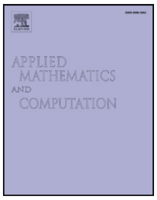
Applied Mathematics and Computation 269 (2015) 196–216

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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



APPLIED MATHEMATICS AND COMPUTATION

Efficient algorithms for solving the fractional ordinary differential equations

Jingwei Deng[a,b], Lijing Zhao[a,*], Yujiang Wu[a]

[a] *School of Mathematics and Statistics, Gansu Key Laboratory of Applied Mathematics and Complex Systems, Lanzhou University, Lanzhou 730000, PR China*

[b] *School of Mathematics and Computer Science, Northwest University for Nationalities, Lanzhou 730030, PR China*

ARTICLE INFO

Keywords:
Fractional ordinary differential equation
Predictor-corrector approach
Short memory principle
Equidistributing meshes

ABSTRACT

Fractional calculus and fractional differential equations are popular in describing anomalous diffusion, ground water flow and transport, and the price fluctuation in finance, etc. Some numerical methods are developed to solve the fractional ordinary differential equations. However, for most of these methods it seems that we always have to make a trade-off between efficiency and accuracy because of the non-local properties of fractional operators. In other words, for ensuring the accuracy, usually the computation cost is hard to accept; on the other hand if the computation cost is reduced then the accuracy is greatly lost. Based on the idea of equidistributing meshes, this paper designs efficient numerical schemes, which have linearly increasing computation cost with time t but not losing the accuracy at the same time. Error estimates for the proposed schemes are performed; and the numerical examples demonstrate the efficacy of our algorithms.

1. Introduction

The notion of fractional calculus goes back to the Leibniz's note in his list to L'Hospital [3,15,19], dated Sep. 30, 1695. In the last two decades, fractional calculus has gained increasing attention in the development of the mathematical models in statistical physics, mathematical finance, polymer modeling, and the rich fractional dynamics are also disclosed, such as, anomalous diffusion, synchronization of chaos, and multi-directional multi-scroll attractors [1,2,4–6,18].

There are already some progresses for developing the numerical methods of fractional (ordinary or partial) differential equations including predictor–corrector approach, finite difference method, and even the Monte Carlo approach [7,8,11,12,16,17]. Comparing with the classical differential equations, one of the big challenges we have to face is the expensiveness of its computation cost besides its complexity, since fractional operators are pseudodifferential operators which are non-local [20]; most of the time, the computation cost is proportional to t^2 for the time dependent problems. The algorithms proposed in this paper have computation cost being linearly increasing with the time t.

We focus on discussing the numerical method of the following time dependent equation

$$D_*^\alpha x(t) = f(t, x(t)), \quad 0 < t < T, \tag{1.1}$$

[*] Corresponding author. Tel.: +86 13919084849.
E-mail addresses: djw_1205@163.com (J. Deng), zhaolj10@lzu.edu.cn, zhaoqin198803@126.com (L. Zhao), myjaw@lzu.edu.cn (Y. Wu).

http://dx.doi.org/10.1016/j.amc.2015.07.048
0096-3003/© 2015 Elsevier Inc. All rights reserved.

with the initial conditions

$$x^{(k)}(0) = x_0^{(k)}, \quad k = 0, \ 1, \ldots, \lceil\alpha\rceil - 1, \tag{1.2}$$

with $\alpha \in (0, \infty)$, $x_0^{(k)}$ are arbitrary real numbers, and D_*^α denotes the fractional derivative in the Caputo sense, defined by

$$D_*^\alpha y(t) = D^{-(n-\alpha)} D^n y(t),$$

where $n = \lceil\alpha\rceil$ is the smallest integer not less than α, D^n is the classical derivative of order n, and $D^{-(n-\alpha)}$ is the Riemann–Liouville integral operator of order $n - \alpha$, described by

$$D^{-(n-\alpha)} y(t) = \frac{1}{\Gamma(n-\alpha)} \int_0^t (t-\tau)^{n-\alpha-1} y(\tau) d\tau.$$

The pioneer method for the numerical solution of (1.1)–(1.2) is the predictor–corrector approach, the widely used method in engineering and physical community, raised by Diethelm et al. [10–12]. This method is improved in [8] (or see the review article [9]), where almost half of the computation cost is reduced and the computation accuracy is improved from $O(h^{\min\{1+\alpha,2\}})$ to $O(h^{\min\{1+2\alpha,2\}})$, but the computation expenditure is still $O(h^{-2})$, which also means that the computation cost is proportional to t^2. Along the direction of reducing the computation cost, some efforts have been made using the nested meshes which arebased on the fixed memory principle [19] and the short memory principle [14] of fractional operator when $\alpha \in (0, 1)$ in (1.1), and the short memory principle is apprehended from a new point of view in [7] where the range can be extended to $\alpha \in (0, 2)$. After applying the nested meshes, the computation cost is reduced from $O(h^{-2})$ to $O(h^{-1} \log(h^{-1}))$ not losing the numerical accuracy, but we have to say this is more theoretical claims rather than numerical practices; in fact the numerical experiments in [7,14] show this. More recently, another very novel idea, so called Jacobian-predictor-corrector approach, is introduced by Zhao and Deng [22], which greatly improves the accuracy while having the linearly increasing computation cost.

In this paper, we dig out the potential of the short memory principle of fractional operators and apply it felicitously to reduce the computation cost by using the idea of equidistributing meshes [21] for numerically solving the initial value problem (1.1) and (1.2); numerical experiments show that the provided algorithms have linearly increasing computation cost with time t, like numerically solving the classical time dependent problem, even though we are computing the nonlocal one. In the next section, we introduce the techniques of equidistributing the meshes and present the detailed numerical schemes. Numerical examples to illustrate the efficacy of the algorithm and conclusions are given in Sections 3 and 4, respectively.

2. Algorithms with equidistributing meshes

The initial value problem (1.1)–(1.2) is equivalent to the following Volterra integral equation [10]:

$$\begin{aligned} x(t) &= \sum_{k=0}^{\lceil\alpha\rceil-1} x_0^{(k)} \frac{t^k}{k!} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, x(\tau)) d\tau \\ &:= x_0(t) + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, x(\tau)) d\tau, \end{aligned} \tag{2.1}$$

where $x_0(t) := \sum_{k=0}^{\lceil\alpha\rceil-1} x_0^{(k)} \frac{t^k}{k!}$. For uniform nodes $t_{n+1} = (n+1)h$, $n = 0, 1, \ldots, N$, with $h = T/N$ being the steplength of numerical computation, the above equation can be recast as [7,8]

$$x(t_{n+1}) = x_0(t_{n+1}) + \frac{1}{\Gamma(\alpha)} \int_{t_n}^{t_{n+1}} (t_{n+1}-\tau)^{\alpha-1} f(\tau, x(\tau)) d\tau + \frac{1}{\Gamma(\alpha)} \int_0^{t_n} (t_{n+1}-\tau)^{\alpha-1} f(\tau, x(\tau)) d\tau, \tag{2.2}$$

or

$$\begin{aligned} x(t_{n+1}) &= x(t_n) + x_0(t_{n+1}) - x_0(t_n) + \frac{1}{\Gamma(\alpha)} \int_{t_n}^{t_{n+1}} (t_{n+1}-\tau)^{\alpha-1} f(\tau, x(\tau)) d\tau \\ &\quad + \frac{1}{\Gamma(\alpha)} \int_0^{t_n} ((t_{n+1}-\tau)^{\alpha-1} - (t_n-\tau)^{\alpha-1}) f(\tau, x(\tau)) d\tau. \end{aligned} \tag{2.3}$$

For the single step integral $\int_{t_n}^{t_{n+1}} (t_{n+1}-\tau)^{\alpha-1} g(\tau) d\tau$, we can employ the rectangle or trapezoidal quadrature formula to approximate it, i.e.,

$$\int_{t_n}^{t_{n+1}} (t_{n+1}-\tau)^{\alpha-1} g(\tau) d\tau \approx \int_{t_n}^{t_{n+1}} (t_{n+1}-\tau)^{\alpha-1} g(t_n) d\tau = \frac{h^\alpha}{\alpha} g(t_n), \tag{2.4}$$

or

$$\begin{aligned} \int_{t_n}^{t_{n+1}} (t_{n+1}-\tau)^{\alpha-1} g(\tau) d\tau &\approx \int_{t_n}^{t_{n+1}} (t_{n+1}-\tau)^{\alpha-1} \frac{g(t_{n+1})(\tau - t_n) + g(t_n)(t_{n+1}-\tau)}{h} d\tau \\ &= \frac{h^\alpha}{\alpha(\alpha+1)} [\alpha g(t_n) + g(t_{n+1})]. \end{aligned} \tag{2.5}$$