East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.080418.200618

Vol. **8**, No. 4, pp. 656-677
November 2018

A Fast Algorithm for the Caputo Fractional Derivative

Kun Wang[1] and Jizu Huang[2,3,*]

[1]*College of Mathematics and Statistics, Chongqing University, Chongqing, 401331, China.*
[2]*LSEC, ICMSEC, Institute of Computational Mathematics and Scientific/ Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China.*
[3]*School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China.*

Received 8 April 2018; Accepted (in revised version) 20 June 2018.

Abstract. A fast algorithm with almost optimal memory for the computation of Caputo's fractional derivative is developed. It is based on a nonuniform splitting of the time interval $[0, t_n]$ and a polynomial approximation of the kernel function $(1-\tau)^{-\alpha}$. Both the storage requirements and the computational cost are reduced from $\mathcal{O}(n)$ to $(K+1)\mathcal{O}(\log n)$ with K being the degree of the approximated polynomial. The algorithm is applied to linear and nonlinear fractional diffusion equations. Numerical results show that this scheme and the corresponding direct methods have the same order of convergence but the method proposed performs better in terms of computational time.

AMS subject classifications: 35R11, 26A33, 33F05

Key words: Caputo derivative, fast algorithm, polynomial approximation, error, graded mesh.

1. Introduction

Fractional differential equations become a popular tool in various applications. They can properly reflect physical processes in biology, ecology and control systems [11–14, 18, 19, 28, 31–33]. The Riemann–Liouville [3, 4] and the Caputo [11, 33, 37, 39, 42] fractional derivatives are commonly used and the latter is better suited to work with fractional partial differential equations (PDEs), since the former is connected with the initial conditions containing limit values of the fractional derivatives at $t = 0$, the physical meaning of which is not quite clear. We consider a fast method for PDEs with the Caputo fractional derivative

*Corresponding author. *Email addresses:* `kunwang@cqu.edu.cn` (K. Wang), `huangjz@lsec.cc.ac.cn` (J.Z. Huang)

$$ {}_{0}^{C}\mathscr{D}_{t}^{\alpha}u(t) = \frac{1}{\Gamma(1-\alpha)} \int_{0}^{t} \frac{u'(\tau)}{(t-\tau)^{\alpha}} d\tau, \quad 0 < \alpha < 1, \tag{1.1} $$

where Γ is the Gamma function and $t \in [0, T]$ with a finite time T.

A popular approach to discretise the fractional derivative (1.1), called $L1$ formula [9, 20, 37], consists in piecewise linear approximation of the integrand $u(t)$ on every chosen subinterval. For $0 < \alpha < 1$, this scheme enjoys $2-\alpha$ order of convergence. Other methods, including the Crank-Nicolson-Type discretisation [42] and the matrix transfer technique [39], with the convergence rate $2-\alpha$ have been also studied. Moreover, Zeng *et al.* [40] constructed a second order unconditionally stable scheme by using fractional linear multistep methods for the Caputo derivative approximation. Gao and Sun [8] proposed an $L1-2$ formula, based on piecewise quadratic interpolation, and achieved $3-\alpha$ accuracy order and Cao *et al.* [2] improved the convergence order to $3+\alpha$. On the other hand, Li and Xu [18] considered a numerical scheme with the spectral accuracy order but this direct method has to keep all the previous solutions and requires $\mathscr{O}(n)$ storage capacity and $\mathscr{O}(n)$ flops at the n-th time step. Therefore, for long time large scale simulations of fractional PDEs, efficient and reliable fast methods are needed. To keep the memory and reduce the computational cost, several fast methods have been developed. Thus Lubich and Schädle [22] introduced a fast convolution method, where the kernel function in the Caputo derivative was represented via the inverse Laplace transform and then calculated — cf. [21, 34, 41]. The storage requirements and the computational cost of such methods are $\mathscr{O}(\log n)$. Ren *et al.* [33] used the Laplace transform method to change the fractional differential equation into an approximate local problem, and Li [17] employed the Gauss–Legendre quadratures to construct a fast algorithm based on the representation

$$ t^{\alpha-1} = \frac{1}{\Gamma(\alpha)\Gamma(1-\alpha)} \int_{0}^{\infty} e^{-\xi t} \xi^{-\alpha} d\xi. $$

In the case $T >> 1$, Jiang *et al.* [11] used a combination of the Gauss–Jacobi and Gauss–Legendre quadratures to improve this method to $\mathscr{O}(\log n)$ storage requirements and $\mathscr{O}(\log n)$ computational cost. The fast scheme turned out to have $2-\alpha$ convergence rate and to be unconditionally stable [11]. Further on, to solve fractional diffusion equations Yan *et al.* [38] employed an $L2-1_{\sigma}$ formula, achieving the second-order accuracy. McLean [26] approximated fractional kernel by degenerate kernels and Baffet and Hesthaven [1] used a kernel compression to discretise the corresponding fractional integral operator.

In this paper, we introduce a fast algorithm with almost optimal memory for the Caputo fractional derivative, which has the same order of convergence as a direct method. At each time step, the fractional derivative is decomposed into local and history parts. The local part, represented by an integral over interval $[t_{n-1}, t_n]$, is calculated by a direct method. To evaluate history part, we split the interval $[0, t_{n-1}]$ into nonuniform subintervals. The corresponding integrals depend on t_n, which causes difficulties in fast calculations. To overcome these difficulties, we approximate the kernel function by a polynomial of K-th degree. The convolutions of the integral of $u'(t)$ with different polynomial basis functions