Far Eastern Mathematical Journal. 2022. V. 22. No 2. P. 245–251

UDC 519.63
MSC2020 35R11 + 65Y05

© M. A. Sultanov[1], V. E. Misilov[2], Y. Nurlanuly[1]

Efficient Parareal algorithm for solving time-fractional diffusion equation

The work is devoted to developing efficient parallel algorithms for solving the initial boundary problem for the time-fractional diffusion equation. Traditional approaches to parallelization are based on the space domain decomposition. In contrast, the parareal method is based on the time domain decomposition and an iterative predictor-corrector procedure. The fast solver on a coarse grid is used to construct the initial approximations for subtasks (solved by accurate solvers on finer grids) and for correcting the solutions of subtasks. The subtasks may be solved independently for each subinterval of time. This allows one to implement the efficient parallel algorithms for various high-performance architectures. Currently, this method is widely used for problems for classical differential equations with integer orders. But it is much less commonly used for the fractional equations. In this work, the parareal algorithm for solving the initial boundary problem for the time-fractional diffusion equation is implemented using the OpenMP technology for multicore processors. The numerical experiments are performed to estimate the efficiency of parallel implementation and compare the parareal algorithm with the traditional space domain decomposition.

Key words: *Caputo fractional derivative, time-fractional diffusion equation, parallel computing, parareal method.*

DOI: https://doi.org/10.47910/FEMJ202233

Introduction

Fractional calculus gained large interest recently [1,2], as it allows one to develop mathematical models for various physical processes with memory and nonlocality effects, such as anomalous diffusion [3].

[1] Kh. Yasavi International Kazakh-Turkish University, Kazakhstan, 487010, Turkestan, pl. Esim-Khan, 2.
[2] N. N. Krasovskii Institute of Mathematics and Mechanics of the Ural Branch of the Russian Academy of Sciences, Russia, 620990, Ekaterinburg, S. Kovalevskaja st., 16.
E-mail: murat.sultanov@ayu.edu.kz (M. A. Sultanov), v.e.misilov@urfu.ru (V. E. Misilov), yerkebulan.nurlanuly@ayu.edu.kz (Y. Nurlanuly).

The numerical methods for solving the direct and inverse problems for fractional differential equations are usually more computationally expensive than for classical differential equation, as they require storing and utilizing the entire history of the process. This makes importance of the parallel computing in implementing the numerical algorithms for such problems.

One of the common problems in this field is the initial-boundary problem for the time-fractional diffusion equation. Several parallel algorithms are designed specifically for this problem [4–6]. Usually, these algorithms are based on space domain decomposition or using parallel algorithms for auxilliary problems, such as solving SLAEs.

In this paper, we implement the parallel algorithm for solving the time-fractional diffusion equation on the base of the parareal method [7,8]. It is based on the predictor-corrector procedure and utilization of two solvers. The former is a fast solver on a coarse grid, and the latter is a precise solver on a fine grids. The precise solver is split into subtasks that can be executed in parallel. Under suitable condition, the correction iteration will converge to a serial solution after a couple of iterations.

The paper is organized as follows. In Section 1, we present the statement of the problem. In Section 2, we describe the parareal method for a time-fractional diffusion equation. The results of numerical experiments are presented in Section 3. Section 4 concludes the paper.

1 Statement of the Problem

Consider the basis time-fractional parabolic partial differential equation in the following form:

$$\frac{\partial^\alpha U(x,t)}{\partial t^\alpha} = a(x)\frac{\partial^2 U(x,t)}{\partial x^2} + b(x)\frac{\partial U(x,t)}{\partial x} + c(x)U(x,t) + d(x,t), \tag{1}$$

where $U(x,t)$ is the sought function, $a(x), b(x), c(x), d(x,t)$ are the known functions or constants, $0 < \alpha < 1$ is the parameter defining the fractional order of the time derivative.

The problem is on the space interval $0 \le x \le \ell$ and time interval $0 \le t \le T$. The boundary and initial conditions are

$$U(0,t) = g_1(t), \quad U(\ell,t) = g_2(t), \quad 0 \le t \le T.$$
$$U(x,0) = g_0(x), \quad 0 \le x \le \ell$$

where $g_0(x), g_1(t), g_2(t)$ are the given functions.

We consider the following definition of the Caputo fractional partial derivative [9]:

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \frac{1}{\Gamma(1-\alpha)}\int\limits_0^\infty \frac{\partial u(x-s)}{\partial t}(t-s)^{-\alpha} ds. \tag{2}$$

Discretization of Equation (1) is performed by splitting the space interval $[0,\ell]$ into a grid of m points with step $h = \Delta x = \ell/m$. The time interval $[0,T]$ is split into the fine grid of N points with step $\delta t = T/N$ and a coarse grid of L points (such as N is a multiple of L) with step $\Delta T = T/L$. Then, we can denote the grid points for space

as $x_i = ih, \quad i \in \{0, 1, ..., m\}$. The grid points for time are denoted as $t_j = j\delta t, \quad j \in \{0, 1, ..., N\}$ and $T_l = l\Delta T, \quad l \in \{0, 1, ..., L\}$. Now, we can denote the values of the sought function $U(x,t)$ at the fine grid points as $\overline{U}_{i,j} = U(x_i, t_j)$, and at the coarse grid points as $\widetilde{U}_{i,l} = U(x_i, T_l)$.

For approximating the Caputo fractional partial derivative in the left-hand part of Equation (1) with time step τ, we use the first-order approximation formula

$$\begin{gathered} D_t^\alpha \overline{U}_{i,n} \cong \sigma(\alpha,\tau) \sum_{j=1}^{n} w_j^{(\alpha)} (\overline{U}_{i,n-j+1} - \overline{U}_{i,n-j}), \\ \sigma(\alpha,\tau) = \frac{1}{\Gamma(1-\alpha)(1-\alpha)\tau^\alpha}, \quad w_j^{(\alpha)} = j^{1-\alpha} - (j-1)^{1-\alpha}. \end{gathered} \tag{3}$$

After applying the implicit finite difference scheme of the second order at the grid point (x_i, t_n) (for either the fine grid or coarse grid), we obtain the difference equation

$$\begin{gathered} \sigma_{\alpha,\tau} \sum_{j=1}^{n} w_j^{(\alpha)} (U_{i,n-j+1} - U_{i,n-j}) = \\ = a_i \frac{U_{i-1,n} - 2U_{i,n} + U_{i+1,n}}{h^2} + b_i \frac{U_{i+1,n} - U_{i-1,n}}{2h} + c_i U_{i,n} + d_{i,n} \ . \end{gathered} \tag{4}$$

Then, after transforming the equation and denoting

$$\begin{gathered} p_i = \frac{a_i}{h^2} - \frac{b_i}{2h}, \quad q_i = \sigma_{\alpha,\tau} - c_i + \frac{2a_i}{h^2}, \quad r_i = \frac{a_i}{h^2} + \frac{b_i}{2h} \\ f_{i,n} = \sigma_{\alpha,\tau} \left(U_{i,n-1} - \sum_{j=2}^{n} w_j^{(\alpha)} (U_{i,n-j+1} - U_{i,n-j}) \right) + d_{i,n}, \quad n > 1, \\ f_{i,1} = \sigma_{\alpha,\tau} U_{i,0} + d_{i,0}, \end{gathered}$$

we can combine equations for all spatial points into a system of linear algebraic equations

$$AU_n = f_n, \tag{5}$$

where

$$\begin{gathered} U_n = [U_{1,n},\ U_{2,n}, ...,\ U_{m-1,n}], \\ f_n = [f_{1,n} + p_1 U_{0,n},\ f_{2,n}, ..., f_{m-2,n}, ...,\ f_{m-1,n} + r_{m-1} U_{m,n}]. \end{gathered}$$

Matrix A is a square tridiagonal matrix of $(m-1) \times (m-1)$ dimension. To solve system (5), we use the sweep method [10]. Thus, solving the initial-boundary problem for time interval $\{T_{\text{start}}; T_{\text{end}}\}$ on either fine or coarse grid is reduced to solving the systems (5) sequentially at each time level with correspondent time step $\tau = \delta t$ or ΔT.

2 Parareal method for fractional differential equations

Let us denote $\mathcal{F}(U_{\text{end}}; T_{\text{start}}; T_{\text{end}})$ as the numerical solution obtained on the fine grid, and $\mathcal{G}(U_{\text{end}}; T_{\text{start}}; T_{\text{end}})$ as the solution obtained on the coarse grid.

The parareal method for the time-fractional diffusion equation is an iterative method given as [8]

$$U_n^{k+1} = \mathcal{G}\left(U_n^{k+1}; T_0; T_n\right) + \mathcal{F}\left(U_n^k; T_0; T_n\right) - \mathcal{G}\left(U_n^k; T_0; T_n\right),$$
$$n \in \{1, 2, \ldots, N\}, k \in \{0, 1, \ldots, \}.$$

Apparently, if $\lim\limits_{k\to\infty} U_n^{k+1}$ exists, than $U_n^{k+1} \to \mathcal{F}(U_n; T_0; T_n)$, *i.e.*, the parareal solution converges to the serial solution on the fine grid.

In contrast with the parareal method for classical differential equations where the fine grid propagators integrate only over the interval $\{T_{n-1}, T_n\}$, the fractional derivatives require the entire history part of the solution. That means that integrating on the fine grid $\{T_0, T_n\}$ with step δt to obtain the $\mathcal{F}(U_n^k; T_0; T_n)$ is equivalent to the serial algorithm. Thus, the parareal method will not give us any performance advantages.

To alleviate this problem, we use the idea described in [11]. It consists in using the composite fine grid solution operators $\widetilde{\mathcal{F}}(U_n^k; T_0; T_n)$, where the latest interval $\{T_{n-1}, T_n\}$ uses the fine grid with the time step δt and the history part $\{T_0, T_{n-1}\}$ is integrated on the coarse grid with the time step ΔT.

The fine grid solutions $\widetilde{\mathcal{F}}(U_n^k; T_0; T_n)$ can be computed independently for points T_n, allowing the parallelization. In our implementation, this work is distributed between the OpenMP threads, while the coarse grid corrections $\mathcal{G}$ are performed in serial mode using '`#pragma omp master`' directive.

3 Numerical experiments

In this section, we apply our parallel implementations of the parareal algorithm to numerical solution of the time-fractional diffusion equation. The experiments were performed on 16-core Intel i7-12900k CPU. This section presents the results of the experiments.

The test problem uses the equation [6]

$$\frac{\partial^\alpha U(x,t)}{\partial t^\alpha} = \frac{\partial^2 U(x,t)}{\partial x^2} + \frac{\Gamma(4+\alpha)}{6} x^2(2-x)t^3 - 4x^2(6-5x)t^{3+\alpha},$$
$$0 < \alpha < 1, \quad 0 \le x \le 2, \quad 0 \le t \le 1,$$

with the boundary and initial conditions

$$U(0,t) = 0, \quad U(2,t) = 0, \quad 0 \le t \le T,$$
$$U(x,0) = 0, \quad 0 \le x \le \ell.$$

The exact solution of this problem is

$$U(x,t) = x^4(2-x)t^{3+\alpha}.$$

The numerical experiments were performed for the order $\alpha = 0.5$ with the grid sizes $m = 4096$, $N = 4096$.

Table 1 presents the results of numerical experiments for the test problem using the serial implementation, the parareal algorithm, and the parallel sweep method implemented

Table 1: Results of experiments for the test problem with grid $m = 4096, N = 4096$.

Method	T_1 [sec]	T_{16} [sec]	K	δ
Serial	6.6			$3 \cdot 10^{-7}$
Parallel sweep	6.6	6.6		$3 \cdot 10^{-7}$
Parareal	6.6	0.5	4	$3.8 \cdot 10^{-7}$

previously in work [6]. It contains the execution times T_1 for serial programs (which is equivalent to the parallel program with a single thread), as well as the times T_{16} for parallel programs run on 16 OpenMP threads. The last column presents the relative error $\delta = \left\| (U - \overline{\overline{U}})/U \right\|_\infty$ of the solution obtained by a given method.

These results show that the parareal method is a very promising approach. The parallel sweep method is based on the spatial decomposition, and shows good performance only for a large spatial grids. $m = 4096$ is too small, resulting in overhead preventing any speedup of parallel algorithm. In contrast, while the parareal method required 4 iteration to converge, the total speedup is significant, over 13 times using 16 threads.

4 Conclusions

In this work, the parallel algorithm for solving the initial boundary problem for the time-fractional diffusion equation is implemented on the base of the parareal method. The algorithm is based on the finite-difference scheme for approximating the differential equation and the sweep method for solving the systems of linear algebraic equations.

The parallel implementation is based on the time domain decomposition and parareal method. The algorithm is implemented for the multicore processors using the OpenMP technology. The numerical experiments were performed to investigate the efficiency of the developed parallel algorithm. The parallel algorithm reduces the computing time up to 13 times using the 16-core processor.

References

[1] J. T. Machado, A. Galhano, J. J. Trujillo, "Science metrics on fractional calculus development since 1966", *Fract. Calc. Appl. Anal.*, **16**, (2013), 479–500.

[2] M. A. Sultanov, D. K. Durdiev, A. A. Rahmonov, "Construction of an Explicit Solution of a Time-Fractional Multidimensional Differential Equation", *Mathematics*, **9**, (2021), 2052.

[3] R. Metzler, J. H. Jeon, A. G. Cherstvy, E. Barkai, "Anomalous diffusion models and their properties: Non-stationarity, non- ergodicity, and ageing at the centenary of single particle tracking", *Phys. Chem. Chem. Phys.*, **16**, (2014), 24128–24164.

[4] P. de Luca, A. Galletti, H. Ghehsareh, L. Marcellino, M. Raei, "A GPU-CUDA framework for solving a two-dimensional inverse anomalous diffusion problem", *Parallel Comput. Technol. Trends*, **36**, (2020), 311.

[5] X. Yang, L. Wu, "A New Kind of Parallel Natural Difference Method for Multi-Term Time Fractional Diffusion Model", *Mathematics*, **8**:4, (2020), 596.

[6] M. A. Sultanov, E. N. Akimova, V. E. Misilov, Y. Nurlanuly, "Parallel Direct and Iterative Methods for Solving the Time-Fractional Diffusion Equation on Multicore Processors", *Mathematics*, **10(3)**, (2022), 323.

[7] J. L. Lions, Y. Maday, G. Turinici, "A "parareal" in time discretization of PDEs", *C.R. Acad. Sci. Paris Ser. I Math.*, **332**, (2001), 661–668.

[8] Q. Xu, J. S. Hesthaven, F. Chen, "A parareal method for time-fractional differential equations", *J. Comp. Phys.*, **293**, (2015), 173–183.

[9] Y. Zhang, "A finite difference method for fractional partial differential equation", *Appl. Math. Comput.*, **215**, (2009), 524–529.

[10] A. A. Samarskii, E. S. Nikolaev, *Numerical methods for grid equations*, Volume I Direct Methods, Birkhauser Basel, Switzerland, 1989, XXXV, 242 pp.

[11] H. Fu, H. Wang, "A preconditioned fast Parareal finite difference method for space-time fractional partial differential equation", *J. Sci. Comput.*, **78**:3, (2018), 1724–1743.

Received by the editors
June 15, 2022

The first author (M.A.S.) and third author (E.N.) were financially supported by the Ministry of Education and Science of the Republic of Kazakhstan (Project AP09258836). The second author (V.E.M.) received no external funding.

Султанов М. А., Мисилов В. Е., Нурланулы Е. Эффективный алгоритм Parareal для решения уравнения диффузии с дробной производной по времени. *Дальневосточный математический журнал.* 2022. Т. 22. № 2. С. 245–251.

АННОТАЦИЯ

Статья посвящена разработке эффективных параллельных алгоритмов решения начально-краевой задачи для уравнения диффузии с дробной производной по времени. Традиционные подходы к распараллеливанию основаны на декомпозиции пространственной области. Метод Parareal, напротив, основан на декомпозиции временной области и итеративной процедуре "предиктор-корректор". Быстрый решатель на грубой сетке используется для построения начальных приближений для подзадач (решаемых точными решателями на более мелких сетках) и для корректировки решений подзадач. Подзадачи могут решаться независимо для каждого подынтервала времени. Это позволяет реализовать эффективные параллельные алгоритмы для различных высокопроизводительных архитектур. В настоящее время данный метод широко используется в задачах для классических дифференциальных уравнений с целыми порядками производных, гораздо реже используется для дробных уравнений. В данной работе алгоритм Parareal для решения начально-краевой задачи для уравнения диффузии с дробной производной по времени реализован для многоядерных процессоров с использованием технологии OpenMP. Проведены численные эксперименты для оценки эффективности параллельной реализации и сравнения алгоритма Parareal с традиционной декомпозицией в пространственной области.

Ключевые слова: *дробная производная Капуто, уравнение диффузии с дробной производной по времени, параллельные вычисления, метод Parareal.*