УДК 519.6

С.В. Баханович, Г.М. Заяц, Н.А. Лиходед, В.А. Цурко

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ ЛОКАЛЬНО-ОДНОМЕРНОГО МЕТОДА ЧИСЛЕННОГО РЕШЕНИЯ ДВУМЕРНЫХ ПАРАБОЛИЧЕСКИХ УРАВНЕНИЙ

Предлагаются алгоритмы и параллельные программы для реализации локально-одномерного метода численного решения двумерных параболических уравнений на суперкомпьютерах с распределенной памятью. Разработанные алгоритмы обладают высокими показателями эффективности параллельной реализации.

Введение

В настоящее время имеется большое число экономичных разностных схем для решения двумерных параболических уравнений. Одним из подходов к построению экономичных разностных схем является сведение двумерной задачи к последовательности одномерных задач. Такой подход к решению двумерных уравнений реализован в виде итерационных разностных методов: метода переменных направлений, метода дробных шагов, локально-одномерного метода. Перечисленные методы обладают устойчивостью и простотой решения. Для решения параболических уравнений применяются также прямые методы, среди которых достаточно универсален метод матричной прогонки, позволяющий решать уравнения с переменными коэффициентами и не накладывающий сильных ограничений на вид граничных условий. В то же время при численном решении практических задач появляются трудности, обусловленные недостаточными мощностью и объемом оперативной памяти персонального компьютера. Возникает задача построения параллельных методов и алгоритмов для численного решения параболических уравнений на суперкомпьютерах.

В статье разработаны алгоритмы и параллельные программы для численного решения двумерных параболических уравнений с краевыми условиями первого рода на суперкомпьютерах с распределенной памятью. Авторам работы неизвестны доступные для свободного применения соответствующие параллельные программы. Параллельные реализации получены путем адаптации к параллельному выполнению предварительно разработанной последовательной программы. Для этого потребовалось решить ряд задач статического распараллеливания: отображение операций на физические процессоры, выбор зерна вычислений, организация коммуникаций и обмена данными. Рассмотрена также модификация параллельного алгоритма, допускающая выполнение некоторых операций на фоне обмена данными. Проведены и проанализированы вычислительные эксперименты на суперкомпьютере СКИФ.

1. Локально-одномерный метод численного решения двумерных параболических уравнений

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

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x_1}(k_1(\mathbf{x},t)\frac{\partial u}{\partial x_1}) + \frac{\partial}{\partial x_2}(k_2(\mathbf{x},t)\frac{\partial u}{\partial x_2}) + f(\mathbf{x},t), \quad \mathbf{x} = (x_1, x_2), \tag{1}$$

с начальными

$$u(\mathbf{x},0) = u_0(\mathbf{x}), \quad x \in G, \tag{2}$$

и краевыми

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

условиями.

Согласно [1] осуществим дифференциальное расщепление задачи (1)–(3). На отрезке $[0 \le t \le T]$ введем сетку узлов $\omega_\tau = \{t_j = j\tau,\ j = 0, 1, ..., j_0,\ j_0\tau = T\}$. Представим функцию $f(\mathbf{x}, t)$ в виде суммы двух функций: $f(\mathbf{x}, t) = f_1(\mathbf{x}, t) + f_2(\mathbf{x}, t)$. На отрезках $t_{j-1} \le t \le t_j$, $j = 1, 2, ..., j_0$, уравнению (1) поставим в соответствие уравнения

$$\frac{\partial \upsilon_{(1)}}{\partial t} = \frac{\partial}{\partial x_1}\left(k_1(\mathbf{x}, t)\frac{\partial \upsilon_{(1)}}{\partial x_1}\right) + f_1(\mathbf{x}, t), \quad \mathbf{x} \in G; \tag{4}$$

$$\frac{\partial \upsilon_{(2)}}{\partial t} = \frac{\partial}{\partial x_2}\left(k_2(\mathbf{x}, t)\frac{\partial \upsilon_{(2)}}{\partial x_2}\right) + f_2(\mathbf{x}, t), \quad \mathbf{x} \in G. \tag{5}$$

Уравнения (4) и (5) связаны соотношениями

$$\upsilon_{(1)}(\mathbf{x}, t_j) = \upsilon_{(2)}(\mathbf{x}, t_j),\ \mathbf{x} \in G,\ j = 1, 2, ..., j_0 - 1; \tag{6}$$

$$\upsilon_{(2)}(\mathbf{x}, t_j) = \upsilon_{(1)}(\mathbf{x}, t_{j+1}),\ \mathbf{x} \in G,\ j = 0, 1, ..., j_0 - 1. \tag{7}$$

При этом

$$\upsilon_{(1)}(\mathbf{x}, 0) = u_0(\mathbf{x}),\ \mathbf{x} \in G. \tag{8}$$

Уравнение (4) дополним краевыми условиями

$$\upsilon_{(1)}(0, x_2, t) = \mu(0, x_2, t), \quad \upsilon_{(1)}(l_1, x_2, t) = \mu(l_1, x_2, t), \quad 0 \le x_2 \le l_2, \quad t_{j-1} < t \le t_j, \tag{9}$$

а уравнение (5) – условиями

$$\upsilon_{(2)}(x_1, 0, t) = \mu(x_1, 0, t), \quad \upsilon_{(2)}(x_1, l_2, t) = \mu(x_1, l_2, t), \quad 0 \le x_1 \le l_1, \quad t_{j-1} < t \le t_j. \tag{10}$$

За приближенное решение задачи (1)–(3) на каждом слое t_j, $j = 1, 2, ..., j_0$, принимается функция $\upsilon_{(2)}(\mathbf{x}, t)$ [1]. Задача (4)–(10) аппроксимирует задачу (1)–(3) в суммарном смысле [1].

Для численного решения задачи (4)–(10) в области G введем сетку узлов $\overline{\omega_h} = \{x_1^{(i_1)} = i_1 h_1,\ i_1 = 0, 1, ..., N_1,\ N_1 h_1 = l_1,\ x_2^{(i_2)} = i_2 h_2,\ i_2 = 0, 1, ..., N_2,\ N_2 h_2 = l_2\}$. Обозначим $y_{(1)}$ и $y_{(2)}$ соответственно численные значения функций $\upsilon_{(1)}$ и $\upsilon_{(2)}$ на сетке $\omega = \omega_\tau \times \overline{\omega_h}$. Введем следующие коэффициенты:

$$a^j_{1, i_1, i_2} = 0,5(k_1(x_{i_1}, x_{i_2}, t_j) + k_1(x_{i_1 - 1}, x_{i_2}, t_j)), \quad i_1 = 1, ..., N_1,\ i_2 = 1, ..., N_2 - 1,\ j = 1, ..., j_0;$$

$$a^j_{2, i_1, i_2} = 0,5(k_2(x_{i_1}, x_{i_2}, t_j) + k_2(x_{i_1}, x_{i_2 - 1}, t_j)), \quad i_1 = 1, ..., N_1 - 1,\ i_2 = 1, ..., N_2,\ j = 1, ..., j_0.$$

Разностная схема для задачи (4), (6), (8), (9) имеет вид [1]

$$\frac{y^j_{(1) i_1, i_2} - y^{j-1}_{(1) i_1, i_2}}{\tau} = \frac{1}{h_1^2}\left(a^j_{1, i_1 + 1, i_2}\left(y^j_{(1) i_1 + 1, i_2} - y^j_{(1) i_1, i_2}\right) - a^j_{1, i_1, i_2}\left(y^j_{(1) i_1, i_2} - y^j_{(1) i_1 - 1, i_2}\right)\right) + f_1(x_{i_1}, x_{i_2}, t_j),$$

$$i_1 = 1, ..., N_1 - 1,\ i_2 = 1, ..., N_2 - 1,\ j = 1, 2, ..., j_0;$$

$$y^{j-1}_{(1) i_1, i_2} = \begin{cases} u_0(x_{i_1}, x_{i_2}), & j = 1, \\ y^{j-1}_{(2) i_1, i_2}, & j = 2, ..., j_0, \end{cases} \qquad i_1 = 1, ..., N_1 - 1,\ i_2 = 1, ..., N_2 - 1; \tag{11}$$

$$y^j_{(1) 0, i_2} = \mu(0, x_{i_2}, t_j), \quad y^j_{(1) N_1, i_2} = \mu(l_1, x_{i_2}, t_j), \quad i_2 = 0, 1, ..., N_2,\ j = 1, 2, ..., j_0.$$

Разностная схема для задачи (5), (7), (10) следующая:

$$\frac{y^j_{(2)i_1,i_2} - y^{j-1}_{(2)i_1,i_2}}{\tau} = \frac{1}{h_2^2}(a^j_{2,i_1,i_2+1}(y^j_{(2)i_1,i_2+1} - y^j_{(2)i_1,i_2}) - a^j_{2,i_1,i_2}(y^j_{(2)i_1,i_2} - y^j_{(2)i_1,i_2-1})) + f_2(x_{i_1}, x_{i_2}, t_j),$$

$$i_1 = 1, ..., N_1 - 1,\ i_2 = 1, ..., N_2 - 1,\ j = 1, 2, ..., j_0; \tag{12}$$

$$y^{j-1}_{(2)i_1,i_2} = y^j_{(1)i_1,i_2},\quad i_1 = 1, ..., N_1 - 1,\ i_2 = 1, ..., N_2 - 1,\ j = 1, 2, ..., j_0;$$

$$y^j_{(2)i_1,0} = \mu(x_{i_1}, 0, t_j),\quad y^j_{(2)i_1,N_2} = \mu(x_{i_1}, l_2, t_j),\quad i_1 = 0, 1, ..., N_1,\ j = 1, 2, ..., j_0.$$

Зафиксировав в (11) j и i_2, получим систему уравнений относительно значений $y^j_{(1)i_1,i_2}$ $(i_1 = 1, ..., N_1 - 1)$, состоящую из $(N_1 - 1)$-го линейного уравнения, которую можно решить методом прогонки [1, с. 40–41]. В целом систему (11) на каждом слое t_j можно представить как $(N_2 - 1)$ независимую задачу (для каждого фиксированного $i_2 = 1, ..., N_2 - 1$), решаемую методом прогонки.

Аналогично решение задачи (12) на каждом слое t_j представляет собой решение $(N_1 - 1)$-й независимой задачи при фиксированном $i_1 = 1, ..., N_1 - 1$. Каждая из указанных задач является системой из $(N_2 - 1)$ линейного уравнения относительно значений $y^j_{(2)i_1,i_2}$ $(i_2 = 1, ..., N_2 - 1)$ и решается методом прогонки.

Величины $y^j_{(2)i_1,i_2}$, $j = 1, 2, ..., j_0$, $i_1 = 0, 1, ..., N_1$, $i_2 = 0, 1, ..., N_2$, – приближенные решения задачи (1)–(3) на слоях t_j, $j = 1, 2, ..., j_0$. Сеточная функция $y^{j_0}_{(2)i_1,i_2}$, $i_1 = 0, 1, ..., N_1$, $i_2 = 0, 1, ..., N_2$, является приближенным решением задачи (1)–(3).

2. Расчетная схема локально-одномерного метода

Обозначим для фиксированных j и i_2:

$$y_{i_1} = y^j_{(1)i_1,i_2};$$

$$A_{i_1} = a^j_{1,i_1,i_2},\quad B_{i_1} = a^j_{1,i_1+1,i_2},\quad C_{i_1} = \frac{h_1^2}{\tau} + a^j_{1,i_1+1,i_2} + a^j_{1,i_1,i_2},\quad F_{i_1} = \frac{h_1^2}{\tau} y^{j-1}_{(1)i_1,i_2} + h_1^2 f_1(x_{i_1}, x_{i_2}, t_j);$$

$$\nu_1 = \mu(0, x_{i_2}, t_j),\quad \nu_2 = \mu(l_1, x_{i_2}, t_j).$$

Тогда, положив $i_1 = i$, $N_1 = N$, систему алгебраических уравнений (11) можно записать в виде разностного трехточечного уравнения

$$A_i y_{i-1} - C_i y_i + B_{i_1} y_{i+1} = -F_i,\qquad i = 1, ..., N - 1, \tag{13}$$

с краевыми условиями

$$y_0 = \nu_1,\quad y_N = \nu_2. \tag{14}$$

Систему уравнений (12) можно также записать в виде (13), (14), если для фиксированных j и i_1 ввести обозначения

$$y_{i_2} = y^j_{(2)i_1,i_2};$$

$$A_{i_2} = a^j_{2,i_1,i_2}, \quad B_{i_2} = a^j_{2,i_1,i_2+1}, \quad C_{i_2} = \frac{h_2^2}{\tau} + a^j_{2,i_1,i_2+1} + a^j_{2,i_1,i_2}, \quad F_{i_2} = \frac{h_2^2}{\tau} y^{j-1}_{(2)i_1,i_2} + h_2^2 f_2(x_{i_1}, x_{i_2}, t_j);$$

$$\nu_1 = \mu(x_{i_1}, 0, t_j), \quad \nu_2 = \mu(x_{i_1}, l_2, t_j)$$

и положить $i_2 = i$, $N_2 = N$.

Для решения уравнений (13), (14) воспользуемся формулами правой прогонки

$$\alpha_1 = 0, \quad \beta_1 = \nu_1, \quad \alpha_{i+1} = \frac{B_i}{C_i - \alpha_i A_i}, \quad \beta_{i+1} = \frac{A_i\beta_i + F_i}{C_i - \alpha_i A_i}, \quad i = 1, 2, ..., N-1;$$

$$y_N = \nu_2, \quad y_{N-i-1} = \alpha_{N-i} y_{N-i} + \beta_{N-i}, \quad i = 0, 1, ..., N-1.$$

Тогда основную часть алгоритма приближенного решения задачи (1)–(3) можно записать в программоподобном виде (циклы, итерации которых заведомо можно выполнять независимо, запишем как *dopar*):

do $j = 1, j_0$

dopar $i_2 = 1, N_2 - 1$

$\alpha(1)=0, \quad \beta(1) = \mu(0, x_{i_2}, t_j), \quad y_{(1)}(N_1, i_2) = \mu(l_1, x_{i_2}, t_j)$

do $i_1 = 1, N_1 - 1$

$$A = a^j_{1,i_1,i_2}, \quad B = a^j_{1,i_1+1,i_2}, \quad C = \frac{h_1^2}{\tau} + a^j_{1,i_1+1,i_2} + a^j_{1,i_1,i_2},$$

$$F = \frac{h_1^2}{\tau} y_{(2)}(i_1, i_2) + h_1^2 f_1(x_{i_1}, x_{i_2}, t_j) \quad (\text{if } j{=}1 \text{ then } y_{(2)}(i_1, i_2) = u_0(i_1, i_2))$$

$$\alpha(i_1 + 1) = \frac{B}{C - \alpha(i_1)A}, \quad \beta(i_1 + 1) = \frac{A\beta(i_1) + F}{C - \alpha(i_1)A}$$

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

enddopar

dopar $i_1 = 1, N_1 - 1$

$\alpha(1) = 0, \quad \beta(1) = \mu(x_{i_1}, 0, t_j), \quad y_{(2)}(i_1, N_2) = \mu(x_{i_1}, l_2, t_j)$

do $i_2 = 1, N_2 - 1$

$$A = a^j_{2,i_1,i_2}, \quad B = a^j_{2,i_1,i_2+1}, \quad C = \frac{h_2^2}{\tau} + a^j_{2,i_1,i_2+1} + a^j_{2,i_1,i_2}, \quad F = \frac{h_2^2}{\tau} y_{(1)}(i_1, i_2) + h_2^2 f_2(x_{i_1}, x_{i_2}, t_j),$$

$$\alpha(i_2 + 1) = \frac{B}{C - \alpha(i_2)A}, \quad \beta(i_2 + 1) = \frac{A\beta(i_2) + F}{C - \alpha(i_2)A}$$

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

enddopar

enddo

Выходные данные: $y_{(2)}(i_1, i_2)$ ($y^{j_0}_{(2)i_1,i_2}$). Заметим, что в представленном алгоритме вычисляются не все граничные значения $y_{(2)}(i_1, i_2)$. При необходимости эти значения можно вычислить для любого фиксированного *j*: $y_{(2)}(i_1, 0) = \mu(x_{i_1}, 0, t_j)$, $y_{(2)}(0, i_2) = \mu(0, x_{i_2}, t_j)$, $y_{(2)}(N_1, i_2) = \mu(l_1, x_{i_2}, t_j)$.

3. Параллельные реализации

Для отображения алгоритмов, задаваемых последовательными программами, на параллельные компьютеры с распределенной памятью требуется распределить операции алгоритма между процессорами, распределить данные (элементы массивов) между процессорами, установить порядок выполнения операций в каждом процессоре, задать зерно вычислений и организовать обмены данными [2, 3]. Под зерном вычислений понимается множество операций алгоритма, выполняемых атомарно: все вычисления, принадлежащие одному зерну, производятся на одном процессоре и не могут прерываться обменом данными. Выбор зерна определяет объем и частоту передачи пакетов данных.

Воспользуемся естественным параллелизмом алгоритма. Пусть P – число процессоров, предназначенных для реализации алгоритма. Каждому процессору при фиксированном параметре *j* самого внутреннего цикла назначим выполнить $B^{(2)} = \left\lceil \frac{N_2 - 1}{P} \right\rceil$ итераций параллельного цикла с параметром i_2 и $B^{(1)} = \left\lceil \frac{N_1 - 1}{P} \right\rceil$ итераций параллельного цикла с параметром i_1. Единый для каждого из P процессоров псевдокод параллельного алгоритма можно записать следующим образом:

Параллельный алгоритм 1

if $(1 \le p \le P)$

do $j = 1, j_0$

do $i_2 = 1 + (p-1)B^{(2)}, \min(pB^{(2)}, N_2 - 1)$

$\alpha(1) = \ldots$, $\beta(1) = \ldots$, $y_{(1)}(N_1, i_2) = \ldots$

do $i_1 = 1, N_1 - 1$

$A = \ldots$, $B = \ldots$, $C = \ldots$, $F = \ldots y_{(2)}(i_1, i_2) \ldots$, $\alpha(i_1 + 1) = \ldots$, $\beta(i_1 + 1) = \ldots$

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

обмен данными (синхронизация)

do $i_1 = 1 + (p-1)B^{(1)}, \min(pB^{(1)}, N_1 - 1)$

$\alpha(1)$=... , $\beta(1) = \ldots$, $y_{(2)}(i_1, N_2) = \ldots$

do $i_2 = 1,\ N_2 - 1$

$A = \ldots$, $B = \ldots$, $C = \ldots$, $F = \ldots y_{(1)}(i_1, i_2) \ldots$, $\alpha(i_2 + 1) = \ldots$, $\beta(i_2 + 1) = \ldots$

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

обмен данными (синхронизация)

enddo

Используются два типа зерна вычислений. Зерно вычислений первого типа составляют операции, выполняемые на $B^{(2)}$ итерациях параллельного цикла уровня вложенности 2 с параметром i_2. Зерно вычислений второго типа составляют операции, выполняемые на $B^{(1)}$ итерациях параллельного цикла уровня вложенности 2 с параметром i_1. Каждый процессор получает данные от всех других процессоров (сборка данных). После выполнения каждого зерна вычислений первого типа p-й процессор должен получить от q-го ($q \neq p$) процессора данные $y_{(1)}(i_1, i_2)$, $1 + (p-1)B^{(1)} \le i_1 \le \min(pB^{(1)}, N_1 - 1)$, $1 + (q-1)B^{(2)} \le i_2 \le \min(qB^{(2)}, N_2 - 1)$. После выполнения каждого зерна вычислений второго типа p-й процессор должен получить от q-го ($q \neq p$) процессора данные $y_{(2)}(i_1, i_2)$, $1 + (q-1)B^{(1)} \le i_1 \le \min(qB^{(1)}, N_1 - 1)$, $1 + (p-1)B^{(2)} \le i_2 \le \min(pB^{(2)}, N_2 - 1)$.

Рассмотрим модификацию параллельного алгоритма 1, при которой можно начинать вычисление операций нового зерна вычислений, не дожидаясь окончания обмена данными между процессорами. Для этого в циклах, в которых вычислялись прогоночные коэффициенты, выделим отдельно операции, зависящие только от входных данных. Эти операции можно вычислять на фоне обмена данными. В этом случае следует хранить больше результатов промежуточных вычислений.

Параллельный алгоритм 2

if $(1 \le p \le P)$

do $j = 1, j_0$

do $i_2 = 1 + (p-1)B^{(2)}, \min(pB^{(2)}, N_2 - 1)$

$\alpha(i_2, 1)$=...

do $i_1 = 1, N_1 - 1$

$A = (i_1, i_2) \ldots$, $B = \ldots$, $C(i_1, i_2) = \ldots$, $\alpha(i_2, i_1 + 1) = \ldots$

enddo

enddo

дождаться конца обмена данными

(при $j = 1$ *обменов нет, требуются только входные данные)*

do $i_2 = 1 + (p-1)B^{(2)}, \min(pB^{(2)}, N_2 - 1)$

$\beta(1) = \ldots$, $y_{(1)}(N_1, i_2) = \ldots$

do $i_1 = 1, N_1 - 1$

$F = \ldots y_{(2)}(i_1, i_2) \ldots$, $\beta(i_1 + 1) = \ldots$

enddo

do $i_1 = 1, N_1 - 1$

$y_{(1)}(N_1 - i_1, i_2) = \alpha(N_1 - i_1 + 1, i_2)\, y_{(1)}(N_1 - i_1 + 1, i_2) + \beta(N_1 - i_1 + 1)$

enddo

enddo

обмен данными (начать, но не дожидаться конца обмена данными)

do $i_1 = 1 + (p-1)B^{(1)}, \min(pB^{(1)}, N_1 - 1)$

$\alpha(i_1, 1) = \ldots$

do $i_2 = 1, N_2 - 1$

$A(i_1, i_2) = \ldots$, $B = \ldots$, $C(i_1, i_2) = \ldots$, $\alpha(i_1, i_2 + 1) = \ldots$

enddo

enddo

дождаться конца обмена данными

do $i_1 = 1 + (p-1)B^{(1)}, \min(pB^{(1)}, N_1 - 1)$

$\beta(1) = \ldots$, $y_{(2)}(i_1, N_2) = \ldots$

do $i_2 = 1, N_2 - 1$

$F = \ldots y_{(1)}(i_1, i_2) \ldots$, $\beta(i_2 + 1) = \ldots$

enddo

do $i_2 = 1, N_2 - 1$

$y_{(2)}(i_1, N_2 - i_2) = \alpha(i_1, N_2 - i_2 + 1)\, y_{(2)}(i_1, N_2 - i_2 + 1) + \beta(N_2 - i_2 + 1)$

enddo

enddo

обмен данными (начать, но не дожидаться конца обмена данными)

enddo

4. Результаты вычислительных экспериментов

Эксперименты, результаты которых представлены в этом разделе, проводились на суперкомпьютере СКИФ (ОИПИ НАН Беларуси) при следующих значениях параметров алгоритмов: $N_1 = N_2 = 1440$, $j_0 = 1440$. Результаты вычислительных экспериментов представлены на рис. 1–3.

Из графиков следует, что алгоритм 1 обладает в целом более качественными показателями, чем алгоритм 2. Это объясняется более значительными накладными расходами на организацию коммуникаций в алгоритме 2 по сравнению с алгоритмом 1. Видно, что оба параллельных алгоритма обладают высокой эффективностью при использовании до 60 процессоров. Под эффективностью в данном случае понимается отношение ускорения, полученного при выполнении алгоритма, к количеству процессоров, на которых алгоритм выполнялся. Из графика на рис. 3 следует, что наибольшая эффективность параллельных реализаций алгоритма при заданных значениях параметров N_1 , N_2 и j_0 наблюдается при P=40±10. Этот факт, как и то, что максимальная эффективность превышает единицу, объясняется более эффективным использованием кешей процессоров при выполнении алгоритма.

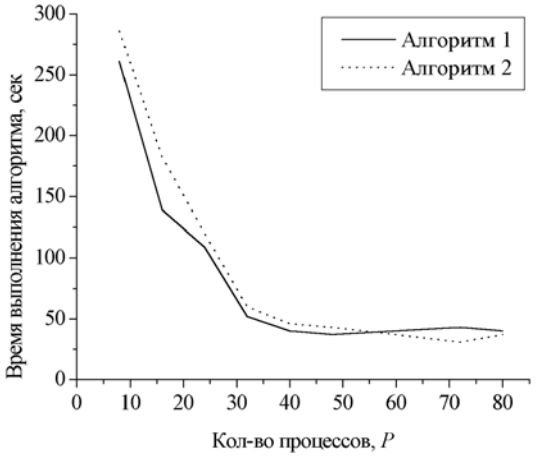

Рис. 1. График зависимости времени выполнения параллельных алгоритмов 1 и 2 от количества процессов

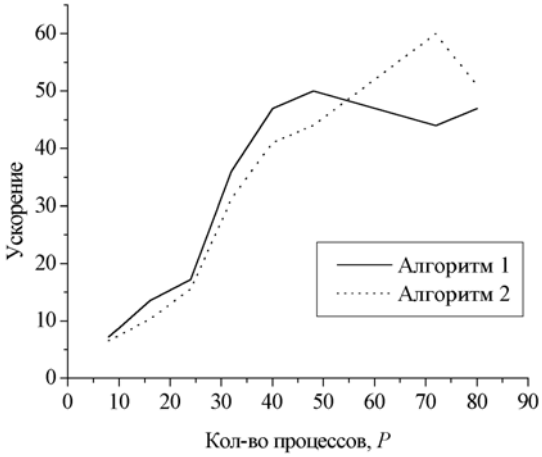

Рис. 2. График зависимости ускорения от количества процессов при выполнении параллельных алгоритмов 1 и 2 (из расчета, что время последовательной реализации алгоритма равно 1880 с)

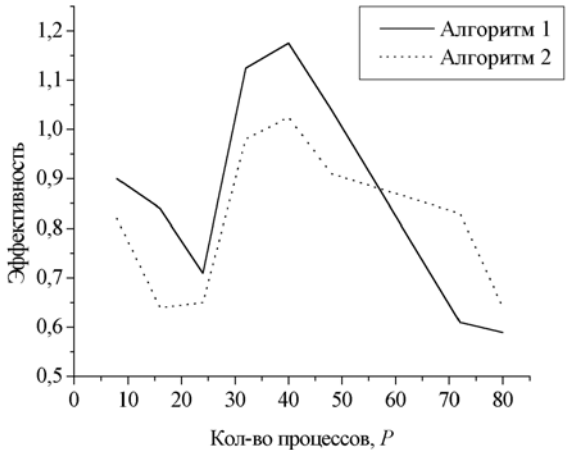

Рис. 3. График эффективности параллельных алгоритмов 1 и 2

Заключение

Разработанные алгоритмы и параллельные программы для реализации локально-одномерного метода численного решения двумерных параболических уравнений на суперкомпьютерах с распределенной памятью обладают высокими показателями эффективности параллельной реализации. В частности, при использовании до 60 процессоров на сетке размера 1440×1440 получена эффективность, близкая к единице, причем в диапазоне от 30 до 50 процессоров эффективность превышает единицу. Получению большей эффективности при увеличении числа используемых для вычислений процессоров препятствует интенсивный обмен данными, происходящий между шагами алгоритма. Перспективным направлением повышения производительности вычислений при параллельной реализации рассмотренного локально-одномерного метода представляется получение параллельного алгоритма, позволяющего локализовать в процессорах суперкомпьютера некоторые промежуточные результаты вычислений.

Список литературы

1. Самарский, А.А. Теория разностных схем / А.А. Самарский. – М. : Наука, 1989. – 616 с.

2. Воеводин, В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин. – СПб. : БХВ-Петербург, 2002. – 600 с.

3. Лиходед, Н.А. Методы распараллеливания гнезд циклов : курс лекций / Н.А. Лиходед. Минск : БГУ, 2008. – 100 с.

Поступила 14.07.10

Институт математики НАН Беларуси,
Минск, Сурганова, 11
e-mail: bsv@im.bas-net.by,
zayats@im.bas-net.by,
likhoded@im.bas-net.by

S.V. Bakhanovich, G.M. Zayats, N.A. Likhoded, V.A. Tsurko

PARALLEL IMPLEMENTATION OF A LOCALLY ONE-DIMENSIONAL METHOD FOR NUMERICAL SOLUTION OF TWO-DIMENSIONAL PARABOLIC EQUATIONS

Algorithms and parallel programs for the implementation of a locally one-dimensional method for numerical solution of two-dimensional parabolic equations on parallel computers with the distributed memory are developed. It is demonstrated that being implemented in a parallel mode, the algorithms are highly efficient compared to the conventional ones.