

Схема «правый угол» и ее распараллеливание для численного решения уравнения переноса с запаздыванием

Е.С. Елькин
evgeny.elkin@urfu.ru

С.В. Свиридов
sergey.sviridov@urfu.ru

УрФУ (Екатеринбург)

Аннотация

Рассматривается уравнение переноса с запаздыванием с одномерной переменной по пространству. В качестве метода для численного решения данного уравнения используется разностная схема «правый угол». В работе рассматривается механизм распараллеливания указанной схемы. Для полноты эксперимента рассматриваются различные виды запаздывания: постоянное, переменное и распределенное. Проведен замер эффективности данного метода для последовательной и параллельной версии на разных количествах узлов по пространству и по времени. Предоставлены результаты проведенных экспериментов.

1 Введение

Уравнение переноса — это уравнение в частных производных первого порядка. В моделях физики такое уравнение часто называют уравнением конвекции, в моделях биологии — уравнением адвекции. Оно может осложняться различными видами запаздывания [1]. В силу сложности объектов, на первый план выходит конструирование численных алгоритмов их решения.

Численные методы решения уравнения переноса без запаздывания достаточно хорошо изучены, например, в работах [3, 2, 5]. Вопросы численного решения уравнений переноса с запаздыванием рассматривались, например, в [8, 9]. Методы распараллеливания рассматривались ранее, например, в [10, 11, 12]. В данной работе исследуются вопросы распараллеливания для конкретного алгоритма.

Рассмотрим уравнение переноса с эффектом наследственности:

$$\frac{\partial u(x, t)}{\partial t} + a \frac{\partial u(x, t)}{\partial x} = f(x, t, u(x, t), u_t(x, \cdot)), \quad (1)$$

$$0 \leq t \leq T, \quad 0 \leq x \leq X,$$

с краевым условием

$$u(0, t) = \gamma(t), \quad 0 \leq t \leq T, \quad (2)$$

и начальным условием

$$u(x, s) = \varphi(x, s), \quad 0 \leq x \leq X, \quad -\tau \leq s \leq 0. \quad (3)$$

Copyright © by the paper's authors. Copying permitted for private and academic purposes.

In: A.A. Makhnev, S.F. Pravdin (eds.): Proceedings of the 47th International Youth School-conference “Modern Problems in Mathematics and its Applications”, Yekaterinburg, Russia, 02-Feb-2016, published at <http://ceur-ws.org>

Здесь x, t — независимые переменные, $u(x, t)$ — искомая функция, $u_t(x, \cdot) = \{u(x, t + s), -\tau \leq s < 0\}$ — функция-предыстория искомой функции к моменту t , $\tau > 0$ — величина запаздывания.

Предполагается, что функционал f , функции $\gamma(t)$, $\varphi(x, s)$ и константа $a > 0$ таковы, что задача (1)–(3) имеет единственное решение.

2 Дискретизация задачи

Пусть шаг h по переменной x такой, что $X/h = N$, где N — натуральное, тогда обозначим через $x_i = ih \in [0, X]$, $i = 0 \dots N$. Пусть шаг Δ по переменной t такой, что $\tau/\Delta = m$, где m — натуральное, пусть $M = \lfloor T/\Delta \rfloor$, обозначим $t_j = j\Delta \in [-\tau, T]$, $j = -m \dots M$. Сеткой назовем набор пар $\{x_i, t_j\}$. Через u_j^i обозначим приближенное значение $u(x_i, t_j)$.

При всяком фиксированном $i = 0, \dots, N$ введем дискретную предысторию к моменту $t_j, j = 0, \dots, M$: $\{u_k^i\}_j = \{u_k^i, j - m \leq k \leq j\}$. *Оператором интерполяции-экстраполяции* назовем оператор, определенный на множестве всех допустимых предысторий и действующий по правилу $I: \{u_k^i\}_j \rightarrow v_j^i(\cdot) \in Q[t_j - \tau, t_j + \Delta]$. Здесь $Q[\alpha, \beta]$ — это набор функций $u(s)$, кусочно-непрерывных на $[\alpha, \beta]$ с конечным числом точек разрыва первого рода. В точках разрыва будем считать функцию непрерывной справа. Определим норму $Q = Q[\alpha, \beta]$ как

$$\|u(\cdot)\|_Q = \max_{s \in [\alpha, \beta]} |u(s)|.$$

Будем полагать, что, во-первых, оператор интерполяции-экстраполяции липшицев, то есть существует такая константа L_I , что для всех предысторий дискретной модели $\{u_k^i\}_j$ и $\{y_k^i\}_j$ выполняется

$$\max_{t_j - \tau \leq t \leq t_j + \Delta} |v_j^i(t) - w_j^i(t)| \leq L_I \max_{j-m \leq l \leq j} |u_l^i - y_l^i|,$$

где $v_j^i(\cdot) = I(\{u_k^i\}_j)$, $w_j^i(\cdot) = I(\{y_k^i\}_j)$.

Во-вторых, положим, что оператор интерполяции-экстраполяции согласован, то есть

$$v_j^i(t_k) = u_k^i, \quad k = j - m, \dots, j.$$

Будем говорить, что оператор интерполяции-экстраполяции имеет порядок p , если существуют константы C_1 и C_2 такие, что $|v_j^i(t) - u(x_i, t)| \leq C_1 \max_{j-m \leq k \leq j} |u_k^i - u(x_i, t_k)| + C_2 \Delta^p$ для всех i, j и $t \in [t_j - \tau, t_{j+1}]$.

Простейший способ интерполяции — кусочно-линейная функция. Простейший способ экстраполяции — экстраполяция продолжением (см. [7]). Будем использовать кусочно-линейную интерполяцию с экстраполяцией продолжением, которая является липшицевым оператором ($L_I = 2$), согласована и имеет порядок $p = 2$ (см. [7]).

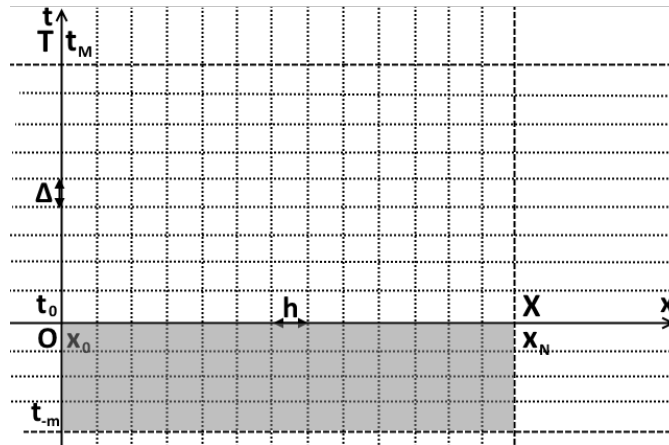


Рис. 1: Сетка

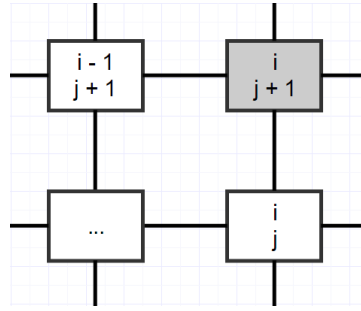


Рис. 2: Изображение разностной схемы «Правый угол»

3 Схема «Правый угол»

Рассмотрим метод:

$$\frac{u_{j+1}^i - u_j^i}{\Delta} + a \frac{u_{j+1}^i - u_{j+1}^{i-1}}{h} = f(x_i, t_j, u_j^i, u_t(x, \cdot)). \quad (4)$$

Каждая производная в (4) представлена как разность приближенных значений функции в узлах сетки, показанных на рис. 2. Значение u_{j+1}^{i-1} может быть выражено из ранее подсчитанных значений, значит, рассматриваемый метод «правый угол» является явным для уравнения (1)–(3).

Обозначим $\varepsilon_j^i = u(x_i, t_j) - u_j^i$, $i = 0, \dots, N$, $j = 0, \dots, M$. Будем говорить, что *метод сходится*, если $\varepsilon_j^i \rightarrow 0$ при $h \rightarrow 0$ и $\Delta \rightarrow 0$ для всех $i = 0, \dots, N$ и $j = 0, \dots, M$. Будем говорить, что метод сходится с порядком $h^p + \Delta^q$, если существует константа C такая, что $|\varepsilon_j^i| \leq C(h^p + \Delta^q)$ для всех $i = 0, \dots, N$ и $j = 0, \dots, M$. В [4] доказано, что рассматриваемый метод сходится с порядком $O(\Delta + h)$. Данная схема входит в семейство методов, рассмотренных в [6].

4 Идея параллельного вычисления

В данной схеме для вычисления очередных неизвестных значений используются значения, стоящие ниже или левее по x и по t , следовательно, мы можем независимо друг от друга вычислять элементы, находящиеся на одной диагонали (далее будем называть эти «диагональные» элементы *фронт*ом). Это дает возможность для параллельного вычисления результирующей функции. Схематичное расположение «фронтов» для параллельного вычисления показано на рис. 3.

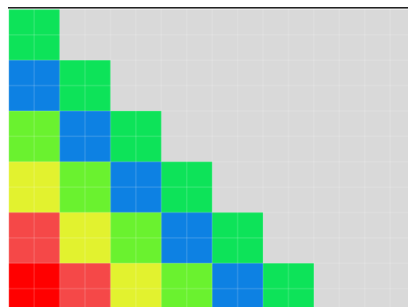


Рис. 3: Схема «Правый угол»

Для реализации данного подхода использовалось многопоточное программирование. В качестве языка было решено использовать C# и платформу .Net, включающую примитивы многопоточного программирования. При реализации использовалась сборка System.Threading.Tasks, в частности метод Parallel.For, позволяющий распараллелить выполнение цикла For между потоками. Все вычисление выполнялось пошагово по диагоналям. На каждом шаге выбиралась следующая диагональ, для узлов которой параллельно вычислялись искомые значения. Так как элементы на диагонали не зависят друг от друга, то данный подход является корректным с точки зрения вычислений. На каждом этапе параллельного вычисления

искомых значений в узлах диагонали платформа порождает 8 потоков и распределяет узлы между ними по мере возможности. Из-за того, что одновременно вычисляется не более 8 потоков и количество элементов диагонали в большинстве случаев больше 8, эффект «узкого фронта распараллеливания» был незначительным, поэтому в большинстве примеров мы рассматривали сетки с одинаковым числом узлов по осям. Так как целью данной работы является исследование возможности ускорения путем распараллеливания вычислений, то мы не прибегали к другим приемам программных и алгоритмических оптимизаций (векторизации, кешированию и др.). В итоге, мы оценивали потенциал ускорения при помощи лишь многопоточного программирования.

5 Численные эксперименты

Во всех таблицах с результатами экспериментов используем следующие обозначения:

Err — максимум модуля разности точного и приближенного решений в узлах сетки. $T1$ — время выполнения последовательной версии (в миллисекундах). $T2$ — время выполнения параллельной версии (в миллисекундах).

5.1 Пример 1. Постоянное запаздывание

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \pi t \cos(\pi x) + (3 - t) \sin(\pi x) u(x, t - 2),$$

$$0 \leq t \leq 4, \quad 0 \leq x \leq 4,$$

с краевым условием:

$$u(0, t) = 0, \quad 0 \leq t \leq 4,$$

с начальным условием:

$$u(x, t) = t \sin(\pi x), \quad 0 \leq x \leq 4, \quad -2 \leq t \leq 0.$$

Точное решение данного уравнения:

$$u(x, t) = t \sin(\pi x).$$

Таблица 1: Численные эксперименты для примера 1

$N =$	40	40	80	80	160	320	640
$M =$	40	80	40	80	160	320	640
Err	1.31	1.2	1.18	0.65	0.32	0.15	0.08
T1	0	1	1	1	6	31	132
T2	12	8	8	10	12	35	72

5.2 Пример 2. Переменное запаздывание

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = u\left(x, \frac{t}{2} - \frac{1}{2}\right) + 6x + 3t^2 + 6t - 3x^3 - \left(\frac{t}{2}\right)^3 - \frac{5}{8},$$

$$0 \leq t \leq 1, \quad 0 \leq x \leq 2,$$

с краевым условием:

$$u(0, t) = 3t^2 + 6t, \quad 0 \leq t \leq 4,$$

с начальным условием:

$$u(x, t) = x^3 + t^3 + 3x^2 + 3t^2, \quad 0 \leq x \leq 1, \quad -\frac{1}{2} \leq t \leq 0.$$

Точное решение данного уравнения:

$$u(x, t) = x^3 + t^3 + 3x^2 + 3t^2.$$

Таблица 2: Численные эксперименты для примера 2

$N =$	40	80	160	320	640
$M =$	40	80	160	320	640
Err	4.15	2.56	1.84	0.98	0.42
T1	4	9	32	196	628
T2	14	19	43	152	439

5.3 Пример 3. Распределенное запаздывание

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = (1-x)\sin(t) - \int_{-\pi}^{-\frac{\pi}{2}} u(x, t + \xi) d\xi,$$

$$0 \leq t \leq 4, \quad 0 \leq x \leq \pi,$$

с краевым условием:

$$u(0, t) = 0, \quad 0 \leq t \leq \pi,$$

с начальным условием:

$$u(x, t) = x \sin(t), \quad 0 \leq x \leq 4, \quad -\pi \leq t \leq 0.$$

Точное решение данного уравнения:

$$u(x, t) = x \sin(t).$$

Таблица 3: Численные эксперименты для примера 3

$N =$	40	80	160	320	640
$M =$	40	80	160	320	640
Err	1.68	0.92	0.65	0.34	0.18
T1	1	4	12	122	680
T2	12	16	39	97	421

Отметим, что численные эксперименты для примера 3 проводились без использования рекуррентных соотношений между интегралами по пересекающимся интервалам. Сделано это было для большего числа вычислений на каждом шаге и анализа эффективности использования параллелизма для численного решения уравнений рассматриваемых типов.

6 Заключение

Все вычисления выполнялись на ноутбуке с процессором Intel(R) Core i7 4700MQ 2.4GHz x 4, поддерживающем одновременное выполнение 8 потоков. Для чистоты эксперимента все вычисления проводились в максимально «стерильных условиях» (на компьютере запущены только необходимые приложения), однако операционная система и фоновые процессы в любом случае влияют на время вычислений, поэтому в примерах указано среднее время выполнения среди десяти запусков. Также можно заметить сильное замедление параллельной программы на малом количестве узлов по сравнению с последовательной версией. Это вызвано временем, которое уходит у платформы на подготовку многопоточной экосистемы и переключение между потоками. Последовательная программа лишена данных недостатков. В качестве эффективности в данной работе рассматривается отношение времени выполнения параллельной программы ко времени выполнения последовательной. В дальнейшем планируется более глубокий анализ алгоритмов параллельного вычисления решений описанных выше уравнений: с учетом различных показателей эффективности, таких как слабая и сильная масштабируемость.

Временные результаты вычислений на сетках с небольшим количеством узлов обусловлены тем, что рассмотренная в статье параллельная версия правого уголка тратит много времени на подготовку потоков и переключение между ними. Однако с увеличением количества узлов видно, что параллельный алгоритм становится быстрее последовательного. Так как наибольший интерес с практической точки зрения представляют более точные вычисления на мелкой сетке, то можем сделать вывод, что полученные преимущества при распараллеливании подсчетов можем считать основанием для применения рассмотренного алгоритма на практике и для ускорения аналогичных явных разностных схем.

Благодарности

Работа поддержана программой ППК (постановление Правительства РФ № 211 от 16.03.2013).

Список литературы

- [1] J. Wu. *Theory and Application of Partial Functional Differential Equations*. Springer-Verlag, New York, 1996.
- [2] N. N. Kalitkin. *Numerical methods. Second edition*. BHV-Peterburg, Sankt-Peterburg, 2011 (in Russian). = Н. Н. Калиткин. *Численные методы. 2-е издание*. БХВ-Петербург, Санкт-Петербург, 2011.
- [3] A. A. Samarsky. *Theory of differential schemes. Third edition*. Nauka, Moscow, 1989 (in Russian). = А. А. Самарский. *Теория разностных схем. 3-е издание*. Наука, Москва, 1989.
- [4] V. G. Pimenov. *Differential methods for solution of functional equations with delay*. Ural State University, Ekaterinburg, 2014 (in Russian). = В. Г. Пименов. *Разностные методы решения уравнений в частных производных с наследственностью*. Издательство уральского университета, Екатеринбург, 2014.
- [5] I. B. Petrov, A. I. Lobanov. *Lectures on computational mathematics*. Vinom, Moscow, 2006 (in Russian). = И. Б. Петров, А. И. Лобанов. *Лекции по вычислительной математике*. Бином, Москва, 2006.
- [6] L. S. Volkanin. Numerical solution of an advection equation with delay. *Teoriya upravleniya i matematicheskoe modelirovanie. Conference proceedings. Izhevsk, 12–13, 2012* (in Russian). = Л. С. Волканин. Численное решение уравнения переноса с эффектом наследственности. *Теория управления и математическое моделирование. Тезисы конференции. Ижевск, 12–13, 2012*.
- [7] A. V. Kim, V. G. Pimenov. *i-smooth analysis and a numerical methods for solving of a functional-differential equations*. RCD, Moscow-Izhevsk, 2004 (in Russian). = А. В. КИМ, В. Г. ПИМЕНОВ. *i-гладкий анализ и численные методы решения функционально-дифференциальных уравнений*. РХД, Москва-Ижевск, 2004.
- [8] V. G. Pimenov, S. V. Sviridov. Grid methods for solving of an advection equation with delay. *Vestnik Udmurtskogo Universiteta. Matematika. Mehanika. Kompyuternye nauki*, 3:59–74, 2014 (in Russian). = В. Г. Пименов, С. В. Свиридов. Сеточные методы решения уравнения переноса с запаздыванием. *Вестник Удмуртского университета. Математика. Механика. Компьютерные науки*, 3:59–74, 2014.
- [9] S.I. Solodushkin. One difference scheme for numerical solving of advection equation with aftereffect. *Proceedings of 8th Conference on Applied Mathematics and Scientific Computing*, Sibenik, Croatia, 10-14 June, 54–55, 2013.
- [10] S.I. Solodushkin, I.F.Yumanova, R.H. De Staelen. A difference scheme for multidimensional transfer equations with time delay. *Journal of Computational and Applied Mathematics*, 318:580–590, 2017.
- [11] S.I. Solodushkin, A.A. Sagoyan, I.F. Iumanova. Parallel variant of numerical algorithm for solving a multidimensional advection equation with time delay. *CEUR Workshop Proceedings*, 1662:315–325, 2016.
- [12] S.I. Solodushkin, A.A. Sagoyan, I.F. Iumanova. One parallel method for solving the multidimensional transfer equation with aftereffect. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10187:617–624, 2017.

Parallelization of «the right corner» scheme for numerical solution of an advection equation with delay

Eugene S. Elkin, Sergey V. Sviridov
Ural Federal University (Yekaterinburg, Russia)

Keywords: advection equation, delay, grid schemes, parallelization.

We consider an advection equation with delay and the right corner scheme for numerical solution of the considered equation. A parallelization method is proposed. Test examples with different types of delay are given. The considered parallel method is compared with a serial implementation of the right corner method. Results for all experiments are given.