

# Модификация алгоритма PARAREAL для решения дифференциальных уравнений дробного порядка

С.Ю. Лукашук

Предлагается модификация алгоритма распараллеливания по временной переменной PARAREAL для численного решения дифференциальных уравнений, содержащих производные по времени дробного порядка типа Римана-Лиувилля. Приводятся описание алгоритма и результаты численных экспериментов.

## 1. Введение

В задачах численного моделирования одним из наиболее распространенных методов построения параллельных алгоритмов является метод декомпозиции расчетной области (domain decomposition). Обычно декомпозиция осуществляется по пространственным переменным, а по временной переменной алгоритм остается последовательным, отражая, тем самым, естественную эволюцию системы. Тем не менее, как показано в работе [1], метод декомпозиции может быть успешно применен и к временной переменной. Предложенный в этой работе итерационный алгоритм PARAREAL позволяет осуществлять распараллеливание по времени для задач эволюционного типа, т.е. описываемых уравнениями с первой производной по времени. Теоретические оценки точности и сходимости этого алгоритма приводятся в [2-5]. В настоящее время имеется ряд модификаций алгоритма PARAREAL для решения различных задач эволюционного типа. Так, в [6] приведена модификация этого алгоритма для решения системы уравнений Навье-Стокса, а в [7] реализован симплектический PARAREAL для решения системы уравнений Гамильтона.

В настоящей работе предлагается модификация алгоритма PARAREAL для решения задач, описываемых дифференциальными уравнениями с дробной производной по времени типа Римана-Лиувилля [8, 9]. Уравнения такого вида позволяют моделировать различные аномальные процессы переноса, в которых корреляции затухают не по экспоненциальному, а по степенному закону [10], например, процессы аномальной диффузии [11]. Поскольку дробная производная по времени отражает эффект полной памяти системы [10], вычислительная трудоемкость численного решения таких уравнений чрезвычайно высока. Кроме того, количество операций, необходимых для вычисления дробной производной по времени при численном расчете, линейно возрастает с увеличением номера временного слоя. Все это делает оправданным использование при решении уравнений с дробными производными многоядерных и многопроцессорных вычислительных систем, а также обуславливает актуальность разработки для них параллельных численных алгоритмов.

## 2. Описание алгоритма

### 2.1. Постановка задачи и последовательный алгоритм

Для упрощения теоретических выкладок приведем описание алгоритма на примере обыкновенного дифференциального уравнения дробного порядка  $\alpha \in (0,1)$ . Соответствующее обобщение на уравнения с частными производного дробного порядка не представляет принципиальной сложности.

Рассмотрим следующую задачу Коши для дифференциального уравнения дробного порядка:

$$\begin{aligned} {}_0 D_t^\alpha y &= f(t, y), \quad 0 < t \leq T, \quad 0 < \alpha < 1; \\ y(0) &= c = \text{const}. \end{aligned} \quad (1)$$

Здесь

$${}_0D_t^\alpha y \equiv \frac{1}{\Gamma(1-\alpha)} \frac{\partial}{\partial t} \int_0^t \frac{y(\tau) d\tau}{(t-\tau)^\alpha} \quad (2)$$

– производная дробного порядка типа Римана-Лиувилля [8].

При численном решении дробная производная (2) заменяется конечно-разностным аналогом, получающимся из определения дробной производной Грюнвальда-Летникова [8]:

$${}_0D_t^\alpha y \approx \frac{1}{h^\alpha} \sum_{j=0}^{\left\lfloor \frac{t}{h} \right\rfloor} (-1)^j \binom{\alpha}{j} y(t-jh), \quad (3)$$

где  $h$  – шаг сетки,

$$\binom{\alpha}{j} = \frac{(-1)^{j-1} \alpha \Gamma(j-\alpha)}{\Gamma(1-\alpha) \Gamma(j+1)}$$

– биномиальные коэффициенты.

Простейшая численная схема решения задачи (1), переходящая в схему Эйлера для обыкновенного дифференциального уравнения первого порядка (т.е. при  $\alpha = 1$ ), имеет вид [9]

$$\frac{1}{h^\alpha} \sum_{j=0}^m w_j^{(\alpha)} y_{m-j} = f_{m-1}, \quad m = 1, 2, \dots, M, \quad y_0 = c,$$

где  $t_j = jh$ ,  $t_M = Mh = T$ ,  $y_j = y(t_j)$ ,  $f_{m-1} = f(t_{m-1}, y_{m-1})$ ,  $w_j^{(\alpha)} = (-1)^j \binom{\alpha}{j}$ . Выражая отсюда значение искомой функции на текущем временном слое  $y_m$ , приходим к следующим формулам явной разностной схемы:

$$y_0 = c, \quad y_m = h^\alpha f_{m-1} - \sum_{j=1}^m w_j^{(\alpha)} y_{m-j}, \quad m = 1, 2, \dots, M. \quad (4)$$

Расчет по формулам (4) может выполняться только последовательно.

## 2.2. Алгоритм PARAREAL для дифференциального уравнения дробного порядка

Следуя идеи [1], лежащей в основе алгоритма PARAREAL, разобьем временной отрезок  $[0, T]$  на  $n$  частей:

$$\Omega_i \equiv [T_{i-1}, T_i] \quad (i = 1, 2, \dots, n), \quad \bigcup_{i=1}^n \Omega_i = [0, T], \quad 0 \equiv T_0 < T_1 < \dots < T_n \equiv T.$$

На отрезке  $\Omega_i$  дробная производная из (1) может быть переписана следующим образом:

$${}_0D_t^\alpha y \equiv \frac{1}{\Gamma(1-\alpha)} \frac{\partial}{\partial t} \int_0^t \frac{y(\tau) d\tau}{(t-\tau)^\alpha} = \frac{1}{\Gamma(1-\alpha)} \frac{\partial}{\partial t} \left( \int_0^{T_{i-1}} \frac{y(\tau) d\tau}{(t-\tau)^\alpha} + \int_{T_{i-1}}^t \frac{y(\tau) d\tau}{(t-\tau)^\alpha} \right) \equiv {}_0I_{T_{i-1}}(t, y) + {}_{T_{i-1}}D_t^\alpha y,$$

где введены обозначения

$${}_0I_{T_{i-1}}(t, y) = \frac{1}{\Gamma(-\alpha)} \int_0^{T_{i-1}} \frac{y(\tau) d\tau}{(t-\tau)^{1+\alpha}}, \quad {}_{T_{i-1}}D_t^\alpha y = \frac{1}{\Gamma(1-\alpha)} \frac{\partial}{\partial t} \int_{T_{i-1}}^t \frac{y(\tau) d\tau}{(t-\tau)^\alpha}.$$

Отметим, что значения интеграла  ${}_0I_{T_{i-1}}(t, y)$  зависят от значений искомой функции  $y(t)$  на предшествующих временных интервалах  $\Omega_j$  ( $j = 1, 2, \dots, i-1$ ). Дробная производная  ${}_{T_{i-1}}D_t^\alpha y$  является обычной производной Римана-Лиувилля порядка  $\alpha$  со смещенным началом и может быть аппроксимирована аналогично (3).

В результате задачу (1) можно представить в виде совокупности  $n$  подзадач, каждая из которых рассматривается на своем временном отрезке  $\Omega_i$ :

$$\begin{aligned} T_{i-1} D_t^\alpha y &= f(t, y) - {}_0 I_{T_{i-1}}(t, y), \quad t \in \Omega_i, \quad 0 < \alpha < 1; \\ y(T_{i-1}) &= Y_{i-1}, \quad i = 1, 2, \dots, n. \end{aligned} \quad (5)$$

Задачи (5) могут решаться независимо (т.е. параллельно) при условии, что определены значения  $Y_{i-1}$  ( $i = 1, 2, \dots, n$ ) и независимо могут вычисляться значения интегралов  ${}_0 I_{T_{i-1}}(t, y)$ . Отметим, что в случае дифференциальных уравнений целого порядка, для которых и был разработан классический алгоритм PARAREAL, эффекты памяти отсутствуют и нет необходимости вычислять интеграл в правой части.

Как и в классическом алгоритме PARAREAL будем рассматривать две временные сетки – грубую и точную. Грубую сетку образуем из совокупности границ и середин отрезков  $\Omega_i$ :

$$\omega_G = \{T_i\}_{i=0}^n \cup \{T_{i-1/2}\}_{i=1}^n, \quad T_{i-1/2} = \frac{T_{i-1} + T_i}{2}.$$

Введение дополнительной средней точки необходимо для более точного вычисления значений интеграла  ${}_0 I_{T_{i-1}}(t, y)$ . Точную сетку  $\omega_F = \{\omega_i\}_{i=0}^n$ ,  $\omega_i = \{t_j^i\}_{j=0}^{m_i}$  для упрощения вычисления дробной производной в пределах каждой области  $\Omega_i$  будем считать равномерной с шагом  $h_i = \frac{T_i - T_{i-1}}{m_i}$ . Грубая сетка используется для построения начального приближения, вычисления значений интегралов  ${}_0 I_{T_{i-1}}(t, y)$  и корректировки получаемых решений. На точной сетке параллельно решаются задачи (5).

В результате приходим к следующей модификации алгоритма PARAREAL для решения дифференциальных уравнений дробного порядка.

1) На грубой сетке  $\omega_G$  последовательно строятся решения задач (5)  $\{y_{i-1/2}^0, y_i^0\}_{i=1}^n$ , которые рассматриваются в дальнейшем как начальные приближения. При этом на  $\Omega_i$  дробная производная аппроксимируется согласно (3). Значения интегралов  ${}_0 I_{T_{i-1}}(t, y)$  рассчитываются на основе аналитических зависимостей, полученных в предположении квадратичной аппроксимации  $y(t)$  на каждом отрезке  $\Omega_j$  ( $j = 1, 2, \dots, i-1$ ) по уже найденным значениями  $\{y_{j-1/2}^0, y_j^0\}_{j=1}^{i-1}$ .

Далее строится обобщенная итерационная процедура для  $k$ -й итерации.

2) Параллельно осуществляется решение задач (5) на точной сетке  $\omega_F$ . При этом в качестве  $Y_{i-1}$  принимается значение  $y_{i-1}^{k-1}$  с предыдущего шага. Дробные производные аппроксимируются согласно (3) на сетках  $\omega_i$ . Значения интегралов  ${}_0 I_{T_{i-1}}(t, y)$  рассчитываются аналогично п.1 по значениям  $\{y_{j-1/2}^{k-1}, y_j^{k-1}\}_{j=1}^{i-1}$ . В результате находится предварительное решение на текущей итерации  $\{\tilde{y}_{j-1/2}^k, \tilde{y}_j^k\}_{j=1}^n$ .

3) Значения  $\tilde{y}_{k-1/2}^k$  и  $\tilde{y}_k^k$  принимаются за окончательные решения. Для областей  $\Omega_i$  ( $i = k+1, \dots, n$ ) новые приближения последовательно находятся по формулам

$$y_{i-1/2}^k = \tilde{y}_{i-1/2}^k + g_{i-1/2}^k - g_{i-1/2}^{k-1}, \quad y_i^k = \tilde{y}_i^k + g_i^k - g_i^{k-1},$$

где  $g_i^k$  – решение задачи на грубой сетке при условии, что в качестве исходных данных выбираются уже рассчитанные значения на текущем временном шаге, т.е.  $\{y_{j-1/2}^k, y_j^k\}_{j=1}^{i-1}$ .

Очевидно, что максимальное число итераций алгоритма равно  $n$ . Однако, преимущество от распараллеливания будет только в том случае, если номер итерации, на котором достигается требуемый уровень точности решения, оказывается существенно меньше  $n$ . Оптимальным случаем, обеспечивающим максимальную эффективность распараллеливания, является сходимость уже после первой итерации. Как показывают численные эксперименты, сходимость достигается за 1-3 итерации.

### 3. Результаты численных экспериментов

Для тестирования предлагаемого алгоритма и оценки его точности рассмотрим простейший случай, когда правая часть уравнения (1) является линейной. Будем решать следующую задачу Коши:

$$\begin{aligned} {}_0D_t^\alpha y &= t - y, \quad 0 < t \leq 1, \quad 0 < \alpha < 1; \\ y(0) &= 0. \end{aligned} \quad (6)$$

Аналитическое решение задачи (6) имеет вид

$$y(t) = t^{1+\alpha} E_{\alpha, \alpha+2}(-t^\alpha), \quad (7)$$

где

$$E_{\alpha, \beta}(t) = \sum_{k=0}^{\infty} \frac{t^k}{\Gamma(\alpha k + \beta)}$$

– функция Миттаг-Леффлера [8]. Графики решения (7) для трех значений  $\alpha$  даны на Рис. 1.

Также аналитически может быть вычислен интеграл

$$\begin{aligned} {}_0I_{T_{i-1}}(t, y) &= \frac{1}{\Gamma(-\alpha)} \int_0^{T_{i-1}} \frac{y(\tau) d\tau}{(t-\tau)^{1+\alpha}} = \\ &= \frac{1}{\Gamma(-\alpha)} \left( \frac{T_{i-1}}{t} \right)^{1+\alpha} \sum_{k=0}^{\infty} \frac{(-1)^k T_{i-1}^{\alpha k + 1}}{\Gamma(\alpha k + \alpha + 3)} {}_2F_1 \left( 1 + \alpha, \alpha k + \alpha + 2; \alpha k + \alpha + 3; \frac{T_{i-1}}{t} \right), \end{aligned}$$

где  ${}_2F_1(a, b; c; x)$  – гипергеометрическая функция. Данное представление использовалось для оценки погрешности аппроксимации интеграла в предлагаемом алгоритме.

На Рис. 2 и 3 приведены графики ошибки решения (модуль разности между точным аналитическим решением задачи (6) и приближенным решением по предложенному алгоритму) для нескольких первых итераций при  $\alpha = 0.1$  и  $\alpha = 0.9$ , соответственно. Данные числовые значения были выбраны с целью изучения поведения предлагаемого алгоритма в зависимости от степени нелинейности решения (как видно из Рис.1, при  $\alpha = 0.1$  решение близко к линейному, в то время как при  $\alpha = 0.9$  оно сильно нелинейно). Грубая сетка выбиралась равномерной с шагом 0.2, точная сетка – равномерной с шагами 0.005 (а) и 0.001 (б). Пунктиром показана ошибка, получаемая при вычислении интегралов  ${}_0I_{T_{i-1}}(t, y)$  по точным данным.

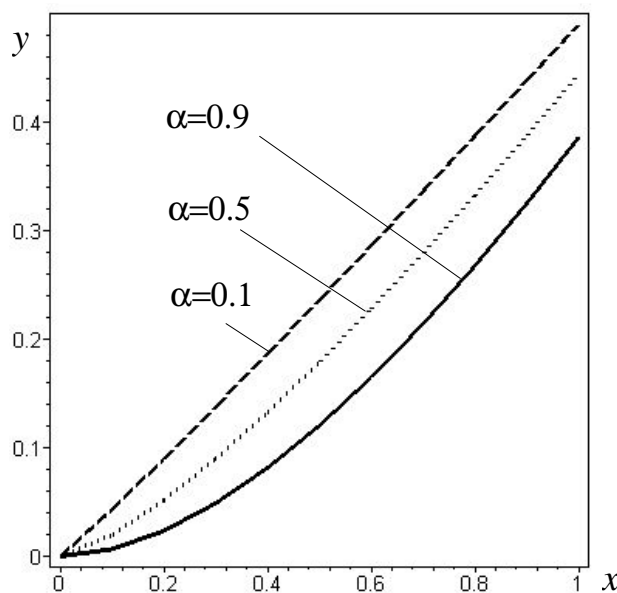
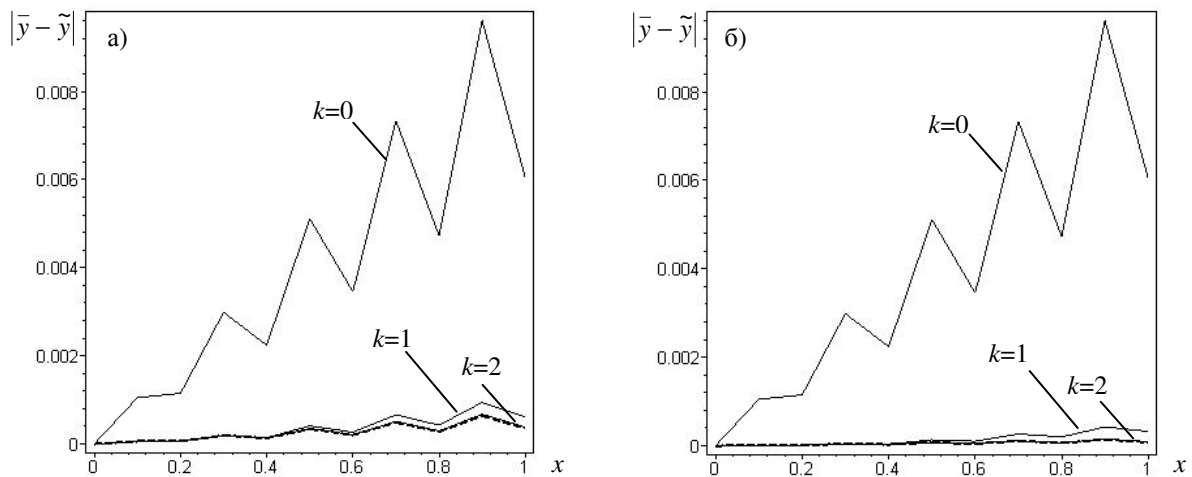
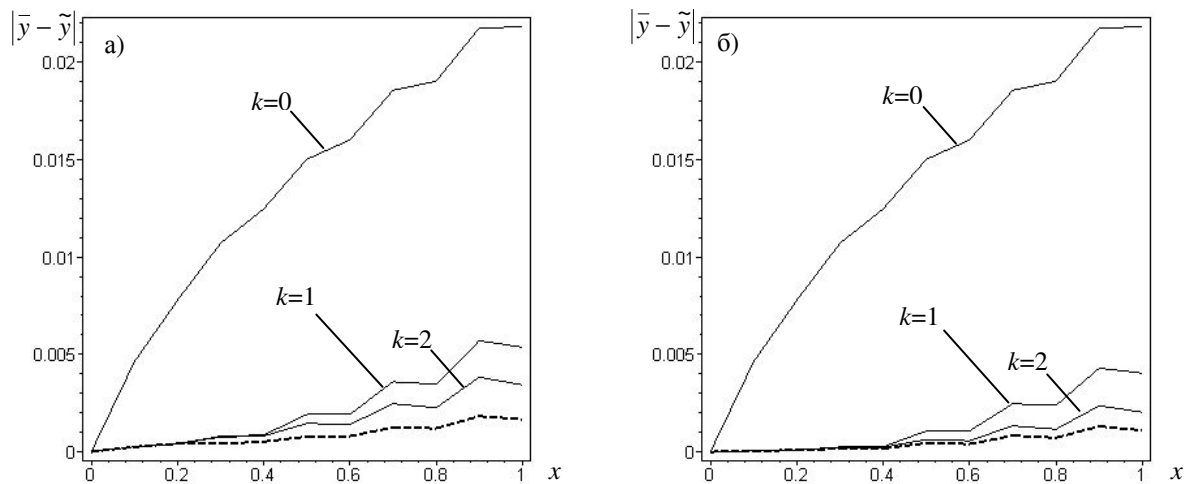


Рис. 1. Графики точного решения



**Рис. 2.** Ошибка решения при  $\alpha = 0.1$ ,  $k$  – номер итерации  
(а – шаг точной сетки 0.005, б – шаг точной сетки 0.001)



**Рис. 3.** Ошибка решения при  $\alpha = 0.9$ ,  $k$  – номер итерации  
(а – шаг точной сетки 0.005, б – шаг точной сетки 0.001)

Из Рис. 2 и 3 видно, что степень дробной производной оказывает существенное влияние на сходимость итерационного процесса. При малых значениях  $\alpha$  удовлетворительный уровень ошибки достигается уже после первой итерации, при больших  $\alpha$  для сходимости требуется 2-3 итерации. Уменьшение шага точной сетки, как и следовало ожидать, приводит к уменьшению уровня ошибок.

## Заключение

Предложена модификация параллельного алгоритма PARAREAL для численного решения дифференциальных уравнений с производными по времени дробного порядка. Проведенные численные эксперименты подтвердили работоспособность алгоритма и показали его быструю сходимость. В ходе дальнейшей работы будут получены теоретические оценки точности и скорости сходимости алгоритма, а также оценки его эффективности для многоядерных и многопроцессорных вычислительных систем.

## Литература

1. Lions J.-L., Maday Y., Turinici G. Resolution d'edp par un schema en temps parareel // C.R.Acad Sci. Paris. – Ser. I Math. – 2001. – Vol. 332. – P. 661–668.
2. Maday Y., Turinici G. The Parareal in Time Iterative Solver: a Further Direction to Parallel Implementation // Lecture Notes in Computational Science and Engineering: Domain Decomposition Methods in Science and Engineering XVIII. – 2005. – Vol. 40. – P. 441–448.
3. Staff G. A., Rønquist E.M. Stability of the Parareal Algorithm // Lecture Notes in Computational Science and Engineering: Domain Decomposition Methods in Science and Engineering XVIII. – 2005. – Vol. 40. – P. 449–456.
4. Gander M.J., Vandewalle S. On the Superlinear and Linear Convergence of the Parareal Algorithm // Lecture Notes in Computational Science and Engineering: Domain Decomposition Methods in Science and Engineering XVI. – 2007. – Vol. 55. – P. 291–298.
5. Gander M.J., Vandewalle S. Analysis of the parareal time-parallel time-integration method // SIAM J. Sci. Comput. – 2007. – Vol. 29, N. 2. – P. 556–578.
6. Fischer P.F., Hecht F., Maday Y. A Parareal in Time Semi-implicit Approximation of the Navier-Stokes Equations // Lecture Notes in Computational Science and Engineering: Domain Decomposition Methods in Science and Engineering XVIII. – 2005. – Vol. 40. – P. 433–440.
7. Bal G., Wu Q. Symplectic Parareal // Lecture Notes in Computational Science and Engineering: Domain Decomposition Methods in Science and Engineering XVII. – 2008. – Vol. 60. – P. 401–408.
8. Самко С.Г., Килбас А.А., Маричев О.И. Интегралы и производные дробного порядка и некоторые их приложения. – Минск: Наука и техника, 1987. – 688 с.
9. Podlubny I. Fractional Differential Equations. An Introduction to Fractional Derivatives, Fractional Differential Equations, Some Methods of Their Solution and Some of Their Applications.– Academic Press, San Diego, California. – 1999. – 368 p.
10. Учайкин В.В. Метод дробных производных. – Ульяновск: Изд-во "Артишок", 2008. – 512 с.
11. Metzler R., Klafter J. The random walks guide to anomalous diffusion: A fractional dynamics approach // Phys. Rep. – 2000. – V. 339. – P. 1–77.