МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

РАБОТА ПРОВЕРЕНА
Рецензент
к.т.н. доцент кафедры ПМиП
_________ Т.Ю. Оленчикова
«___»___________ 2018 г.

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой,
д.ф.-м.н., профессор
__________Л.Б. Соколинский
«___»___________ 2018 г.

ПРИМЕНЕНИЕ МЕТОДА ПРОЕКТИРОВАНИЯ Q-ЭФФЕКТИВНОЙ ПРОГРАММЫ ДЛЯ РЕШЕНИЯ СИСТЕМЫ СЕТОЧНЫХ УРАВНЕНИЙ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.03.02.2018.115-026.ВКР

Научный руководитель,
к.ф.-м.н., доцент кафедры СП
__________В.Н. Алеева

Автор работы,
студент группы КЭ-402
__________Л.А. Баженова

Ученый секретарь
(нормоконтроллер)
________________О.Н. Иванова
«___»___________ 2018 г.

Челябинск–2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Актуальность

Параллельные вычисления – способ организации компьютерных вычислений, при котором программы разрабатываются, как набор взаимодействующих вычислительных процессов, работающих асинхронно и при этом одновременно. Параллельное программирование необходимо для того, чтобы максимально эффективно использовать имеющиеся ресурсы вычислительных систем. Именно поэтому важны разработки в сфере поиска единого подхода к решению задачи параллельного программирования. Концепция Q-детерминанта может решить проблему повышения эффективности выполнения программ на параллельных вычислительных системах.

Цель и задачи исследования

Целью данной работы является применение метода проектирования Q-эффективной программы для решения системы сеточных уравнений.

Для достижения поставленной цели было необходимо решить следующие задачи:

1) изучение подхода к распараллеливанию алгоритмов, основанного на представлении алгоритмов в форме Q-детерминанта;

2) изучение метода проектирования параллельных программ на основе концепции Q-детерминанта;

3) изучение метода прогонки для решения системы линейных трехточечных уравнений;

4) изучение метода Фурье для решения краевой задачи;

5) представление метода прогонки для системы линейных трехточечных уравнений и метода Фурье в форме Q-детерминанта;

6) изучение технологии OpenMP;

7) разработка Q-эффективной программы для решения системы линейных трехточечных уравнений методом прогонки;

8) разработка Q-эффективной программы для решения краевой задачи методом Фурье;

9) анализ динамических характеристик разработанных Q-эффективных программ.

Структура и объем работы

Выпускная квалификационная работа состоит из введения, четырех разделов, заключения, библиографии. Объем работы составляет 30 страниц, объем библиографии – 16 наименований.

Содержание работы

Первый раздел «Анализ предметной области» описывает теоретические сведения о концепции Q-детерминанта и способах проектирования параллельных программ.

Второй раздел «Проектирование» содержит описания исследуемых алгоритмов и диаграммы вариантов использования.

Третий раздел «Реализация» содержит последовательные и Q-эффективные программы метода прогонки системы линейных трехточечных уравнений и метода Фурье.

Четвертый раздел «Вычислительные эксперименты» включает результаты тестирования и динамические характеристики разработанных программ.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Концепция Q-детерминанта

Пусть α – это алгоритм для решения алгоритмической проблемы $\bar{y} = F(N, B)$, где $N = \{n_1, ..., n_k\}$ – множество параметров размерности проблемы или пустое множество, B – множество входных данных, $\bar{y} = (y_1, ..., y_m)$ – множество выходных данных. $\bar{N}$ – вектор $(\bar{n}_1, ..., \bar{n}_k)$, где $\bar{n}_i$ – некоторое заданное значение параметра n_i . $\{\bar{N}\}$ – множество всевозможных векторов $\bar{N}$. Q– множество операций, используемых алгоритмом α.

Любое однозначное отображение $w: \{\bar{N}\} \to V$, где V – множество всех выражений над B и Q, называется безусловным Q-термом. Если при любом $\bar{N} \in \{\bar{N}\}$ и любой интерпретации переменных B $w(\bar{N})$ принимает значение логического типа, то w называется безусловным логическим Q-термом. Пусть $u_1, ..., u_l$ – безусловные логические Q-термы, $w_1, ..., w_l$ – безусловные Q-термы. Множество пар (u_i, w_i), где $i = 1, ..., l$, обозначается $(\bar{u}, \bar{w}) = \{(u_i, w_i)\}_{i=1,...,l}$ и называется условным Q-термом длины l. Счетное множество пар безусловных Q-термов $(\bar{u}, \bar{w}) = \{(u_i, w_i)\}_{i=1,2,...}$ называется условным бесконечным Q-термом, если $\{(u_i, w_i)\}_{i=1,...,l}$ является условным Q-термом для любого $l < \infty$. Если не имеет значения, является ли Q-терм безусловным, условным или условным бесконечным, то его можно называть Q-термом.

Под вычислением безусловного Q-терма w при интерпретации B следует понимать вычисление выражения $w(\bar{N})$ при некотором $\bar{N} \in \{\bar{N}\}$. Для вычисления при заданной интерпретации B и $\bar{N} \in \{\bar{N}\}$ условного Q-терма $(\bar{u}, \bar{w}) = \{(u_i, w_i)\}_{i=1,...,l}$ необходимо найти такие $u_{i_0}(\bar{N}), w_{i_0}(\bar{N})$, что $u_{i_0}(\bar{N})$ принимает значение *true*, а значение $w_{i_0}(\bar{N})$ определено. В качестве значения $(\bar{u}, \bar{w})$ нужно взять $w_{i_0}(\bar{N})$. Если установлено, что выражений $u_{i_0}(\bar{N}), w_{i_0}(\bar{N})$ не существует, то значение $(\bar{u}, \bar{w})$ для данной интерпретации B и $\bar{N}$ не определено. Вычисление условного бесконечного Q-терма определяется аналогично.

Предположим, что I_1, I_2, I_3 – подмножества множества $I = (1,...,m)$ такие, что: одно или два из множеств $I_i\ (i = 1,2,3)$ могут быть пустыми, $I_1 \cup I_2 \cup I3 = I$, $I_i \cap I_j = \varnothing\ (i \neq j; i, j = 1,2,3)$. Множество Q-термов $\{f_i\}_{i \in I}$ удовлетворяет условиям: $f_{i_1}\ (i_1 \in I_1)$ – безусловный Q-терм, $f_{i_1} = w^{i_1}$; $f_{i_2}\ (i_2 \in I_2)$ – условный Q-терм, $f_{i_2} = \{(u_j^{i_2}, w_j^{i_2})\}_{j=1,...,l_{i_2}}$, l_{i_2} является вычислимой функцией параметров N; $f_{i_3}\ (i_3 \in I_3)$ – условный бесконечный Q-терм, $f_{i_3} = \{(u_j^{i_3}, w_j^{i_3})\}_{j=1,2,...}$. Если алгоритм α состоит в том, что для определения $y_i\ (i \in I)$ требуется вычислить Q-терм f_i, то множество Q-термов $y_i\ (i \in I)$ называется Q-детерминантом алгоритма α, а представление алгоритма в виде $y_i = f_i\ (i \in I)$ представлением в форме Q-детерминанта.

Реализацией алгоритма α, представленного в форме Q-детерминанта $y_i = f_i\ (i \in I)$, называется вычисление Q-термов $f_i\ (i \in I)$ при заданной интерпретации B. Если реализация такова, что выражения $W(\overline{N}) = \{w^{i_1}(\overline{N})\ (i_1 \in I_1); u_j^{i_2}(\overline{N}), w_j^{i_2}(\overline{N})\ (i_2 \in I_2, j = 1,...,l_{i_2}); u_j^{i_3}(\overline{N}), w_j^{i_3}(\overline{N})\ (i_3 \in I_3, j = 1,2,...)\}$ вычисляются одновременно и при их вычислении операции выполняются по мере готовности, то реализация называется Q-эффективной. Если алгоритм допускает распараллеливание, то его Q-эффективная реализация полностью использует ресурс параллелизма алгоритма, поэтому является максимально параллельной реализаций алгоритма. Реализация алгоритма α называется выполнимой, если одновременно необходимо выполнять конечное число операций [1, 5].

1.2. Проектирование параллельных программ на основе Q-детерминанта

Подход к разработке программы, выполняющей Q-эффективную реализацию алгоритма, основан на следующих утверждениях:

1) Q-детерминант можно построить для любого численного алгоритма;

2) Q-детерминант позволяет описать Q-эффективную реализацию алгоритма;

3) если Q-эффективная реализация алгоритма является выполнимой, то можно разработать программу для ее выполнения.

Процесс разработки программы состоит из трех этапов:

1) построение Q-детерминанта алгоритма;

2) описание Q-эффективной реализация алгоритма;

3) если Q-эффективная реализация выполнима, то для нее разрабатывается программа.

Разработанную программу будем называть Q-эффективной, а процесс ее разработки – Q-эффективным программированием. Q-эффективная программа полностью использует ресурс параллелизма алгоритма, т.к. выполняет его Q-эффективную реализацию. В связи с этим она не допускает дальнейшего распараллеливания. Q-эффективная программа – программа, выполняющая Q-эффективную реализацию алгоритма на определенной вычислительной архитектуре. В данной работе для рассматриваются следующие особенности архитектуры: наличие общей либо распределенной памяти, а также количество независимых вычислительных узлов.

Если готовы к выполнению несколько операций цепочки, то они выполняются по схеме сдваивания, которая ускоряет вычисления, изображенной на рис. 1.

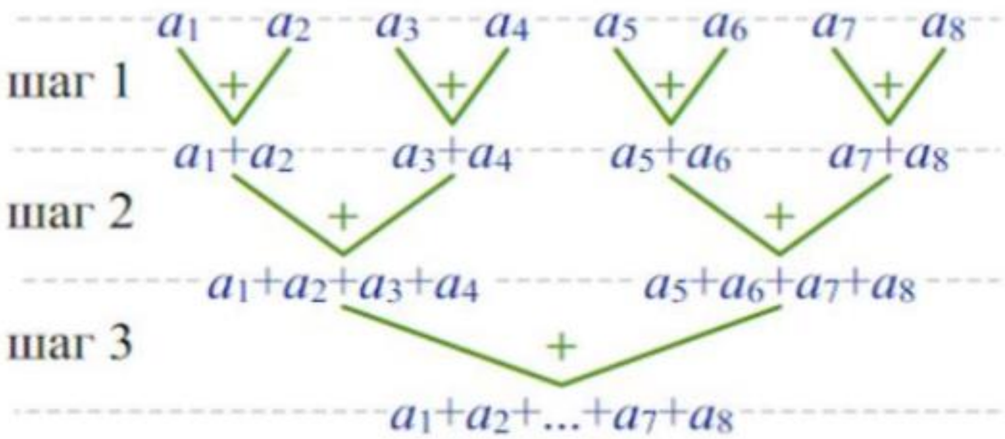

Рис. 1. Схема сдваивания

1.3. Распараллеливание вычислений

OpenMP – открытый стандарт для распараллеливания программ. Разработку спецификации OpenMP ведут несколько крупных производителей вычислительной техники и программного обеспечения.

Распараллеливание в OpenMP выполняется явно при помощи вставки в текст программы специальных директив, а также вызова вспомогательных функций. При использовании OpenMP предполагается SPMD-модель (Single Program Multiple Data) параллельного программирования, в рамках которой для всех параллельных нитей используется один и тот же код. Программа начинается с последовательной области – сначала работает один процесс (нить), при входе в параллельную область порождается ещё некоторое число процессов, между которыми в дальнейшем распределяются части кода. По завершении параллельной области все нити, кроме одной (нити- мастера), завершаются, и начинается последовательная область. В программе может быть любое количество параллельных и последовательных областей. Кроме того, параллельные области могут быть также вложенными друг в друга. В отличие от полноценных процессов, порождение нитей является относительно быстрой операцией, поэтому частые порождения и завершения нитей не так сильно влияют на время выполнения программы. Для написания эффективной параллельной программы необходимо, чтобы все нити, участвующие в обработке программы, были равномерно загружены полезной работой. Это достигается тщательной балансировкой загрузки, для чего предназначены различные механизмы OpenMP. Существенным моментом является также необходимость синхронизации доступа к общим данным. Само наличие данных, общих для нескольких нитей, приводит к конфликтам при одновременном несогласованном доступе. Поэтому значительная часть функциональности OpenMP предназначена для осуществления различного рода синхронизаций работающих нитей. OpenMP не выполняет синхронизацию доступа различных нитей к одним и тем же файлам. Если это необходимо для корректности программы, пользователь должен явно использовать директивы синхронизации или соответствующие библиотечные функции. При доступе каждой нити к своему файлу никакая синхронизация не требуется [8, 10, 16].

Отличия последовательной и параллельной программы изображены на рис. 2.

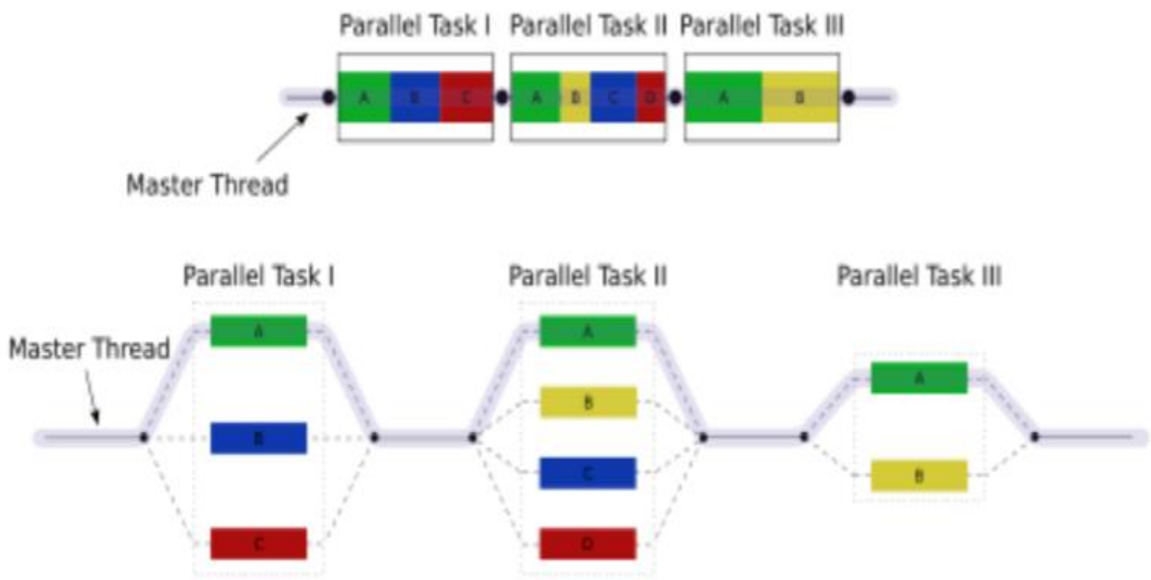

Рис. 2. Разбиение на потоки в OpenMP

1.4. Сеточные уравнения

Для решения задач линейной алгебры существует много различных численных методов, непрерывно ведется работа по их усовершенствованию, проводится переоценка методов, разрабатываются новые методы. В результате оказывается, что значительная часть имеющихся методов имеет право на существование, обладает своей областью применимости. Поэтому для решения конкретной задачи на ЭВМ существует проблема выбора одного метода на множестве допустимых методов решения данной задачи. Этот метод должен, очевидно, обладать наилучшими характеристиками, такими, как минимум времени решения задачи на ЭВМ (или минимум числа арифметических и логических операций при отыскании решения), вычислительная устойчивость, т. е. устойчивость по отношению к ошибкам округления, и др.

Значительное число задач физики и техники приводят к дифференциальным уравнениям в частных производных (уравнениям математической физики). Установившиеся процессы различной физической природы описываются уравнениями эллиптического типа. Точные решения краевых задач для эллиптических уравнений удается получить лишь в частных случаях. Поэтому эти задачи в основном решают приближенно. Одним из наиболее универсальных и эффективных методов, получивших в настоящее время широкое распространение для приближенного решения уравнений математической физики, является метод конечных разностей или метод сеток.

Применение различных численных методов (разностных, вариационно-разностных, проекционно-разностных методов, в том числе метода конечных элементов) для решения дифференциальных уравнений приводит к системе линейных алгебраических уравнений специального вида – к разностным уравнениям. Эта система обладает следующими специфическими чертами:

1) она имеет высокий порядок, равный числу узлов сетки;

2) система плохо обусловлена (отношение максимального собственного значения матрицы системы к минимальному велико; так, для разностного оператора Лапласа это отношение обратно пропорционально квадрату шага сетки);

3) матрица системы является разреженной – в каждой ее строке отлично от нуля несколько элементов, число которых не зависит от числа узлов;

4) ненулевые элементы матрицы расположены специальным образом – матрица является ленточной.

При аппроксимации на сетке интегральных и интегро-дифференциальных уравнений мы получаем систему уравнений относительно функции, заданной на сетке (сеточной функции). Такие уравнения естественно называть сеточными уравнениями:

$$\sum_{\vartheta \in \omega} a(x, \vartheta) y(\vartheta) = f(x), \qquad x \epsilon \omega,$$

где суммирование проводится по всем узлам сетки ω, т. е. по дискретному множеству точек. Матрица $a(x, \vartheta)$ сеточного уравнения является, в общем случае, заполненной.

2. ПРОЕКТИРОВАНИЕ

2.1. Проектирование и создание Q-эффективной реализации метода прогонки для системы линейных трехточечных уравнений

2.1.1. Постановка задачи

Пусть дана система:

$$\begin{cases} c_0 y_0 - b_0 y_1 = f_1, i = 0, \\ -a_i y_{i-1} + c_i y_i - b_i y_{i+1} = f_i, 1 \leq\ i\ \leq\ N-1, \\ -a_N y_{N-1} + c_N y_N = f_N, i = N, \end{cases}$$

или в векторном виде AY = F, где Y = $(y_0, y_i, \ldots, y_N)$ – вектор неизвестных, F = $(f_0, f_i, \ldots, f_N)$ – вектор правых частей, A – квадратная матрица размера $(N+1) \times (N+1)$.

$$A = \begin{vmatrix} c_0 & -b_0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 \\ -a_1 & c_1 & -b_1 & 0 & \ldots & 0 & 0 & 0 & 0 \\ 0 & -a_2 & c_2 & -b_2 & \ldots & 0 & 0 & 0 & 0 \\ . & . & . & . & \ldots & . & . & . & . \\ 0 & 0 & 0 & 0 & \ldots & -a_{N-2} & c_{N-2} & -b_{N-2} & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & -a_{N-1} & c_{N-1} & -b_{N-1} \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 & -a_N & c_N \end{vmatrix}.$$

Системы такого вида возникают при трехточечной аппроксимации краевых задач для обыкновенных дифференциальных уравнений второго порядка с постоянными и переменными коэффициентами, а также при реализации разностных схем для уравнений в частных производных. В последнем случае обычно требуется решить не единичную задачу, а серию задач с различными правыми частями, причем число задач в серии может равняться нескольким десяткам и сотням при числе неизвестных в каждой задаче N ≈ 100. Поэтому необходимо разработать экономичные методы решения задач, число действий для которых пропорционально числу неизвестных. Для данной системы таким методом является метод прогонки.

Умножим первое уравнение на a_1, сложим со вторым и разделим на c_1. Получаем укороченную систему, которая не содержит неизвестное y_0. На втором шаге будет исключено неизвестное y_1, на третьем - y_2 и т.д.

Найдем неизвестные по формулам:

$$y_i = \alpha_{i+1}y_{i+1} + \beta_{i+1},\ i = N\text{-}1, N\text{-}2, \ldots, 0,\ y_N = \beta_{N+1},$$

где α_i и β_i находятся по рекуррентным формулам:

$$\alpha_{i+1} = \frac{b_i}{c_i - a_i\alpha_i}, i = 1,2, \ldots, N-1,\ \alpha_1 = b_0/c_0,$$

$$\beta_{i+1} = \frac{f_i + a_i\beta_i}{c_i - a_i\alpha_i}, i = 1,2, \ldots, N,\ \beta_1 = f_0/c_0.$$

2.1.2. Проектирование параллельной программы для Q-эффективной реализации алгоритма прогонки

Теперь построим Q-детерминант метода прогонки и спроектируем параллельную Q-эффективную реализацию метода.

Количество Q-термов в Q-детерминанте алгоритма зависит от количества входных данных. Q-детерминант метода прогонки состоит из N безусловных Q-термов.

Q-детерминант имеет вид:

$$y_i = \alpha_{i+1}y_{i+1} + \beta_{i+1},$$

$$Y_n = \beta_{n+1},\ (i=N\text{-}1, N\text{-}2, \ldots, 0), \text{ где}$$

$$\alpha_{i+1} = \frac{b_i}{\hat{c}_i - a_i\alpha_i},\ i=1,2,\ldots,n,\ \alpha_1 = \frac{b_0}{c_0}$$

$$\beta_{i+1} = \frac{\hat{f}_i + a_i\beta_i}{\hat{c}_i - a_i\alpha_i},\ i=1,2,\ldots,n,\ \beta_1 = \frac{f_0}{c_0}$$

$$\hat{f}_n = \frac{f_{n-1}a_n}{\hat{c}_{n-1}} + f_n, n = 1,\ldots,N$$

$$\hat{c}_n = \frac{-b_{n-1}a_n}{\hat{c}_{n-1}} + c_n, \qquad n = 1,\ldots,N$$

2.2. Проектирование и создание Q-эффективной реализации метода Фурье для решения краевой задачи

2.2.1. Постановка задачи

Одним из методов отыскания решений сеточных многомерных задач, допускающих разделение переменных, является разложение искомого решения в конечную сумму Фурье по собственным функциям соответствующих сеточных операторов. Эффективность этого метода существенно зависит от того, как быстро можно вычислить коэффициенты Фурье заданной сеточной функции и восстановить искомую функцию по заданным коэффициентам Фурье.

Рассмотрим разностную задачу:

$$\frac{u_{n-1} - 2u_n + u_{n+1}}{\tau^2} = -f_n, n = 1, \dots, N-1, u_0 = u_n = 0, \tau N = L.$$

Рассмотрим ее решение в виде разложения по базису собственных функций разностного оператора:

$$P^\tau(u) = \frac{u_{n-1} - 2u_n + u_{n+1}}{\tau^2}.$$

Этот оператор имеет полную ортонормированную систему собственных функций:

$$\omega_\mathrm{k}(\mathrm{t_n}) = \sqrt{\frac{2}{L}}\, sin \frac{Pi\ k\ (\tau n)}{L}, k = 1 \dots N-1, \tau_n = t_n,$$

которые соответствуют собственным значениям оператора $P^\tau(u)$:

$$\lambda_k = \frac{4}{\tau^2} sin^2 \frac{Pi\ k\ \tau}{2L}.$$

Будем искать решение в виде:

$$u_n = u(t_n) = \sum_{k=1}^{N-1} c_k w_k(t_n),$$

(n = 1…N-1),

где: $\mathrm{c_k}$ – пока неизвестные коэффициенты Фурье.

Для нахождения этих коэффициентов представим правую часть разностного уравнения в виде суммы Фурье

$$f_n = \sum_{k=1}^{N-1} \hat{f}_k\, \omega_k(t_n),$$

$$\hat{f}_k = \sum_{i=1}^{N-1} f_i w_k(t_i)\tau.$$

Подставим выражение для $u_n f_n$ в виде сумм Фурье в исходное разностное уравнение:

$$P^{\tau}\left[\sum_{k=1}^{N-1} c_k w_k(t_n)\right] = -\sum_{k=1}^{N-1} c_k w_k(t_n),$$

откуда с учетом соотношения $P^{\tau}(\omega_k) = -\lambda\omega_k$, получим:

$$-\sum_{k=1}^{N-1} c_k(\lambda_k\omega_k) = -\sum_{k=1}^{N-1} \hat{f}_k\omega_k, \quad c_k\lambda_k = \hat{f}_k, c_k = \frac{\hat{f}_k}{\lambda_k}.$$

2.2.2. Проектирование параллельной программы для Q-эффективной реализации алгоритма Фурье

Теперь построим Q-детерминант метода Фурье и спроектируем параллельную Q-эффективную реализацию метода.

Количество Q-термов в Q-детерминанте алгоритма зависит от количества входных данных. Q-детерминант метода Фурье состоит из N безусловных Q-термов.

Представление метода прогонки в форме Q-детерминанта имеет вид:

$$u_{n=}\sum_{k=1}^{N-1}\frac{\sum_{i=1}^{N-1} f_i\sqrt{\frac{2}{L}}sin\frac{Pi\ k\ (\tau i)}{L}}{\frac{4}{\tau^2}sin^2\frac{Pi\ k\ \tau}{2L}} * \sqrt{\frac{2}{L}}sin\frac{Pi\ k\ (\tau n)}{L}, n = 1, \ldots, N.$$

Поскольку неизвестные не зависят друг от друга, они могут вычисляться параллельно.

2.3. Варианты использования

Реализуемые Q-эффективные программы представляются в виде приложений, позволяющих решать систему линейных трехточечных уравнений методом прогонки и краевую задачу методом Фурье. Единственным актером данной системы является пользователь системы, выполняющий вычислительные эксперименты.

Пользователь может:

1) задавать размерность входные данные;

2) решать систему уравнений последовательно;
3) решать систему уравнений параллельно.

Диаграмма вариантов использования представлена на рис. 3.

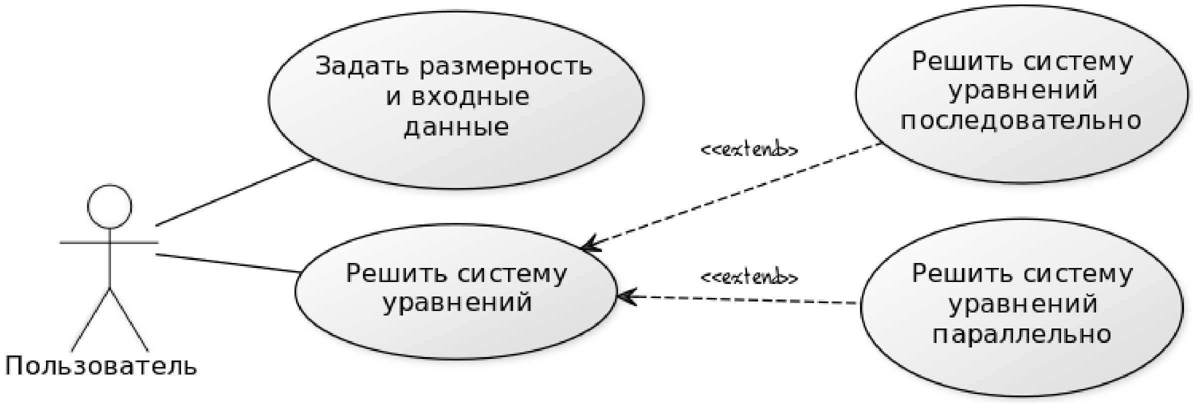

Рис. 3. Диаграмма вариантов использования

3. РЕАЛИЗАЦИЯ

В ходе выполнения работы было написано 4 программы: последовательная и параллельная реализация метода прогонки для решения системы линейных трехточечных уравнений и последовательная и параллельная реализация метода Фурье для решения краевой задачи.

3.1. Реализация метода прогонки для решения системы линейных трехточечных уравнений

Q-детерминант алгоритма имеет вид:

$$y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1},$$

$$Y_n = \beta_{n+1},\ (i{=}N\text{-}1,\ N\text{-}2, \ldots, 0),\ \text{где}$$

$$\alpha_{i+1} = \frac{b_i}{\hat{c}_i - a_i \alpha_i},\ i{=}1{,}2, \ldots, n,\ \alpha_1 = \frac{b_0}{c_0}$$

$$\beta_{i+1} = \frac{\hat{f}_i + a_i \beta_i}{\hat{c}_i - a_i \alpha_i},\ i{=}1{,}2, \ldots, n,\ \beta_1 = \frac{f_0}{c_0}$$

$$\hat{f}_n = \frac{f_{n-1} a_n}{\hat{c}_{n-1}} + f_n,\ n = 1, \ldots, N$$

$$\hat{c}_n = \frac{-b_{n-1} a_n}{\hat{c}_{n-1}} + c_n,\ n = 1, \ldots, N$$

Поскольку при вычислении каждого элемента матрицы используется предыдущий элемент матрицы, то параллелизмом обладает только вычисление каждого из элементов.

На рис. 4 представлен листинг вычисления неизвестных y_i.

```
for (int j = 1; j < N + 1; j++)
{
	Alph[j] = B1 / (C1 - A1 * Alph[j - 1]);
	Bet[j] = (F0 + A1 * Bet[j - 1]) / (C1 - A1 * Alph[j - 1]);
}
Ans[N - 1] = Bet[N];
for (int j = N - 2; j > 0; j--)
{
	F[j] = Alph[j + 1] * F[j + 1] + Bet[j + 1];
}
```

Рис. 4. Вычисление неизвестных y_i

3.2. Реализация метода Фурье для решения краевой задачи

Q-детерминант метода Фурье имеет вид

$$u_{n=}\sum_{k=1}^{N-1}\frac{\sum_{i=1}^{N-1} f_i\sqrt{\frac{2}{L}}sin\frac{Pi\,k\,(\tau i)}{L}}{\frac{4}{\tau^2}sin^2\frac{Pi\,k\,\tau}{2L}} * \sqrt{\frac{2}{L}}sin\frac{Pi\,k\,(\tau n)}{L}, n = 1, \ldots, N$$

В отличии от метода прогонки, в методе Фурье все неизвестные не зависят друг от друга и могут вычисляться одновременно.

Листинг вычисления u_n представлен на рис. 5.

```
#pragma omp parallel for
for (int j = 1; j < N-2; j++)
{
	for (int k = 0; k < N; k++)
		{
			for (int i = 0; i < N; i++)
				{
					vars = F[i] * sqrt(2 / L)*sin(PI * k * i / L);
					vargs = vargs + vars;
				}
			F0 = vargs * sqrt(2 / L) * sin(PI * k * n1 * tao / L) /
(4 /
			(tao * tao) * sin(PI * k * tao / 2 / L) * (PI * k * tao /
2 /
			L));
			SG = SG + F0;
		}
	Ans[j] = SG;
	n1++;
}
```

Рис. 5. Листинг вычисления u_n

4. ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

4.1. Тестирование

Для проверки корректности работы разработанных программ было проведено функциональное тестирование.

Функциональное тестирование – это тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности ПО в определенных условиях решать задачи, нужные пользователям. Функциональные требования определяют, какие задачи ПО решает. В рамках данной работы были разработаны программы с единственной функциональной возможностью: решить систему уравнений методом прогонки/методом Фурье. Исходные данные считываются из файла, а результаты выводятся в другой файл. Программы были протестированы на системах уравнений малых размерностей с заранее известными результатами. Для нахождения ускорения использовались системы уравнений больших размерностей.

4.2. Эксперименты

Для исследования было решено использовать такие данные, как S_p – ускорение работы параллельной программы и E_p – эффективность параллельной программы, которые вычисляются как $S_p = T_1/T_p$ и $E_p = S_p/p$, где T_1 – время работы последовательной программы, T_p – время работы параллельной программы, а p – количество ядер.

4.2.1. Результаты тестирования программы для решения системы линейных трехточечных уравнений методом прогонки

На рис. 6 можно увидеть зависимость времени выполнения последовательной и параллельных программ от количества входных данных. Исследование показало, что распараллеливать метод прогонки для решения системы линейных трехточечных уравнений не имеет смысла, поскольку различия во времени исполнения не существенны.

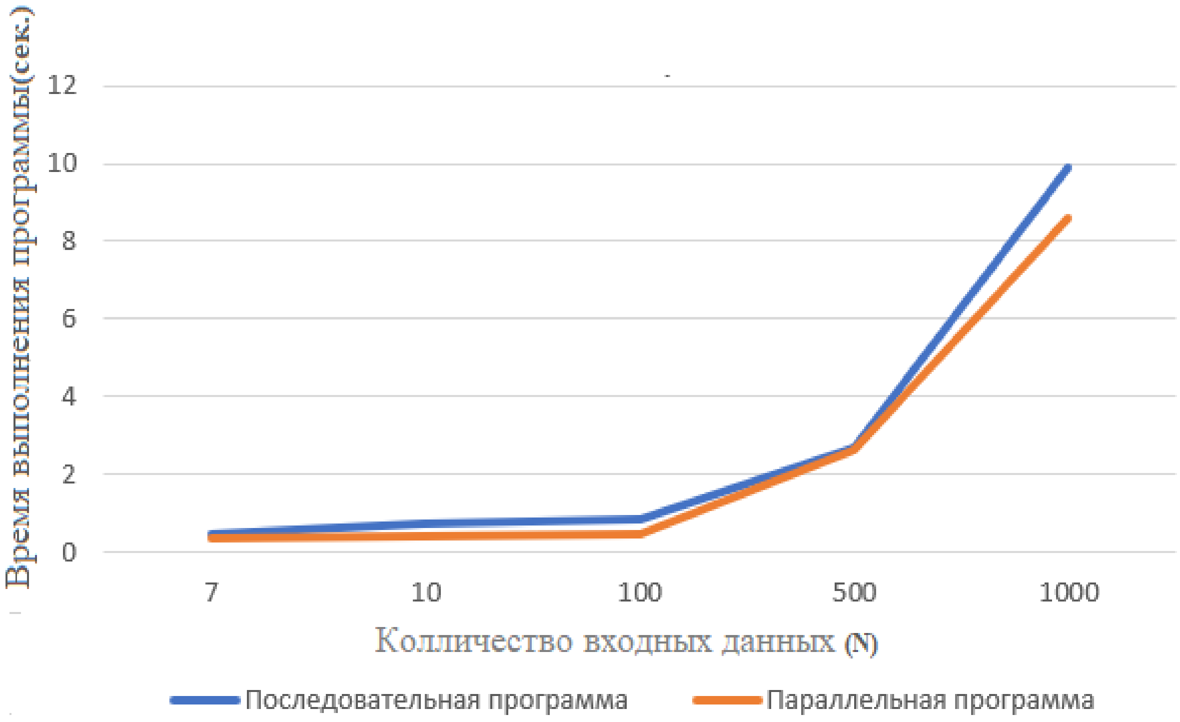

Рис. 6. График зависимости времени выполнения последовательной и параллельных программ от количества входных данных методом прогонки

На рис. 7 и 8 представлены графики ускорения и эффективности для метода прогонки.

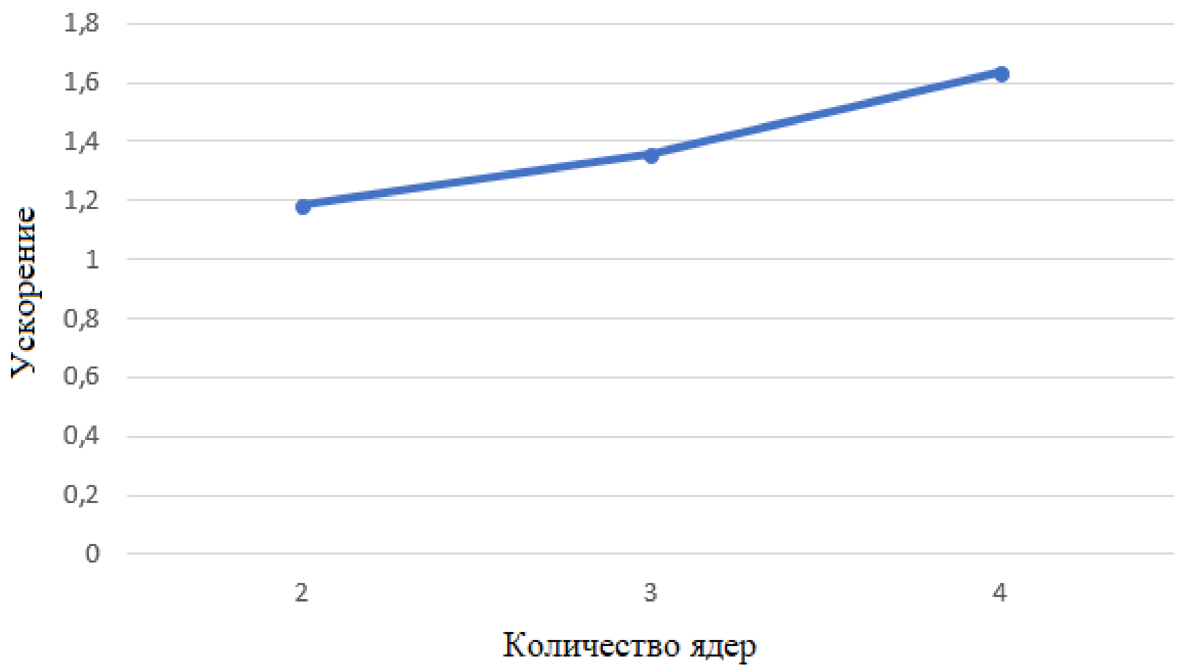

Рис. 7. Ускорение Q-эффективной программы для решения системы линейных трехточечных уравнений методом прогонки

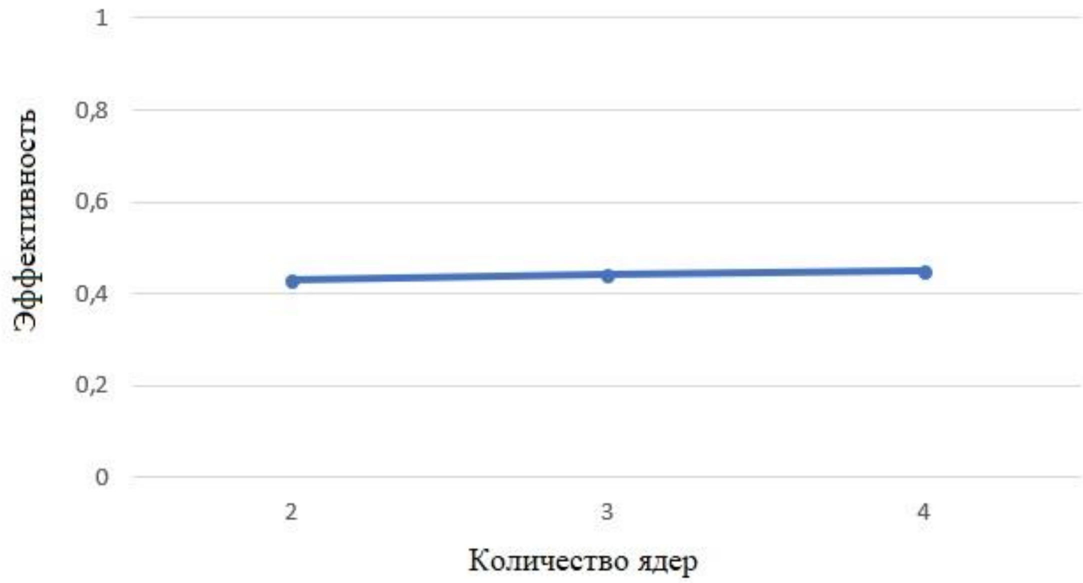

Рис. 8. Эффективность Q-эффективной программы для решения системы линейных трехточечных уравнений методом прогонки

4.2.2. Результаты тестирования программы для решения краевой задачи методом Фурье

На рис. 9 можно увидеть зависимость времени выполнения последовательной и параллельных программ от количества входных данных.

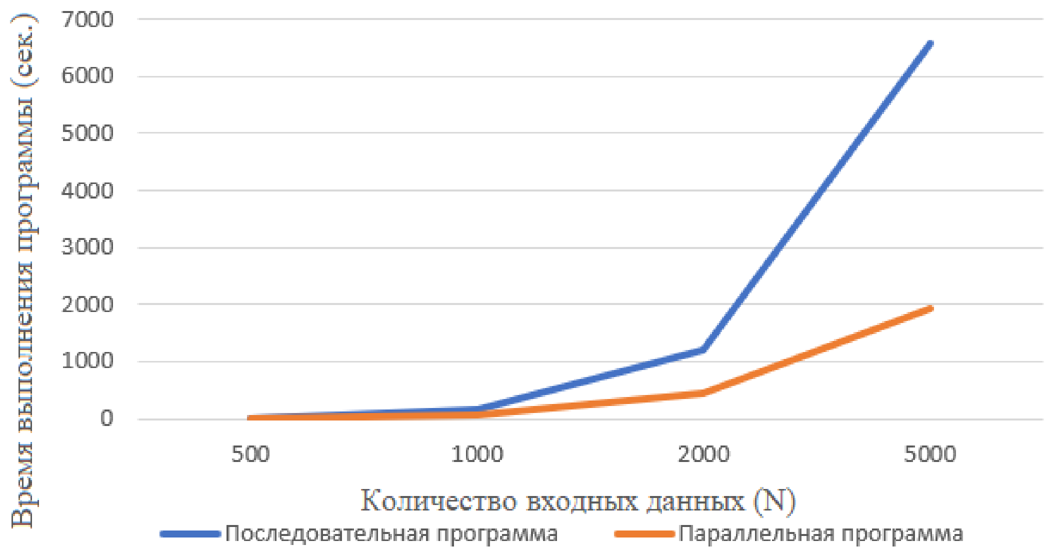

Рис. 9. График зависимости времени выполнения последовательной и параллельных программ от количества входных данных методом Фурье

На рис. 10 и 11 представлены графики ускорения и эффективности для метода Фурье.

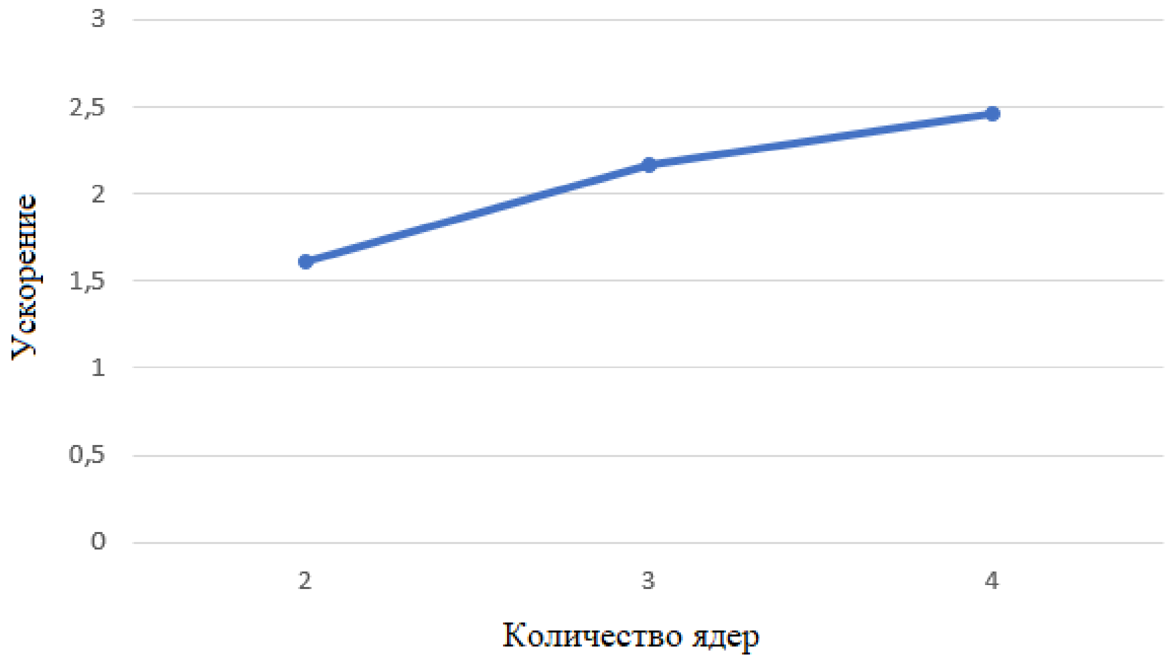

Рис. 10. Ускорение Q-эффективной программы для решения краевой задачи методом Фурье

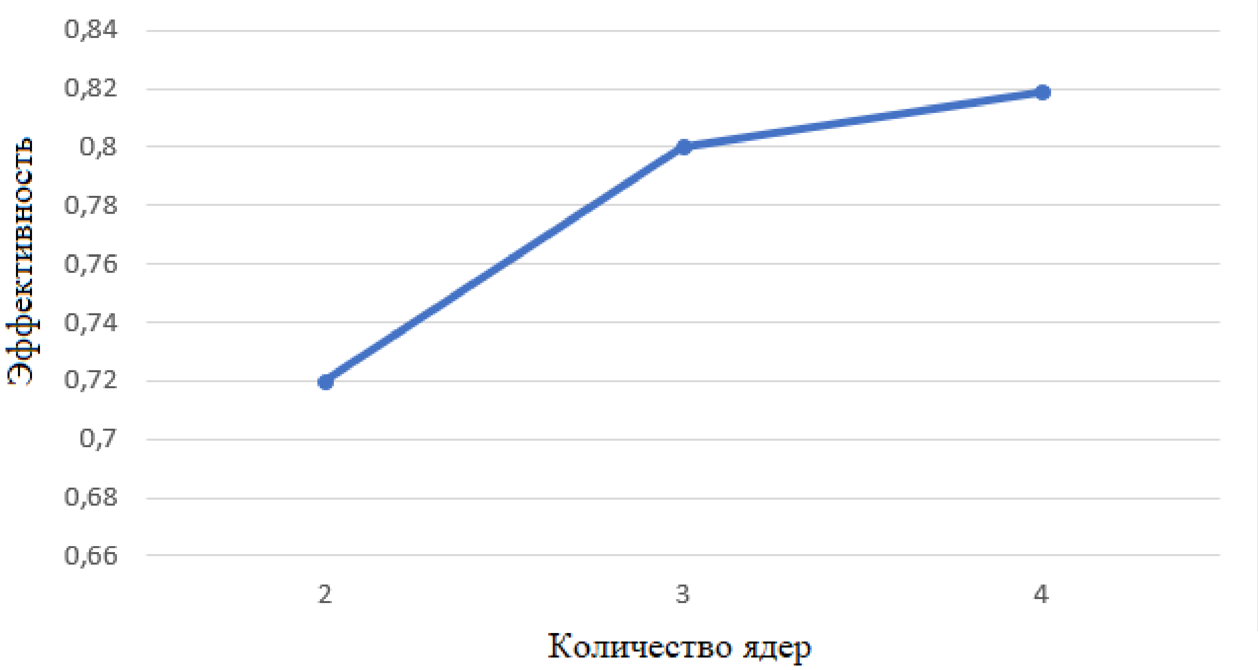

Рис. 11. Ускорение Q-эффективной программы для решения краевой задачи методом Фурье

4.2.3. Ускорение и эффективность программ

С помощью формул $S_p = T_1/T_p$ и $E_p = S_p/p$ рассчитаем ускорение и эффективность программ.

Таблица 1 содержит время выполнения, ускорение и эффективность программ.

Табл. 1. Время выполнения, ускорение и эффективность программ

Количество входных данных	7	10	100	500	1000
Время выполнения метода прогонки (послед.)	0.446	0.723	0.816	4.75	9.988
Время выполнения метода прогонки (паралл.)	0.365	0.411	0.456	2.652	5.506
Время выполнения метода Фурье (послед.)	0.854	0.957	1.766	15.209	153.615
Время выполнения метода Фурье (паралл.)	0.728	0.817	0.945	7.778	62.417
Ускорение метода прогонки, S_p	1.2219	1.7591	1.7895	1.7911	1.814
Ускорение метода Фурье, S_p	1.1731	1.1714	1.8688	1.9554	2.4611
Эффективность метода прогонки, E_p	0.3055	0.4398	0.4474	0.4478	0.4535
Эффективность метода Фурье, E_p	0.2933	0.2928	0.4672	0.4888	0.6153

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы были разработаны Q-эффективные реализации алгоритма прогонки для трехточечных уравнений и алгоритма Фурье. Было проведено сравнение быстродействия полученных алгоритмов с последовательной реализацией этого метода. Были решены следующие задачи.

1. Изучен подход к распараллеливанию алгоритмов, основанный на представлении алгоритмов в форме Q-детерминанта.

2. Изучен метод проектирования параллельных программ на основе концепции Q-детерминанта.

3. Изучен метод прогонки для решения системы линейных трехточечных уравнений.

4. Изучен метод Фурье для решения краевой задачи.

5. Метод прогонки для системы линейных трехточечных уравнений и метод Фурье представлены в форме Q-детерминанта.

6. Изучена технология OpenMP.

7. Разработана Q-эффективная программа для решения системы линейных трехточечных уравнений методом прогонки.

8. Разработана Q-эффективная программа для решения краевой задачи методом Фурье.

9. Проведен анализ динамических характеристик разработанных Q-эффективных программ.

Основной целью работы являлась разработка Q-эффективных программ для решения системы линейных сеточных уравнений методом прогонки и решения краевой задачи методом Фурье.

В процессе построения Q-детерминанта метода прогонки для системы линейных трехточечных уравнений стало очевидно, что достичь параллельности можно только при вычислении одного неизвестного, т.к. неизвестные зависят друг от друга и не могут вычисляться независимо.

В методе Фурье для решения краевой задачи все неизвестные могут вычисляться независимо друг от друга, благодаря чему возможно получить максимально параллельную программу.

СПИСОК ЛИТЕРАТУРЫ

1. Aleeva V.N., Sharabura I.S., Suleymanov D.E. Software System for Maximal Parallelization of Algorithms on the Base of the Conception of Qdeterminant. // 13th International Conference on Parallel Computing Technologies, PaCT 2015, Petrozavodsk, Russian Federation, 31 August - 4 September 2015; Proceedings. Lecture Notes in Computer Science, Springer, 2015. – Vol. 9251. – P. 3-9.

2. Алеева В.Н. Анализ параллельных численных алгоритмов – Новосибирск: Препринт ВЦ СО АН СССР, 1985. – № 590. – 23 с.

3. Алеева, В.Н. А458 Архитектура вычислительных систем: курс лекций / В.Н. Алеева. – Челябинск: Издательский центр ЮУрГУ, 2013.– 96 с.

4. Алеева В.Н. Расширенная модель концепции Q-детерминанта и ее применение для реализации ресурса параллелизма численных алгоритмов. / В.Н. Алеева, Н.В. Валькевич, Ю.С. Лаптева, Д.Е. Тарасов // «Современные проблемы математического моделирования, обработки изображений и параллельных вычислений 2017» (СПММОИиПВ-2017): труды Международной научной конференции (пос. Дивноморское, 4–11 сентября 2017 г.). Том I; Донской гос. техн. ун-т. – Ростов-на-Дону: ООО «ДГТУ-Принт», 2017. – С. 23–32.

5. Борзунов С.В. Практикум по параллельному программированию. – СПб.: Изд-во БВХ-Петербург, 2017. – 265 с.

6. Гергель В.П. Теория и практика параллельных вычислений. Учебное пособие. – ИНТУИТ, Бином. Лаборатория знаний, 2007. – 40 с.

7. Игнатьев С.В. Определение ресурса параллелизма алгоритмов на базе концепции Q-детерминанта: Вып. квалиф. работа магистра прикладной математики и информатики: 010500.68 /Южно-Уральский государственный университет. – Челябинск, 2009. – 75 с.

8. Левин М.П. Параллельное программирование с использование OpenMP. Учебное пособие. – М.: БИНОМ. Лаборатория знаний, Интернет-Университет Информационных Технологий (ИНТУИТ), 2008. – 120 с.

9. Параллельное программирование с использованием технологии OpenMP: Учебное пособие. – М.: Изд-во МГУ, 2009. – 77 с.

10. Самарский А. А., Гулин А. В. Численные методы: Учеб, пособие для вузов. – М.: Наука. Гл. ред. физ-мат. лит., 1989. – 432 с. – ISBN 5-02-013996-3.

11. Самарский А.А., Николасо Е.С. Методы решения сеточных уравнений. Главная редакция физико-математической литературы изд-ва «На ука*, М., 1978.

12. Свирихин Д.И., Алеева В.Н. Определение максимально эффективной реализации алгоритма на основе концепции Q-детерминанта. // Параллельные вычислительные технологии (ПаВТ'2013): труды международной научной конференции (1–5 апреля 2013 г., г. Челябинск). – Челябинск: Издательский центр ЮУрГУ, 2013. – С. 617.

13. Свирихин, Д.И. Определение ресурса параллелизма алгоритма и его эффективного использования для конечного числа процессоров. // Научный сервис в сети Интернет: поиск новых решений: труды Международной суперкомпьютерной конференции (17-22 сентября 2012 г., г. Новоросийск). – М.: изд. МГУ, 2012. – С. 257–260.

14. Свирихин Д.И. Построение эффективной реализации алгоритма на основе концепции Q-детерминанта: Вып. квалиф. работа магистра: 010300.68 – Челябинск: Южно-Уральский государственный университет, 2013. – 47 с.

15. Официальный сайт проекта OpenMP. [Электронный ресурс] URL: http://www.openmp.org (дата обращения: 18.02.2018).

16. Официальный сайт проекта НОУ ИНТУИТ. Курс: введение в вычислительную математику. Лекция 11: численное решение краевых задач для систем обычных дифференциальных уравнений. [Электронный ресурс] URL: https://www.intuit.ru/studies/courses/1012/168/lecture/4608 (дата обращения: 26.02.2018).