

Двухсеточные параллельные алгоритмы для решения дробно-дифференциальных уравнений аномальной диффузии

С.Ю. Лукашук

Уфимский государственный авиационный технический университет

Приводятся описание и анализ параллельных алгоритмов решения начально-краевых задач для уравнений аномальной диффузии, содержащих производные дробного порядка типа Римана-Лиувилля по пространственным и/или временной переменным. Параллельные алгоритмы построены на основе двухсеточных методов и, в случае распараллеливания по оси времени, представляют собой модификации известного алгоритма PARAREAL.

1. Введение

Во многих неупорядоченных сложных средах, таких как пористые и трещиновато-пористые среды, аморфные полупроводники, жидкие полимеры и стекла, перколяционные кластеры и самоподобные фрактальные среды, турбулентные потоки жидкости, газа и плазмы, кинетика протекания процессов диффузионного переноса часто отличается от классической, соответствующей нормальному закону распределения [1, 2]. Такие процессы принято называть процессами аномальной диффузии. Аномальный диффузионный перенос обычно обусловлен эффектами памяти среды, пространственной нелокальности и перемежаемости, имеет немарковскую стохастическую природу и поэтому не может быть описан классическими эволюционными уравнениями переноса [3].

Одним из эффективных и широко используемых подходов для описания процессов аномальной диффузии является использование аппарата интегро-дифференцирования дробного порядка [4–6]. В этом случае уравнение процесса является интегро-дифференциальным уравнением, содержащим производные дробного порядка по временной и/или пространственным переменным.

Основной проблемой численного моделирования процессов аномальной диффузии, описываемых уравнениями дробного порядка, является нелокальный характер последних. При использовании конечно-разностной дискретизации обычных диффузионных уравнений значение в текущем узле расчетной сетки зависит только от значений в соседних узлах. При конечно-разностной дискретизации уравнений дробного порядка значение в текущем узле будет зависеть от всех остальных пространственных узлов (при наличии в уравнении дробных производных по пространственным переменным) и всех предыдущих временных слоев (при наличии дробной производной по времени) [4]. В результате при численных расчетах аномального переноса объем вычислений оказывается существенно (на порядки!) большим по сравнению с объемом вычислений в расчетах классических диффузионных процессов. При наличии дробной производной по времени объем вычислений также растет с увеличением номера временного слоя, поскольку увеличивается диапазон памяти системы, которую нужно учитывать. Существенно возрастает и объем необходимой оперативной памяти для хранения результатов расчета. Использование параллельных алгоритмов, построенных только на принципах пространственной декомпозиции расчетной области (domain decomposition), ситуацию не спасает, поскольку приводит к чрезвычайно большим объемам передачи данных между процессорами.

Указанные проблемы делают актуальной задачу разработки новых последовательных и параллельных численных алгоритмов решения уравнений аномальной диффузии с производными дробного порядка. В данной работе предлагается подход к построению таких

алгоритмов, основанный на использовании идеи двухсеточных методов. Грубая сетка используется в этом случае как основа для расчета эффектов пространственного и временного дальнего действия, а мелкая сетка служит для непосредственных вычислений. По узлам грубой сетки может производиться пространственная и, при необходимости, временная декомпозиция расчетной области, что приводит к легко реализуемым параллельным алгоритмам. При распараллеливании по временной оси данный подход приводит к модификации алгоритма PARAREAL для уравнений с производными дробного порядка [8].

2. Постановка задачи

При описании процессов аномальной диффузии наиболее часто используются левосторонняя и правосторонняя производные дробного порядка α типа Римана-Лиувилля [5]:

$$({}_a D_x^\alpha y)(x) = \frac{1}{\Gamma(n-\alpha)} \left(\frac{d}{dx}\right)^n \int_a^x \frac{y(\xi)}{(x-\xi)^{\alpha-n+1}} d\xi, \quad (1)$$

$$({}_x D_b^\alpha y)(x) = \frac{(-1)^n}{\Gamma(n-\alpha)} \left(\frac{d}{dx}\right)^n \int_x^b \frac{y(\xi)}{(\xi-x)^{\alpha-n+1}} d\xi, \quad (2)$$

где $n = [\alpha] + 1$ и $[\alpha]$ – целая часть числа α .

В дальнейшем для иллюстрации предлагаемого подхода будем рассматривать уравнение аномальной диффузии дробного порядка следующего вида:

$$({}_0 D_t^\alpha u)(x, t) = \gamma ({}_0 D_x^{1+\beta} u)(x, t) + (1-\gamma) ({}_x D_L^{1+\beta} u)(x, t) + f(x, t), \quad (3)$$

$$t \in (0, T], \quad x \in (0, L), \quad \alpha, \beta, \gamma \in (0, 1).$$

Для уравнения (3) поставим первую краевую задачу:

$$u(x, 0) = u^0(x), \quad u(0, t) = u_0(t), \quad u(L, t) = u_L(t), \quad (4)$$

где $u^0(x)$, $u_0(t)$, $u_L(t)$ – заданные функции. Отметим, что постановка таких краевых условий предъявляет определенные дополнительные требования к виду функции $f(x, t)$, которые однако не являются важными для рассматриваемых алгоритмов и поэтому здесь не приводятся.

При построении численных схем для дробно-дифференциальных уравнений используются различные конечно-разностные аппроксимации дробных производных. Для левосторонней дробной производной (1) в случае равномерной сетки все они могут быть представлены в следующем общем виде:

$$({}_a D_x^\alpha y)(x_k) \approx h^{-\alpha} \sum_{j=0}^{k+m} A_j y_{k+m-j}, \quad y_j = y(x_j), \quad x_j = a + jh \quad (5)$$

(здесь h – шаг сетки). Аналогичная формула имеет место и для правосторонней дробной производной (2). При $m = 0$ и $A_j = (-1)^j \binom{\alpha}{j}$ имеем наиболее часто используемую классическую аппроксимацию, получаемую из формулы Грюнвальда-Летникова [5], и обеспечивающую первый порядок точности аппроксимации дробной производной для аналитических функций [4]. Повысить порядок точности можно использованием полиномиальной интерполяции искомой функции между узлами расчетной сетки с последующим аналитическим вычислением соответствующих интегралов. Случай линейной междуузельной интерполяции, обеспечивающий второй порядок точности, рассмотрен в [7].

3. Декомпозиция задачи на основе двухсеточного подхода

Введем в рассматриваемой пространственно-временной области $\bar{\Omega} = [0, L] \times [0, T]$ равномерную конечно-разностную сетку с шагом ΔX по пространственной и ΔT по временной

переменным:

$$\omega_G = \{(X_i = i\Delta X, T_j = j\Delta T), i = 0, 1, \dots, N, j = 0, 1, \dots, M, \Delta X = L/N, \Delta T = T/M\}. \quad (6)$$

Сетку (6) в дальнейшем будем называть грубой сеткой. Каждую ячейку, ограниченную смежными узлами грубой сетки, будем рассматривать как отдельную расчетную подобласть:

$$\bar{\Omega}_{i,j} = [X_{i-1}, X_i] \times [T_{j-1}, T_j], \quad \bar{\Omega} = \bigcup_{i=1}^N \bigcup_{j=1}^M \bar{\Omega}_{i,j}.$$

Для $(t, x) \in \Omega_{i,j}$ производные дробного порядка, входящие в уравнение (3), могут быть записаны следующим образом:

$$\begin{aligned} {}_0D_t^\alpha u &= {}_{T_{j-1}}D_t^\alpha u + {}_0I_{T_{j-1}}u, \\ {}_0D_x^{1+\beta} u &= {}_{X_{i-1}}D_x^{1+\beta} u + {}_0J_{X_{i-1}}u, \quad {}_xD_L^{1+\beta} u = {}_xD_{X_i}^{1+\beta} u + {}_{X_i}J_L u, \end{aligned}$$

где введены следующие обозначения для интегралов:

$$\begin{aligned} {}_0I_{T_{j-1}}u &= \frac{1}{\Gamma(-\alpha)} \int_0^{T_{j-1}} \frac{u(x, \tau)}{(t - \tau)^{\alpha+1}} d\tau, \\ {}_0J_{X_{i-1}}u &= \frac{1}{\Gamma(-\beta - 1)} \int_0^{X_{i-1}} \frac{u(\xi, t)}{(x - \xi)^{\beta+2}} d\xi, \quad {}_{X_i}J_L u = \frac{1}{\Gamma(-\beta - 1)} \int_{X_i}^L \frac{u(\xi, t)}{(\xi - x)^{\beta+2}} d\xi. \end{aligned}$$

Уравнение (3) в области $\Omega_{i,j}$ может быть теперь переписано в следующем виде:

$${}_{T_{j-1}}D_t^\alpha u = \gamma {}_{X_{i-1}}D_x^{1+\beta} u + (1 - \gamma) {}_xD_{X_i}^{1+\beta} u + f(x, t) + Lu, \quad (7)$$

где

$$L = \gamma {}_0J_{X_{i-1}} + (1 - \gamma) {}_0J_{X_{i-1}} - {}_0I_{T_{j-1}}$$

- линейный интегральный оператор. Принципиально важным является тот факт, что вычисление Lu в области $\Omega_{i,j}$ производится по значениям функции $u(x, t)$ при $(x, t) \notin \Omega_{i,j}$.

Обозначим через $U = \{U_{i,j}, i = 0, 1, \dots, N, j = 0, 1, \dots, M\}$ сеточную функцию решения задачи на грубой сетке ω_G . В предлагаемом подходе ее значения используются для постановки начальных и граничных условий для уравнения (7), а также вычисления интегралов, входящих в Lu .

После того, как задача для уравнения (7) замыкается с помощью начальных и граничных условий и указан способ вычисления Lu по значениям U , численное решение задач в каждой подобласти $\Omega_{i,j}$ или соответствующим образом выбранной группе подобластей может выполняться параллельно. Для этого в каждой подобласти вводится точная сетка (для простоты считаем, что разбиение всех подобластей идентично):

$$\omega_F = \left\{ (x_k = k\delta x, t_l = l\delta t), k = 0, 1, \dots, n, l = 0, 1, \dots, m, \delta x = \frac{\Delta X}{n}, \delta t = \frac{\Delta T}{m} \right\}. \quad (8)$$

С использованием аппроксимаций вида (5) на сетке ω_F строится конечно-разностная аппроксимация уравнений (7) требуемого уровня точности. Как и в случае классического уравнения диффузии, могут быть построены явные, неявные или полу-явные схемы. Сеточную функцию задачи на точной сетке будем обозначать через $u = \{u_{k,l}, k = 0, 1, \dots, n, l = 0, 1, \dots, m\}$.

Таким образом, точная сетка ω_F используется для параллельного решения задач в отдельных подобластях, которые могут выполняться параллельно, а грубая сетка ω_G используется для постановки краевых условий и вычисления дополнительной функции источника Lu .

На основе такого двухсеточного подхода возможно построение алгоритмов двух видов. В алгоритмах первого вида счет ведется последовательно (например, по времени) и в узлы грубой сетки просто копируются значения, полученные при расчете на точной сетке. Затем эти значения рассылаются по всем подобластям, в которых они необходимы для вычисления соответствующих элементов. Второй вид алгоритмов в более полной мере реализует идею двухсеточного подхода и основан на схемах типа „предиктор-корректор“. В этом случае каким-либо образом рассчитываются начальные приближения в узлах грубой сетки, а затем проводятся расчеты (параллельные) на точных сетках и уточняются приближения на грубой сетке. К алгоритму второго типа относится, в частности, модификация алгоритма PARAREAL для распараллеливания по времени. Далее будут рассмотрены характерные представители алгоритмов обоих видов.

4. Описание параллельных алгоритмов

4.1. Параллельный алгоритм с декомпозицией по пространству

Данный алгоритм относится к алгоритмам первого вида. Расчет по времени ведется последовательно, поэтому шаги по времени грубой и точной сеток совпадают: ($\delta t = \Delta T$, $M = m$). Характерными особенностями алгоритма, обеспечивающими требуемый уровень точности вычислений, являются: 1) использование кубической сплайн-интерполяции по значениям в узлах грубой сетки на одном временном слое, 2) использование полиномов второй и третьей степени для аппроксимации в окрестности границ подобластей $\Omega_{i,j}$ производных дробного порядка по пространственной переменной.

Сохранение заданной точности вычислений производных дробного порядка по пространству после декомпозиции на подобласти $\Omega_{i,j}$ является главной проблемой при разработке соответствующих параллельных алгоритмов. Использование классических аппроксимаций вида (5) (типа Грюнвальда-Летникова) для смещенных дробных производных, входящих в уравнение (7), может приводить к катастрофической потере точности вычислений в окрестности границ областей $\Omega_{i,j}$ несмотря на сохранение общего порядка точности аппроксимации. Для подавления ошибки предлагается использовать следующие вычислительные приемы.

Во-первых, необходимо выполнить сдвиг вычисляемой функции на величину соответствующего граничного значения:

$${}_{X_{i-1}}D_x^{1+\beta}u = {}_{X_{i-1}}D_x^{1+\beta}[u(x,t) - u(X_{i-1},t)] + \frac{1}{\Gamma(-\beta)} \frac{u(X_{i-1},t)}{(x - X_{i-1})^{\beta+1}}, \quad (9)$$

$${}_x D_{X_i}^{1+\beta}u = {}_x D_{X_i}^{1+\beta}[u(x,t) - u(X_i,t)] + \frac{1}{\Gamma(-\beta)} \frac{u(X_i,t)}{(X_i - x)^{\beta+1}}. \quad (10)$$

В результате основная составляющая производной оказывается вычисленной аналитически, а по приближенным формулам рассчитываются уже относительно небольшие поправки.

Во-вторых, при расчете левосторонней производной из правой части (9) в точке x_1 и правосторонней производной из правой части (10) в точке x_{n-1} коэффициенты A_r соответствующих аппроксимаций вида (5) находятся на основе явного вычисления этих производных по трехузловой квадратичной интерполяции. Для следующих точек x_2 и x_{n-2} используется четырехузловая кубическая интерполяция. Для остальных точек может быть использована классическая аппроксимация первого или второго порядка точности.

В-третьих, вычисление интегралов ${}_0 J_{X_{i-1}}u$ и ${}_{X_i} J_L u$ производится аналитически для кубической сплайн-интерполяции, построенной по значениям на грубой сетке U . Известно [9], что оценка погрешности аппроксимации функции $f(x)$ кубическим сплайном $s(x)$ на равномерной сетке с шагом H имеет вид

$$|s(x) - f(x)| \leq \frac{5}{384} AH^4,$$

где $|f^{IV}(x)| \leq A$. Тогда легко получается оценка точности вычисления интеграла ${}_0J_{X_{i-1}}u$:

$$|{}_0J_{X_{i-1}}u(x, t_l) - s(x, t_l)| \leq \frac{5}{384} \frac{A(t)(\Delta X)^4}{(\delta x)^{1+\beta}},$$

где $|u_{xxxx}(x, t)| \leq A(t)$. Аналогичная оценка справедлива для интеграла ${}_{X_i}J_Lu$. Поскольку множитель $(\delta x)^{-1-\beta}$ является общим множителем при вычислении дробных производных на точной сетке по аппроксимациям вида (5), то точность вычисления интегралов по сплайн-интерполяции в разностной схеме решения уравнения (7) на точной сетке имеет порядок $O(\Delta X^4)$. Если для аппроксимации дробных производных используется аппроксимация первого порядка точности по пространству, то для сохранения этого порядка у всей схемы для уравнения (7) необходимо выполнение следующего очевидного неравенства: $(\Delta X)^4 \leq \delta x$. При использовании аппроксимации второго порядка точности неравенство становится более жестким: $(\Delta X)^2 \leq \delta x$.

Далее представлены основные шаги алгоритма при использовании явной схемы для дискретизации уравнений (7).

1. По заданному начальному условию задачи $u^0(x)$ (4) каждый процессор параллельно вычисляет значения сеточной функции U на нулевом временном шаге.
Следующие шаги алгоритма являются едиными для временных слоев $l = 1, 2, \dots, m$.
2. По значениям U для $(l-1)$ -го временного слоя каждый процессор вычисляет коэффициенты соответствующего кубического интерполяционного сплайна.
3. На точной сетке все процессоры параллельно выполняют расчет $u_{k,l}$ ($k = 1, 2, \dots, n-1$) на временном слое l . При этом производная по времени аппроксимируется формулой Грюнвальда-Летникова

$$({}_0D_t^\alpha u)(x_k, t_l) \approx (\delta t)^{-\alpha} \sum_{r=0}^l (-1)^r \binom{\alpha}{r} u_{k,l-r},$$

а производные по пространству аппроксимируются на $(l-1)$ -м временном слое с учетом (9) и (10) по формулам

$$\begin{aligned} \left({}_{X_{i-1}}D_x^{1+\beta} u \right) (x_k, t_{l-1}) &\approx (\delta x)^{-(1+\beta)} \sum_{r=0}^{k+1} A_r (u_{k+1-r, l-1} - u_{0, l-1}) + \\ &+ \frac{1}{\Gamma(-\beta)} \frac{u_{0, l-1}}{(x_k - X_{i-1})^{1+\beta}}, \\ \left({}_x D_{X_i}^{1+\beta} u \right) (x_k, t_{l-1}) &\approx (\delta x)^{-(1+\beta)} \sum_{r=0}^{n-k+1} A_r (u_{k-1+r, l-1} - u_{0, l-1}) + \\ &+ \frac{1}{\Gamma(-\beta)} \frac{u_{0, l-1}}{(X_i - x_k)^{1+\beta}}, \end{aligned}$$

Коэффициенты A_r выбираются так, как описано выше. Интегралы ${}_0J_{X_{i-1}}u$ и ${}_{X_i}J_Lu$ вычисляются по аналитическим формулам для сплайн-интерполяции, построенной на шаге 2.

4. Процессор, отвечающий за расчет подобласти $\Omega_{i+1,l}$ ($i = 1, 2, \dots, N-1$), пересылает процессору, отвечающему за расчет соседней подобласти $\Omega_{i,l}$ значение приращения за временной шаг искомой функции в первом узле точной сетки: $u_{1,l}^{i+1} - u_{1,l-1}^{i+1}$ (здесь верхний индекс соответствует пространственному номеру области). Процессоры, получившие значения, вычисляют сеточную функцию в узлах грубой сетки на l -м временном слое по формуле

$$U_{i,l} = U_{i,l-1} + 0.5 \left(u_{1,l}^{i+1} - u_{1,l-1}^{i+1} + u_{n-1,l}^i - u_{n-1,l-1}^i \right).$$

5. Выполняется взаимный обмен рассчитанными значениями сеточной функции U таким образом, чтобы полный вектор этих значений находился на каждом процессоре.

Далее переход к следующему временному слою и повтор алгоритма с шага 2.

Отметим, что условие устойчивости использованной явной конечно-разностной схемы имеет вид

$$\frac{(\delta t)^\alpha}{(\delta x)^{1+\beta}} \leq \frac{\alpha}{1+\beta} \quad (11)$$

(для случая $\alpha = 1$ это условие приведено в [7]). В предельном случае классического уравнения диффузии $\alpha = 1$, $\beta = 1$ это условие переходит в хорошо известное условие устойчивости явной разностной схемы для уравнения параболического типа.

4.2. Параллельный алгоритм с декомпозицией по времени

Известно, что параллельные алгоритмы для решения эволюционных задач могут быть построены не только на основе декомпозиции пространственной расчетной области, но и на основе декомпозиции по временной оси. Одним из наиболее широко используемых алгоритмов данного вида является алгоритм PARAREAL, впервые предложенный в работе [10]. В настоящее время данный алгоритм хорошо изучен, получены теоретические оценки его точности и сходимости [11–14], предложен целый ряд его модификаций для решения различных классов задач эволюционного типа [11, 15, 16]. В работе [8] была предложена модификация алгоритма PARAREAL для решения обыкновенных дифференциальных уравнений с производной дробного порядка $\alpha \in (0, 1)$. В данном разделе дается описание новой модификации алгоритма PARAREAL, совмещенной с приведенным в предыдущем разделе алгоритмом декомпозиции по пространству, и предназначенной для решения уравнений вида (3).

Алгоритм базируется на пространственно-временной декомпозиции расчетной области Ω на основе грубой сетки ω_G , представленной в разделе 3, и двумерной сплайн-интерполяции решения по значениям сеточной функции U в узлах грубой сетки.

Пусть с использованием некоторого сеточного метода G , который в дальнейшем будем называть „грубым“, построено приближенное решение U краевой задачи (3), (4) в узлах грубой сетки. Рассмотрим теперь задачу для уравнения (7) в области $\Omega_{i,j}$. Обозначим через $u^{i,j}$ функцию решения этого уравнения. С использованием сплайн-интерполяции по значениям U на $(j-1)$ -м временном слое можно записать начальное условие для уравнения (7) в виде

$$u^{i,j}(x, T_{j-1}) = \phi(x, U_{0,j-1}, U_{1,j-1}, \dots, U_{N,j-1}). \quad (12)$$

Пусть интерполяция по временной переменной поля значений U выполняется таким образом, что значения интерполяционных коэффициентов для произвольного $(j-1)$ -го слоя зависят только от значений U на этом и предыдущих временных слоях. Эта интерполяция используется для вычисления интегралов ${}_0I_{T_{j-1}}$. Тогда решение уравнения (7) с начальным условием (12) будет зависеть только от значений U на первых $(j-1)$ -м временных слоях:

$$u^{i,j} = u^{i,j}(x, t, U_{k,l}), \quad k = 0, 1, \dots, N, \quad l = 0, 1, \dots, j-1, \quad (x, t) \in \Omega_{i,j}. \quad (13)$$

Отметим, что для получения решений вида (13) необходимо, вообще говоря, на каждом временном слое решать систему уравнений вида (7) для всех подобластей, принадлежащих данному временному слою, с условиями сопряжения решений на границах подобластей. Однако, при численном решении этой задачи на мелкой сетке может быть эффективно использован алгоритм, приведенный в п. 4.1.

Зная решения (13), можно записать уравнения для нахождения значений в узлах грубой сетки:

$$U_{i,j} = u^{i,j}(X_i, T_j, U_{k,l}), \quad k = 0, 1, \dots, N, \quad l = 0, 1, \dots, j-1, \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, M. \quad (14)$$

Система (14) является системой нелинейных уравнений относительно сеточной функции U и может быть решена методом Ньютона:

$$U_{i,j}^{q+1} = u^{i,j}(X_i, T_j, U^q) + \sum_{k=0}^N \sum_{l=0}^{j-1} \frac{\partial u^{i,j}(X_i, T_j, U^q)}{\partial U_{k,l}} (U_{k,l}^{q+1} - U_{k,l}^q), \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, M \quad (15)$$

(здесь q – номер итерации метода Ньютона).

Ключевым моментом алгоритма PARAREAL является способ вычисления производных, входящих в (15):

$$\sum_{k=0}^N \sum_{l=0}^{j-1} \frac{\partial u^{i,j}(X_i, T_j, U^q)}{\partial U_{k,l}} (U_{k,l}^{q+1} - U_{k,l}^q) \approx G(X_i, T_j, U^{q+1}) - G(X_i, T_j, U^q), \quad (16)$$

где через G обозначено решение задачи на грубой сетке, получаемое с помощью „грубого“ метода. Таким образом, для вычисления производных на каждом следующем временном слое используется разность „грубых“ решений задачи, полученных по старым и новым, уже вычисленным на предыдущих временных слоях, значениям U .

Обозначим через $F(X_i, T_j, U)$ численное решение задачи на точной сетке в области $\Omega_{i,j}$. Полагая $u^{i,j}(X_i, T_j, U) \approx F(X_i, T_j, U)$, на основе (15), (16) получаем основную расчетную формулу алгоритма PARAREAL:

$$U_{i,j}^{q+1} = F(X_i, T_j, U_{k,l}^q) + G(X_i, T_j, U_{k,l}^{q+1}) - G(X_i, T_j, U_{k,l}^q), \quad (17)$$

$$k = 0, 1, \dots, N, \quad l = 0, 1, \dots, j - 1, \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, M.$$

Данная вычислительная процедура позволяет параллельно по времени итерационно находить приближенные решения задачи в узлах грубой сетки, сопоставимые по точности со значениями, получаемыми при расчете на точной сетке во всей расчетной области.

В результате приходим к следующему параллельному алгоритму (полагаем, что имеется $N \times M$ -процессорная решетка $\{p_{i,j}\}$ и за вычисление каждой области $\Omega_{i,j}$ отвечает свой процессор).

1. С использованием классической явной конечно-разностной схемы G , построенной на основе аппроксимаций дробных производных вида (5), каждый процессор находит начальное приближение решения в узлах грубой сетки U^0 .
2. Организуется глобальная итерационная по q процедура метода Ньютона. На первом шаге $q = 0$.
3. По значениям $U_{k,l}^q$ ($k = 0, 1, \dots, N, \quad l = 0, 1, \dots, j - 1$) каждый процессор $p_{i,j}$ строит двумерную кубическую по каждой переменной сплайн-интерполяцию поля грубого решения.
4. Все процессоры параллельно производят расчет каждый своей подобласти на точной сетке. При этом группы процессоров, отвечающие за расчет одинаковых временных слоев, реализуют алгоритм с декомпозицией по пространству, приведенный в п. 4.1. Интегралы, образующие Lu в уравнениях (3), рассчитываются по аналитическим формулам, полученным по принятому виду сплайн-интерполяции.
5. Последовательно (по временным слоям) рассчитывается следующее приближение в узлах грубой сетки U^{q+1} по формуле (17). При этом каждый процессор $p_{i,j}$ отвечает за вычисление одного значения $U_{i,j}^{q+1}$ и выполняет один дополнительный расчет „грубым“ методом $G(X_i, T_j, U_{k,l}^{q+1})$. После расчета каждого нового временного слоя полученные обновленные значения U^{q+1} рассылаются каждому процессору.

6. Проверяется условие остановки итерационного процесса: $\|U^{q+1} - U^q\| \leq \varepsilon$. Если условие не выполнено, то возврат к шагу 2. При этом временной слой $q + 1$ из расчета исключается (на нем уже получено решение, соответствующее решению на точной сетке, которое не может быть уточнено по (17)).

Описанный алгоритм допускает простую модификацию для случая, когда количество доступных процессоров достаточно для расчета только нескольких временных слоев одновременно. Кроме того, данный алгоритм существенно упрощается в следующих двух случаях. 1) При $\alpha = 1$, т.е. когда исходное уравнение (3) содержит обыкновенную (не дробную) производную по времени первого порядка – в этом случае отсутствует необходимость интерполяции по времени и алгоритм становится прямой комбинацией алгоритма п. 4.1 и классического алгоритма PARAREAL. 2) Если отсутствует пространственная декомпозиция – в этом случае отсутствует необходимость интерполяции по пространственной переменной и алгоритм становится прямой комбинацией модифицированного на уравнения дробного порядка алгоритма PARAREAL и классического последовательного алгоритма конечно-разностного решения таких уравнений.

5. Оценка параллельной эффективности алгоритмов

5.1. Оценка эффективности параллельного алгоритма с декомпозицией по пространству

Сначала приведем оценку трудоемкости последовательного алгоритма решения задачи (3), (4), построенного по явной схеме с аппроксимацией дробных производных по формуле Грюнвальда-Летникова. Шаг сетки по пространственной и временной переменным для последовательного алгоритма примем равным соответствующим шагам точной сетки параллельного алгоритма (в этом случае последовательное и параллельное решения будут иметь одинаковый порядок точности). Таким образом, имеем $Nn - 1$ расчетных узлов по пространству и M расчетных узлов по времени. Расчет l -го временного слоя по последовательному алгоритму требует $2(Nn - 1)(Nn + l + 4)$ операций. Принимая, как обычно при таких оценках, время выполнения одной операции за единицу, получаем время выполнения последовательного алгоритма равным $T_1 = M(Nn - 1)(2Nn + M + 9)$.

Теперь оценим время выполнения параллельного алгоритма, приведенного в п. 4.1. Будем полагать, что количество используемых для его выполнения процессоров равно количеству пространственных подобластей: $P = N$. Количество операций, выполняемых каждым процессором на l -м временном слое, складывается из четырех составляющих: 1) операций, необходимых для вычисления дробных производных: $2(n - 1)(n + l + 11)$, 2) операций, необходимых для вычисления интегралов ${}_0J_{X_{i-1}}u$ и X_iJ_Lu по явным аналитическим формулам через сплайн-интерполяцию: $228(n - 1)(N - 1)$, 3) операций, необходимых для расчета значений в узлах грубой сетки: $5(N - 1)$ и 4) операций, необходимых для вычисления коэффициентов интерполяционного сплайна: $22(N - 1)$ (для решения системы используется метод прогонки). Таким образом, трудоемкость расчета l -го временного слоя в параллельном алгоритме составляет $2(n - 1)(n + l + 11) + [228(n - 1) + 27](N - 1)$ операций, которые выполняются параллельно. Тогда время расчета всех M временных слоев в параллельном алгоритме составит $T_p = M[(n - 1)(2n + M + 228N - 206) + 27(N - 1)]$. Долей последовательного расчета, выполняемого на этапе инициализации алгоритма до начала основной вычислительной процедуры, в данной оценке пренебрегаем.

Тогда для ускорения параллельного алгоритма получаем следующую оценку:

$$S_p \equiv \frac{T_1}{T_p} = \frac{(Nn - 1)(2Nn + M + 9)}{(n - 1)(2n + M + 228N - 206) + 27(N - 1)} \approx \frac{N(2Nn + M)}{2n + M}$$

(последняя оценка становится справедливой при $N \ll n$). Заметим, что число временных слоев M при использовании явной схемы может оказаться достаточно большим.

При стремлении к нулю шага точной сетки, т.е. при $n \rightarrow \infty$, и неизменных остальных параметрах получаем следующую асимптотическую оценку ускорения $S_p \approx N^2 \equiv P^2$. Таким образом, по сравнению с классическим алгоритмом, предлагаемый параллельный алгоритм демонстрирует асимптотически квадратичное ускорение. Причина такого нелинейного ускорения очевидна: в то время как классический алгоритм использует для расчета все пространственные узлы, параллельный алгоритм рассчитывает далекодействующие нелокальные эффекты с использованием сплайн-аппроксимации, что приводит к существенному уменьшению количества операций в параллельном алгоритме по сравнению с последовательным.

Приведенная оценка не учитывает времени, затрачиваемого на обмен данными между процессорами, и является справедливой для систем с общей памятью. Для систем с распределенной памятью это время необходимо учитывать. Тем не менее, на величину асимптотической оценки это время влияния не оказывает, поскольку пропорционально $M(N-1)$.

5.2. Оценка эффективности параллельного алгоритма с декомпозицией по времени

Время выполнения последовательного алгоритма T_1 оценивается аналогично п. 5.1 и составляет $T_1 = Mm(Nn-1)(2Nn+Mm+9)$ (оценка изменилась, поскольку теперь имеем Mm слоев точной сетки по времени).

Оценим время исполнения параллельного алгоритма. Как и ранее, долей последовательного расчета, выполняемого на этапе инициализации алгоритма до начала основной вычислительной процедуры, пренебрегаем. Расчет выполняется на грубой ω_G и точной ω_F сетках, определенных в (6) и (8), соответственно.

Наибольшее число операций будут выполнять процессоры, отвечающие за расчет подобластей $\Omega_{i,M}$, соответствующих последнему M -му временному слою, поэтому именно они и будут определять время реализации алгоритма. Для каждой итерации q количество операций, выполняемых каждым таким процессором, складывается из четырех составляющих: 1) операций, необходимых для вычисления дробных производных: $m(n-1)(2n+m+23)$, 2) операций, необходимых для вычисления интегралов ${}_0J_{X_{i-1}}u$, $X_i J_L u$ и ${}_0I_{T_{M-1}}$ по явным аналитическим формулам через сплайн-интерполяцию: $114m(n-1)[2(N-1)+(M-1)]$, 3) операций, необходимых для расчета на грубой сетке по „грубому“ методу и уточнению значений U по формуле (17): $M(N-1)(2N+M+9)+2$ и 4) операций, необходимых для вычисления коэффициентов интерполяционного сплайна: $22(2MN-M-1)$. Сумма этих значений дает трудоемкость одной итерации метода Ньютона в параллельном алгоритме. Если для реализации алгоритма требуется Q итераций, то общее время выполнения параллельного алгоритма равно

$$T_p = Q[m(n-1)(2n+m+128N+114M-319) + M(N-1)(2N+M+53) + 22(M-1)].$$

По классической формуле $S_p = T_1/T_p$ может быть легко оценено ускорение параллельного алгоритма. Если $N \ll n$, $M \ll m$, то $T_1 \approx MNmn(2Nn+Mm)$, $T_p \approx Qmn(2n+m)$ и получаем следующую оценку ускорения:

$$S_p = \frac{MN(2Nn+Mm)}{Q(2n+m)}.$$

Для случая квадратной грубой сетки ($M=N$) получаем $S_p = N^3/Q$. Учитывая, что общее количество процессоров, задействованных для выполнения параллельного алгоритма, равно $P = MN = N^2$, получаем следующую асимптотическую оценку ускорения: $S_p \approx P^{3/2}/Q$. Максимально возможное количество итераций $Q = M = \sqrt{P}$. Однако на практике количество итераций всегда оказывается существенно меньше, поэтому действительная асимптотическая оценка ускорения получается сверхлинейной.

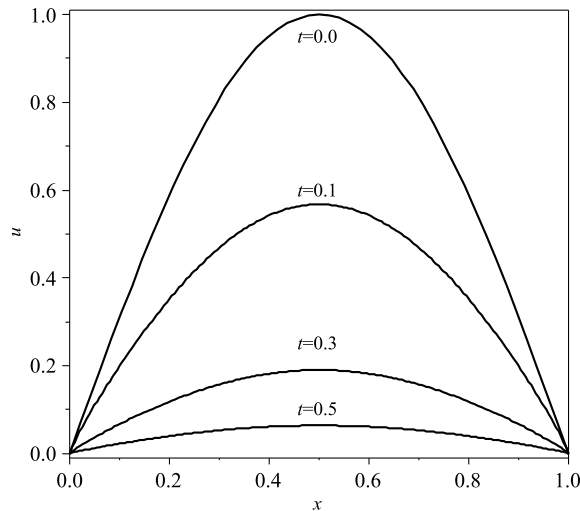


Рис. 1. Решение тестовой задачи для различных моментов времени

6. Результаты тестовых расчетов

Точность и эффективность предлагаемых алгоритмов была проверена рядом тестовых расчетов. Ниже представлены некоторые результаты проведенных вычислительных экспериментов.

Решается уравнение (3) в единичном квадрате ($L = 1$, $T = 1$) при нулевых граничных условиях $u_0(t) = 0$, $u_L(t) = 0$, без функции источника $f(x, t) = 0$, с начальным условием $u^0(x) = \sin(\pi x)$. На Рис. 1 показано решение этой задачи в различные моменты времени при $\alpha = 0.95$, $\beta = 0.7$, $\gamma = 0.5$.

На Рис. 2 показаны ошибки приближенных решений Δu для двух моментов времени $t = 0.1$ и $t = 0.3$, полученных с помощью классического последовательного алгоритма при $m = 10^5$ шагов по времени и количестве узлов сетки по пространству а) $n = 200$ и б) $n = 400$. Как и следовало ожидать, измельчение сетки приводит к уменьшению ошибки при неизменном характере ее распределения.

На Рис. 3 показаны ошибки приближенных решений Δu для тех же моментов времени, полученных при использовании параллельного алгоритма с декомпозицией по пространству при $m = M = 10^5$ и а) количестве процессоров $P = N = 4$ и количестве узлов сетки по пространству в каждой подобласти $n = 100$; б) количестве процессоров $P = N = 8$ и количестве узлов $n = 50$. Расчеты проводились на одном узле суперкомпьютера Уфимского государственного авиационного технического университета, объединяющего на общей памяти объемом 4 Гб два четырехядерных процессора Intel Xeon 5300. Из приведенных рисунков видно, что параллельный алгоритм показывает тот же порядок точности, что и последовательный.

7. Заключение

Разработанные двухсеточные параллельные алгоритмы решения уравнений аномальной диффузии дробного порядка показывают сверхлинейное ускорение по сравнению с классическим последовательным конечно-разностным алгоритмом и обеспечивают тот же порядок точности вычислений при условии согласованного выбора шагов точной и грубой сеток. Это свидетельствует о неоптимальности классического алгоритма. Поэтому представляется целесообразным и при последовательных расчетах использовать двухсеточные алгоритмы, получаемые очевидной редукцией предложенных параллельных алгоритмов на однопроцессорный случай.

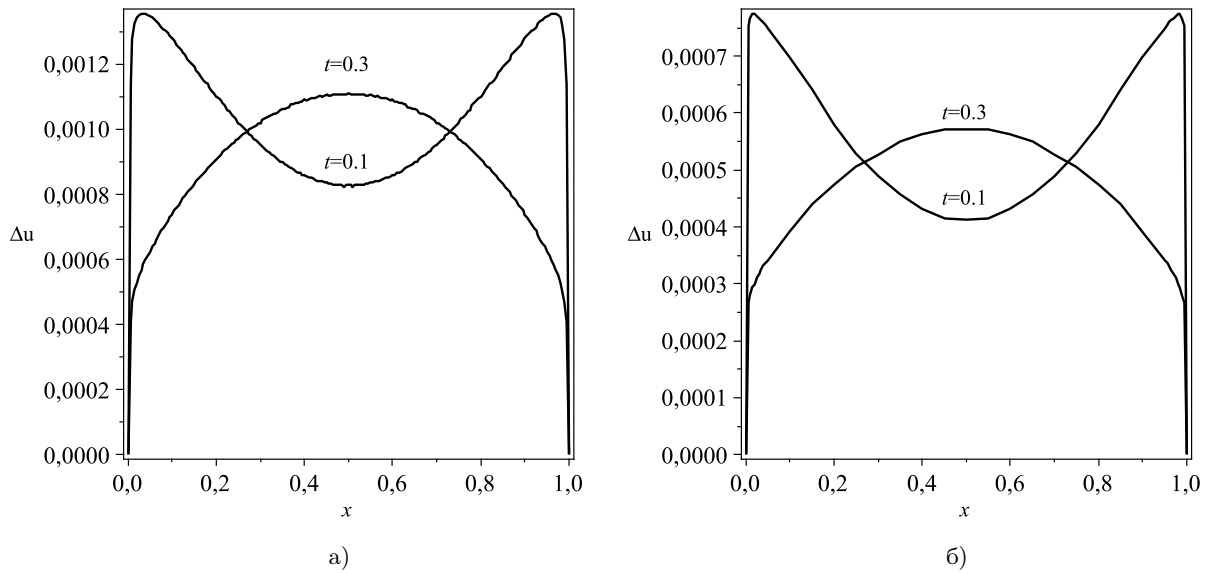


Рис. 2. Ошибка решения Δu , полученного по классическому последовательному алгоритму: а) $n = 200$, б) $n = 400$

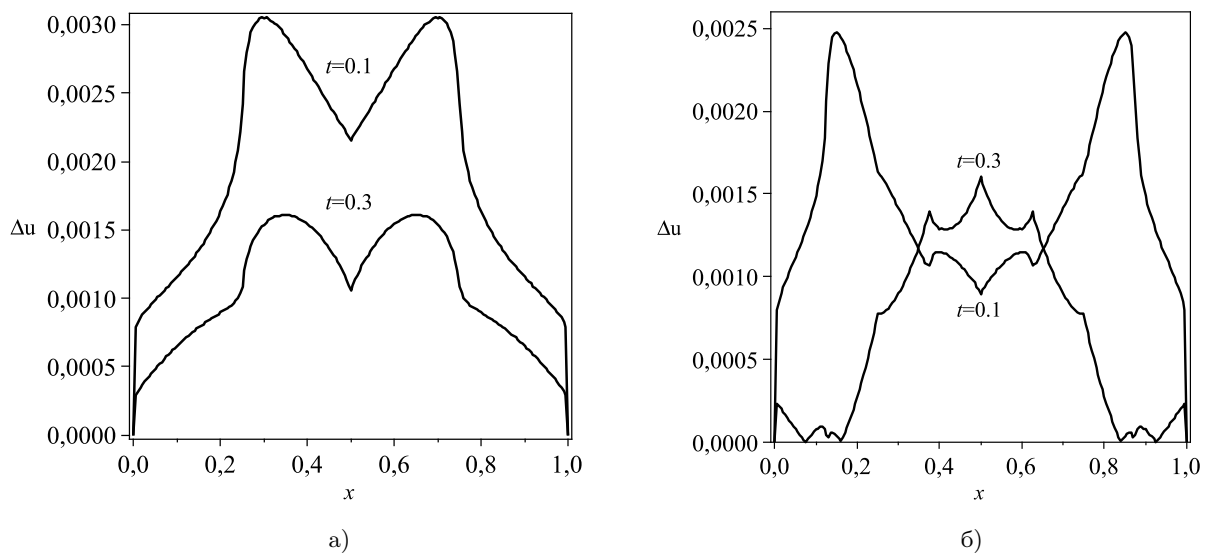


Рис. 3. Ошибка решения Δu , полученного с использованием параллельного алгоритма с декомпозицией по пространству: а) $P = N = 4$, $n = 100$, б) $P = N = 8$, $n = 50$

Литература

1. Bouchaud J.P., Georges A. Anomalous diffusion in disordered media: statistical mechanisms, models and physical applications // *Phys. Rep.* 1990. Vol. 195. P. 127–293.
2. Metzler R., Klafter J. The random walk's guide to anomalous diffusion: A fractional dynamic approach // *Phys. Rep.* 2000. Vol. 339. P. 1–77.
3. Учайкин В.В. Автомодельная аномальная диффузия и устойчивые законы. // *УФН.* 2003. Т. 173, No. 8. С. 847–876.
4. Podlubny I. *Fractional differential equations.* Academic press, San Diego, 1999. 365 p.
5. 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.
6. Учайкин В.В. *Метод дробных производных.* Ульяновск: Изд-во „Артишок“, 2008. 512 с.
7. Головизнин В.М., Киселев В.П., Короткин И.А., Юрков Ю.И. Прямые задачи неклассического переноса радионуклидов в геологических формациях // *Известия Академии наук, серия "Энергетика".* 2004. №. 4. С. 121–132.
8. Лукашук С.Ю. Модификация алгоритма PARAREAL для решения дифференциальных уравнений дробного порядка. // *Параллельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции.* Челябинск: Изд. ЮУрГУ, 2010. С. 519–524.
9. Завьялов Ю.С., Квасов Б.И., Мирошниченко В.Л. *Методы сплайн-функций.* М.: Наука, 1980. 352 с.
10. Lions J.-L., Maday Y., Turinici G. Resolution d'edp par un schema en temps parareal // *C.R. Acad Sci. Paris. Ser. I Math.* 2001. Vol. 332. P. 661–668.
11. 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.
12. Staff G.A., Ronquist 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.
13. 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.
14. 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.
15. 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.
16. 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.