

УДК 519.633

А. А. Верлань, канд. техн. наук,

Д. С. Смаковский, аспирант,

И. Ю. Михайлова, аспирант

Национальный технический университет Украины «КПИ», г. Киев

## ЧИСЛЕННОЕ РЕШЕНИЕ ДВУМЕРНОГО УРАВНЕНИЯ ТЕПЛОПРОВОДНОСТИ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ OPENMP

Рассмотрены результаты применения технологии OpenMP для распараллеливания численного решения двумерного неоднородного уравнения теплопроводности методом переменных направлений.

**Ключевые слова:** дифференциальные уравнения в частных производных, технология OpenMP, метод переменных направлений.

**Введение.** В связи с ростом требований к продуктивности компьютерных систем, в частности, расчетных комплексов, и появлением процессоров нового поколения, имеющих два и более ядра, стала актуальной проблема как можно более полного использования вычислительных возможностей процессоров. В отличие от одноядерного процессора,  $n$ -ядерный позволяет работать с многопоточными приложениями в параллельном режиме, при этом операционная система распределяет программные потоки по ядрам. Благодаря этому появляется возможность значительно увеличить производительность вычислений. Это, в свою очередь, позволяет сократить время расчета, однако требует иногда значительной переработки алгоритма и кода приложения с целью разделения инструкций на несколько независимых потоков команд, которые исполняются на самостоятельных вычислительных блоках. В данной работе рассматривается технология OpenMP, которая позволяет при помощи директив препроцессора, библиотечных функций и переменных окружения реализовать параллельные вычисления с помощью многопоточности. При этом главный поток создает набор подчиненных потоков и задача распределяется между ними.

**Постановка задачи.** Исследование эффективности параллельного алгоритма решения и его реализации с применением технологии OpenMP проведем на примере численного решения двумерного нестационарного уравнения теплопроводности [1]. Данное уравнение может описывать передачу тепла за счет теплопроводности в плоской пластине с геометрическими размерами  $x_{\max} \times y_{\max}$  :

$$\frac{\partial U(x, y, t)}{\partial t} = \frac{\partial^2 U(x, y, t)}{\partial x^2} + \frac{\partial^2 U(x, y, t)}{\partial y^2} + RU(x, y, t) + F(x, y, t),$$

где  $U(x, y, t)$  — температура пластины в точке  $(x, y)$  в момент времени  $t$ ;  $R(x, y) = -20$  — коэффициент, определяющий конвективный теплообмен;  $F(x, y) = \begin{cases} 2, & x < 0, 2, y < 0, 2 \\ 0 & \end{cases}$  — коэффициент, определяющий тепловой поток, подводимый к части пластины;

$x \in [0, x_{\max}]$ ;  $y \in [0, y_{\max}]$ ;  $t \in [0, t_{\max}]$ .

Краевые условия соответствуют отсутствию теплообмена на торцах пластины в любой момент времени:

$$\left. \frac{\partial U}{\partial x} \right|_{x=0} = \left. \frac{\partial U}{\partial x} \right|_{x=x_{\max}} = \left. \frac{\partial U}{\partial y} \right|_{y=0} = \left. \frac{\partial U}{\partial y} \right|_{y=y_{\max}} = 0.$$

Начальное условие — начальная температура детали в каждом узле сетки на начальном этапе:  $U(x, y, 0) = 0$ .

**Методы решения.** Для решения дифференциальных уравнений в частных производных (ДУЧП) чаще всего используются две группы методов: конечных элементов (МКЭ) и конечных разностей (МКР). Эти методы основаны на дискретизации области определения с достаточно мелким шагом. Одним из отличий этих двух групп методов является конструкция используемых сеток. МКР использует регулярные сетки, в которых особенности геометрии учитываются только на границах области. Благодаря этому МКР чаще всего используется для анализа задач с прямолинейными границами областей определения функций. МКЭ использует разбиение с учетом геометрических особенностей области, начиная от границ. Таким образом, аппроксимация будет более точной. Благодаря этому МКЭ используется для решения задач со сложной геометрией областей определения. Поскольку в данной задаче расчетная область достаточно проста, имеет смысл использовать МКР.

Метод конечных разностей состоит из трех этапов: дискретизация расчетной области, замена ДУЧП системами алгебраических уравнений, решение этих систем.

Поскольку в данной задаче расчетная область достаточно проста и температура меняется плавно, для дискретизации расчетной области, т.е. построения сетки, используется постоянный шаг. Однако, при изменении условий задачи, можно перейти на адаптивную сетку, которая уплотняет узлы в зонах, где функция резко меняется, и разрежает в остальных случаях.

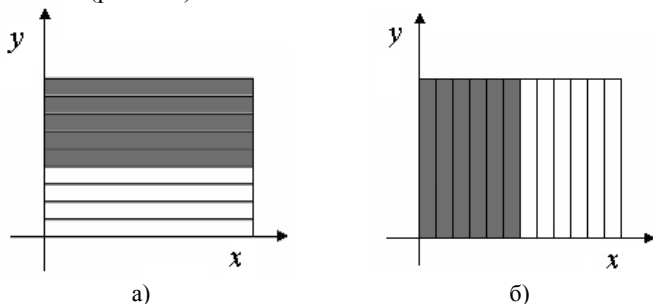
Для перехода от ДУЧП к системам алгебраических уравнений применим схему переменных направлений [2], которая трансформирует двумерную задачу в последовательность одномерных. Для данной задачи используется разностная схема, в которой поочередно используется неявный метод для одного из пространственных направлений и явный метод для другого направления. Схема переменных направлений – абсолютно устойчивая, т.е. допускает применение большего временного шага в сравнении с явной схемой.

Введем на оси  $OX$  сетку с шагом  $h_x$ :  $x_i = i \cdot h_x$ ,  $i = 0, 1, \dots, N_1$ . Аналогично введем сетки по осям  $OY$  и  $Ot$  с шагами  $h_y$  и  $\tau$  соответственно:  $y_j = j \cdot h_y$ ,  $j = 0, 1, \dots, N_2$  и  $t_k = k \cdot \tau$ ,  $k = 0, 1, \dots, N_3$ . Обозначим значение приближенного решения в точке  $(x_i, y_j, t_k)$  через  $u_{ij}^k$ . Каждый шаг схемы переменных направлений, которая имеет второй порядок точности по времени и пространственным переменным, реализуется на равномерной сетке переход с  $k$ -го временного слоя на  $(k+1)$ -й и заключается в выполнении двух этапов.

На первом этапе для одной из переменных неявно аппроксимируется ее производная на  $k + \frac{1}{2}$  слое. Для второй переменной неявная аппроксимация используется на  $k$  слое. Аналогично на втором этапе для второй переменной неявно аппроксимируется ее производная. Таким образом, получаем следующие два уравнения, каждое из которых отвечает неявной схеме по одному из координатных направлений:

$$\begin{aligned} \frac{u_{ij}^{k+\frac{1}{2}} - u_{ij}^k}{\frac{\tau}{2}} &= \frac{u_{i-1,j}^{k+\frac{1}{2}} - 2u_{ij}^{k+\frac{1}{2}} + u_{i+1,j}^{k+\frac{1}{2}}}{h_x^2} + \\ &+ \frac{u_{i,j-1}^k - 2u_{ij}^k + u_{i,j+1}^k}{h_y^2} + Ru_{ij}^{k+\frac{1}{2}} + F^{k+\frac{1}{2}}, \\ \frac{u_{ij}^{k+1} - u_{ij}^{k+\frac{1}{2}}}{\frac{\tau}{2}} &= \frac{u_{i-1,j}^{k+\frac{1}{2}} - 2u_{ij}^{k+\frac{1}{2}} + u_{i+1,j}^{k+\frac{1}{2}}}{h_x^2} + \\ &+ \frac{u_{i,j-1}^{k+1} - 2u_{ij}^{k+1} + u_{i,j+1}^{k+1}}{h_y^2} + Ru^{k+1} + F^{k+\frac{1}{2}}. \end{aligned}$$

На первом этапе осуществляется проход по строкам одного временного слоя (рис. 1 а), на втором этапе — по столбцам одного временного слоя (рис. 1 б).



**Рис. 1.** Этапы решения СЛАУ: а — прогонка по строкам (расчет  $k + \frac{1}{2}$ -го слоя на основе  $k$ -го); б — прогонка по столбцам (расчет  $k+1$ -го слоя на основе  $k + \frac{1}{2}$ )

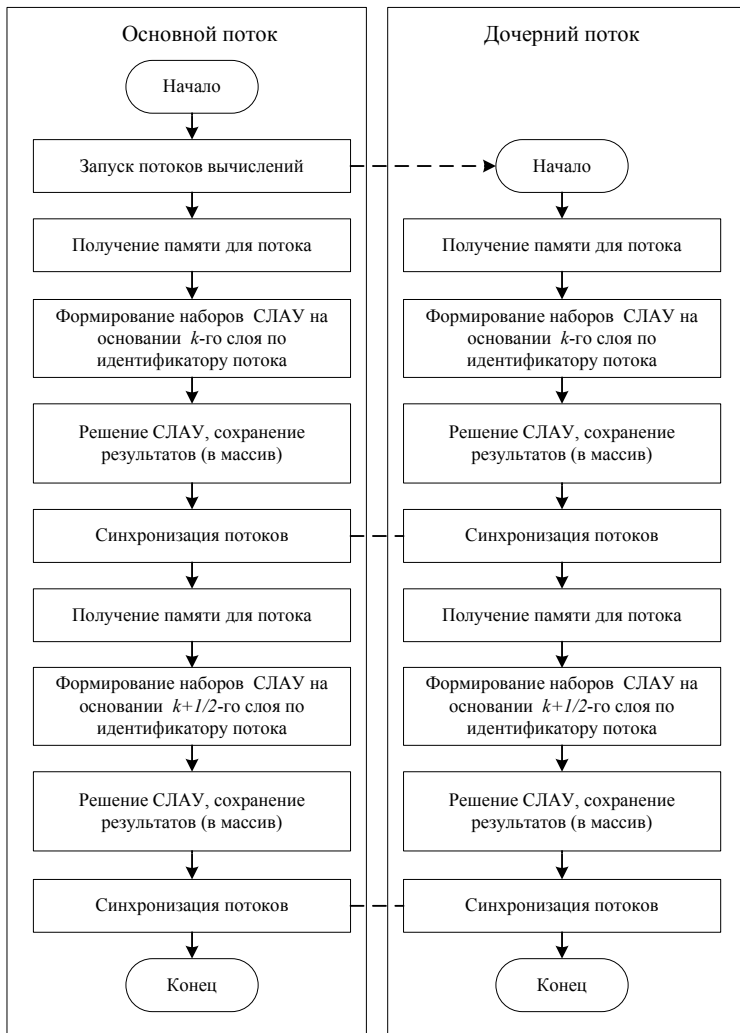
Подставив соответствующие значения в уравнения, получим трехдиагональную матрицу, решив которую находим одну из строк или столбцов матрицы, в которой хранятся соответствующие значения искомой функции  $U(x, y, t)$ .

**Параллельный алгоритм.** Рассмотрим алгоритм параллельной реализации метода переменных направлений. Выполнение перехода на следующий временной слой возможно только после получения результата на предыдущем, т.е. после формирования наборов СЛАУ, их решения и сохранения полученных результатов в матрицу решения. Таким образом, распараллеливание можно осуществлять только при переходе от предыдущего временного слоя к следующему (рис. 2).

Распараллеливание процесса вычислений производится на этапе формирования и решения отдельных СЛАУ из набора, поскольку отдельные СЛАУ не зависят друг от друга. По завершении решения набора СЛАУ для перехода с одного временного слоя на полуцелый и с полуцелого на следующий, необходимо произвести синхронизацию потоков, поскольку расчет следующего слоя зависит от результатов решения предыдущего.

Системы параллельного программирования делятся на 2 группы: системы с общей памятью и системы передачи сообщений. Преимущество систем с общей памятью (таких как OpenMP) состоит в том, что все общие данные можно хранить в общем пространстве памяти и не тратить время на передачу данных между параллельными процессами.

С другой стороны, для обеспечения такой функциональности необходимо осуществлять синхронизацию вычислительных потоков.



*Рис. 2. Схема распараллеливания процесса расчета*

Процедура параллельного решения наборов СЛАУ с использованием библиотеки OpenMP требует создания в каждом потоке массивов для хранения трехдиагональной матрицы СЛАУ, вектора свободных членов и вектора результатов расчета. Выделить память для хранения данных массивов можно несколькими способами:

- объявить приватные для потока переменные, которые будут храниться в стеке вычислительного потока;
- выделить динамическую память в каждом потоке;
- выделить динамическую память в главном потоке для всех потоков, а в дочерних потоках использовать часть выделенной в главном потоке памяти.

С точки зрения выполнения программы первый способ наиболее быстрый, однако, при статическом выделении памяти при компиляции программы количество узлов сетки еще неизвестно, что приводит к избыточности выделенной памяти. Выделение и освобождение динамической памяти при выполнении параллельного блока требует значительных затрат машинного времени. Кроме того, операции с памятью в разных потоках могут блокировать друг друга, поскольку потоки используют общую «кучу» процесса. Учитывая вышесказанное, целесообразно выделить динамическую память заранее для всех параллельных потоков и использовать соответствующую часть памяти в параллельном блоке, в зависимости от идентификатора потока.

**Программное обеспечение** разработано на основе вышеизложенных методов и реализовано на языке C++ в среде программирования Microsoft Visual Studio с использованием стандарта OpenMP 2.0 [3].

Исходные данные для расчета:  $h_x$ ,  $h_y$ ,  $\tau$  — величины шагов по осям  $OX$ ,  $OY$  и  $Ot$  соответственно;  $x_{\max}$ ,  $y_{\max}$ ,  $t_{\max}$  — максимальные значения по осям  $OX$ ,  $OY$  и  $Ot$  соответственно.

В процессе решения дифференциального уравнения значения температурного поля записываются в файл с возможностью последующего построения графика зависимости температуры от пространственных координат (рис. 3).

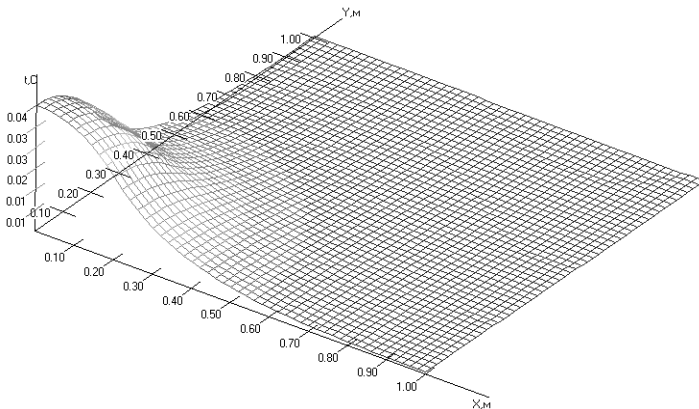


Рис. 3. Температура в момент времени  $t = 1$  с

**Результаты расчета.** Расчет проводился для прямоугольной области размерами  $x \in [0,1]$ ,  $y \in [0,1]$  для времени  $t \in [0,1]$ ,  $h_x = 0,002$ ;  $h_y = 0,002$ ;  $\tau = 0,001$ .

Расчеты проводились на компьютерах с разными конфигурациями (табл. 1). Эффективность параллельного расчета определяется коэффициентами ускорения и эффективности:

$$S_m = T_1 / T_m, \quad E_m = S_m / m,$$

где  $T_m$  — время выполнения  $m$  — поточного приложения,  $T_1$  — однопоточного.

Таблица 1

*Характеристики компьютеров*

№ комп.	Процессор	Тактовая частота, ГГц	Оперативная память, Гб
1	Intel Core Quad	2,67 x4	3
2	Intel Core i3	2,27x2	3
3	Intel Core 2 Duo	1,96x2	2
4	Intel Core 2 Duo	1,86x2	2
5	Intel Pentium 4 (Hyper-threading)	3,2x2	3
6	AMD Athlon 64 X2 Dual Core	2,4x2	1

Сравнение характеристик расчетного процесса (табл. 2) с одинаковыми входными параметрами показывают, что многопоточные вычисления дают возможность сократить время расчетов в 1, 2, 3 раза и повысить эффективность на 60—80% в зависимости от архитектуры.

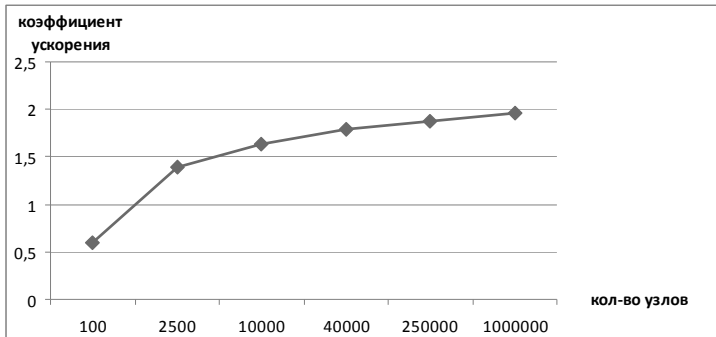
Таблица 2

*Характеристики расчетного процесса*

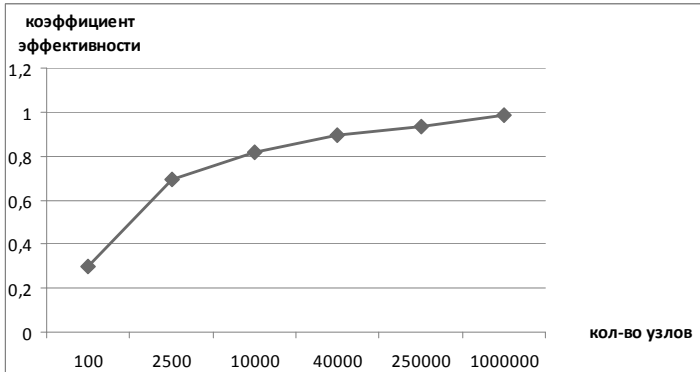
№ комп.	Кол-во потоков	Время расчета, мс	Загрузка процессора, %	Коэффициент ускорения	Коэффициент эффективности
1	1	63781	27	3,14	0,78
	4	20328	97		
2	1	105363	26	3,24	0,81
	4	32495	94		
3	1	100547	52	1,47	0,74
	2	68203	90		
4	1	101140	65	1,34	0,67
	2	75313	100		
5	1	43259	65	1,22	0,61
	2	35553	100		
6	1	34750	50	1,86	0,93
	2	18641	100		

Кроме того, были проведено исследование зависимости эффективности параллельного алгоритма от количества узлов сетки. С уве-

личением количества узлов сетки возрастают коэффициент ускорения (рис. 4) и эффективности (рис. 5). Однако, на сетке с небольшим количеством узлов однопоточный вариант быстрее многопоточного, что объясняется временными затратами на создание потоков. При применении адаптивных сеток необходимо динамически выбирать многопоточный или однопоточный алгоритм решения в зависимости от количества узлов сетки. Следует отметить, что в данном тестовом примере была использована функция  $F(x, y, t)$ , которая не требует значительных затрат на вычисление. При более сложных функциях эффективность параллельного решения будет выше. Расчеты сделаны на компьютере с параметрами: AMD Athlon 64 X2 Dual Core, 1 Гб ОЗУ.



*Рис. 4. Зависимость коэффициента ускорения от общего числа узлов сетки*



*Рис. 5. Зависимость коэффициента эффективности от общего числа узлов сетки*

**Заключение.** Решение двумерного нестационарного неоднородного уравнения теплопроводности показало, что для различных компьютеров с одинаковыми входящими параметрами использование технологии OpenMP для реализации многопоточных параллельных



вычислений позволяет выполнить вычислительный процесс в 1, 2, 3 раза быстрее по сравнению с однопоточным последовательным процессом расчета благодаря почти 100% загрузке всех ядер компьютера. Применение технологии OpenMP целесообразно для решения задач на сетках со значительным числом узлов. Алгоритм решения можно применить для решения трехмерных задач.

### Список использованной литературы:

1. Тихонов А. Н. Уравнения математической физики / А. Н. Тихонов, А. А. Самарский. — М. : Наука, 1977. — 735 с.
2. Самарский А. А. Методы решения сеточных уравнений / А. А. Самарский, Е. С. Николаев. — М. : Наука, 1978. — 590 с.
3. К. С. Гэтлин, П. Айсенс. OpenMP и C++. — Режим доступа: <http://www.microsoft.com/Rus/MSDN/Magazine/2005/10/OpenMP.mspx>.

Results of application of OpenMP technology to parallelize numerical solving of heterogeneous 2D heat equation using alternating directions implicit (ADI) method are reviewed.

**Key words:** partial differential equations, OpenMP technology, alternating directions implicit method.

Отримано 25.09.2010

УДК 519.64

Д. А. Верлань, студент

КНУ ім. Тараса Шевченка, факультет кібернетики, м. Київ

## АЛГОРИТМИ РЕАЛІЗАЦІЇ МЕТОДУ ВИРОДЖЕНИХ ЯДЕР ПРИ РОЗВ'ЯЗАННІ ІНТЕГРАЛЬНИХ РІВНЯНЬ ВОЛЬТЕРРИ

У статті розглянуто чисельний алгоритм розв'язку інтегрального рівняння Вольтерри II роду з ядром довільного вигляду. При цьому довільне ядро попередньо приводиться до виродженого вигляду за допомогою ітераційно-варіаційного методу. Шляхом обчислювальних експериментів оцінено даний метод.

**Ключові слова:** інтегральне рівняння Вольтерри, вироджене ядро, апроксимація, функція двох змінних.

**Вступ.** Інтегральні рівняння Вольтерри II роду

$$y(t) + \int_a^t K(t, \tau) y(\tau) d\tau = f(t), \quad (1)$$

де  $y(t)$  — шукана функція, ядро  $K(t, \tau)$  та  $f(t)$  — відомі функції, отримують все більше розповсюдження в якості динамічних моделей