

## Улучшение локальности параллельных алгоритмов численного решения двумерных квазилинейных параболических уравнений

С. В. Баханович\*, Н. А. Лиходед†, П. А. Мандрик†

\* *Институт математики НАН Беларуси  
ул. Сурганова, д. 11, Минск, Республика Беларусь, 220072*

† *Факультет прикладной математики и информатики  
Белорусский государственный университет  
пр. Независимости, д. 4, Минск, Республика Беларусь, 220030*

Уравнения параболического типа описывают процессы нелинейной теплопроводности, диффузии заряженных частиц в плазме, диффузии и дрейфа примесных атомов в полупроводниковых структурах, в химической кинетике. При численном решении практических задач такого рода появляются трудности, обусловленные недостаточными мощностью и объёмом оперативной памяти персонального компьютера. Возникает задача построения параллельных методов и алгоритмов для численного решения параболических уравнений на суперкомпьютерах. Одним из методов численного решения многомерных параболических уравнений является локально-одномерный метод. В работе предлагается параллельная реализация локально-одномерного метода численного решения линейных и квазилинейных двумерных параболических уравнений с краевыми условиями первого рода на суперкомпьютерах с распределённой памятью. Параллельный алгоритм построен с учётом локализации данных — операции и данные перераспределены между процессами таким образом, что значительная часть данных приватизирована процессами и не требует коммуникационных операций. Приведены результаты численных экспериментов.

**Ключевые слова:** параллельные вычисления, локализация данных, параболические уравнения, локально-одномерный метод.

### 1. Введение

Уравнения параболического типа описывают процессы нелинейной теплопроводности, диффузии заряженных частиц в плазме, диффузии и дрейфа примесных атомов в полупроводниковых структурах, фильтрации газов и жидкостей в пористых средах, в химической кинетике. При численном решении практических задач такого рода появляются трудности, обусловленные недостаточными мощностью и объёмом оперативной памяти персонального компьютера. Возникает задача построения параллельных методов и алгоритмов для численного решения параболических уравнений на суперкомпьютерах.

В работах [1, 2] предложены параллельные реализации локально-одномерного метода численного решения линейных и квазилинейных двумерных параболических уравнений с краевыми условиями первого рода на суперкомпьютерах с распределённой памятью. Параллельные реализации основаны на естественном параллелизме локально-одномерного метода. Вычислительные процессы на каждом временном слое могут выполняться независимо, но при переходе к новому слою требуется обращение каждого процесса к локальной памяти всех других процессов. С ростом числа используемых процессов это приводит к большим накладным расходам на коммуникационные операции.

В этой работе предлагается отказаться от использования части естественного параллелизма и получить параллельные алгоритмы с улучшенной локальностью. Локальность алгоритма — это его вычислительное свойство, отражающее степень использования памяти с быстрым доступом [3, 4]. При многопроцессорной обработке памятью с быстрым доступом считается локальная память процессора. Локальность параллельного алгоритма, предназначенного для реализации на компьютерах с распределённой памятью, напрямую определяет коммуникационные затраты при выполнении программы.

## 2. Вычислительный алгоритм локально-одномерного метода

Рассмотрим в области  $Q_T = G \times [0 < t \leq T]$ ,  $G = [0 < x_1 < l_1] \times [0 < x_2 < l_2]$ , квазилинейное параболическое уравнение

$$\frac{\partial u}{\partial t} = \sum_{m=1}^2 \left( \frac{\partial}{\partial x_m} \left( k_m(x, t, u) \frac{\partial u}{\partial x_m} \right) + \frac{\partial}{\partial x_m} (r_m(x, t, u)u) \right) - q(x, t, u)u + f(x, t, u), \quad (1)$$

$$q(x, t, u) \geq 0, \quad x = (x_1, x_2),$$

с начальными

$$u(x, 0) = u_0(x), \quad x \in G, \quad (2)$$

и краевыми

$$u(x, t)|_{S_t} = \mu(x, t), \quad (x, t) \in S_t \equiv \partial Q_T, \quad (3)$$

условиями.

Введём на отрезке  $[0 \leq t \leq T]$  сетку узлов  $\omega_\tau = \{t_j = j\tau, j = 0, 1, \dots, j_0, j_0\tau = T\}$ ; в области  $G$  введём сетку узлов  $\bar{\omega}_h = \{x_1^{(i_1)} = i_1 h_1, i_1 = 0, 1, \dots, N_1, N_1 h_1 = l_1, x_2^{(i_2)} = i_2 h_2, i_2 = 0, 1, \dots, N_2, N_2 h_2 = l_2\}$ . Если согласно [5] осуществить дифференциальное расщепление задачи (1)–(3), затем построить итерационный процесс и применить метод прогонки для решения возникающих систем линейных алгебраических уравнений, то получим алгоритм приближенного решения задачи (1)–(3). Данный подход к решению задачи подробно описан в работе [2].

Алгоритм решения задачи (1)–(3) схематично можно представить следующим образом (циклы, итерации которых заведомо можно выполнять независимо, запишем как `dopar`):

```

do j = 1, j_0
  dopar i_2 = 1, N_2 - 1
    do s = 0, s_M
      do i_1 = 1, N_1 - 1
         $\check{y}_{(1)}^s(i_1, i_2) = y_{(1)}(i_1, i_2)$ 
      enddo
      do i_1 = 1, N_1 - 1
         $\alpha(i_1 + 1) = F_1(\alpha(i_1), \check{y}_{(1)}^s(i_1 - 1, i_2), \check{y}_{(1)}^s(i_1, i_2), \check{y}_{(1)}^s(i_1 + 1, i_2))$ 
         $\beta(i_1 + 1) = F_2(\alpha(i_1), \beta(i_1), y_{(2)}(i_1, i_2), \check{y}_{(1)}^s(i_1 - 1, i_2), \check{y}_{(1)}^s(i_1, i_2), \check{y}_{(1)}^s(i_1 + 1, i_2))$ 
      enddo
      do i_1 = 1, N_1 - 1
         $y_{(1)}(N_1 - i_1, i_2) = \alpha(N_1 - i_1 + 1)y_{(1)}(N_1 - i_1 + 1, i_2) + \beta(N_1 - i_1 + 1)$ 
      enddo
    enddo(s)
  enddopar
  dopar i_1 = 1, N_1 - 1
    do s = 0, s_M
      do i_2 = 1, N_2 - 1
         $\check{y}_{(2)}^s(i_1, i_2) = y_{(2)}(i_1, i_2)$ 
      enddo
      do i_2 = 1, N_2 - 1
         $\alpha(i_2 + 1) = F_3(\alpha(i_2), \check{y}_{(2)}^s(i_1, i_2 - 1), \check{y}_{(2)}^s(i_1, i_2), \check{y}_{(2)}^s(i_1, i_2 + 1))$ 
         $\beta(i_2 + 1) = F_4(\alpha(i_2), \beta(i_2), y_{(1)}(i_1, i_2), \check{y}_{(2)}^s(i_1, i_2 - 1), \check{y}_{(2)}^s(i_1, i_2), \check{y}_{(2)}^s(i_1, i_2 + 1))$ 
      enddo
      do i_2 = 1, N_2 - 1
         $y_{(2)}(i_1, N_2 - i_2) = \alpha(N_2 - i_2 + 1)y_{(2)}(i_1, N_2 - i_2 + 1) + \beta(N_2 - i_2 + 1)$ 
      enddo
    enddo(s)
  enddopar
enddo(j)

```

**enddopar**

**enddo** ( $j$ )

Здесь  $\alpha$  и  $\beta$  — коэффициенты прогонки, возникающие при решении промежуточных систем линейных алгебраических уравнений. Число  $s_M$  в алгоритме задаёт максимальное количество итераций. Конечные значения  $y_{(2)}(i_1, i_2)$  являются приближенным решением задачи (1)–(3).

### 3. Параллельный алгоритм с улучшенной локальностью

Для отображения алгоритмов на параллельные компьютеры с распределённой памятью требуется распределить операции алгоритма и данные между процессами, установить порядок выполнения операций в каждом процессе и организовать обмены данными [6].

На основе алгоритма, описанного в предыдущем разделе, можно построить параллельный алгоритм, в котором операции и данные перераспределены между процессами таким образом, что значительная часть данных приватизирована процессами и не требует коммуникационных операций. При этом теряется использование части естественного параллелизма, но за счёт локализации данных время реализации параллельного алгоритма может существенно уменьшиться.

Пусть  $P$  число процессов, предназначенных для реализации алгоритма, числа  $Q_2$  и  $r_2$  связаны соотношением  $r_2 = \lceil (N_2 - 1)/Q_2 \rceil$ . Будем предполагать, что  $r_1 = (N_1 - 1)/P$  является целым числом. Приведём псевдокод выполнения алгоритма.

Для каждого процесса  $p$ ,  $0 \leq p \leq P - 1$ :

```

do  $j = 1, j_0$ 
  do  $s = 0, s_M$ 
    do  $q_2 = 0, Q_2 - 1$ 
      do  $i_2 = 1 + q_2 r_2, \min((q_2 + 1)r_2, N_2 - 1)$  // Начало тайла 0-типа
        do  $i_1 = 1 + p r_1, (p + 1)r_1$ 
           $\overset{s}{y}_{(1)}(i_1, i_2) = y_{(1)}(i_1, i_2)$ 
        enddo
        do  $i_1 = 1 + p r_1, (p + 1)r_1$ 
           $\alpha(i_2, i_1 + 1) = F_1(\alpha(i_2, i_1), \overset{s}{y}_{(1)}(i_1 - 1, i_2), \overset{s}{y}_{(1)}(i_1, i_2), \overset{s}{y}_{(1)}(i_1 + 1, i_2))$ 
           $\beta(i_2, i_1 + 1) = F_2(\alpha(i_2, i_1), \beta(i_2, i_1), y_{(2)}(i_1, i_2), \overset{s}{y}_{(1)}(i_1 - 1, i_2),$ 
             $\overset{s}{y}_{(1)}(i_1, i_2), \overset{s}{y}_{(1)}(i_1 + 1, i_2))$ 
        enddo
      enddo ( $i_2$ ) // Конец тайла 0-типа. Обмен данными
    enddo ( $q_2$ )
    do  $q_2 = 0, Q_2 - 1$ 
      do  $i_2 = 1 + q_2 r_2, \min((q_2 + 1)r_2, N_2 - 1)$  // Начало тайла 1-типа
        do  $i_1 = 1 + (P - p - 1)r_1, (P - p)r_1$ 
           $y_{(1)}(N_1 - i_1, i_2) = \alpha(i_2, N_1 - i_1 + 1)y_{(1)}(N_1 - i_1 + 1, i_2) + \beta(i_2, N_1 - i_1 + 1)$ 
        enddo
      enddo ( $i_2$ ) // Конец тайла 1-типа. Обмен данными
    enddo ( $q_2$ )
  enddo ( $s$ )
  do  $i_1 = 1 + p r_1, (p + 1)r_1$  // Начало тайла 2-типа
    do  $s = 0, s_M$ 
      do  $i_2 = 1, N_2 - 1$ 
         $\overset{s}{y}_{(2)}(i_1, i_2) = y_{(2)}(i_1, i_2)$ 
      enddo
      do  $i_2 = 1, N_2 - 1$ 
         $\alpha(i_2 + 1) = F_3(\alpha(i_2), \overset{s}{y}_{(2)}(i_1, i_2 - 1), \overset{s}{y}_{(2)}(i_1, i_2), \overset{s}{y}_{(2)}(i_1, i_2 + 1))$ 
         $\beta(i_2 + 1) = F_4(\alpha(i_2), \beta(i_2), y_{(1)}(i_1, i_2), \overset{s}{y}_{(2)}(i_1, i_2 - 1), \overset{s}{y}_{(2)}(i_1, i_2), \overset{s}{y}_{(2)}(i_1, i_2 + 1))$ 
      enddo
      do  $i_2 = 1, N_2 - 1$ 
         $y_{(2)}(i_1, N_2 - i_2) = \alpha(N_2 - i_2 + 1)y_{(2)}(i_1, N_2 - i_2 + 1) + \beta(N_2 - i_2 + 1)$ 
      enddo
    enddo ( $s$ )
  enddo ( $i_1$ )
enddo ( $j$ )

```

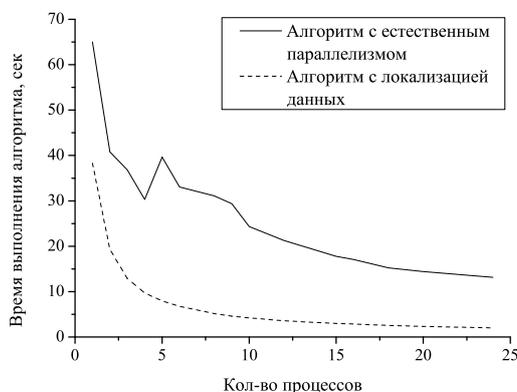
**enddo** ( $i_1$ ) // Конец тайла 2-типа. Обмен данными не требуется.

**enddo** ( $j$ )

Используются три типа зерна вычислений (три типа тайлов). После вычислений тайлов 0-го и 1-го типов требуются коммуникационные операции небольшого объёма. Для выполнения операций тайлов 2-го типа коммуникационных операций не требуется вообще, так как все необходимые для вычислений данные находятся в локальной памяти процесса.

Отметим, что для численного решения линейных двумерных параболических уравнений (частный случай уравнения (1)) не требуется строить итерационный процесс с итерациями по  $s$ .

На рис. 1 приведены результаты вычислительных экспериментов на кластере СКИФ-БГУ при  $N_1 = N_2 = 1440$ ,  $j_0 = 25$ ,  $r_2 = 15$ . Улучшение локальности привело к значительному уменьшению времени выполнения алгоритма.



**Рис. 1.** Результаты выполнения алгоритмов с естественным параллелизмом и локализацией данных на суперкомпьютере СКИФ-БГУ

#### 4. Заключение

Построен параллельный алгоритм, в котором операции и данные перераспределены между процессами таким образом, что значительная часть данных приватизирована процессами и не требует коммуникационных операций. Предложенным способом можно улучшить локальность и некоторых других параллельных алгоритмов, требующих групповых коммуникационных операций (обращение каждого процесса к локальной памяти всех других процессов). В частности, можно построить новые параллельные алгоритмы, реализующие метод переменных направлений.

#### Литература

1. Параллельная реализация локально-одномерного метода численного решения двумерных параболических уравнений / Н. А. Лиходед, С. В. Баханович, Г. М. Заяц, В. А. Цурко // Информатика. — 2010. — Т. 17, № 4. — С. 72–80. [Parallel Implementation of Locally One-Dimensional Method for Numerical Solution of Two-Dimensional Parabolic Equations / N. A. Likhoded, S. V. Bakhanovich, G. M. Zayats, V. A. Tsurko // Informatics. — 2010. — Vol. 17, No 4. — Pp. 72–80. ]

2. Параллельная реализация локально-одномерного метода численного решения двумерных параболических уравнений / С. В. Баханович, Г. М. Заяц, Н. А. Лиходед, С. А. Артемчик // Международный конгресс по информатике: информационные системы и технологии CSIST'2011, Минск, Беларусь. БГУ. — 2011. — Т. 2. — С. 76–81. [Parallel Implementation of Locally One-Dimensional Method for Numerical Solution of Two-Dimensional Parabolic Equations / S. V. Bakhanovich, G. M. Zayats, N. A. Likhoded, S. A. Artemchik // International Congress on Computer Science: Information Systems and Technologies (CSIST'2013). — 2011. — Vol. 2. — Pp. 76–81. ]
3. Garcia J., Ayguade E., Labarta J. A Framework for Integrating Data Alignment, Distribution and Redistribution in Distributed Memory Multiprocessors // IEEE Transactions on Parallel and Distributed Systems. — 2001. — Vol. 12, No 4. — Pp. 416–430.
4. Адуткевич Е. В., Лиходед Н. А., Сикорский А. О. К распараллеливанию последовательных программ: распределение массивов между процессорами и структуризация коммуникаций // Кибернетика и системный анализ. — 2012. — № 1. — С. 144–163. [Adutskevich E. V., Likhoded N. A., Sikorskiy A. O. By Parallelization of Sequential Programs: Distribution of Arrays Between Processors and Structuring Communications // Cybernetics and Systems Analysis. — 2012. — No 4. — Pp. 144–163. ]
5. Самарский А. А. Теория разностных схем. — М.: Наука, 1989. [Samarskiy A. A. The Theory of Difference Schemes. — Moscow: Nauka, 1989. ]
6. Воеводин В. В., Воеводин В. В. Параллельные вычисления. — СПб.: БХВ-Петербург, 2002. [Voevodin V. V., Voevodin V. V. Parallel Computing. — Saint Petersburg: BHV-Peterburg, 2002. ]

UDC 519.6

## Improvement of Locality of Parallel Algorithms of the Numerical Solutions of the Two-Dimensional Quasilinear Parabolic Equations

S. V. Bakhanovich\*, N. A. Likhoded<sup>†</sup>, P. A. Mandrik<sup>†</sup>

\* *Institute of Mathematics, NAS of Belarus,  
11 Surganov str., Minsk, Republic of Belarus, BY-220072*

<sup>†</sup> *Faculty of Applied Mathematics and Computer Science  
Belarusian State University  
4, Nezavisimosti ave., Minsk, Republic of Belarus, 220030*

The equations of parabolic type describe processes of nonlinear heat conductivity, diffusions of the loaded particles in plasma, diffusion and drift of impurity atoms in semiconductor structures, in chemical kinetics. At the numerical solution of practical tasks such there are the difficulties caused by the insufficient capacity and volume of memory of the personal computer. There is a problem of creation of parallel methods and algorithms for the numerical decision the parabolic equations on supercomputers. One of methods of the numerical solution of the multidimensional parabolic equations the locally one dimensional method is. Parallel realization of a locally one method for numerical solutions of the linear and quasilinear two-dimensional parabolic equations on supercomputers with the distributed memory is offered. The parallel algorithm is constructed taking into account locality of data — the operations and data are distributed between processes in such a way that the considerable part of data is privatized by processes and doesn't need communication operations. Results of numerical experiments are given.

**Key words and phrases:** parallel computation, data locality, data alignment, parabolic equations, locally one dimensional method.