^{n} a_k x^{\lambda_k}$ with real coefficients are dense in $L^2[0, 1]$ if and only if $\sum_{k=1}^{\infty} \lambda_k^{-1} = +\infty$. If the constant function 1 belongs to the system, that is if $\lambda_0 = 0$, then the same result holds for $C[0, 1]$ with the uniform norm. A proof of the Müntz–Szász theorem can be found in [34].

In this paper, we consider the Müntz–Legendre polynomials, which are orthogonal over the interval $(0, 1)$ with respect to the weight function $w(x) = 1$.

3.1. Müntz–Legendre polynomials

Let the complex numbers from the set $\Lambda_n = \{\lambda_0, \lambda_1, \ldots, \lambda_n\}$ satisfy the condition $\mathrm{Re}(\lambda_k) > -1/2$. Then the Müntz–Legendre polynomials on the interval $(0, 1]$ are defined by (see [25–27])

$$P_n(x) := P_n(x; \Lambda_n) = \sum_{k=0}^{n} C_{n,k} x^{\lambda_k}, \quad C_{n,k} = \frac{\prod_{\nu=0}^{n-1} (\lambda_k + \bar{\lambda}_\nu + 1)}{\prod_{\nu=0, \nu \ne k}^{n} (\lambda_k - \lambda_\nu)}. \tag{14}$$

For the Müntz–Legendre polynomials (14), the orthogonality relation

$$(P_n, P_m) = \int_0^1 P_n(x) P_m(x) \mathrm{d}x = \frac{\delta_{nm}}{\lambda_n + \bar{\lambda}_n + 1}$$

holds true. It is easy to prove that $P_n(1) = 1$ and $P_n'(1) = \lambda_n + \sum_{k=0}^{n-1} (\lambda_k + \bar{\lambda}_k + 1)$ [25].

In this paper, we consider the case when the powers of the Müntz basis elements build an arithmetic progression. In other words, we assume that there exists a positive number α such that $\lambda_k = k\alpha$. In this case, the Müntz–Legendre polynomials on the interval $[0, T]$ are represented by the formula

$$L_n(t; \alpha) := \sum_{k=0}^{n} C_{n,k} \left(\frac{t}{T}\right)^{k\alpha}, \quad C_{n,k} = \frac{(-1)^{n-k}}{\alpha^n k! (n-k)!} \prod_{\nu=0}^{n-1} ((k+\nu)\alpha + 1). \tag{15}$$

The functions $L_k(t; \alpha)$, $k = 0, 1, \ldots, n$ form an orthogonal basis for $\mathbb{M}_{n,\alpha}$, where $\mathbb{M}_{n,\alpha}$ is defined by

$$\begin{aligned} \mathbb{M}_{n,\alpha} &= \mathrm{span}\{1, t^\alpha, \ldots, t^{n\alpha}\}, \quad t \in [0, T], \\ &= \{c_0 + c_1 t^\alpha + \cdots + c_n t^{n\alpha} : c_k \in \mathbb{R},\ t \in [0, T]\}. \end{aligned}$$

3.2. Numerical evaluation of $L_n(t;\alpha)$

A direct evaluation of Müntz–Legendre polynomials in the form (14) can be problematic in finite arithmetic, especially when n is a large number and x is close to 1. These problems have been addressed by Milovanović in [27] and arise from the fact that the coefficients $C_{n,k}$ become very large when n increases, but their sums are always equal to 1. For example, for $n = 10$ we get

$$\begin{aligned} L_{10}(t;1/2) &= 11 - 660t^{1/2} + 12\,870t - 120\,120t^{3/2} + 630\,630t^2 - 2\,018\,016t^{5/2} \\ &\quad + 4\,084\,080t^3 - 5\,250\,960t^{5/2} + 4\,157\,010t^4 - 1\,847\,560t^{7/2} + 352\,716t^5. \end{aligned}$$

In this section, we present a stable method for evaluating the Müntz–Legendre polynomials defined by (15). Our technique is based on a three-term recurrence relation that is induced from the following theorem.

Theorem 3.1. *Let $\alpha > 0$ be a real number and $t \in [0, T]$. Then the representation*

$$L_n(t;\alpha) = P_n^{(0,1/\alpha-1)}\left(2\left(\frac{t}{T}\right)^{\alpha} - 1\right) \tag{16}$$

holds true.

Proof. By substituting $x = 2\left(\frac{t}{T}\right)^{\alpha} - 1$ in (7), we get

$$\begin{aligned} P_n^{(0,1/\alpha-1)}\left(2\left(\frac{t}{T}\right)^{\alpha} - 1\right) &= \sum_{k=0}^{n} \frac{(-1)^{n-k}(1/\alpha)_{n+k}}{k!(n-k)!(1/\alpha)_k}\left(\frac{t}{T}\right)^{k\alpha} \\ &= \sum_{k=0}^{n} \frac{(-1)^{n-k}}{\alpha^k k!(n-k)!}\prod_{\nu=0}^{n-1}((k+\nu)\alpha+1)\left(\frac{t}{T}\right)^{k\alpha} \\ &= \sum_{k=0}^{n} C_{n,k}\left(\frac{t}{T}\right)^{k\alpha} = L_n(t;\alpha), \end{aligned}$$

and the proof is completed. □

So, in view of (8) and (9), the Müntz–Legendre polynomials $L_n(t;\alpha)$ can be evaluated by means of the three-term recursion

$$\begin{aligned} &L_0(t;\alpha) = 1, \qquad L_1(t;\alpha) = \left(\frac{1}{\alpha}+1\right)\left(\frac{t}{T}\right)^{\alpha} - \frac{1}{\alpha}, \\ &b_{1,n}L_{n+1}(t;\alpha) = b_{2,n}(t)L_n(t;\alpha) - b_{3,n}L_{n-1}(t;\alpha), \end{aligned} \tag{17}$$

where

$$b_{1,n} = a_{1,n}^{0,1/\alpha-1}, \qquad b_{2,n}(t) = a_{2,n}^{0,1/\alpha-1}\left(2\left(\frac{t}{T}\right)^{\alpha} - 1\right), \qquad b_{3,n} = a_{3,n}^{0,1/\alpha-1}.$$

The stability of the three-term recurrence relation (17) compared to that of the power form (15) is illustrated in Fig. 1, where we plotted the absolute errors in the values of $L_n(T;\alpha)$ for $\alpha = 0.5$ obtained by (17) and (15) for $n = 1, \ldots, 40$. Note that in all cases the relation $L_n(T;\alpha) = 1$ holds true. As we can see, the values obtained by using the power form (15) are quite inaccurate for $n \geq 20$. In contrast, with the recurrence relation (17) the values of $L_n(t;\alpha)$ can be evaluated accurately even for large values of n.

4. Numerical evaluation of $D_{\star}^{\alpha}L_n(t;\alpha)$

Let $L_n(t;\alpha)$ be defined by (15). Then the Caputo fractional derivative of $L_n(t;\alpha)$ can be represented in the form

$$D_{\star}^{\alpha}L_n(t;\alpha) := \sum_{k=1}^{n} D_{n,k}\left(\frac{t}{T}\right)^{(k-1)\alpha}, \qquad D_{n,k} = \frac{\Gamma(1+k\alpha)}{\Gamma(1+k\alpha-\alpha)T^{\alpha}}C_{n,k}. \tag{18}$$

The Eq. (18) was obtained from (15) by means of the well-known formula

$$D_{\star}^{\alpha}t^{\beta} = \frac{\Gamma(1+\beta)}{\Gamma(1+\beta-\alpha)}t^{\beta-\alpha}, \quad \alpha > 0,\ \beta > -1,\ t > 0. \tag{19}$$

It is important to notice that $D_{\star}^{\alpha}L_n(t;\alpha) \in \mathbb{M}_{n,\alpha}$. Again, the coefficients $D_{n,k}$ become very large when n increases and direct evaluation of $D_{\star}^{\alpha}L_n(t;\alpha)$ can be problematic, too. In what follows, we introduce a stable numerical method for evaluating of $D_{\star}^{\alpha}L_n(t;\alpha)$. One of the important components of this method is presented in the following theorem.

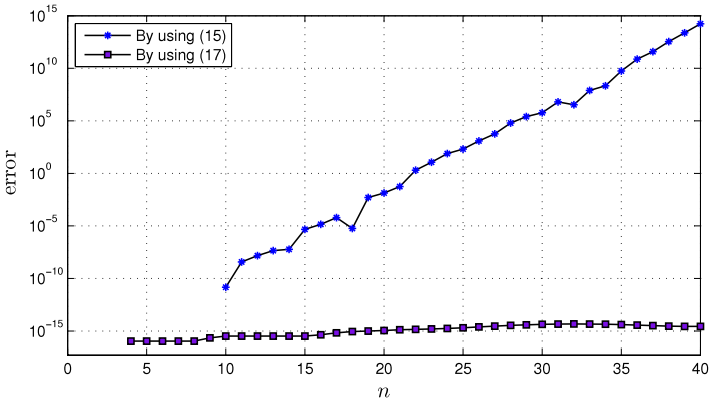

Fig. 1. Errors in the values of $L_n(T; 0.5)$ versus n.

Theorem 4.1. *Let $0 < \alpha < 1$ be a real number and $t \in [0, T]$. Then the representation*

$$D^{\alpha}_{\star}L_n(t;\alpha) = \frac{1+n\alpha}{\alpha\Gamma(1-\alpha)T^{\alpha}}\int_0^1 (1-x^{1/\alpha})^{-\alpha}P^{(1,1/\alpha)}_{n-1}\left(2\left(\frac{t}{T}\right)^{\alpha}x-1\right)\mathrm{d}x \tag{20}$$

holds true.

Proof. By definition, we get

$$D^{\alpha}_{\star}L_n(t;\alpha) = \frac{1}{\Gamma(1-\alpha)}\int_0^t (t-\tau)^{-\alpha}\frac{\mathrm{d}}{\mathrm{d}\tau}L_n(\tau;\alpha)\mathrm{d}\tau. \tag{21}$$

The Eqs. (16) and (10) lead to the formula

$$\frac{\mathrm{d}}{\mathrm{d}\tau}L_n(\tau;\alpha) = \frac{1}{T^{\alpha}}(1+n\alpha)\tau^{\alpha-1}P^{(1,1/\alpha)}_{n-1}\left(2\left(\frac{\tau}{T}\right)^{\alpha}-1\right). \tag{22}$$

Substituting (22) into (21) we arrive at the representation

$$D^{\alpha}_{\star}L_n(t;\alpha) = \frac{1+n\alpha}{\Gamma(1-\alpha)T^{\alpha}}\int_0^t (t-\tau)^{-\alpha}P^{(1,1/\alpha)}_{n-1}\left(2\left(\frac{\tau}{T}\right)^{\alpha}-1\right)\tau^{\alpha-1}\mathrm{d}\tau. \tag{23}$$

Formula (20) follows now from (23) by applying the change of variables

$$\tau = tx^{1/\alpha}, \quad x \in [0,1]. \quad \square$$

To calculate the integral on the right-hand side of (20), N-point Gaussian quadrature rule

$$\int_0^1 (1-t^{1/\alpha})^{-\alpha}f(t)\mathrm{d}t = \sum_{k=1}^{N}\omega_k^{(\alpha)}f(\tau_k^{(\alpha)}), \qquad f \in \mathbb{P}_{2N-1} \tag{24}$$

is used. Unfortunately, the weight function $w(t;\alpha) := (1-t^{1/\alpha})^{-\alpha}$ is a nonclassical one and no explicit formulae are known for $\omega_k^{(\alpha)}$ and $\tau_k^{(\alpha)}$. But we can use the Chebyshev and Golub–Welsch algorithms to calculate the nodes and weights in (24) as discussed in 2nd Section. Let us note that the quadrature rule (24) with $N = \lceil n/2 \rceil$ becomes exact for computing the integral in (20). After obtaining the nodes $\tau_k^{(\alpha)}$ and weights $\omega_k^{(\alpha)}$, the fractional derivative $D^{\alpha}_{\star}L_n(t;\alpha)$ can be computed by using the formula

$$D^{\alpha}_{\star}L_n(t;\alpha) = \frac{1+n\alpha}{\alpha\Gamma(1-\alpha)T^{\alpha}}\sum_{k=1}^{\lceil n/2\rceil}\omega_k^{(\alpha)}P^{(1,1/\alpha)}_{n-1}\left(2\left(\frac{t}{T}\right)^{\alpha}\tau_k^{(\alpha)}-1\right). \tag{25}$$

Now, in the following three steps, we elaborate the computation of the nodes $\tau_k^{(\alpha)}$ and weights $\omega_k^{(\alpha)}$ in (25).
Step 1. Determine the moments μ_r given by (4). This can be done by using the following theorem.

Theorem 4.2. *For the moments μ_r that correspond to the weight $w(t;\alpha) = (1-t^{1/\alpha})^{-\alpha}$ the formula*

$$\mu_r = \alpha\mathrm{B}(\alpha r+\alpha, 1-\alpha) \tag{26}$$

is valid, where $\mathrm{B}(\cdot,\cdot)$ is the Beta function.

Proof. From (4), we have

$$\mu_r = \int_0^1 t^r w(t;\alpha)\mathrm{d}t = \int_0^1 t^r (1-t^{1/\alpha})^{-\alpha}\mathrm{d}t. \tag{27}$$

If we set $t = z^\alpha$, Eq. (27) becomes

$$\mu_r = \alpha \int_0^1 z^{\alpha r+\alpha-1}(1-z)^{-\alpha}\mathrm{d}z.$$

By noting that [4]

$$\mathrm{B}(p,q) = \int_0^1 z^{p-1}(1-z)^{q-1}\mathrm{d}z, \quad p,q>0,$$

the proof is completed. □

Step 2. Calculate the coefficients $\alpha_k^{(\alpha)}$ and $\beta_k^{(\alpha)}$ in the three-term recurrence relation (5) by using the moments (26) and the *Chebyshev algorithm*. Unfortunately, the use of moments is numerically problematic inasmuch as they give rise to severe ill-conditioning [28]. Thus, in order to obtain the double precision results at this step, the authors have performed the computation in MAPLE with about 40 digits operations.

Step 3. Construct the tridiagonal Jacobi matrix $\mathbf{J}_N$ as mentioned in (13) by using the coefficients $\alpha_k^{(\alpha)}$ and $\beta_k^{(\alpha)}$. The Jacobi matrix allows then to compute the nodes and the weights in (25).

5. The collocation method

In this section, the collocation method [23,21] is applied for solving an initial value problem for a nonlinear fractional differential equation

$$D_\star^\alpha y(t) = f(t, y(t)), \quad t \in (0, T], \tag{28}$$

$$y(0) = y_0. \tag{29}$$

Under certain conditions on the function f, the initial value problem (28)–(29) possesses unique solution y in an appropriate space of functions (see e.g. [8,18] for details). Numerical evaluation of this solution is the aim of this section. At first, the solution y is approximated by $\tilde{y}_n \in \mathbb{M}_{n,\alpha}$ as the finite sum

$$\tilde{y}_n(t) := \sum_{k=0}^{n} a_k L_k(t;\alpha), \tag{30}$$

where a_k are unknown coefficients. It is worthwhile to note that if $\tilde{y}_n \in \mathbb{M}_{n,\alpha}$, then $D_\star^\alpha \tilde{y}_n$ belongs to $\mathbb{M}_{n,\alpha}$, too. This key property is crucial for application of the collocation method to the initial value problem (28)–(29).

The unknown coefficients a_k in approximation (30) are obtained from the initial condition

$$\tilde{y}_n(0) = y_0, \tag{31}$$

and the fact that $\tilde{y}_n(t)$ should satisfy the fractional differential equation in some suitably chosen collocation points θ_i, $i = 1, \ldots, n$. More precisely, the relations

$$D_\star^\alpha \tilde{y}_n(\theta_i) = f(\theta_i, \tilde{y}_n(\theta_i)), \quad i = 1, \ldots, n \tag{32}$$

have to be valid.

It is a well-established fact, that a proper choice of collocation points is crucial for the accuracy of the numerical solution and for its computational stability [23,22]. In our case, a particularly convenient choice for the collocation points θ_i is $\theta_i = t_i^{1/\alpha}$, $i = 1, \ldots, n$, where t_i are Chebyshev–Gauss–Lobatto points associated with the interval $[0, T]$, i.e.,

$$t_i = \frac{T}{2} - \frac{T}{2}\cos\frac{\pi i}{n}, \quad i = 0, \ldots, n.$$

Substituting (30) into (31), the equation

$$g_0(a_0, \ldots, a_n) := \sum_{k=0}^{n} a_k L_k(0;\alpha) - y_0 = 0 \tag{33}$$

is obtained. In its turn, Eq. (32) can be presented in form of n algebraic equations

$$g_i(a_0, \ldots, a_n) := \sum_{k=0}^{n} a_k D_\star^\alpha L_k(\theta_i;\alpha) - f\left(\theta_i, \sum_{k=0}^{n} a_k L_k(\theta_i;\alpha)\right) = 0, \quad i = 1, \ldots, n. \tag{34}$$

Note that $D^{\alpha}_{\star}L_k(\theta_i;\alpha)$ in (34) can be computed by using (25). The Eqs. (34) and (33) are nothing else than a system of $(n+1)$ equations for the $(n+1)$ unknown coefficients a_k that can be written in the form

$$\mathbf{G}(\mathbf{a}) = \mathbf{0}, \tag{35}$$

where $\mathbf{a} = [a_0, a_1, \ldots, a_n]^T$ and $\mathbf{G} : \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}$ is defined as

$$\mathbf{G}(\mathbf{a}) = \begin{pmatrix} g_0(a_0, \ldots, a_n) \\ g_1(a_0, \ldots, a_n) \\ \vdots \\ g_n(a_0, \ldots, a_n) \end{pmatrix}.$$

In the case of linear fractional differential equations, (35) becomes a linear system. The coefficients a_k are obtained by solving this system of algebraic equations with one of the known methods. Substituting them into (30) leads to an approximated solution of the fractional initial value problems (28)–(29).

6. Numerical results

This section is devoted to presentation of some numerical simulations obtained by applying the collocation method. The algorithm for numerical approximation of solutions to the initial value problems for the fractional differential equations that we discussed in the previous sections was implemented with MATLAB. In the case of nonlinear equations, the MATLAB function fsolve was used for solving the nonlinear system (35). For generating the nodes and weights in (25) the OPQ MATLAB routines written by Gautschi [28] were employed.

In the case the exact solution y to a problem is known, the dependence of approximation errors on the discretization parameter n was estimated in 2-norm

$$e_n = \sqrt{\sum_{k=0}^{n} (\tilde{y}_n(\theta_k) - y(\theta_k))^2},$$

where $\tilde{y}_n$ is the approximated solution corresponding to the discretization parameter n.

Example 6.1. We start with a simple linear problem [4]

$$D^{\alpha}_{\star}y(t) + y(t) = 0, \qquad y(0) = 1 \quad 0 < \alpha \leq 1.$$

The exact solution for this problem is given by $y(t) = E_{\alpha}(-t^{\alpha})$, where $E_{\alpha}(z)$ is the Mittag-Leffler function defined by the power series

$$E_{\alpha}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(k\alpha + 1)}, \quad \alpha > 0.$$

Algorithms for evaluation of the Mittag-Leffler function were presented e.g. in [35–38]. In our simulations, we used the Matlab routines from [37] implemented according to the algorithm suggested in [35]. In Fig. 2, the approximation error e_n as a function of the discretization parameter n for $\alpha = 0.25, 0.50, 0.75$ and $T = 1$ is plotted in a semi-log coordinate system. As expected, the error e_n shows an exponential decay that corresponds to its linear dependence from the discretization parameter n. This result demonstrates the spectral accuracy of the collocation method.

Example 6.2. Our next example corresponds to the case when the solution y has a smooth derivative of order $0 < \alpha < 1$ [8,19]. Specifically, we consider the equation

$$D^{\alpha}_{\star}y(t) = \frac{40\,320}{\Gamma(9-\alpha)}t^{8-\alpha} - 3\frac{\Gamma(5+\alpha/2)}{\Gamma(5-\alpha/2)}t^{4-\alpha/2} + \frac{9}{4}\Gamma(\alpha+1) + \left(\frac{3}{2}t^{\alpha/2} - t^4\right)^3 - [y(t)]^{3/2}$$

with a nonlinear and nonsmooth right-hand side. The analytical solution subject to the initial condition $y(0) = 0$ is given by

$$y(t) = t^8 - 3t^{4+\alpha/2} + \frac{9}{4}t^{\alpha}.$$

The approximation error e_n for $\alpha = 0.25, 0.50, 0.75$ and $T = 1$ is plotted in Fig. 3. As we can see, e_n decreases rapidly until it takes a value about 10^{-15}. This error can be explained by the computer round-off errors that prevent any further accuracy improvements.

In Fig. 4, the approximated solutions obtained by our method for $\alpha = 0.50$ and $n = 6, 12$ as well as the exact solution are plotted. As one can see, the collocation method provides accurate results even with a small number of nodes ($n = 12$ in the example).

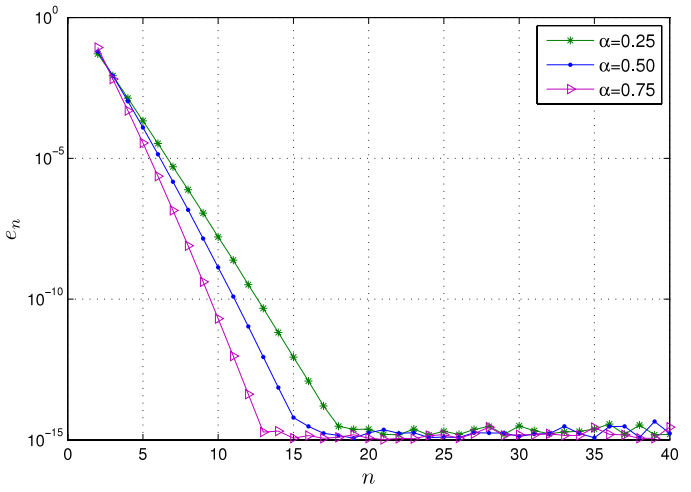

Fig. 2. Approximation error e_n versus n for various values of α (Example 6.1).

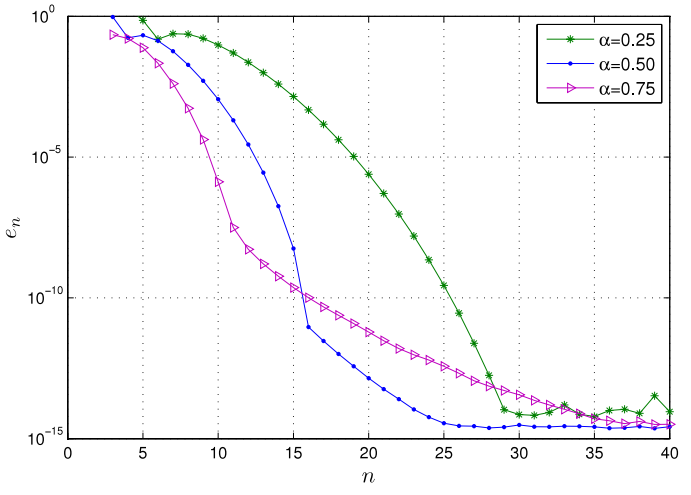

Fig. 3. Approximation error e_n versus n for various values of α (Example 6.2).

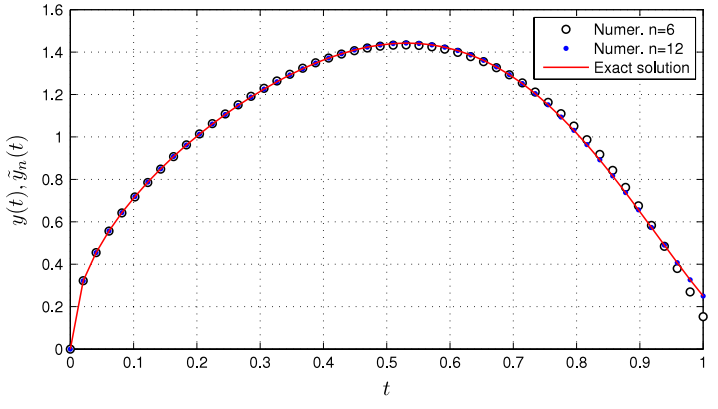

Fig. 4. Approximated solutions for $\alpha = 0.5$ with $n = 6$, 12 nodes and the exact solution (Example 6.2).

Example 6.3. Next the fractional Riccati equation [39]

$$D^{\alpha}_{\star} y(t) = 1 + 2y(t) - [y(t)]^2, \quad 0 < \alpha \leq 1,$$

subject to the initial condition $y(0) = 0$ is considered.

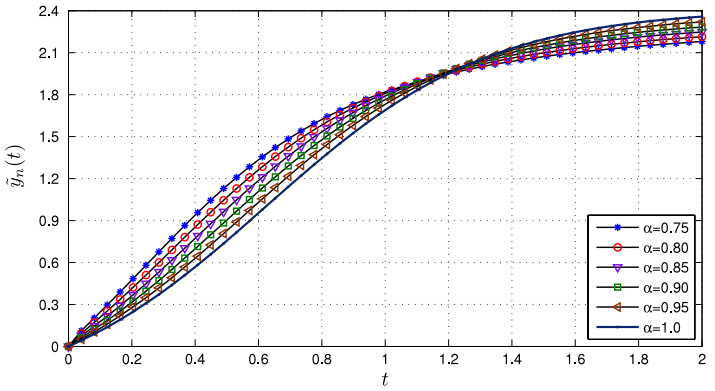

Fig. 5. Approximated solution for $\alpha = 0.75, 0.80, 0.85, 0.90, 0.95$ with $n = 15$ nodes and the exact solution for $\alpha = 1.0$ (Example 6.3).

Table 1
Approximated solution for $\alpha = 0.5$ and with $n = 10, 20, 30, 40$ nodes (Example 6.3). Correct decimal places of the approximations are highlighted in the bold face.

t	$n = 10$	$n = 20$	$n = 30$	$n = 40$
0.2	**0.93**32	**0.9331**83	**0.933182830**	**0.933182830272**0
0.4	**1.34**66	**1.3466**49	**1.346650336**	**1.346650335369**3
0.6	**1.57**05	**1.5705**65	**1.570564952**	**1.570564952065**3
0.8	**1.70**69	**1.7068**75	**1.706874594**	**1.706874593776**6
1.0	**1.79**82	**1.7982**14	**1.798214731**	**1.798214731107**9
1.2	**1.86**37	**1.8638**72	**1.863872124**	**1.863872123621**2
1.4	**1.91**35	**1.9135**59	**1.913559556**	**1.913559555523**1
1.6	**1.95**28	**1.9526**43	**1.952642348**	**1.952642348486**6
1.8	**1.98**40	**1.9843**13	**1.984313450**	**1.984313449952**8
2.0	**2.01**16	**2.0105**93	**2.010589863**	**2.010589863573**2

The approximated solutions are evaluated for $\alpha = 0.75, 0.80, 0.85, 0.90, 0.95$ with $n = 15$ nodes. The results of the numerical simulations are plotted in Fig. 5. In contrast to a fractional α $(0 < \alpha < 1)$, for $\alpha = 1$ the exact solution of the Riccati equation with the initial condition $y(0) = 0$ is known:

$$y(t) = 1 + \sqrt{2}\tanh\left(\sqrt{2}t + \frac{1}{2}\ln\left(\frac{\sqrt{2}-1}{\sqrt{2}+1}\right)\right).$$

This exact solution is plotted in Fig. 5, too. One can see that as α approaches 1, the numerical solution converges to that of the Riccati equation with $\alpha = 1$. In Table 1, numerical results for the problem from Example 6.3 for $\alpha = 0.5$ obtained by the collocation method with various values of n in some points $t \in [0, 2]$ are given. The accuracy and convergence of the method is illustrated by highlighting all correct decimal places of the approximations in the bold face.

Example 6.4. The final example is the nonlinear equation [8]

$$D^{\alpha}_{\star}y(t) = (0.5 - t)\sin y(t) + 0.8t^3, \quad 0 < \alpha \leq 1,$$

subject to the initial condition $y(0) = y_0$.

The approximated solutions for $\alpha = 0.28$ with $n = 15$ nodes are plotted in Fig. 6. The obtained numerical results are in very good agreement with the results presented in [8]. In Table 2, numerical results for the problem from Example 6.4 for $\alpha = 0.28$ and the initial condition $y(0) = 1.6$ obtained by the collocation method with various values of n in some points $t \in [0, 1]$ are given. The accuracy and convergence of the method is illustrated by highlighting all correct decimal places of the approximations in the bold face.

7. Conclusions

In this paper, the collocation method has been adopted for numerical solution of the initial value problems for fractional differential equations. A special family of the Müntz–Legendre polynomials was used as an approximation basis. The collocation method is characterized by its simplicity, efficiency, and high accuracy. This method can be readily implemented, too, especially in the framework of the so called computer algebra systems. The accuracy and validity of the presented method were demonstrated through numerical simulations for a number of examples. Of course, obtaining some theoretical estimates for the approximation errors would be desirable. This work is currently in progress. Another direction for further

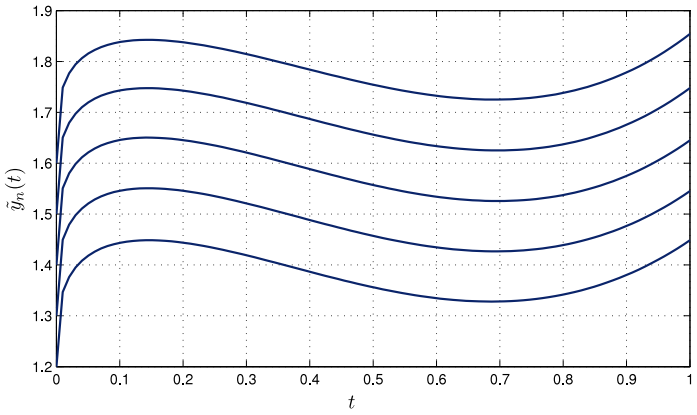

Fig. 6. Approximated solutions for $\alpha = 0.28$ with $n = 15$ nodes with five different initial values $y(0) = 1.2, 1.3, 1.4, 1.5, 1.6$ (Example 6.4).

Table 2
Approximated solution for $\alpha = 0.28$, $y(0) = 1.6$ and with $n = 10, 20, 30, 40$ nodes (Example 6.4). Correct decimal places of the approximations are highlighted in the bold face.

t	$n = 10$	$n = 20$	$n = 30$	$n = 40$
0.1	**1.83**82	**1.83820**86	**1.8382086235**	**1.83820862358424**
0.2	**1.83**77	**1.83773**83	**1.8377383455**	**1.83773834555485**
0.3	**1.81**45	**1.81459**65	**1.8145965052**	**1.81459650519912**
0.4	**1.78**40	**1.78399**32	**1.7839932850**	**1.78399328504674**
0.5	**1.75**43	**1.75438**45	**1.7543845332**	**1.75438453330184**
0.6	**1.73**25	**1.73263**31	**1.7326330909**	**1.73263309096366**
0.7	**1.72**50	**1.72512**77	**1.7251277702**	**1.72512777030217**
0.8	**1.73**82	**1.73819**01	**1.7381899697**	**1.73818996972004**
0.9	**1.77**87	**1.77852**88	**1.7785290416**	**1.77852904152860**
1.0	**1.85**32	**1.85418**28	**1.8541838725**	**1.85418387310940**

research would be to extend the presented method to the systems of fractional differential equations, the so called multi-term fractional differential equations, and fractional differential equations of the distributed order.

References

[1] B. Ross, The development of fractional calculus 1695–1900, Hist. Math. 4 (1) (1977) 75–89.
[2] R. Hilfer (Ed.), Applications of Fractional Calculus in Physics, World Scientific Publishing Company, Singapore, 2000.
[3] K.B. Oldham, J. Spanier, The Fractional Calculus, Academic Press, New York, 1974.
[4] I. Podlubny, Fractional Differential Equations, Academic Press, San Diego, CA, 1999.
[5] Y.A. Rossikhin, M.V. Shitikova, Application of fractional calculus for dynamic problems of solid mechanics: novel trends and recent result, Appl. Mech. Rev. 63 (2010) 52 pages.
[6] Yu. Luchko, M. Rivero, J.J. Trujillo, M.P. Velasco, Fractional models, non-locality, and complex systems, Comput. Math. Appl. 59 (3) (2010) 1048–1056.
[7] J.T. Machado, V. Kiryakova, F. Mainardi, Recent history of fractional calculus, Commun. Nonlinear Sci. Numer. Simul. 16 (3) (2011) 1140–1153.
[8] K. Diethelm, The Analysis of Fractional Differential Equations, Springer-Verlag, Berlin, 2010.
[9] N.J. Ford, J.A. Connolly, Comparison of numerical methods for fractional differential equations, Commun. Pure Appl. Anal. 5 (2) (2006) 289–307.
[10] K. Diethelm, J.M. Ford, N.J. Ford, M. Weilbeer, Pitfalls in fast numerical solvers for fractional differential equations, J. Comput. Appl. Math. 186 (2) (2006) 482–503.
[11] O.P. Agrawal, P. Kumar, Comparison of five numerical schemes for fractional differential equations, in: Advances in Fractional Calculus, Springer-Verlag, Dordrecht, 2007, pp. 43–60.
[12] K. Diethelm, N.J. Ford, A.D. Freed, Yu. Luchko, Algorithms for the fractional calculus: a selection of numerical methods, Comput. Methods Appl. Mech. Engrg. 194 (2005) 743–773.
[13] S. Esmaeili, M. Shamsi, A pseudo-spectral scheme for the approximate solution of a family of fractional differential equations, Commun. Nonlinear Sci. Numer. Simul. 16 (9) (2011) 3646–3654.
[14] R. Garrappa, M. Popolizio, On accurate product integration rules for linear fractional differential equations, J. Comput. Appl. Math. 235 (5) (2011) 1085–1097.
[15] F. Ghoreishi, S. Yazdani, An extension of the spectral Tau method for numerical solution of multi-order fractional differential equations with convergence analysis, Comput. Math. Appl. 61 (1) (2011) 30–43.
[16] C. Li, Y. Wang, Numerical algorithm based on Adomian decomposition for fractional differential equations, Comput. Math. Appl. 57 (10) (2009) 1672–1681.
[17] C. Lubich, Discretized fractional calculus, SIAM J. Math. Anal. 17 (3) (1986) 704–719.
[18] Yu. Luchko, R. Gorenflo, An operational method for solving fractional differential equations with the Caputo derivatives, Acta Math. Vietnam. 24 (2) (1999) 207–233.
[19] A. Saadatmandi, M. Dehghan, A new operational matrix for solving fractional-order differential equations, Comput. Math. Appl. 59 (3) (2010) 1326–1336.
[20] X. Li, C. Xu, A space–time spectral method for the time fractional diffusion equation, SIAM J. Numer. Anal. 47 (3) (2009) 2108–2131.
[21] A. Quarteroni, A. Valli, Numerical Approximation of Partial Differential Equations, Springer-Verlag, Berlin, 1997.
[22] L.N. Trefethen, Spectral Methods in Matlab, SIAM, Philadelphia, 2000.

[23] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods. Fundamentals in Single Domains, Springer-Verlag, Berlin, 2006.
[24] A.H. Khater, R.S. Temsah, Numerical solutions of the generalized Kuramoto–Sivashinsky equation by Chebyshev spectral collocation methods, Comput. Math. Appl. 56 (6) (2008) 1465–1472.
[25] P. Borwein, T. Erdélyi, J. Zhang, Müntz systems and orthogonal Müntz–Legendre polynomials, Trans. Amer. Math. Soc. 342 (2) (1994) 523–542.
[26] P.C. McCarthy, J.E. Sayre, B.L.R. Shawyer, Generalized legendre polynomials, J. Math. Anal. Appl. 177 (1993) 530–537.
[27] G.V. Milovanović, Müntz orthogonal polynomials and their numerical evaluation, in: Applications and Computation of Orthogonal Polynomials, in: Internat. Ser. Numer. Math., vol. 131, Birkhäuser, Basel, 1999, pp. 179–194.
[28] W. Gautschi, Orthogonal Polynomials: Computation and Approximation, Oxford University Press, New York, 2004.
[29] G. Szegő, Orthogonal Polynomials, 4th ed., AMS Colloq. Publ, 1975.
[30] W. Gautschi, On generating orthogonal polynomials, SIAM J. Sci. Stat. Comput. 3 (3) (1982) 289–317.
[31] G.H. Golub, J.H. Welsch, Calculation of Gauss quadrature rules, Math. Comp. 23 (1969) 221–230.
[32] G.H. Golub, G. Meurant, Matrices, Moments and Quadrature with Applications, Princeton University Press, Princeton, NJ, 2010.
[33] E.L. Ortiz, A. Pinkus, Herman Müntz: a mathematician's odyssey, Math. Intelligencer 27 (1) (2005) 22–31.
[34] E.W. Cheney, Introduction to Approximation Theory, AMS Chelsea Publishing, Providence, RI, 1998, Reprint of the second (1982) edition.
[35] R. Gorenflo, J. Loutchko, Yu. Luchko, Computation of the Mittag-Leffler function and its derivatives, Fract. Calc. Appl. Anal. 5 (2002) 491–518.
[36] V. Kiryakova, The multi-index Mittag-Leffler functions as an important class of special functions of fractional calculus, Comput. Math. Appl. 59 (5) (2010) 1885–1895.
[37] I. Podlubny, M. Kacenak, MATLAB implementation of the Mittag-Leffler function, 2005–2009. http://www.mathworks.com.
[38] H. Seybold, R. Hilfer, Numerical algorithm for calculating the generalized Mittag-Leffler function, SIAM J. Numer. Anal. 47 (1) (2008) 69–88.
[39] Z. Odibat, S. Momani, Modified homotopy perturbation method: application to quadratic Riccati differential equation of fractional order, Chaos Solitons Fractals 36 (1) (2008) 167–174.