

Институт информационных и вычислительных технологий МОН РК

Казахский Национальный Университет имени аль-Фараби

Университет Туран

Люблинский технический университет, Польша

«Ғылым ордасы»



МАТЕРИАЛЫ

IV международной научно-практической конференции
"Информатика и прикладная математика",
посвященной 70-летию юбилею профессоров
Биярова Т.Н., Вальдемара Вуйцика
и 60-летию профессора Амиргалиева Е.Н.
25-29 сентябрь 2019, Алматы, Казахстан

Часть 1

Алматы 2019

Кенжебек Е.Г., Иманкулов Т.С., Маткерим Б., Ахмед-Заки Д.Ж.	ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ МЕТОДА ПРОГОНКИ ДЛЯ 2D УРАВНЕНИЯ ТЕПЛОПРОВОДНОСТИ	261
Кудайкулов А., Ташев А.А., Аршидинова М.Т., Бегалиева К.Б.	ЧИСЛЕННОЕ ИССЛЕДОВАНИЕ ТЕРМОМЕХАНИЧЕСКОГО СОСТОЯНИЯ СТЕРЖНЯ ПЕРЕМЕННОГО СЕЧЕНИЯ ПРИ НАЛИЧИИ ЛОКАЛЬНЫХ ИСТОЧНИКОВ ТЕПЛА.	271
Мансурова М.Е., Мукимбеков М.Ж., Накибаева М.Т.	МОДЕЛИРОВАНИЕ ТРЕХМЕРНОЙ ЗАДАЧИ В РАЗРАБОТКЕ НЕФТЯНЫХ МАССИВОВ	279
Набиева Г.С., Калижанова А.У., Кожасбаева А.	МЕТОДЫ РЕШЕНИЯ ПРИКЛАДНЫХ ЗАДАЧ ДИСКРЕТНОГО ПРОГРАММИРОВАНИЯ	285
Нуртазин А.Т.	ЭКЗИСТЕНЦИАЛЬНО ЗАМКНУТЫЕ ПРИМАРНЫЕ ГРУППЫ	294
Пирожников А.Г., Дюсенбаев Д.С.	НЕКОТОРЫЕ СЛЕДСТВИЯ ТЕОРЕМЫ ВИЛЬСОНА	301
Родионов А.С.	К РАСЧЁТУ ВЕРОЯТНОСТИ СВЯЗНОСТИ И ПОЛУЧЕНИЮ ПОЛИНОМА НАДЁЖНОСТИ СЛУЧАЙНОГО ГРАФА. НОВЫЕ ВОЗМОЖНОСТИ	304
Сансызбай Л.Ж.	СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ РЕАЛИЗАЦИИ СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ	314
Темирбекова Л.Н.	РЕГУЛЯРИЗАЦИОННЫЙ АЛГОРИТМ ЧИСЛЕННОГО РЕШЕНИЯ УРАВНЕНИЯ ГЕЛЬФАНДА-ЛЕВИТАНА-КРЕЙНА	321
Тулешов А.К., Кошекков К.Т., Шадыманова А.А.	АВТОМАТИЗИРОВАННЫЙ МОНИТОРИНГ И УПРАВЛЕНИЕ ЗЕРНОХРАНИЛИЩАМИ С ГОРИЗОНТАЛЬНЫМИ СИЛОСАМИ	322
Хисамиев З.Г.	КОМПАЬОНЫ АБЕЛЕВОЙ ГРУППЫ ЦЕЛЫХ ЧИСЕЛ И ЭКЗИСТЕНЦИАЛЬНО ЗАМКНУТЫЕ АБЕЛЕВЫ ГРУППЫ БЕЗ КРУЧЕНИЯ	332

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ МЕТОДА ПРОГОНКИ ДЛЯ 2D УРАВНЕНИЯ ТЕПЛОПРОВОДНОСТИ

¹Кенжебек Е.Г., ²Иманкулов Т.С., ³Маткерим Б., ⁴Ахмед-Заки Д.Ж.

e-mail: kenzhebekyerzhan@gmail.com

^{1,2,3}Казахский Национальный Университет имени аль-Фараби, Казахстан

⁴Университет Международного Бизнеса

***Аннотация.** В данной работе рассмотрена параллельная реализация метода прогонки для 2D уравнения теплопроводности с использованием технологий MPI. В работе приведены описания распараллеливания данной задачи с помощью метода Яненко при использовании 1D и 2D декомпозиции данных. Полученные результаты тестирования представлены и проанализированы, на основании чего описаны особенности использованных декомпозиций.*

***Ключевые слова:** высокопроизводительные вычисления, метод прогонки, метод Яненко, параллельные вычисления, MPI.*

Введение

В настоящее время все более широкое применение в различных отраслях науки находит моделирование процессов с помощью численного решения дифференциальных уравнений. Так как, к решению таких уравнений способствует развитие вычислительной техники и численных методов. Наиболее распространенные методы сводят дифференциальную задачу к системе линейных алгебраических уравнений (СЛАУ). К методам, решающим такие системы относят различные варианты прогонки.

Метод прогонки является прямым методом и привлекает своей простотой реализации в последовательном решении. Появление и развитие вычислительных систем, использующих многоядерные процессоры и графические ускорители, актуализирует задачу распараллеливания прогонки.

В данной работе приводятся описания численного решения уравнения теплопроводности и распараллеливания данной задачи для 1D и 2D декомпозиции с помощью метода Яненко, на втором этапе которого был использован метод правой прогонки. Так же приводятся особенности их реализации для работы на вычислительной системе.

Обзор литературы

В настоящее время многие задачи, описывающие физические процессы сводятся к необходимости численного решения систем линейных алгебраических уравнений (СЛАУ). Имеются многие работы о вопросах разрешимости и сходимости разностных схем, среди которой можно отметить работу [1]. Для нахождения решения таких СЛАУ используются такие методы, как метод прогонки и метод циклической редукции. Метод циклической редукции является более сложным в реализации, но на него меньше влияние, по сравнению с прогонкой, оказывают погрешности округления [2].

В настоящее время есть множество статей на данную тему. Среди них можно выделить следующие работы: в [3,4] предложено сочетание параллельной циклической редукции [5] и правой прогонки, чтобы обеспечить параллелизм и вычислительную сложность. В [6] работе предложен вычислительный решатель на основе алгоритма SPIKE [7]. В [8] работе сформулированы и проанализированы параллельные алгоритмы для решения СЛАУ методом встречных прогонок. Так же, известен метод, предложенный Н.Н. Яненко [9], который позволяет редуцировать исходную систему с большим числом неизвестных к системе с числом неизвестных, равным числу процессоров. Такая система, состоящая из параметрических гранично-процессорных точек, решается методом прогонки. Так же, можно применить такие методы, как метод встречной прогонки, метод параллельно-циклической редукции и параллельно конвейерный метод [10].

В работах [11-14] показано применение метода Яненко при реализации на вычислительной системе с графическими ускорителями. В работе [15] рассмотрено двухуровневое распараллеливание прогонки (на общей памяти средствами OpenMP и на распределенной памяти средствами MPI) для решения трехдиагональных систем, возникающих при моделировании двумерных и трехмерных физических процессов. А также, в данной работе подробно описан метод Яненко, параллельно-конвейерный метод и различные методы для второго этапа метода Яненко. В работе [16,17] показано сравнение параллельно-конвейерного метода и метода Яненко, на втором этапе которого использовалось метод правой прогонки, так же приведены результаты эффективности распараллеливания.

Параллельный алгоритм метода прогонки

Идея метода прогонки предложенным Яненко Н.Н. получила название параметрической прогонки. Метод параметрической прогонки Яненко реализуется с помощью распределения системы сеточных уравнений, где на границе процессорных элементов выделяются так называемые параметрические неизвестные.

Рассмотрим систему линейных уравнений следующего вида:

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i, \quad i = 1, \dots, n-1, \quad (1)$$

$$b_0 x_0 + c_0 x_1 = d_0, \quad a_n x_{n-1} + b_n x_n = d_n.$$

Пусть на каждом процессоре будет одинаковое количество точек $m = N/size$, где N – число неизвестных и $size$ – число процессоров. Таким образом, на процессоре с номером j будет находиться лишь часть уравнений системы (1) с номерами от $(j-1) * m + 1$ до $j * m$, где j – номер процессора. Обозначим x_{j*m} через z_j и будем искать решение системы (1) в следующем виде:

$$x_{(j-1)*m+i} = u_i z_{j-1} + v_i z_j + w_i, \quad i = 1, \dots, m-1, \quad j = 1, \dots, size \quad (2)$$

где u, v, w – решения следующих систем:

$$\begin{aligned}
 a_i u_{i-1} + b_i u_i + c_i u_{i+1} &= 0, \quad u_{(j-1)*m} = 1, \quad u_{j*m} = 0; & j = 1, \dots, size, \\
 a_i v_{i-1} + b_i v_i + c_i v_{i+1} &= 0, \quad v_{(j-1)*m} = 0, \quad v_{j*m} = 1; \\
 a_i w_{i-1} + b_i w_i + c_i w_{i+1} &= d_i, \quad w_{(j-1)*m} = 0, \quad w_{j*m} = 0;
 \end{aligned} \tag{3}$$

$$i = (j - 1) * m + 1, \dots, j * m - 1.$$

Решения этих трех систем можно найти методом прогонки, причем независимо на каждом процессоре. Будем называть этот этап решения задачи – этап нахождения предрешений.

В уравнения с номерами $j * m$ из системы (1) подставляем вместо x комбинации (2). Таким образом, получаем систему трехточечных уравнений для нахождения z_j имеющую следующий вид:

$$A_j z_{j-1} + B_j z_j + C_j z_{j+1} = D_j, \quad j = 1, \dots, size - 1 \tag{4}$$

с коэффициентами:

$$B_0 = b_0 + c_0 u_1, \quad C_0 = c_0 v_1, \quad D_0 = d_0 - c_0 w_1$$

$$A_j = a_{j*m} u_{j*m-1}, \quad B_j = a_{j*m} v_{j*m-1} + b_{j*m} + c_{j*m} u_{j*m+1}$$

$$C_j = c_{j*m} v_{j*m+1}, \quad D_j = d_{j*m} - a_{j*m} w_{j*m-1} - c_{j*m} w_{j*m+1} \quad j = 1, \dots, size - 1,$$

$$A_{size} = a_{size*m} u_{size*m-1}, \quad B_{size} = b_{size*m} + a_{size*m} v_{size*m-1},$$

$$D_{size} = d_{size*m} - a_{size*m} w_{size*m-1}.$$

Будем называть этот этап этапом нахождения гранично-процессорных решений. Размерность этой системы уравнений равна количеству процессоров [15].

После нахождения гранично-процессорных решений восстанавливаем окончательное решение по формуле (2).

Метод Яненко содержит три этапа: 1) нахождения предрешений, 2) нахождения гранично-процессорных решений, 3) восстановление решения.

Данный вычислительный алгоритм обладает высокой степенью параллелизма, так как первый и третий этапы выполняются параллельно на каждом процессоре, но решение системы для параметрических неизвестных (4) выполняется последовательно только одним процессором и требует коммуникаций между MPI-процессами.

В данной работе для решения второго этапа метода Яненко, состоящая из гранично-процессорных точек, был использован метод правой прогонки.

Формулы для метода правой прогонки показано следующим образом:

$$\alpha_0 = -\frac{c_0}{b_0}, \quad \beta_0 = \frac{d_0}{b_0},$$

$$\alpha_i = -\frac{c_i}{b_i + \alpha_i \alpha_{i-1}}, \quad \beta_i = -\frac{d_i - \alpha_i \beta_{i-1}}{b_i + \alpha_i \alpha_{i-1}}, \quad i = 1, \dots, size - 1,$$

$$x_{size} = \frac{d_{size} - \alpha_{size} \beta_{size-1}}{b_{size} + \alpha_{size} \alpha_{size-1}}, \quad x_i = \alpha_i x_{i+1} + \beta_i, \quad i = size - 1, \dots, 0.$$

Реализация распараллеливания метода прогонки

Реализация на распределенной памяти была выполнена с использованием стандарта MPI. Как мы знаем, при решении двумерных задач, использующий метод продольно-поперечной прогонки возникает множество систем трехточечных уравнений. Для нахождения решений таких систем применяется метод Яненко, на втором этапе которого был использован метод правой прогонки. Данная задача была выполнена с одномерной и двумерной декомпозицией.

При использовании одномерной декомпозиции сеточная область разделяется на горизонтальные полосы (Рис. 1).

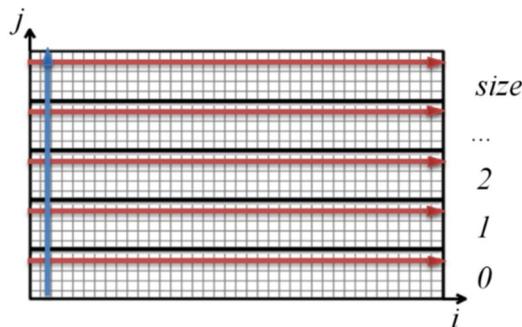


Рис. 1 Одномерная декомпозиция

Реализация прогонки состоит из двух дробных шагов. На Рисунке 1 первый дробный шаг обозначен красным цветом, а второй дробный шаг обозначен синим цветом. На первом дробном шаге, прогонка выполняется для каждой строки независимо на каждом MPI-процессе. Поэтому на первом дробном шаге пересылка между MPI-процессами при использовании метода прогонки не требуется. Однако на втором дробном шаге, где необходимо трехточечные уравнения решать по сеточным линиям $i = const$, мы используем метод Яненко.

Пример декомпозиции вдоль каждого направления прогонки показано на Рис. 2. Гранично-процессорные параметрические неизвестные, которые определяются на втором этапе метода Яненко обозначены зеленым цветом [18].

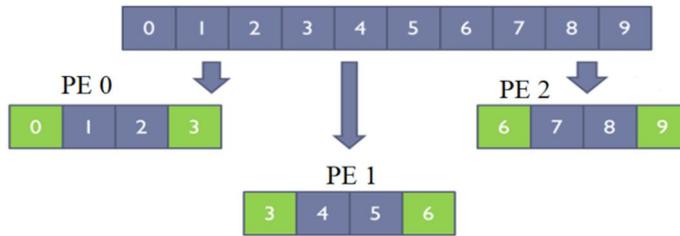


Рис. 2 Пример декомпозиции между тремя MPI-процессами

Второй этап метода Яненко требует дополнительных MPI-коммуникации. Для выполнения второго этапа, из каждого MPI-процесса с помощью функции `MPI_Gather` было собрано граничные элементы предрешений u, v, w . После последовательного нахождения гранично-процессорных решений на одном процессе с помощью метода правой прогонки, была выполнена широковежательная рассылка данных с помощью функции `MPI_Bcast`. Затем, каждый процесс имея свои параметрические граничные элементы и элементы предрешений u, v, w восстанавливает окончательное решение.

В вычислительной системе с распределенной памятью при использовании двумерной декомпозиции, сеточная область разделяется на блоки (Рис. 3). Для этого мы построили двумерную декартовую топологию.

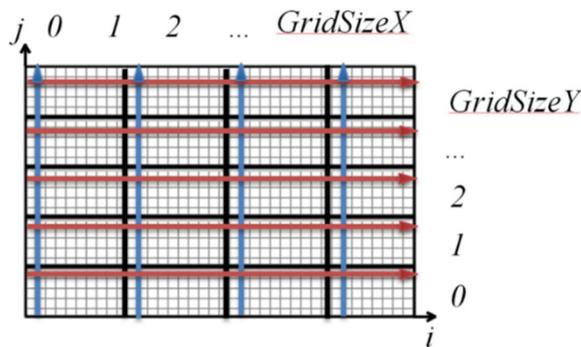


Рис. 3 Двумерная декомпозиция

На Рисунке 3 первый дробный шаг обозначен красным цветом, а второй дробный шаг обозначен синим цветом. На первом и втором дробном шаге, где необходимо трехточечные уравнения решать по сеточным линиям i и j , мы используем метод Яненко. То есть, вдоль каждого направления прогонки была вызвана функция метода Яненко. При использовании двумерной декомпозиции $GridSize$ – число MPI-процессов вдоль направления прогонки.

Численные эксперименты

В качестве тестовой задачи было взято двумерное уравнение теплопроводности следующего вида:

$$\frac{\partial U}{\partial t} = \alpha \left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + f \right)$$

Функция следующего вида:

$$f = -3$$

Начальные условия:

$$U(x, y, 0) = x^2 + y^2$$

Граничные условия:

$$\begin{aligned} U(0, y, t) &= y^2 + t \\ U(1, y, t) &= 1 + y^2 + t \\ U(x, 0, t) &= x^2 + t \\ U(x, 1, t) &= 1 + x^2 + t \end{aligned}$$

Результаты и обсуждения

Тестирование проведено на двумерной задаче теплопроводности с неявной схемой аппроксимации, где в каждом направлении сетки было взято 512, 1024, 2048, 4096 точек.

Тестирование проводилось на вычислительном кластере Новосибирского Государственного Университета (НГУ) [19]. В каждом вычислительном узле находилось 2 Intel Xeon CPU E5-2603 (4 вычислительных ядер в каждом).

При программной конфигурации был использован компилятор Intel C++ Compiler 15.0.2 (уровень оптимизации -O3). Следующие таблицы показывают усредненные значения времени, основанные на нескольких измерениях.

Таблица 1. Время выполнения параллельной программы для 1D декомпозиции

Размер задачи	Количество процессов				
	1	2	4	8	16
512×512	1,26	1,87	1,51	1,75	3,69
1024×1024	6,57	6,33	4,43	4,61	8,04
2048×2048	30,01	22,8	13,63	11,53	15,68
4096×4096	125,01	93,32	54,89	42,92	39,88

Таблица 2. Время выполнения параллельной программы для 2D декомпозиции

Размер задачи	Количество процессов					
	1	4	8	16	32	64
512×512	1,26	1,09	0,78	0,61	0,56	1,12
1024×1024	6,57	3,99	2,64	1,61	1,36	2,27
2048×2048	30,01	16,2	10,26	5,83	3,87	4,93
4096×4096	125,0	65,7	40,9	20,8	12,87	10,7

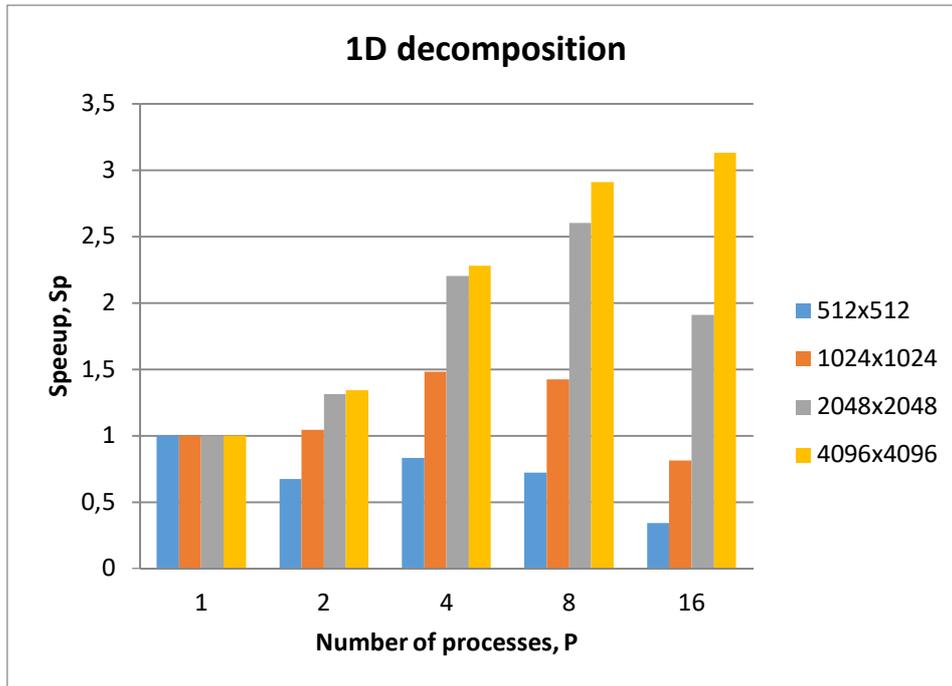


Рис. 4 Ускорение параллельной программы для 1D декомпозиции

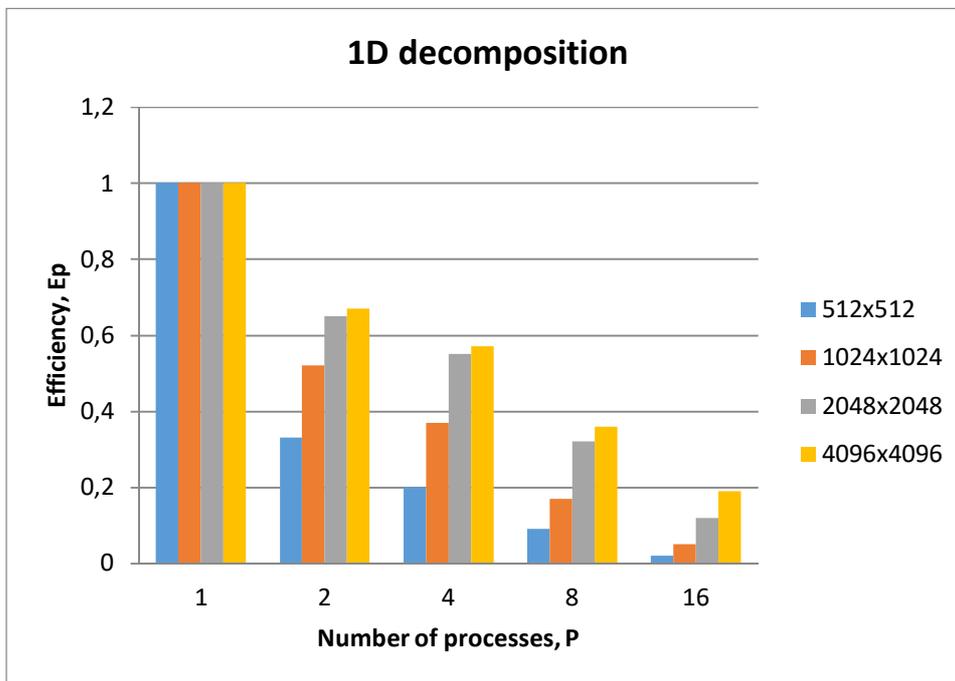


Рис. 5 Эффективность параллельной программы для 1D декомпозиции

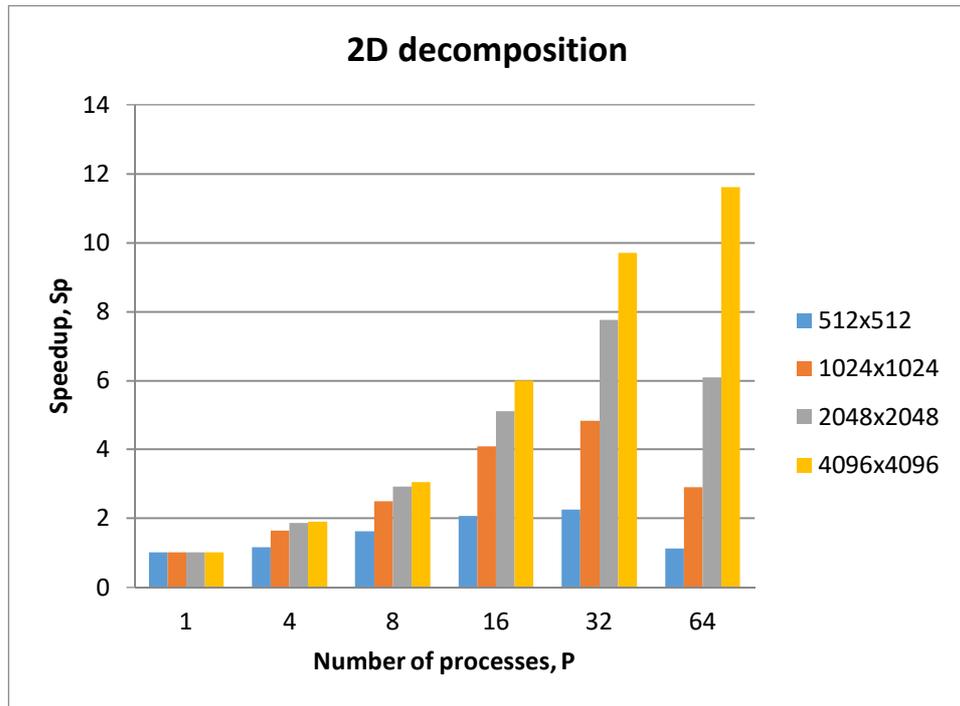


Рис. 6 Ускорение параллельной программы для 2D декомпозиции

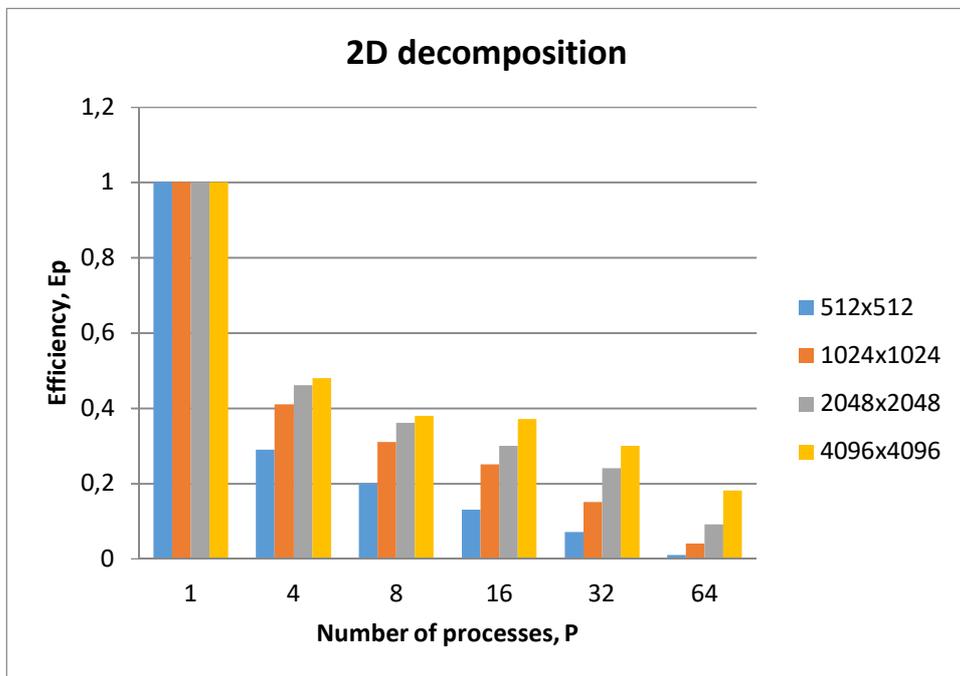


Рис. 7 Эффективность параллельной программы для 2D декомпозиции

На рисунках 4 и 6 представлены ускорение параллельных программ для 1D и 2D декомпозиции при разных количествах MPI-процессов. На этих рисунках можно

заметить, что на каждом процессе при увеличении количество точек задачи ускорение растет. Однако для 1D декомпозиции при использовании 16 MPI-процессов ускорение падает. Это объясняется тем, что при увеличении количество процессов необходимые коммуникационные затраты растут за счет второго этапа метода Яненко. Поэтому, для одномерной декомпозиции максимум используемых процессов было равно 16. А для 2D декомпозиции ускорение увеличивается до 64 MPI-процессов. Так как, при применении двумерной декомпозиции уменьшается число процессов вдоль каждого направления прогонки.

На рисунках 5 и 7 представлены эффективность параллельных программ для 1D и 2D декомпозиции при разных количествах MPI-процессов. Можно заметить, что при увеличении количество процессов эффективность снижается. Для одномерной декомпозиции хорошие показатели эффективности показаны при использовании 2 и 4 MPI-процессов. Для двумерной декомпозиции при увеличении числа MPI-процессов показатели эффективности более стабильны чем при использовании одномерной декомпозиции. Поэтому, при использовании двумерной декомпозиции задача была запущена до 64 процессов, что дает преимущество для оптимизации времени. Двумерная декомпозиция данных при больших сетках является более эффективным по сравнению с одномерной декомпозицией, так как при увеличении числа MPI-процессов в 1D декомпозиции, время выполнения работы второго этапа метода Яненко увеличивается за счет коммуникации между процессами. Например, если запустить задачу при использовании 16 MPI-процессов, на втором дробном шаге, при одномерной декомпозиции на втором этапе метода Яненко коммуникация выполняется между 16 MPI-процессами. А при использовании двумерной декомпозиции коммуникация происходит между 4 MPI-процессами вдоль каждого направления прогонки.

Заключение

Данная работа посвящена параллельной реализации метода прогонки для решения 2D уравнения теплопроводности. Численное решение двумерной задачи теплопроводности был решен с помощью метода продольно-поперечной прогонки. Реализован метод распараллеливание прогонки для решения уравнения теплопроводности с помощью метода Яненко при использовании 1D и 2D декомпозиции данных. В частности, при 2D декомпозиции, вдоль каждого направления прогонки был использован метод Яненко. Было выявлено, что параллельная программа для 1D декомпозиции является эффективным при использовании небольших MPI-процессов. А параллельная программа для 2D декомпозиции является более эффективным при использовании больших вычислительных систем для больших размеров задач.

В дальнейшем планируется:

- реализация для трехмерной задачи с 3D декомпозицией;
- распараллеливание второго этапа метода Яненко;
- применение алгоритма для решения задач нефтедобычи.

Литература

- [1] Самарский А.А., Николаев Е.С. Методы решения сеточных уравнений – М.: Наука, 1987. – С.130.
- [2] Самарский А.А. Введение численные методы. – СПб.: Лань, 2005.
- [3] Davidson A., Zhang Y., D. Owens J. An auto-tuned method for solving large tridiagonal systems on the GPU. Conference Parallel & Distributed Processing Symposium (IPDPS), 2011 IEEE International, 956-965
- [4] Kim H.-S., Wu S., Chang L.-W., Hwu W.-M. A scalable tridiagonal solver for gpus. In Parallel Processing (ICPP), 2011 International Conference on. P. 444–453.
- [5] Hockney R. W., Jesshope C. R. Parallel computers: architecture, programming and algorithm. Hilger. Bristol. 1986. P. 274-280.
- [6] Chang L.-W., Stratton J. A., Kim H.-S., Hwu W.-M. A scalable, numerically stable, highperformance tridiagonal solver using GPUs. High Performance Computing, Networking, Storage and Analysis (SC), 2012 International Conference for. IEEE Computer Society Press, 2012. 11 p.
- [7] Polizzi E., Sameh A. A parallel hybrid banded system solver: The SPIKE algorithm // Parallel Computing. 2006. Vol. 32, No. 2. P. 177–194.
- [8] Головашкин Д.Л. Применение метода встречных прогонок для синтеза параллельного алгоритма решения сеточных уравнений трехдиагонального вида. Журнал Компьютерная оптика, 2002
- [9] Яненко Н.Н., Коновалов А.Н., Бугров А.Н., Шустов Г.В. Об организации параллельных вычислений и “распараллеливании” прогонки // Численные методы механики сплошной среды. 1978. Т. 9. №7. С. 139-146.
- [10] Сапронов И.С., Быков А.Н. Параллельно-конвейерный алгоритм // Атом. 2009. № 44. С. 24- 25
- [11] Быков А. Н., Ерофеев А. М., Сизов Е. А., Федоров А. А. Метод распараллеливания прогонки на гибридных ЭВМ // Вычислительные методы и программирование. 2013. Т. 14. Раздел 2. С. 43-47.
- [12] Jeffers J., Reinders J. Итеуд Xeon Phi Coprocessor High-Performance Programming. Morgan Kaufmann Publishers Inc. San Francisco. 2013. 409 p.
- [13] Goddeke D., Strzodka R. Cyclic reduction tridiagonal solvers on GPUs applied to mixed-precision multigrid // IEEE Transactions on Parallel and Distributed Systems. 2011. Vol. 22. P. 22–32.
- [14] Старченко А.В., Берцун В.Н. Методы параллельных вычислений – Томск: Изд-во Том. ун-та, 2013. С. 223.
- [15] Федоров А. А., Быков А.Н. Метод двухуровневого распараллеливания прогонки для решения трехдиагональных линейных систем на гибридных ЭВМ с многоядерными сопроцессорами// Вычислительные методы и программирование. 2016. Т. 17. С. 234-244.
- [16] Ильин С.А. Старченко А.В. Распараллеливание схемы покомпонентного расщепления для численного решения уравнения теплопроводности // Параллельные вычислительные технологии (ПаВТ’ 2015): Труды международной научной конференции (Екатеринбург, 31 марта – 2 апреля 2015 г.). Челябинск: Издательский центр ЮУрГУ, 2015. С. 399-402.

[17] Быков А.Н., Веселов В.А., Воронин Б.Л., Ерофеев А.М. Методика РАМЗЕС-КП для расчета пространственных движений многокомпонентных теплопроводных сред в эйлерово-лагранжевых координатах. Вып. 13. Саров, 2008. С. 50–57.

[18] Кенжебек Е.Г. Алгоритм Яненко для одномерного уравнения теплопроводности. Алматы: Вестник КазНИТУ, 2019. С. 512-516.

[19] Информационно-вычислительный центр Новосибирского государственного университета. – <http://nusc.nsu.ru/wiki/doku.php> (10.08.19).

ИССЛЕДОВАНИЕ МЕТОДА И ВЫЧИСЛИТЕЛЬНОГО АЛГОРИТМА УЧЕТА ЛОКАЛЬНЫХ ПОВЕРХНОСТНЫХ ТЕПЛОБМЕНОВ В СТЕРЖНЯХ ОГРАНИЧЕННОЙ ДЛИНЫ ПЕРЕМЕННОГО СЕЧЕНИЯ

Кудайкулов А.К.¹, Ташев А.А.¹, Аршидинова М.Т.², Бегалиева К.Б.²
e-mail: mukaddas_arshidi@mail.ru, kudaykulov2006@mail.ru, azattash@mail.ru

¹Институт информационных и вычислительных технологий КН МОН

²Казахский Национальный Университет имени Аль-Фараби

Аннотация

Рассматривается установившееся термомеханическое состояние стержня ограниченной длины переменного сечения при одновременном наличии локальных поверхностных теплообменов, теплоизоляции и тепловых потоков. При исследовании стержня используются фундаментальные законы сохранения энергии и квадратичные сплайн функции в местной системе координат.

Ключевые слова

Удлинение, осевая сила, сечения, температура, перемещение, деформация, напряжение.

Введение

Рассматривается горизонтальный стержень ограниченной длины L [см] и переменного сечения $F(x)$ [см²]. Радиус сечения стержня меняется линейно по его длине, т.е. $r = ax + b$ [см], ($0 \leq x \leq L$), где a и b постоянные. Площадь поперечного сечения стержня меняется нелинейно по длине стержня следующим образом $F(x) = \pi r^2 = \pi(a^2 x^2 + 2abx + b^2)$ [см²], ($0 \leq x \leq L$).

Физико-механические свойства материала стержня характеризуется коэффициентами теплопроводности K_{xx} [$\frac{watt}{cm \cdot K}$], и теплового расширения α [$\frac{1}{K}$].

Модулем упругости материала стержня E [$\frac{кГ}{см^2}$]. Предположим, что на площадь поперечного сечения левого конца стержня подведен тепловой поток постоянной