



ВЕСТНИК

ЮЖНО-УРАЛЬСКОГО № 47 (306)
ГОСУДАРСТВЕННОГО
УНИВЕРСИТЕТА 2012

ISSN 2305-9052

СЕРИЯ

«ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И ИНФОРМАТИКА»

Выпуск 2

Решением ВАК России включен в Перечень ведущих рецензируемых научных журналов и изданий

Учредитель — Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Южно-Уральский государственный университет» (национальный исследовательский университет)

Основной целью издания является пропаганда научных исследований в следующих областях:

- Вычислительная математика и численные методы
- Математическое программирование
- Распознавание образов
- Вычислительные методы линейной алгебры
- Решение обратных и некорректно поставленных задач
- Доказательные вычисления
- Численное решение дифференциальных и интегральных уравнений
- Исследование операций
- Теория игр
- Теория аппроксимации
- Информатика
- Математическое и программное обеспечение высокопроизводительных вычислительных систем
- Системное программирование
- Распределенные вычисления, облачные и грид-технологии
- Технология программирования
- Машинная графика
- Интернет-технологии
- Системы электронного обучения
- Технологии обработки баз данных и знаний
- Интеллектуальный анализ данных

Редакционная коллегия

д.ф.-м.н., проф. Соколинский Л.Б.,
отв. редактор

д.ф.-м.н., проф. Танана В.П.,
зам. отв. редактора

к.ф.-м.н., доц. Цымблер М.Л.,
отв. секретарь

д.ф.-м.н., проф. Карачик В.В.

д.ф.-м.н., проф. Менихес Л.Д.

д.ф.-м.н., проф. Панюков А.В.

Пан К.С., *техн. секретарь*

Редакционный совет

д.ф.-м.н., акад. РАН Бердышев В.И.,
председатель

д.ф.-м.н., чл.-кор. РАН Воеводин В.В.

д.ф.-м.н., акад. РАН Ерёмин И.И.

д.ф.-м.н., акад. РАН Куржанский А.Б.

д.ф.-м.н., чл.-кор. РАН Романов В.Г.

д.ф.-м.н., профессор Томилин А.Н.

д.ф.-м.н., чл.-кор. РАН Третьяков В.Е.

д.ф.-м.н., чл.-кор. РАН Федотов А.М.

д.ф.-м.н., профессор Ухоботов В.И.

д.ф.-м.н., чл.-кор. РАН Ушаков В.Н.

д.ф.-м.н., профессор Хачай М.Ю.



BULLETIN

OF SOUTH URAL
STATE UNIVERSITY

№ 47 (306)
2012

ISSN 2305-9052

SERIES

«COMPUTATIONAL MATHEMATICS
AND SOFTWARE ENGINEERING»

Issue 2

South Ural State University

The main purpose of the series is publicity of scientific researches in the following areas:

- Numerical analysis and methods
- Mathematical optimization
- Pattern recognition
- Numerical methods of linear algebra
- Reverse and ill-posed problems solution
- Computer-assisted proofs
- Numerical solutions of differential and integral equations
- Operations research
- Game theory
- Approximation theory
- Computer science
- High performance computer software
- System programming
- Distributed, cloud and grid computing
- Programming technology
- Computer graphics
- Internet technologies
- E-learning
- Database and knowledge processing
- Data mining

Editorial Board

L.B. Sokolinsky, South Ural State University (Chelyabinsk, Russian Federation)
V.P. Tanana, South Ural State University (Chelyabinsk, Russian Federation)
M.L. Zymbler, South Ural State University (Chelyabinsk, Russian Federation)
V.V. Karachik, South Ural State University (Chelyabinsk, Russian Federation)
L.D. Menikhes, South Ural State University (Chelyabinsk, Russian Federation)
A.V. Panyukov, South Ural State University (Chelyabinsk, Russian Federation)
C.S. Pan, South Ural State University (Chelyabinsk, Russian Federation)

Editorial Council

V.I. Berdyshev, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russian Federation)
V.V. Voevodin, Lomonosov Moscow State University (Moscow, Russian Federation)
I.I. Eremin, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russian Federation)
A.B. Kurzhanovsky, Lomonosov Moscow State University (Moscow, Russian Federation)
V.G. Romanov, Sobolev Institute of Mathematics, Siberian Branch of the RAS (Novosibirsk, Russian Federation)
A.N. Tomilin, Institute for System Programming of the RAS (Moscow, Russian Federation)
V.E. Tretyakov, Ural Federal University (Yekaterinburg, Russian Federation)
A.M. Fedotov, Institute of Computational Technologies, SB RAS (Novosibirsk, Russian Federation)
V.I. Ukhobotov, Chelyabinsk State University (Chelyabinsk, Russian Federation)
V.N. Ushakov, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russian Federation)
M.Yu. Khachay, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russian Federation)

Содержание

ПРОГНОЗИРОВАНИЕ ЧИСЛЕННОСТИ MICROCYSTIS AERUGINOSA НА ОСНОВЕ ПРАВИЛ НЕЧЕТКОЙ ЛОГИКИ И НЕЧЕТКИХ НЕЙРОННЫХ СЕТЕЙ А.О. Гаязова, С.М. Абдуллаев	5
О ЕДИНСТВЕННОСТИ РЕШЕНИЯ ОБРАТНОЙ ЗАДАЧИ НЕСТАЦИОНАРНОЙ ФИЛЬТРАЦИИ А.В. Боков	12
МЕТОДЫ ПАРАЛЛЕЛЬНОГО РЕШЕНИЯ СЛАУ НА СИСТЕМАХ С РАСПРЕДЕЛЕННОЙ ПАМЯТЬЮ В БИБЛИОТЕКЕ KRYLOV Д.С. Бутюгин, В.П. Ильин, Д.В. Перевозкин	22
ИССЛЕДОВАНИЕ УДАРНОГО ВЗАИМОДЕЙСТВИЯ ИНДЕНТОРА С ТКАНЕВЫМИ БРОНЕПЛАСТИНАМИ, РАСПОЛОЖЕННЫМИ НА ПЛАСТИЛИНОВОМ ОСНОВАНИИ Н.Ю. Долганина	37
КАЧЕСТВЕННОЕ ПОСТРОЕНИЕ РАСЧЕТНОЙ СЕТКИ ДЛЯ РЕШЕНИЯ ЗАДАЧ АЭРОДИНАМИКИ В ПРОГРАММНОМ КОМПЛЕКСЕ FLOWVISION П.И. Карасев, А.С. Шишаева, А.А. Аксенов	46
ОБРАБОТКА ЗАПРОСОВ НА КЛАСТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ С МНОГОЯДЕРНЫМИ УСКОРИТЕЛЯМИ П.С. Костенецкий	59
ПОДХОДЫ К ОПТИМИЗАЦИИ И РАСПАРАЛЛЕЛИВАНИЮ ВЫЧИСЛЕНИЙ В ЗАДАЧЕ ДЕТЕКТИРОВАНИЯ ОБЪЕКТОВ РАЗНЫХ КЛАССОВ НА ИЗОБРАЖЕНИИ Е.А. Козинев, В.Д. Кустикова, И.Б. Мееров, А.Н. Половинкин, А.А. Сиднев	68
ДВУХСЕТОЧНЫЕ ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ ДЛЯ РЕШЕНИЯ ДРОБНО-ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ АНОМАЛЬНОЙ ДИФФУЗИИ С.Ю. Лукашук	83
АНАЛИЗ УСТОЙЧИВОСТИ ПОЛОЖЕНИЯ РАВНОВЕСИЯ МОДЕЛИ НЕЙМАНА ПРИ ИНТЕРВАЛЬНОЙ НЕОПРЕДЕЛЕННОСТИ А.В. Панюков, А.Т. Латипова	99
ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ КАТАЛИТИЧЕСКОЙ РЕАКЦИИ (CO + O ₂ → CO ₂) С ПОМОЩЬЮ АСИНХРОННОГО КЛЕТОЧНОГО АВТОМАТА А.Е. Шарифулина	112
Краткие сообщения	
ПОДХОД К РАЗБИЕНИЮ СВЕРХБОЛЬШИХ ГРАФОВ С ПОМОЩЬЮ ПАРАЛЛЕЛЬНЫХ СУБД К.С. Пан	127

Contents

FORECASTING OF MICROCYSTIS AERUGINOSA SEASONAL DYNAMICS USING THE FUZZY LOGIC AND FUZZY NEURAL NETWORKS A.O. Gayazova, S.M. Abdullaev	5
ON THE UNIQUENESS OF THE INVERSE PROBLEM OF UNSTEADY FILTRATION A.V. Bokov	12
PARALLEL METHODS FOR SLAE SOLUTION ON THE SYSTEMS WITH DISTRIBUTED MEMORY IN KRYLOV LIBRARY D.S. Butyugin, V.P. Il'in, D.V. Perevozkin	22
INVESTIGATION OF IMPACT INTERACTION OF THE INDENTER WITH FABRIC ARMOR PLATE WHICH LOCATED ON THE CLAY BASIS N.Yu. Dolganina	37
ALGORITHM OF MESH ADAPTATION FOR ACCURATE AERODYNAMIC SIMULATION IN CFD CODE FLOWVISION P.I. Karasev, A.S. Shishaeva, A.A. Aksenov	46
QUERY PROCESSING ON CLUSTER BASED SYSTEMS WITH MULTICORE ACCELERATORS P.S. Kostenetskiy	59
APPROACHES TO THE OPTIMIZATION AND PARALLELIZATION OF COMPUTATIONS IN THE PROBLEM OF DETECTING OBJECTS OF DIFFERENT CLASSES IN THE IMAGE E.A. Kozinov, V.D. Kustikova, I.B. Meyerov, A.N. Polovinkin, A.A. Sidnev	68
PARALLEL TWO-GRIDS ALGORITHMS FOR SOLUTION OF ANOMALOUS DIFFUSION EQUATIONS OF FRACTIONAL ORDER S.Yu. Lukashchuk	83
STABILITY ANALYSIS OF EQUILIBRIUM POSITION OF VON NEUMANN'S MODEL UNDER INTERVAL UNCERTAINTY A.V. Panyukov, A.T. Latipova	99
PARALLEL IMPLEMENTATION OF CATALYTIC REACTION ($\text{CO} + \text{O}_2 \rightarrow \text{CO}_2$) BY ASYNCHRONOUS CELLULAR AUTOMATA A.E. Sharifulina	112

Brief Reports

APPLYING PARALLEL DBMS TO VERY LARGE GRAPH PARTITIONING C.S. Pan	127
---	-----

ПРОГНОЗИРОВАНИЕ ЧИСЛЕННОСТИ *MICROCYSTIS AERUGINOSA* НА ОСНОВЕ ПРАВИЛ НЕЧЕТКОЙ ЛОГИКИ И НЕЧЕТКИХ НЕЙРОННЫХ СЕТЕЙ

А.О. Гаязова, С.М. Абдуллаев

Описана процедура поиска потенциальных предикторов и создания прогнозных правил нечеткой логики и нечетких нейронных сетей для последующего прогнозирования вспышек численности синезеленой водоросли *M. aeruginosa*. В результате натурных наблюдений ряда биотических и абиотических параметров водной среды, проведенных на озере Смолино (г. Челябинск) за теплый период 2009 и 2011 года получены временные ряды численности *M. aeruginosa* и значений сопутствующих параметров. С помощью кросс-корреляционного анализа данных установлено, что потенциальными предикторами квазипериодических колебаний численности *M. aeruginosa* с периодом 12-20 дней могут выступать численность водоросли *P. duplex*, температура воды и концентрация нитрат-иона. По результатам кросс-корреляционного анализа заданы прогнозныe правила и функции принадлежности в диапазоне изменений предиктанта и предиктора от нуля до 1. Для «автоматического» задания прогнозных правил и функций принадлежности с помощью специально написанной программы произведено обучение нечеткой нейронной сети на данных о значениях предиктанта и отобранных в ходе предварительного анализа параметров-предикторов. Для сравнения результатов дополнительно осуществлена линейная экстраполяция данных о численности предиктанта. Выявлено, что экстраполяционный прогноз хорошо работает на квазилинейных интервалах изменения численности, а алгоритмы нечеткой логики потенциально способны определить время наступления интенсивных вспышек численности предиктанта.

Ключевые слова: *M. aeruginosa*, прогнозирование вспышек, квазипериодические колебания, линейная экстраполяция, нечеткая логика, нечеткие нейронные сети

Введение

В мониторинге окружающей среды все большую актуальность приобретают краткосрочные и среднесрочные прогнозы массового развития макро- и микроводорослей в морских и пресноводных экосистемах. Например, прогноз численности синезеленых водорослей сроком на несколько дней необходим для оптимизации работ систем водоочистки в период массового развития водорослей в водоемах — источниках питьевого водоснабжения. Как показывает опыт [1, 2] и многих других работ наиболее адекватными являются методы прогноза, основанные на математической обработке временных рядов численности прогнозируемой водоросли с учетом ряда гидрологических и биотических факторов, характеризующих текущее состояние водоема.

Специфические особенности экосистем, ограниченность набора первичной информации о параметрах среды, очевидно, ставят вопрос о выборе оптимального метода прогноза в условиях конкретного региона. В частности в работе [1] для включения дополнительных факторов используются правила нечеткой логики, а в [2] поиск прогнозных правил производится с помощью искусственных нейронных сетей. В настоящей работе оценивается сравнительная результативность методов линейной экстраполяции, правил нечеткой логики и нечетких нейронных сетей для краткосрочного прогнозирования численности синезеленой водоросли *Microcystis (M.) aeruginosa* на основе данных натурных наблюдений за изменением значений ее численности и оптимального набора сопутствующих физико-химических и биотических параметров.

1. Данные и методы их обработки

В качестве данных для прогноза используются данные натуральных наблюдений, проведенных первым автором в летне-осенний период 2009 и 2011 годов в прибрежной части озера Смолино, г. Челябинск. В ходе мониторинга одновременно с отбором проб на фитопланктон, определялись гидрохимические показатели, фиксировались локальные гидрометеорологические условия. Методы наблюдений, отбора и обработки проб описаны в предыдущей работе авторов [3, 4]. В результате наблюдений были получены ряды данных общей длиной в 138-159 дней с разрешением во времени от 2 до 7 дней. В целях удобства математического моделирования из исходных нерегулярных рядов посредством кусочно-линейной интерполяции получены регулярные временные серии с разрешением в один день. Кроме этого, поскольку различные данные имеют весьма широкий диапазон изменений величин (от долей единиц для физико-химических параметров водной среды до сотен тысяч для численностей водорослей) поэтому предварительно проведено масштабирование рядов с нормировкой значений на максимальную величину наблюдаемую в ряде.

Как видно из рисунка 1 численность *M. aeruginosa* за 2009 (2011) год обнаруживает значительные внутрисезонные колебания амплитуды с периодом 10 (20) дней. Для оценки наличия связей между численностью водорослей и факторами среды, был произведен кросскорреляционный анализ временных рядов, предварительно избавленных от смещения путем вычитания трехточечного скользящего среднего с равными весами. Найдено, что динамика численности *M. aeruginosa* в 2009 году относительно тесно связана с двумя факторами. Первый фактор — это численность водоросли *Pediastrum (P.) duplex* с коэффициентом кросс-корреляции $k \approx 0,9$ при временном сдвиге между рядами на шесть дней. Вторым значимым фактором — минерализация воды со значением $k \approx 0,6$ при нулевом временном сдвиге. Анализ связей между численностью *M. aeruginosa* и факторами среды в 2011 году показал, что для этой выборки характерны высокие корреляции ($k \approx 0,6$) предиктанта с концентрацией нитрат-иона, температурой воды и численностью *P. duplex* с различными временными сдвигами от 20 до 23 дней. Результаты этого анализа использовались для выработки двух видов нечеткого прогноза. Программная реализация всех алгоритмов прогноза осуществлялась в среде MATLAB. Для экстраполяции на срок 2–6 дней использовалось приближение данных полиномом первой степени по методу наименьших квадратов с помощью функции «polyfit». Алгоритм нечеткой логики осуществлен с помощью редактора правил нечеткой логики «fuzzy». Согласно [5] модели гибридных сетей, объединяют достоинства нейронных сетей и систем нечеткого вывода и используют процедуры обучения нейронных сетей на выборках данных с тем, чтоб определить наилучшие параметры функций принадлежности, соответствующие определенной системе нечеткого вывода. Для реализации этого метода разработана специальная программа.

2. Результаты

Результаты прогнозирования тремя методами представлены на рис. 1. Для оценки успешности прогнозов рассчитывали среднюю абсолютную и относительную ошибки методом, изложенным в руководстве по краткосрочным прогнозам погоды [6].

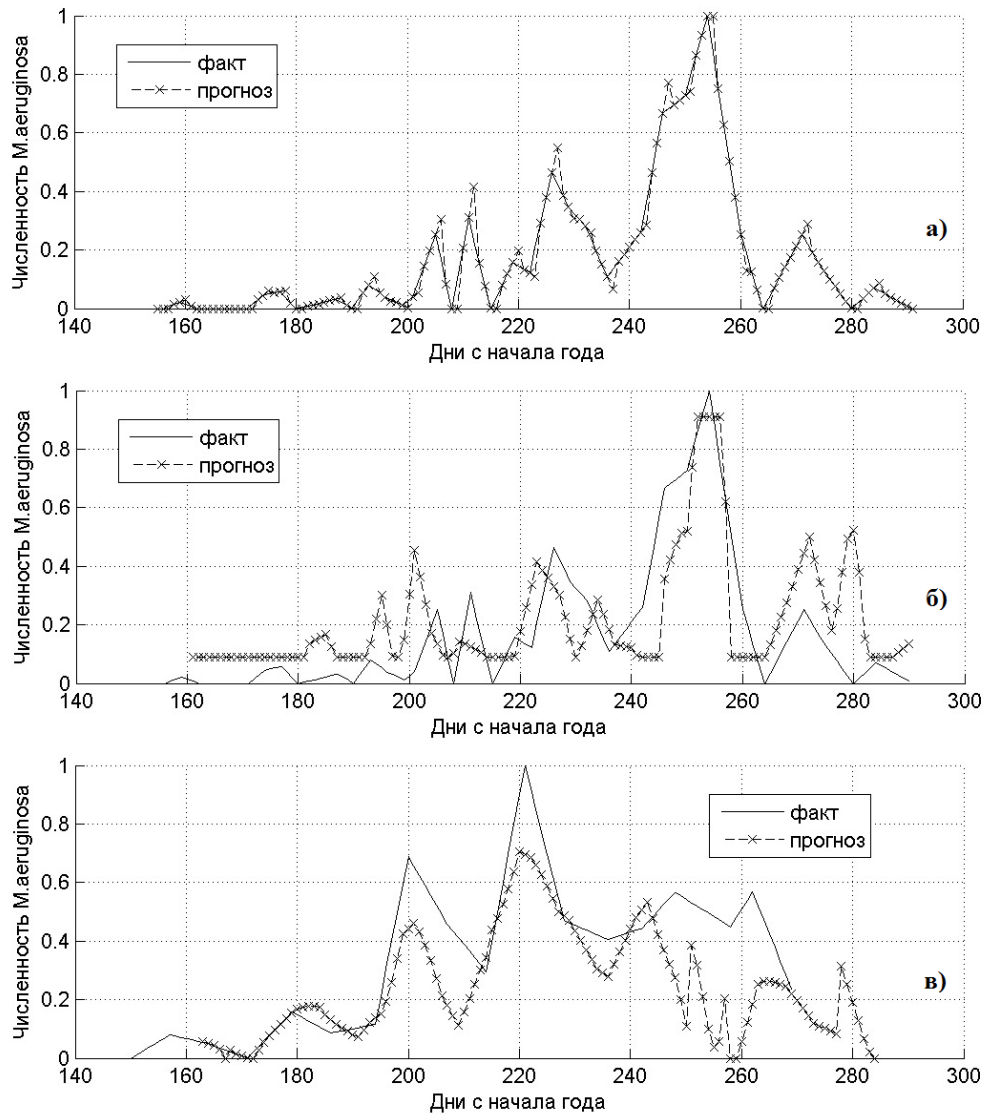


Рис. 1. Результаты прогнозирования численности *M. aeruginosa* различными методами: результаты линейной экстраполяции данных 2009 года (а), прогнозирования по правилам нечеткой логики для 2009 года (б) и (в) прогноза, осуществленного самообучающейся нечеткой нейронной сетью по данным 2011 года

Экстраполяционный прогноз. Экстраполяция ряда численности является самым простым методом прогноза, для применения которого достаточно данных о численности прогнозируемой водоросли и предположения о наличии некоторой инерционности процессов ее увеличения или снижения. Результат линейной экстраполяции данных 2009 года, представлен на рисунке 1 (а). Средняя абсолютная ошибка экстраполяции с использованием последних двух наблюдений на 1 день вперед за два года была в диапазоне 0,6–1,2 %, на 2 дня 3,6–3,7 %, на 3 дня 3,6–7,2 %. Несмотря на то, что результаты экстраполяционного прогноза, судя по значениям абсолютных ошибок кажутся вполне удовлетворительным, очевидно, что при экстраполяции далеко за крайний узел ошибка будет велика. При этом с увеличением числа точек использованных для интерполяции, происходит ухудшение качества прогноза. Так при ис-

пользовании последних 6 наблюдений для экстраполяции на 1 день ошибка достигает 5,3 %, а на 3 дня уже 11 %. Очевидно, что в точках значительного изменения скорости роста численности, и особенно в момент изменения ее знака, экстраполяционный прогноз будет наименее успешным и должен корректироваться. Вопрос заключается в том, возможно ли предугадать момент времени, когда возможен перелом в эволюции численности.

Прогноз с использованием правил нечеткой логики. Выявленные в ходе кросс-корреляционного анализа зависимости, позволили составить правила нечеткой логики, которые легли в основу прогноза численности *M. aeruginosa* для 2009 и 2011 годов. При этом срок прогноза задавался значением сдвига предиктора, определенным в ходе кросс-корреляционного анализа. Так, прогнозирование по данным 2009 года осуществлялось с помощью одного правила: если численность *P. duplex* сегодня «высокая» («низкая»), то численность *M. aeruginosa* через 6 дней будет «высокая» («низкая»). Лингвистические оценки «низкая», «высокая» задавались с помощью редактора функций принадлежности на отрезке [0, 1] и представляют собой множества значений переменных «численность *P. duplex*» и «численность *M. aeruginosa*». Таким образом, заблаговременность прогноза с учетом данных наблюдений за последние сутки достигает 5 дней. Результаты прогнозирования (рис. 1б) демонстрируют угадывание «основных» пиков нормированной численности водоросли. Средняя абсолютная ошибка прогноза для этой модели в 2009 году составила 13 %.

Для выборки 2011 года были использованы три правила: 1) если температура воды сегодня «высокая» («низкая»), то численность *M. aeruginosa* через 20 дней будет высокая («низкая»); 2) Если численность *P. duplex* сегодня «высокая» («низкая»), то численность *M. aeruginosa* через 23 дня будет «высокая» («низкая»); 3) если концентрация нитрат-иона сегодня «высокая» («низкая»), то численность *M. aeruginosa* через 20 дней будет «высокая» («низкая»). Средняя абсолютная ошибка нечеткого прогноза для данных 2011 года составила 15 %.

Прогноз с использованием нечеткой нейронной сети. В отличие от правил нечеткой логики, параметры функций принадлежности в данном случае не задавались «вручную», а определялись автоматически в результате обучения нейронной сети на заранее проанализированных данных. Для обучения нейронной сети использовались пары «вход-выход». В качестве «входа» использовали данные о значении отобранных в ходе кросс-корреляционного анализа параметров с соответствующими сдвигами во времени, а в качестве выхода — данные о численности *M. aeruginosa* на момент прогноза. При этом необходимо заметить, что сдвиги параметров определяют срок прогноза.

Для 2009 года в ходе обучения получены две модели — с учетом минерализации и с учетом численности *P. duplex*. Для 2011 года в ходе обучения получены четыре модели, дающие удовлетворительные результаты. Три однопараметрические с учетом концентрации нитрат-иона; с учетом температуры воды; с учетом численности *P. duplex* и одна двухпараметрическая с учетом численности *P. duplex* и температуры воды. Лучшие результаты со средней ошибкой прогноза 13 % дает модель с двумя параметрами (см. рис. 1в). Как видно из рис. 1в, модель на основе нейро-нечеткой сети не позволяет точно оценить численность водорослей, но достаточно уверенно определяет время и интенсивность максимальной вспышки численности с заблаговременностью до 20 дней. Очевидно, что относительно невысокие результа-

ты прогноза с использованием самообучающейся нейро-нечеткой сети обусловлены короткой длиной выборкой, недостаточной для ее полноценного обучения. С другой стороны, модель требует очень тонкой настройки и тщательного выбора индикаторных факторов. По мере накопления новых данных о состоянии среды и численности водорослей, некоторые факторы будут отбраковываться, другие проявляться.

3. Заключение

Проведенная оценка результатов краткосрочного прогнозирования численности *M. aeruginosa*, методами экстраполяции и методами основанными на правилах нечеткой логики и нейронных сетях показала, что каждый из них обладает своими достоинствами и недостатками. Оценка качества прогнозов на один-два дня вперед выполненных различными методами, показала, что экстраполяция хорошо работает на квазилинейных интервалах изменения численности. С другой стороны, при увеличении средних ошибок прогноза на малые сроки методы с использованием прогнозных правил с большей заблаговременностью предсказывают времена и амплитуды вспышек численности. Очевидно, что искомые правила нечетких методов о связи численности предиктанта с параметрами окружающей среды должны быть уточнены и дополнены после накопления дополнительных натуральных наблюдений. Однако уже на данном этапе это не исключает применения этого метода для оптимизации работы очистных сооружений водопровода в летний период, когда ежедневно производится анализ целого спектра биологических и гидрохимических показателей, необходимых для настройки методов.

Работа выполнена при финансовой поддержке Министерства образования и науки Российской Федерации в рамках федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009-2013 годы (Соглашение № 14.В37.21.0613).

Литература

1. Marsili-Libelli, S. Fuzzy Prediction of the Algal Blooms in the Orbetello Lagoon / S. Marsili-Libelli, G. Pacini, C. Barresi // Environmental Modeling & Software 19. – 2004. – P. 799–808.
2. Oh, H.-M. Community Patterning and Identification of Predominant Factors in Algal Bloom in Daechung Reservoir (Korea) Using Artificial Neural Networks / H.-M. Oh, Ch.-Y. Ahn, J.-W. Lee, T.-S. Chon, K.H. Choi, Y.-S. Park // Ecological modeling. – 2007. – Vol. 203. – P. 109–118.
3. Гаязова, А.О. Исследование сезонной динамики массовых видов фитопланктона озера Смолино / А.О. Гаязова, С.М. Абдуллаев // Вестник Челябинского государственного университета. Экология и природопользование. – 2011. – № 5(220). Вып. 5. – С. 70–75.
4. Гаязова, А.О. Оценка факторов среды, определяющих массовое развитие *Microcystis aeruginosa* в озере Смолино (г. Челябинск) / А.О. Гаязова // Биоиндикация в мониторинге пресноводных экосистем II. Сборник материалов международной конференции. – СПб.: Любавич, 2011. – С. 114–118.

5. Леоненков, А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH / А.В. Леоненков. – СПб.: БХВ-Петербург, 2005. – 736 с.
6. Руководство по краткосрочным прогнозам погоды. / под. ред. С.Л. Белоусова, А.А. Васильева, Н.Ф. Вельтищева и др. – Л.: Гидрометеоиздат, 1986. – Ч. I. – 702 с.

Гаязова Анна Олеговна, гидробиолог, муниципальное предприятие «Производственное объединение водоснабжения и водоотведения», why.ann@mail.ru.

Абдуллаев Санжар Муталович, доктор географических наук, профессор кафедры вычислительной математики, факультет Вычислительной математики и информатики, Южно-Уральский государственный университет, sanjar@mail.ru.

FORECASTING OF MICROCYSTIS AERUGINOSA SEASONAL DYNAMICS USING THE FUZZY LOGIC AND FUZZY NEURAL NETWORKS

A.O. Gayazova, municipal enterprise “Production association of water supply and distribution” (Chelyabinsk, Russian Federation),

S.M. Abdullaev, South Ural State University (Chelyabinsk, Russian Federation)

The retrieval of potential predictors of blue-green algae *M. aeruginosa* blooms and bloom prediction using fuzzy logic rules and fuzzy neural networks is discussed. Time series of seasonal dynamics of *M. aeruginosa* quantities and parameter values were obtained through field observations of biotic and abiotic parameters of the water environment held at Lake Smolino (Chelyabinsk) in the warm season of 2009 and 2011. The cross-correlation analysis of the data revealed the potential predictors of *M. aeruginosa* abundance quasi-periodic oscillations with a period of 12-20 days: algae *P. duplex* abundance, water temperature and the concentration of nitrates. According to the results of cross-correlation analysis a number of rules and membership functions in a range of changes from zero to one is set forward. Specially written program was used to train the fuzzy neural network in data on the values of predictant and selected predictors to apply the predictive rules and membership functions automatically. To compare the results additionally performed a linear extrapolation of the predictant abundance. Seasonal development of *M. aeruginosa* was well predicted by the extrapolation forecast only on quasilinear intervals of *M. aeruginosa* abundance evolution, whereas fuzzy logic theory was good to predict the *M. aeruginosa* intense outbreaks.

Keywords: *M. aeruginosa*, algae bloom forecasting, quasi-periodic oscillations, linear extrapolation, fuzzy logic, fuzzy artificial neural networks.

References

1. Marsili-Libelli S., Pacini G., Barresi. C. Fuzzy Prediction of the Algal Blooms in the Orbetello Lagoon. Environmental Modeling & Software, 2004. Vol. 19, P. 799–808.
2. Oh H.-M., Ahn Ch.-Y., Lee J.-W., Chon T.-S., Choi K.H., Park Y.-S. Community Patterning and Identification of Predominant Factors in Algal Bloom in Daechung Reservoir (Korea) Using Artificial Neural Networks. Ecological modeling. 2007. Vol. 203, P. 109–118

3. Gayazova A.O., Abdullaev S.M. Issledovanie sezonnoj dinamiki massovyh vidov fitoplanktona ozera Smolino [Study of Dominant Algae Species Seasonal Abundance in Smolino Lake]. Vestnik Cheliabinskogo gosudarstvennogo universiteta. Seriya: Ekologiya i Prirodopolzovanie [Bulletin of Chelyabinsk State University. Series: Ecology and Nature management]. 2011. Vol. 220, No. 5. P. 70–75.
4. Gayazova A.O. Otsenka faktorov sredi opredeliaushih massovoe razvitie Microcystis aeruginosa v ozere Smolino (Chelyabinsk) [Evaluation of Environmental Factors that Determine the Mass Development of Microcystis aeruginosa in Lake Smolino (Chelyabinsk)]. Bioindikatsia v monitoringe presnovodnyh ekosistem II. Sbornik materialov mezhdunarodnoj konferentsii [Bioindication in Monitoring of Fresh Water Ecosystems II. Book of Papers of the International Conference]. St. Petersburg, "Lubavich", 2011. P. 114–118.
5. Leonenkov A.V. Nechetkoe modelirovanie v srede MATLAB i fuzzyTECH [Fuzzy Modeling with MATLAB and FuzzyTECH]. St. Petersburg, BHV-Petersburg, 2005. 736 p.
6. Rukovodstvo po kratkosrochnym prognozam pogody. Chast 1. [Guidance on Short-term Weather Forecasts. Part 1]. Edited by S.L. Belousov, A.A. Vasilyev, N.F. Veltishev et al. Leningrad, Gidrometeoizdat, 1986. 702 p.

Поступила в редакцию 5 ноября 2012 г.

О ЕДИНСТВЕННОСТИ РЕШЕНИЯ ОБРАТНОЙ ЗАДАЧИ НЕСТАЦИОНАРНОЙ ФИЛЬТРАЦИИ

А.В. Бокков

При определении коэффициента гидропроводности нефтяного пласта методом гидродинамического прослушивания скважин возникает необходимость решения обратной задачи фильтрации. При этом важно поставить задачу таким образом, чтобы обеспечить единственность решения. В статье определяются условия, достаточные для единственности решения обратной задачи.

Ключевые слова: обратная задача фильтрации, преобразование Лапласа, задача Штурма – Лиувилля.

Введение

Исследованию обратных коэффициентных задач посвящено большое количество работ (см., например, [1, 2]). Особое внимание при этом уделяется доказательству сходимости метода решения и формулировке условий, обеспечивающих существование и единственность решения.

В данной работе формулируются такие условия для обратной задачи фильтрации со смешанными граничными условиями для дифференциального оператора более общего вида, чем в [1].

1. Постановка прямой задачи

Процесс нестационарной фильтрации жидкости к одиночной скважине в осесимметричном случае описывается уравнением

$$\frac{\partial p}{\partial t} = \frac{1}{\rho} \frac{\partial}{\partial \rho} \left[\rho \sigma(\rho) \frac{\partial p}{\partial \rho} \right], \quad (1)$$

где $p = p(\rho, t)$ – давление в пласте, $\sigma = \sigma(\rho)$ – коэффициент гидропроводности.

Решение уравнения (1) будем искать в области изменения переменных (ρ, t) : $t \geq 0$, $0 < r_0 \leq \rho \leq \bar{r}$.

Сделаем следующие предположения:

а) известно начальное давление в пласте

$$p(\rho, 0) = p_0; \quad (2)$$

б) на границе пласта выполняется условие «непротекания»

$$\frac{\partial p(\bar{r}, t)}{\partial \rho} = 0, \quad t \geq 0; \quad (3)$$

в) известно забойное давление

$$p(r_0, t) = f_1(t), \quad t \geq 0. \quad (4)$$

Предположим, что коэффициент гидропроводности $\sigma(\rho)$ удовлетворяет условию

$$\sigma(\rho) \geq d > 0 \text{ при } \rho \in [r_0, \bar{r}]. \quad (5)$$

Задачу (1) – (4) называют прямой задачей фильтрации. При известной функции $\sigma(\rho)$, удовлетворяющей условию (5), и при дополнительных предположениях о гладкости функций $\sigma(\rho)$, $f_1(t)$ и $p(\rho, t)$ эта задача имеет единственное решение.

2. Постановка обратной задачи

Обратная задача заключается в определении неизвестного коэффициента $\sigma(\rho)$ в уравнении (1) по дополнительной информации о решении задачи (1) – (4).

Предположим, что нам известен дебит скважины

$$\frac{\partial p(r_0, t)}{\partial \rho} = g(t), \quad (6)$$

где $g(t)$ – ограниченная и непрерывная функция, $t \geq 0$.

Так как при неизвестной функции $\sigma(\rho)$ решение $p(\rho, t)$ задачи (1) – (4) также неизвестно, то обратную задачу сформулируем как задачу определения двух функций $\sigma(\rho)$ и $p(\rho, t)$, удовлетворяющих условиям (1) – (6).

Сделав в задаче (1) – (6) замену переменной $u(\rho, t) = p(\rho, t) - p_0$, перейдем к новой задаче

$$\frac{\partial u}{\partial t} = \frac{1}{\rho} \frac{\partial}{\partial \rho} \left[\rho \sigma(\rho) \frac{\partial u}{\partial \rho} \right], \quad (7)$$

$$u(\rho, 0) = 0, \quad (8)$$

$$\frac{\partial u(\bar{r}, t)}{\partial \rho} = 0, \quad (9)$$

$$u(r_0, t) = f(t), \quad (10)$$

$$\frac{\partial u(r_0, t)}{\partial \rho} = g(t), \quad (11)$$

где $0 < r_0 \leq \rho \leq \bar{r}$, $t \geq 0$, $f(t) = f_1(t) - p_0$.

Будем предполагать, что функция $f(t) \in C^2[0, \infty)$ удовлетворяет условиям

$$f(0) = f'(0) = 0, \quad f(t) = 0 \text{ при } t \geq t_0, \quad f(t) \neq 0 \text{ при } t \geq 0, \quad (12)$$

а функция $\sigma(\rho)$ удовлетворяет условию (5) и

$$\sigma(\rho) \in C^2[r_0, \bar{r}]. \quad (13)$$

Определение 1. Решением обратной задачи (7) – (11) назовем пару функций $\sigma(\rho)$ и $u(\rho, t)$ таких, что $\sigma(\rho)$ удовлетворяет условиям (5) и (13), $u(\rho, t)$ имеет непрерывные частные производные $(u(\rho, t), u'_\rho(\rho, t), u''_{\rho\rho}(\rho, t), u'_t(\rho, t) \in C[(r_0, \bar{r}) \times (0, \infty)])$,

функции $\sigma(\rho)$ и $u(\rho, t)$ удовлетворяют уравнению (7), а функция $u(\rho, t)$ удовлетворяет условиям (8) – (11).

Лемма 1. Если выполнены сформулированные выше условия, то при $t \rightarrow \infty$ $u(\rho, t) \rightarrow 0$, $u'_\rho(\rho, t) \rightarrow 0$, $u'_t(\rho, t) \rightarrow 0$ и $\frac{1}{\rho} \frac{\partial}{\partial \rho} [\rho \sigma(\rho) \frac{\partial u(\rho, t)}{\partial \rho}] \rightarrow 0$ равномерно на отрезке $[r_0, \bar{r}]$.

Доказательство. Пусть $\sigma(\rho)$ и $u(\rho, t)$ – решение обратной задачи (7) – (11). Тогда ввиду того, что $f(t) = 0$ при $t \geq t_0$, функция $u(\rho, t)$ для $t \geq t_0$ является решением следующей задачи:

$$\frac{\partial u}{\partial t} = \frac{1}{\rho} \frac{\partial}{\partial \rho} \left[\rho \sigma(\rho) \frac{\partial u}{\partial \rho} \right], \quad 0 < r_0 \leq \rho \leq \bar{r}, \quad t \geq t_0,$$

$$u(r_0, t) = u'_\rho(\bar{r}, t) = 0, \quad t \geq t_0,$$

$$u(\rho, t_0) = \psi(\rho), \quad r_0 \leq \rho \leq \bar{r},$$

где $\psi(\rho) \in C^1[r_0, \bar{r}]$, $\psi(r_0) = \psi'(\bar{r}) = 0$.

Используя для решения этой задачи метод разделения переменных ($u(\rho, t) = y(\rho)v(t)$), получим, что

$$u(\rho, t) = \sum_{n=0}^{\infty} \left(\int_{r_0}^{\bar{r}} \psi(\xi) y_n(\xi) \xi d\xi \right) e^{-\lambda_n(t-t_0)} y_n(\rho), \quad (14)$$

где $\lambda_n \geq 0$ и $y_n(\rho)$ – собственные значения и нормированные собственные функции задачи Штурма-Лиувилля

$$\begin{aligned} [\rho \sigma(\rho) y'(\rho)]' + \lambda \rho y(\rho) &= 0, \\ y(r_0) = y'(\bar{r}) &= 0. \end{aligned} \quad (15)$$

Покажем, что собственные значения $\lambda_n > 0$ при $n \geq 0$. Так как предполагается, что $\lambda_n \geq 0$ упорядочены по возрастанию, то достаточно показать, что $\lambda_0 > 0$ ($\lambda_0 \neq 0$).

Предположим противное, $\lambda_0 = 0$. Тогда

$$(\rho \sigma(\rho) y'_0(\rho))' = 0, \quad (16)$$

а

$$y_0(r_0) = 0, \quad y'_0(\bar{r}) = 0. \quad (17)$$

Из (16) следует, что

$$\rho \sigma(\rho) y'_0(\rho) = C_1,$$

и второе из условий (17) влечет $C_1 = 0$. Поэтому

$$y'_0(\rho) = 0, \quad (18)$$

и значит,

$$y_0(\rho) = C_2.$$

Используя первое из условий (17), получим, что $C_2 = 0$, а $y_0(\rho) \equiv 0$, что противоречит условию нормировки собственных функций $y_n(\rho)$. Таким образом, $\lambda_n > 0$ для $n \geq 0$.

Из положительности собственных значений λ_n и разложения (14) следует утверждение леммы. \square

Введем функцию $V(\rho, s)$, являющуюся преобразованием Лапласа от $u(\rho, t)$ по переменной t

$$V(\rho, s) = \int_0^{\infty} e^{-st} u(\rho, t) dt.$$

На основании леммы 1 из (7) – (10) следует, что для значений комплексного параметра s таких, что $\operatorname{Re} s \geq s_0 > 0$, функция $V(\rho, s)$ является решением краевой задачи

$$\frac{1}{\rho} [\rho \sigma(\rho) V'_\rho]'_\rho - sV = 0, \quad (19)$$

$$V'_\rho(\bar{r}, s) = 0, \quad (20)$$

$$V(r_0, s) = \varphi(s), \quad (21)$$

где $\varphi(s) = \int_0^{\infty} e^{-st} f(t) dt$. Кроме того функция $V(\rho, s)$ удовлетворяет дополнительному условию

$$V'_\rho(r_0, s) = \mu(s), \quad (22)$$

где $\mu(s) = \int_0^{\infty} e^{-st} g(t) dt$.

Рассмотрим функцию $W(\rho, s)$, являющуюся решением задачи Коши для уравнения (19) с краевыми условиями

$$W(\bar{r}, s) = 1, \quad (23)$$

$$W'_\rho(\bar{r}, s) = 0.$$

Так как функции $V(\rho, s)$ и $W(\rho, s)$ являются решениями дифференциального уравнения (19) и $V'_\rho(\bar{r}, s) = W'_\rho(\bar{r}, s)$, то из теоремы [3, с. 179] следует, что они линейно зависимые, то есть

$$V(\rho, s) = C(s) W(\rho, s). \quad (24)$$

Тогда из (21) и (24) следует, что

$$\varphi(s) = C(s) W(r_0, s), \quad (25)$$

а из (22) и (24) следует, что

$$\mu(s) = C(s) W'_\rho(r_0, s). \quad (26)$$

Из последнего равенства можно выразить $C(s)$:

$$C(s) = \frac{\mu(s)}{W'_\rho(r_0, s)}. \quad (27)$$

Лемма 2. При сформулированных выше условиях множество нулей функции $W(r_0, s)$ не пересекается с множеством нулей функции $W'_\rho(r_0, s)$.

Доказательство. Предположим противное, то есть найдется s_0 такое, что

$$W(r_0, s_0) = W'_\rho(r_0, s_0) = 0. \quad (28)$$

Тогда $W(\rho, s_0)$ является решением задачи Коши для уравнения (19) и начальными условиями (28). Следовательно, для любого $\rho \in [r_0, \bar{r}]$ $W(\rho, s_0) = 0$, что противоречит условию $W(\bar{r}, s_0) = 1$. \square

Рассмотрим дифференциальное уравнение

$$-\frac{1}{\rho} [\rho\sigma(\rho) y'(\rho)]' = \lambda y(\rho). \quad (29)$$

с краевыми условиями

$$y(r_0) = 0, \quad y'(\bar{r}) = 0 \quad (30)$$

или

$$y'(r_0) = 0, \quad y'(\bar{r}) = 0. \quad (31)$$

Лемма 3. При сформулированных выше условиях значения $s = s_0$ и $s = \bar{s}_0$ являются, соответственно, нулями функций $W(r_0, s)$ и $W'_\rho(r_0, s)$ тогда и только тогда, когда числа $\lambda_0 = -s_0$ и $\bar{\lambda}_0 = -\bar{s}_0$ являются собственными значениями задач Штурма-Лиувилля (29), (30) и (29), (31).

Доказательство. Так как функция $W(\rho, s_0)$ является решением уравнения (19) с $s = s_0$, то функция $y_0(\rho) = W(\rho, s_0)$ является решением уравнения (29) с $\lambda_0 = -s_0$. Второе краевое условие в (30) для $y_0(\rho)$ следует из (23), а первое из того, что s_0 является нулем функции $W(r_0, s)$. Таким образом, $y_0(\rho)$ является решением задачи (29), (30). Это решение нетривиальное, так как в силу (23) $y_0(\bar{r}) = W(\bar{r}, s_0) = 1$. Следовательно, λ_0 является собственным значением задачи (29), (30).

Очевидно, что справедливо и обратное утверждение. Действительно, если λ_0 является собственным значением, а $y_0(\rho)$ – собственной функцией задачи (29), (30), то на основании первого из условий (30) значение $s_0 = -\lambda_0$ является нулем функции $W(r_0, s)$.

Теперь предположим, что $s = \bar{s}_0$ является нулем функции $W'_\rho(r_0, s)$. Тогда из того, что функция $W(\rho, \bar{s}_0)$ является решением уравнения (19) с $s = \bar{s}_0$, следует, что функция $\bar{y}_0(\rho) = W(\rho, \bar{s}_0)$ является решением уравнения (29) с $\bar{\lambda}_0 = -\bar{s}_0$. Второе краевое условие в (31) следует из (23), а первое – из того, что \bar{s}_0 является нулем функции $W'_\rho(r_0, s)$.

Таким образом, $\bar{y}_0(\rho)$ является решением задачи (29), (31). Это решение нетривиальное, так как в силу (23) $\bar{y}_0(\bar{r}) = W(\bar{r}, \bar{s}_0) = 1$. Следовательно, $\bar{\lambda}_0$ является собственным значением задачи Штурма-Лиувилля (29), (31).

Обратно, если $\bar{\lambda}_0$ является собственным значением, а $\bar{y}_0(\rho)$ – собственной функцией задачи (29), (31), то $\bar{s}_0 = -\bar{\lambda}_0$ на основании первого из условий (31) есть ноль функции $W'_\rho(r_0, s)$. \square

Предположим теперь, что функция $\sigma(\rho) \in C^2[r_0, \bar{r}]$ удовлетворяет условию (5), и $\sigma'(\bar{r}) = 0$. Вернемся к решению $W(\rho, s)$ задачи Коши (23) для уравнения (19).

Лемма 4. При сформулированных выше условиях на функцию $\sigma(\rho)$ функции $W(\rho, s)$ и $W'_\rho(\rho, s)$ для каждого фиксированного $\rho \in [r_0, \bar{r}]$ являются целыми функциями комплексного параметра s .

Доказательство. Применяя преобразование Лиувилля ([4], с.35) к уравнению (19), заменим функцию $W(\rho, s)$ на функцию $z(x, s)$, определяемую параметрически следующим образом:

$$x = \frac{1}{c} \int_{r_0}^{\rho} \frac{d\xi}{\sqrt{\sigma(\xi)}}, \quad (32)$$

где

$$c = \frac{1}{\pi} \int_{r_0}^{\bar{r}} \frac{d\xi}{\sqrt{\sigma(\xi)}}, \quad (33)$$

а x – новая независимая переменная, и

$$z(\rho, s) = \sqrt{\rho} \sigma^{\frac{1}{4}}(\rho) W(\rho, s). \quad (34)$$

Подставляя выражения (32) – (34) в уравнение (19), приведем его к уравнению

$$\frac{\partial^2 z}{\partial x^2} + a(x) z - c^2 s z = 0, \quad 0 \leq x \leq \pi, \quad (35)$$

где c задается формулой (33), а

$$a(x) = -\frac{1}{\theta(x)} \frac{d^2 \theta(x)}{dx^2}, \quad (36)$$

где $\theta = \theta(x)$ определена параметрически формулой (32) и равенством

$$\theta(\rho) = \sqrt{\rho} \sigma^{\frac{1}{4}}(\rho). \quad (37)$$

Теперь сделаем замену в краевых условиях (23):

$$z(\pi, s) = \sqrt{\bar{r}} \sigma^{\frac{1}{4}}(\bar{r}) \quad (38)$$

и

$$z'_x(\pi, s) = \frac{c}{2\sqrt{\bar{r}}} \sigma^{\frac{3}{4}}(\bar{r}). \quad (39)$$

Далее введем новую переменную $\tau = \pi - x$ и перейдем от функции $z(x, s)$ к функции $w(\tau, s)$, которую определим формулой

$$w(\tau, s) = \frac{1}{\sqrt{\bar{r}} \sqrt{\sigma(\bar{r})} + \frac{c^2 \sigma^{\frac{3}{2}}(\bar{r})}{4\bar{r}}} z(\pi - \tau, s). \quad (40)$$

Используя формулу (40), легко проверить, что функция $w(\tau, s)$ удовлетворяет уравнению

$$\frac{\partial^2 w}{\partial \tau^2} + a(\pi - \tau) w - c^2 s w = 0, \quad 0 \leq \tau \leq \pi. \quad (41)$$

При этом условия (38) и (39) для функции $w(\tau, s)$ будут выглядеть следующим образом:

$$w(0, s) = \sin \alpha, \quad (42)$$

а

$$w'_\tau(0, s) = -\cos \alpha, \quad (43)$$

где

$$\alpha = \arcsin \left[\frac{\sqrt{\bar{r}} \sigma^{\frac{1}{4}}(\bar{r})}{\sqrt{\bar{r} \sqrt{\sigma(\bar{r})} + \frac{c^2 \sigma^{\frac{3}{2}}(\bar{r})}{4\bar{r}}}} \right]. \quad (44)$$

Из теоремы [5, с. 14–15] следует, что для каждого фиксированного $\tau \in [0, \pi]$ функции $w(\tau, s)$ и $w'_\tau(\tau, s)$ являются целыми функциями s , а из формул (34) и (40) будет следовать утверждение леммы. \square

Так как функция $W(\rho, s)$ представляет собой решение уравнения (19) с условиями (23), то из леммы 4 следует, что $W(\rho, s)$ и $W'_\rho(\rho, s)$ при фиксированном ρ являются аналитическими функциями комплексного переменного s во всей комплексной плоскости.

В дальнейшем всюду будем предполагать, что $\sigma(\rho) \in C^2[r_0, \bar{r}]$, $\sigma'(\bar{r}) = 0$ и выполняется условие (5).

Пусть $\sigma_i(\rho)$ и $u_i(\rho, t)$ ($i = 1, 2$) – решения обратной задачи (7) – (11). Обозначим через $V_i(\rho, s)$ преобразование Лапласа от $u_i(\rho, t)$, а через $W_i(\rho, s)$ – решения задачи Коши для уравнения (19) с $\sigma(\rho) = \sigma_i(\rho)$ и начальными условиями (23). Для краткости $(W_i(\rho, s))'_\rho|_{\rho=r_0}$ будем обозначать $W'_i(r_0, s)$.

Из (10) следует, что $V_1(r_0, s) = V_2(r_0, s)$. Тогда, используя формулы (24) и (27), получим, что при $Re s \geq s_0 > 0$

$$\frac{W_1(r_0, s)}{W'_1(r_0, s)} = \frac{W_2(r_0, s)}{W'_2(r_0, s)}. \quad (45)$$

Из леммы 4 следует, что функции $W_i(r_0, s)/W'_i(r_0, s)$ при $i = 1, 2$ являются аналитическими функциями комплексного переменного s во всей комплексной плоскости за исключением нулей $W'_i(r_0, s)$, являющихся особыми точками.

Из (45) следует, что нули и особые точки функций $W_1(r_0, s)/W'_1(r_0, s)$ и $W_2(r_0, s)/W'_2(r_0, s)$ совпадают.

Используя лемму 2, окончательно получим, что все нули функций $W_1(r_0, s)$ и $W_2(r_0, s)$ совпадают, и все нули функций $W'_1(r_0, s)$ и $W'_2(r_0, s)$ также совпадают.

Таким образом, на основании леммы 3 для любого n выполняются соотношения

$$\lambda_n^1 = \lambda_n^2 \quad (46)$$

и

$$\bar{\lambda}_n^1 = \bar{\lambda}_n^2, \quad (47)$$

где $\{\lambda_n^i\}$ при $i = 1, 2$ – все собственные значения задачи Штурма – Лиувилля (29) – (30) с $\sigma(\rho) = \sigma_i(\rho)$, упорядоченные по возрастанию, а $\{\bar{\lambda}_n^i\}$ – все собственные значения задачи (29), (31) с $\sigma(\rho) = \sigma_i(\rho)$, также упорядоченные по возрастанию.

Теорема 1. *Предположим, что функция $f(t)$ удовлетворяет условиям (12). Тогда, если $\sigma_i(\rho)$ и $u_i(\rho, t)$, $i = 1, 2$ – решения обратной задачи (7)–(11) такие, что $\sigma'_1(\bar{r}) = \sigma'_2(\bar{r}) = \sigma'_1(r_0) = \sigma'_2(r_0) = 0$, $\int_{r_0}^{\bar{r}} \frac{d\xi}{\sqrt{\sigma_1(\xi)}} = \int_{r_0}^{\bar{r}} \frac{d\xi}{\sqrt{\sigma_2(\xi)}}$, а также $\sigma_1(r_0) = \sigma_2(r_0) = \sigma_1(\bar{r}) = \sigma_2(\bar{r})$, и значение $\sigma(r_0)$ нам известно, то $\sigma_1(\rho) = \sigma_2(\rho)$ для $\rho \in [r_0, \bar{r}]$ и $u_1(\rho, t) = u_2(\rho, t)$ для $\rho \in [r_0, \bar{r}]$, $t \geq 0$*

Доказательство. Сделаем в уравнении (29) замену переменных Лиувилля (32), (33) и

$$z(\rho) = \sqrt{\rho} \sigma^{\frac{1}{4}}(\rho) y(\rho), \quad (48)$$

перейдем от функции $y_i(\rho, \lambda)$ к функции $z_i(x, \lambda)$, удовлетворяющей уравнению

$$-\frac{\partial^2 z_i}{\partial x^2} + q_i(x) z_i = c^2 \lambda z_i, \quad 0 \leq x \leq \pi, \quad i = 1, 2, \quad (49)$$

где функции $q_i(x) = -a_i(x)$ определяются формулами (32), (33), (36), (37) и являются непрерывными на отрезке $[0, \pi]$. Сделаем еще одну замену

$$w_i(\tau, \lambda) = z_i(\pi - \tau, \lambda), \quad \tau = \pi - x, \quad 0 \leq \tau \leq \pi, \quad i = 1, 2,$$

получим уравнение для функции $w_i(\tau, \lambda)$

$$-\frac{\partial^2 w_i}{\partial \tau^2} + q_i(\pi - \tau) w_i = c^2 \lambda w_i, \quad 0 \leq \tau \leq \pi, \quad i = 1, 2. \quad (50)$$

Теперь преобразуем граничные условия (30) и (31). Получим, что условиям (30) соответствует пара граничных условий

$$w_i(\pi, \lambda) = 0, \quad i = 1, 2, \quad (51)$$

$$w_i(0, \lambda) \cos(\beta) - w'_i(0, \lambda) \sin \beta = 0, \quad i = 1, 2, \quad (52)$$

где $\beta = \arcsin \left(1 / \sqrt{1 + \left(\frac{c}{2\bar{r}} \right)^2 \sigma(\bar{r})} \right)$ и $0 < \beta < \frac{\pi}{2}$.

Условиям (31) соответствуют граничные условия (52) и

$$w_i(\pi, \lambda) \cos(\gamma) - w'_i(\pi, \lambda) \sin \gamma = 0, \quad i = 1, 2, \quad (53)$$

где $\gamma = \arcsin \left(1 / \sqrt{1 + \left(\frac{c}{2r_0} \right)^2 \sigma(r_0)} \right)$ и $0 < \gamma < \frac{\pi}{2}$.

Таким образом, задача Штурма – Лиувилля (50) – (52) порождает возрастающую последовательность собственных значений $\{\mu_n^i\}$. При этом из формулы (46) и условий теоремы следует, что для любого n

$$\mu_n^1 = \mu_n^2. \quad (54)$$

Аналогично, задача Штурма – Лиувилля (50), (52) и (53) порождает возрастающую последовательность собственных значений $\{\bar{\mu}_n^i\}$ такую, что на основании (47) и условий теоремы следует, что для любого n

$$\bar{\mu}_n^1 = \bar{\mu}_n^2. \quad (55)$$

Из теоремы, приведенной в [6], и равенств (54) и (55) следует, что для любого $x \in [0, \pi]$

$$a_1(x) = a_2(x). \quad (56)$$

Таким образом, обозначив $a_i(x)$ через $a(x)$ и воспользовавшись формулами (36) и (37), получим, что

$$\frac{d^2\theta_i(x)}{dx^2} + a(x)\theta_i(x) = 0, \quad i = 1, 2, \quad (57)$$

где

$$\theta_i(0) = \sqrt{r_0} \sigma^{\frac{1}{4}}(r_0) \quad (58)$$

и

$$\theta'_i(0) = \frac{c}{2\sqrt{r_0}} \sigma^{\frac{3}{4}}(r_0). \quad (59)$$

Задача Коши (57) – (59) для линейного дифференциального уравнения второго порядка имеет единственное решение, и, следовательно,

$$\theta_1(x) = \theta_2(x).$$

Используя формулу (37), получаем, что

$$\sigma_1(\rho) = \sigma_2(\rho) \text{ при } \rho \in [r_0, \bar{r}].$$

Но тогда и решения прямой задачи (7) – (10) при соответствующих ограничениях на функции $u_i(\rho, t)$ и $f(t)$ будут при любых значениях $\rho \in [r_0, \bar{r}]$ и $t \geq 0$ удовлетворять условиям

$$u_1(\rho, t) = u_2(\rho, t),$$

что и доказывает теорему. □

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (грант p_урал_a №10-01-96000).

Литература

1. Денисов, А.М. Введение в теорию обратных задач / А.М. Денисов. – М.: Изд-во МГУ, 1994. – 208 с.
2. Кабанихин, С.И. Обратные и некорректные задачи / С.И. Кабанихин. – Новосибирск: Сибирское научное издательство, 2009. – 457 с.
3. Степанов, В.В. Курс дифференциальных уравнений / В.В. Степанов. – М.: Редакция технико-теоретической литературы, 1938. – 376 с.
4. Мартыненко, Н.А. Конечные интегральные преобразования и их применение к исследованию систем с распределенными параметрами / Н.А. Мартыненко, Л.М. Пустыльников. – М.: Наука, 1986. – 304 с.
5. Левитан, Б.М. Введение в спектральную теорию / Б.М. Левитан, И.С. Саргосян. – М.: Наука, 1970. – 672 с.
6. Levinson, N. The inverse Sturm–Liouville problem / N. Levinson // Math. Tidsskr. Ser. B. – 1949. – P. 25–30.

Александр Викторович Боков, старший преподаватель, кафедры вычислительной математики, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), bokov@susu.ac.ru.

ON THE UNIQUENESS OF THE INVERSE PROBLEM OF UNSTEADY FILTRATION

A. V. Bokov, South Ural State University (Chelyabinsk, Russian Federation)

When determining the coefficient of the hydraulic conductivity of oil layer by the method of the hydrodynamic listening of mining holes well is necessary to solve the inverse task of filtration. Thus it is important to set the task so that to provide the uniqueness of the decision. In this article sufficiency conditions for the uniqueness of the inverse problem are defined.

Keywords: inverse problem of filtration, Laplace transform, Sturm-Liouville problem.

References

1. Denisov A.M. Vvedenie v teoriyu obratnykh zadach [Introduction to the Theory of Inverse Problems]. Moscow, Publishing of the Moscow State University, 1994. 208 P.
2. Kabanikhin S.I. Obratnye i hekorrektnye zadachi [Inverse and Ill-posed Problems]. Novosibirsk, Science Press, 2009. 457 P.
3. Stepanov V.V. Kurs differentsialnykh uravnenij [Course on Differential Equations]. Moscow, Publishing house of technical and theoretical literature, 1938. 376 P.
4. Martynenko N.A., Pustyl'nikov L.M. Konechnye integralnye preobrazovaniya i ikh primeneniye k issledovaniyu sistem s raspredelyonnymi parametrami [Finite Integral Transformations and their Application to the Study of Systems with Distributed Parameter]. Moscow, Publishing House "Nauka", 1986. 304 P.
5. Levitan B.M., Sargosyan I.S. Vvedenie v spektralnuyu teoriyu [Introduction to Spectral Theory]. Moscow, Publishing House "Nauka", 1970. 672 P.
6. Levinson, N. The Inverse Sturm–Liouville Problem. // Math. Tidsskr. Ser. B., 1949. P. 25–30.

Поступила в редакцию 8 ноября 2012 г.

МЕТОДЫ ПАРАЛЛЕЛЬНОГО РЕШЕНИЯ СЛАУ НА СИСТЕМАХ С РАСПРЕДЕЛЕННОЙ ПАМЯТЬЮ В БИБЛИОТЕКЕ KRYLOV¹

Д.С. Бутюгин, В.П. Ильин, Д.В. Первозкин

Рассматривается подход к созданию итерационного black-box («черного ящика») параллельного решателя, использованный в библиотеке Krylov для систем линейных алгебраических уравнений (СЛАУ) с разреженными матрицами высокого порядка, возникающими при сеточных аппроксимациях многомерных краевых задач и представленными в сжатом строчном формате CSR. Предлагается вариант алгебраической одномерной декомпозиции СЛАУ. Алгоритм основан на обходе в ширину графа матрицы системы и позволяет привести ее к блочно-трехдиагональному виду. За основу алгебраического решателя системы взят аддитивный метод Шварца, который естественным образом ложится на архитектуру вычислительных систем с распределенной памятью. Полученные алгебраические системы в подпространстве следов, образованных переменными на внутренних границах подобластей, решаются с помощью обобщенного метода минимальных невязок. Вспомогательные системы в подобластях решаются с помощью прямого алгоритма PARDISO из библиотеки Intel MKL, использующего распараллеливание над общей памятью средствами OpenMP. Реализованные алгоритмы апробированы на численном решении ряда задач вычислительной математики, таких как задачи гидродинамики, диффузионно-конвективные уравнения, задачи электромагнетизма и др. Приведенные результаты численных экспериментов демонстрируют эффективность предлагаемых решений для многопроцессорных вычислительных систем с распределенной памятью.

Ключевые слова: итерационные алгоритмы, методы декомпозиции областей, распараллеливание, алгебраические системы, разреженные матрицы, численные эксперименты, аддитивный метод Шварца.

Введение

Системы линейных алгебраических уравнений с разреженной матрицей возникают при численном решении большого числа задач вычислительной математики. Вычислительная сложность таких задач часто требует кластерных расчетов. В настоящее время существует ряд подходов к декомпозиции задач и, соответственно, матриц систем, а также итерационных алгоритмов, позволяющих свести решение исходной задачи к серии решений задач в подобластях. К методам декомпозиции задачи можно отнести геометрические методы декомпозиции расчетной области и последующую аппроксимацию задачи в каждой из подобластей, а также алгебраические методы декомпозиции матрицы системы, например, реализованные в библиотеке METIS [1].

В рамках данной работы предлагается вариант алгебраической одномерной декомпозиции СЛАУ. Алгоритм основан на обходе в ширину графа матрицы системы и позволяет привести ее к блочно-трехдиагональному виду. За основу алгебраического решателя системы взят аддитивный метод Шварца, который естественным образом ложится на архитектуру вычислительных систем с распределенной памятью. К данному методу применяется крыловское ускорение [2], заключающееся в замене стационарного процесса $x_{n+1} = Tx_n + g$ решением системы $(I - T)x = g$, которое

¹Статья рекомендована к публикации программным комитетом международной научной конференции «Параллельные вычислительные технологии 2012»

осуществляется при помощи итерационных методов в подпространствах Крылова, таких как GMRES. При этом итерации осуществляются в подпространстве следов, образованных переменными на границах подобластей. Решение систем в подобластях осуществляется параллельным прямым решателем разреженных систем PARDISO из библиотеки Intel® MKL [3]. Использование такого подхода позволяет реализовать black-box решатель, не требующий дополнительной информации для решения СЛАУ и имеющий небольшое число конфигурационных параметров. Реализация алгоритма на языке C++ с использованием средств MPI и высокоуровневой библиотеки линейной алгебры Eigen [4] включена в библиотеку Krylov [5].

Тестирование алгоритма проводилось на большом числе методических и практических задач вычислительной физики. Было произведено исследование быстродействия решателя и числа итераций в зависимости от различных параметрах решателя. К варьируемым параметрам относились параметры алгебраической декомпозиции, такие как перекрытие подобластей и их количество, а к параметрам времени исполнения — число потоков в PARDISO при решении задач в подобластях, а также число MPI процессов на узел. Кроме того, была продемонстрирована точность получаемых решений для представленных задач.

1. Алгоритмы решения СЛАУ

1.1. Метод алгебраической декомпозиции СЛАУ

Аппроксимация различных краевых задач соответствующих дифференциальных уравнений различными методами, такими как методы конечных разностей, конечных объемов и конечных элементов, приводит к системе линейных алгебраических уравнений вида

$$Ax = b, \quad x, b \in \mathcal{R}^N, \quad A \in \mathcal{R}^{N,N}, \quad \det |A| \neq 0. \quad (1)$$

Решение СЛАУ на кластере требует предварительной ее декомпозиции и распределения блоков матрицы, а также векторов неизвестных и правой части между узлами кластера. В работе предлагается использовать следующую модификацию метода одно-направленной алгебраической декомпозиции задачи [2], основными достоинствами которой являются простота, высокая скорость работы алгоритма и возможность получить распределенную блочно-трехдиагональную матрицу на выходе.

Сначала ищется псевдо-периферийная вершина v графа G , ассоциированного с матрицей системы (1).

Алгоритм 1. Поиск псевдо-периферийной вершины v графа G . Запускается обход в ширину [6] с произвольной вершины графа матрицы и находятся наиболее удаленные от нее вершины и расстояние до них. Далее снова запускается обход в ширину, начиная уже с произвольной вершины среди множества наиболее удаленных вершин, полученных на предыдущем шаге. Обходы в ширину повторяются до тех пор, пока увеличивается максимальное расстояние от стартовой вершины до остальных. Как только расстояние перестанет увеличиваться, алгоритм останавливается и в качестве псевдо-периферийной вершины v возвращает вершину, с которой начинался последний обход в ширину.

Время работы алгоритма 1 равно $O(E \cdot K)$, где E — количество ребер в графе матрицы. Для большинства конечных схем аппроксимации уравнений число ребер

пропорционально количеству вершин сетки в расчетной области. Этот факт следует из того, что соответствующие конечные базисы финитны и, как правило, используются сетки, у которых имеются ограничения на минимальные величины плоских и телесных внутренних углов. Число K в оценке равно количеству итераций, совершенных алгоритмом 1. Для произвольных графов в худшем случае это число может быть довольно велико, однако для графов практических задач алгоритму обычно требуется всего лишь несколько итераций [2]. За счет этого на практике алгоритм поиска псевдо-периферийных вершин оказывается весьма эффективным. Более того, во многих случаях оказывается достаточно выбрать произвольную, например, первую вершину графа, и вообще не запускать алгоритм поиска псевдо-периферийных вершин.

После того, как выбрана некоторая вершина v , начиная с нее запускается обход графа G в ширину. Все вершины разделяются на множества — «фронты» — согласно расстоянию от них до вершины v . То есть, во фронт F_k ($k = 0 \dots d$, где d — псевдо-диаметр графа G) попадают все вершины, равноудаленные от вершины v на расстояние k .

Далее рассмотрим способ объединения фронтов в алгебраические подобласти Ω_p . Мы будем объединять только последовательные фронты в подобласти, так что

$$\Omega_p = \bigcup_{k=l_p}^{r_p} F_k,$$

где l_p и r_p — границы подобласти Ω_p . Пусть требуется построить P подобластей (например, по числу используемых узлов кластера) для системы (1) с N неизвестными. Желательно разбить неизвестные в подобласти приблизительно по N/P неизвестных. Однако поскольку мы объединяем в подобласти не отдельные вершины, а фронты, то точное разбиение неизвестных по N/P в каждой подобласти может не получиться. Тем не менее, мы можем при помощи бинарного поиска минимизировать максимальный размер подобласти либо максимизировать минимальный размер.

Рассмотрим, например, максимизацию минимального размера, которая может использоваться для того, чтобы, по возможности, более равномерно загрузить все узлы кластера. Легко видеть, что функция зависимости максимально возможного числа подобластей при фиксированном минимально допустимом размере монотонна. Действительно, при фиксированном минимальном размере разделение вершин графа разбиение на подобласти можно произвести при помощи следующего жадного [6] алгоритма.

Алгоритм 2. Алгоритм построения максимального числа подобластей. К первой подобласти добавляем последовательно фронты, начиная с первого, пока размер подобласти не станет больше либо равным минимально допустимому размеру. В этот момент можно прекратить увеличение первой подобласти и приступить к построению второй подобласти. Этот процесс продолжается до тех пор, пока все фронты не окажутся в какой-либо подобласти. При этом может оказаться так, что в конце окажется несколько фронтов, которых недостаточно для формирования отдельной подобласти. Такие фронты можно отнести к последней построенной подобласти.

Ясно, что приведенный выше способ строит максимальное число подобластей. Действительно, если бы можно было построить больше подобластей при заданном

минимальном размере подобласти, то тогда, начиная с некоторого i , подобласть Ω_i в «жадном» разбиении и подобласть Ω'_i отличались бы. Рассмотрим минимальное такое i . Если в подобласти Ω'_i больше фронтов, чем в Ω_i , то тогда оптимальное разбиение можно перестроить таким образом, чтобы в подобласти i было столько же фронтов, что и в оптимальном. Значит, в подобласти Ω'_i должно быть меньше фронтов, чем в Ω_i . Однако Ω_i строилось таким образом, что в него поместили минимально возможное число фронтов при фиксированном начальном. Получаем противоречие, и значит, построенное алгоритмом разбиение является максимальным.

Из приведенных выше рассуждений становится ясно, что с увеличением минимального размера число подобластей не может увеличиться. Значит, функция зависимости их числа от минимального размера действительно монотонна, и можно использовать двоичный поиск максимального минимального размера подобласти, чтобы получить P подобластей. Кроме того, описанный выше жадный алгоритм позволяет собственно построить эти подобласти уже после определения минимального размера.

Представленный алгоритм генерирует подобласти без пересечений. Однако получить разбиение с пересечениями довольно легко — достаточно просто расширить каждую из подобластей на несколько фронтов слева и справа. Также можно при начале генерации следующей подобласти отступить от границы предыдущей внутрь необходимое число фронтов.

Заметим, что ребра в графе матрицы могут соединять вершины либо одного, либо двух соседних фронтов. Тогда получается, что граничными для подобласти p будут вершины, принадлежащие двум фронтам $F_{l_{p-1}}$ и $F_{r_{p+1}}$. Можно считать, что эти фронты принадлежат соседним подобластям Ω_{p-1} и Ω_{p+1} соответственно. Получается, что если последовательно занумеровать переменные в каждой из подобластей, а в каждой из подобластей — последовательно в каждом из фронтов, мы получим следующую блочно-трехдиагональную СЛАУ:

$$\bar{A}\bar{x} = \begin{bmatrix} D_1 & U_1 & & & \\ L_1 & D_2 & \ddots & & \\ & \ddots & \ddots & U_{P-1} & \\ & & & L_{P-1} & D_P \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_P \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_P \end{bmatrix} = \bar{f}. \quad (2)$$

Будем полагать, что диагональные блоки D_p , соответствующие СЛАУ в подобластях Ω_p , не вырождены. При этом если разбиение исходной СЛАУ выполнялось с пересечениями, то в полученной СЛАУ (2) порядка N' окажется по несколько переменных, соответствующих одной и той же вершине графа, но лежащим в разных подобластях; в этом случае $N' > N$.

1.2. Крыловское ускорение аддитивного метода Шварца

После декомпозиции задачи на подобласти и пересылки блоков матрицы в соответствующие узлы кластера мы имеем распределенную блочную СЛАУ (2). Для решения таких СЛАУ широко используются методы Шварца. Одна итерация аддитивного метода Шварца в матричном виде может быть записана как (см. [2])

$$\bar{x}^{i+1} = \bar{x}^i + \sum_p R_p^T D_p^{-1} R_p (f - \bar{A}\bar{x}^i),$$

где $R_p : \mathcal{R}^{N'} \rightarrow \mathcal{R}^{N_p}$ — оператор сужения на подобласть Ω_p . Здесь N' — размерность СЛАУ (2), а N_p — количество неизвестных в подобласти Ω_p . С использованием блочного вида СЛАУ итерацию метода Шварца можно переписать следующим образом:

$$x_p^{i+1} = x_p^i + D_p^{-1}(f_p - L_p x_{p-1}^i - U_p x_{p+1}^i),$$

где для $p = 1$ и $p = P$ полагаем, что $L_p x_{p-1}^i = 0$ и $U_p x_{p+1}^i = 0$ соответственно. В общем случае члены $L_p x_{p-1}^i$ и $U_p x_{p+1}^i$ являются аналогами условий Дирихле для краевых задач.

Заметим, что на самом деле итерировать достаточно только переменные, образующие внешние границы подобластей, поскольку, зная граничные значения, легко определить значения внутри подобластей, решив соответствующую подзадачу. Для этого в произведениях $L_p x_{p-1}^i$ и $U_p x_{p+1}^i$ достаточно использовать только части векторов из подобластей Ω_{p-1} и Ω_{p+1} , соответствующие фронтам $l_p - 1$ и $r_p + 1$. Эти переменные и будут образовывать внешнюю границу области Ω_p .

Обозначим за u^b вектор граничных переменных для всех подобластей, и введем оператор сужения на граничные переменные $\hat{R} : \mathcal{R}^{N'} \rightarrow \mathcal{R}^{N_b}$, где N_b — размерность пространства следов. Тогда итерация метода Шварца в подпространстве следов выглядит следующим образом:

$$u_{i+1}^b = u_i^b + \hat{R} \sum_p R_p^T D_p^{-1} R_p (f - \bar{A} \hat{R}^T u_i^b).$$

Эту итерацию можно представить в виде $u_{i+1}^b = S(u_i^b) = T u_i^b + g$, где

$$T = I - \hat{R} \sum_p R_p^T D_p^{-1} R_p \bar{A} \hat{R}^T, \quad g = \hat{R} \sum_p R_p^T D_p^{-1} R_p f.$$

Сходимость стационарных итераций имеется при спектральном радиусе $\rho(T) < 1$. В то время как для эллиптических уравнений это условие выполнено, для произвольных уравнений это не так. Однако можно заметить, что решение u_*^b должно удовлетворять $u_*^b = T u_*^b + g$, что равносильно системе

$$[I - T] u^b = g. \tag{3}$$

Данную систему можно решать итерационными методами в подпространствах Крылова, например обобщенным методом минимальных невязок (GMRES) [2]. Для этого достаточно, чтобы матрица $[I - T] = \hat{R} \sum_p R_p^T D_p^{-1} R_p \bar{A} \hat{R}^T$ была невырожденной, что является существенно более слабым условием, по сравнению с условием $\rho(T) < 1$. Заметим также, что матрица $[I - T]$ не является симметричной, поэтому нельзя использовать экономичные методы в подпространствах Крылова, такие как метод сопряженных градиентов, ориентированные на решение симметричных СЛАУ и использующие короткие рекурсии для построения базиса подпространств Крылова.

Метод GMRES рассчитан на решение произвольных невырожденных СЛАУ вида $Ax = b$, при этом он минимизирует невязку $r_j = b - Ax_j$ в подпространстве Крылова

$$K_m(A, r_0) = \text{span}\{r_0, Ar_0, A^2 r_0, \dots, A^{m-1} r_0\}.$$

Псевдокод алгоритма следующий:

- $r_0 \leftarrow b - Ax_0$, $\beta = \|r_0\|$, $q_0 \leftarrow r_0/\|r_0\|$, $\xi \leftarrow (1, 0, \dots, 0)^T$
- $j \leftarrow 0, 1, \dots$ while $\|r_j\| > \varepsilon\beta$
 - $\tilde{q}_{j+1} \leftarrow Aq_j$
 - For $k \in [0, \dots, j]$
 - * $H_{k,j} \leftarrow (\tilde{q}_{j+1}, q_k)$
 - * $\tilde{q}_{j+1} \leftarrow \tilde{q}_{j+1} - H_{k,j}q_k$
 - $H_{j+1,j} \leftarrow \|\tilde{q}_{j+1}\|$, $q_{j+1} \leftarrow \tilde{q}_{j+1}/H_{j+1,j}$
 - For $k \in [0, \dots, j-1]$
 - * $\begin{bmatrix} H_{k,j} \\ H_{k+1,j} \end{bmatrix} \leftarrow \begin{bmatrix} c_k & s_k \\ -\bar{s}_k & c_k \end{bmatrix} \begin{bmatrix} H_{k,j} \\ H_{k+1,j} \end{bmatrix}$
 - $c_j \leftarrow |H_{j,j}|/\sqrt{|H_{j,j}|^2 + |H_{j+1,j}|^2}$
 - $\bar{s}_j \leftarrow c_j H_{j+1,j}/H_{j,j}$
 - $\begin{bmatrix} \xi_j \\ \xi_{j+1} \end{bmatrix} \leftarrow \begin{bmatrix} c_j & s_j \\ -\bar{s}_j & c_j \end{bmatrix} \begin{bmatrix} \xi_j \\ \xi_{j+1} \end{bmatrix}$
 - $H_{j,j} \leftarrow c_j H_{j,j} + s_j H_{j+1,j}$, $H_{j+1,j} \leftarrow 0$
 - $\|r_{j+1}\| \leftarrow \beta|\xi_{j+1}|$
- $y_j \leftarrow \beta H^{-1}\xi$
- $x_j \leftarrow x_0 + [q_0 \dots q_{j-1}]y_j$.

Из псевдокода видно, что для алгоритма требуется только операция умножения матрицы системы на вектор. Запишем результат умножения $I - T$ на вектор v через результат одной итерации метода аддитивного Шварца $S(v)$:

$$[I - T]v = v - Tv = v - S(v) + g, \quad g = S(0).$$

Таким образом, для решения СЛАУ (3) методом GMRES достаточно реализовать обычный метод аддитивного Шварца. Для решения задач в подобластях $x_p = D_p^{-1}b_p$ можно использовать какой-нибудь прямой решатель для разреженных систем. При этом LU -разложение матриц D_p достаточно вычислить всего один раз.

Оценим эффективность данного алгоритма для простого случая — решения трехмерного уравнения Пуассона в параллелепипеде со структурированной сеткой. Пусть имеется сетка с n вершинами по каждой из координатных осей. Тогда матрица системы будет иметь порядок $N = n^3$. При решении такой системы прямыми методами требуется $O(N^{5/3})$ памяти и $O(N^{7/3})$ операций [7], поскольку ширина ленты матрицы k в этом случае оказывается пропорциональной n^2 .

Предположим теперь, что декомпозиция задачи на P подобластей выполнена равномерно с небольшими перехлестами, то есть порядок СЛАУ в каждой подобласти есть $O(N/P)$. Размерность пространства следов будем считать равной $O(P \cdot n^2) = O(P \cdot N^{2/3})$. Пусть для решения СЛАУ (3) требуется K итераций. Считая, что в подобластях сохраняются оценки для прямых решателей, получим, что общие затраты памяти M будут равны

$$M = O(P \cdot (N/P)^{5/3} + K \cdot P \cdot N^{2/3}) = O(N^{5/3}/P^{2/3} + K \cdot P \cdot N^{2/3}),$$

или, в пересчете на один узел (при условии использования распределенного GMRES),

$$M_p = O((N/P)^{5/3} + K \cdot N^{2/3}).$$

Время работы решателя T будет складываться из максимального времени факторизации t_{fact} , времени работы итерационного метода t_{iter} и времени на коммуникации

t_{comm} ($T = t_{fact} + t_{iter} + t_{comm}$). Для них можно получить следующие оценки:

$$\begin{aligned} t_{fact} &= O((N/P)^{7/3}), \\ t_{iter} &= O(K \cdot (N/P)^{5/3} + K^2 \cdot N^{2/3}), \\ t_{comm} &= O(K \cdot P \cdot N^{2/3}). \end{aligned}$$

Данные оценки справедливы для распределенного GMRES. При использовании GMRES на одном узле оценка для t_{iter} увеличиваются до

$$t_{iter} = O(K \cdot (N/P)^{5/3} + K^2 \cdot P \cdot N^{2/3}).$$

Тогда имеем

$$T = O((N/P)^{7/3} + K \cdot (N/P)^{5/3} + K^2 \cdot P \cdot N^{2/3}).$$

Поскольку, как правило, K растет с ростом P , получаем, что при фиксированном N для T имеется оптимум при некотором значении P . Этот оптимум достигается, с одной стороны, за счет уменьшения времени факторизации блоков, и, с другой стороны, за счет увеличения временных затрат на ортогонализацию векторов и межпроцессные коммуникации.

2. Технологии реализации решателя

При выборе языка программирования, сторонних библиотек и технологий для реализации решателя учитывалась специфика используемых в решателе алгоритмов. Необходим был высокоуровневый язык и набор библиотек, хорошо подходящий для реализации различных алгоритмов на графах, манипуляций с векторами и разреженными матрицами. Дополнительное требование заключалось в том, чтобы компиляторы языка генерировали высоко-эффективный код. В итоге выбор был сделан в пользу языка C++ и использования стандартной библиотеки STL, из которой использовались различные контейнеры, а также библиотеки линейной алгебры Eigen [4]. Последняя имеет в своем составе классы разреженных матриц и динамических векторов, предоставляет удобный API для манипуляции ими, а также имеет встроенную векторизацию алгоритмов с использованием инструкций SSE.

Для решения задач в подобластях требуется эффективный решатель для СЛАУ с разреженными матрицами невысокого порядка, причем необходимо решать СЛАУ с различными правыми частями. Кроме того, обусловленность диагональных матриц D_p может быть довольно плохой. Для решения таких систем хорошо подходят прямые решатели разреженных систем. В качестве такого решателя был выбран PARDISO (PARallel DIrect SOLver) из библиотеки Intel® MKL [3]. Данный решатель является многопоточным, что позволяет реализовать гибридный подход и повысить производительность: на каждом используемом узле кластера можно решать задачи для одной подобласти, причем факторизацию СЛАУ, выполняющуюся при инициализации решателя, и решение серии систем с разными правыми частями можно выполнять в многопоточном режиме с использованием всех доступных процессорных ядер.

Для организации взаимодействия процессов на узлах кластера был использован интерфейс MPI, который широко используется при написании распределенных программ. При этом решатель был построен на основе master-slave архитектуры, с выделенным основным узлом (с рангом MPI, равным 0). На этом узле решателю

передается матрица, выполняется декомпозиция на подобласти, а также работает решатель GMRES. Остальные MPI процессы работают как ведомые, принимая от мастер-процесса матрицу системы на этапе инициализации и граничные переменные в процессе работы для решения задач в подобластях. Граничные для соседних подобластей переменные после решения отсылаются назад. Использование такого подхода позволяет на этапе итерационного решения ограничиться вызовами трех функций MPI на одну итерацию: MPI_scatter и MPI_gather для рассылки и сбора граничных переменных и MPI_Bcast для управления ведомыми процессами.

Представленный решатель был включен в состав библиотеки Krylov [5]. Интерфейс решателя, позволяющий вызывать его из процедур, написанных на различных языках программирования, представлен на рисунке.

```
void cschwarz_solver(MPI_Comm *pcomm, const int *n,
                    const double *a, const int *ia, const int *ja,
                    double *u, const double *f,
                    const double *eps, const int *ovl,
                    const int *nmax, const int *nres, int *niter,
                    double *time, int *err);
void cschwarz_client(MPI_Comm *pcomm);
```

Интерфейс решателя

Параметры решателя следующие:

pcomm — MPI коммуникатор, процессы которого участвуют в решении СЛАУ; процедура `cschwarz_solver(...)` должна вызываться в процессе с рангом 0, в остальных процессах необходимо вызывать `cschwarz_client(...)`;

n — порядок СЛАУ;

a, ia, ja — матрица системы в формате CSR (Compressed Sparse Row format) [3];

u — вектор размера **n**, в который записывается полученное решение системы;

f — правая часть СЛАУ, должна иметь размер **n**;

eps — критерий останова итераций ε (**eps**) для решателя GMRES, решение считается найденным, если

$$\|g - [I - T]u^b\| < \varepsilon \|g\|;$$

ovl — параметр пересечения подобластей; после построения разбиения на подобласти без пересечений каждая подобласть расширяется на **ovl** фронтов влево и вправо, соответственно пересечение соседних подобластей состоит из $2 \cdot \mathbf{ovl}$ фронтов;

nmax — максимальное число итераций решателя GMRES;

nres — число итераций решателя GMRES до рестарта, позволяет ограничить длину последовательности q_j базисных векторов подпространства Крылова; после выполнения каждых **nres** итераций, даже если не была достигнута необходимая

точность решения, текущее приближение пересчитывается по формулам

$$y_j = \beta H^{-1} \xi, \quad u_j^b \leftarrow u_0^b + [q_0 \dots q_{j-1}] y_j,$$

и новое значение u_j^b используется в качестве начального приближения для решателя;

niter — выходной параметр, количество совершенных решателем итераций;

time — время, затраченное на решение системы;

err — код возврата решателя; ненулевое значение свидетельствует об ошибке.

3. Численные эксперименты

В последующих таблицах приняты следующие обозначения: N — порядок системы, N_{nz} — количество ненулевых элементов в исходной матрице, N_b — размерность пространства следов, t_{fac} — время (в секундах) факторизации матриц D_p , t_{tot} — общее время решения и n — количество итераций. Эксперименты проводились на вычислительных узлах HP BL2x220c G6 кластера НКС-30Т [8], каждый из которых оборудован двумя четырехъядерными процессорами Intel Xeon E5540 и 16-ю гигабайтами памяти. В целях увеличения объема памяти, доступной PARDISO для хранения факторов матриц D_p , запускалось по одному MPI-процессу на узел, а для ускорения факторизации были задействованы все имеющиеся на узлах ядра. Параметр пересечения подобластей при решении всех задач был установлен таким образом, чтобы подобласти пересекались максимум по 8 фронтам, за исключением второй модельной задачи, где пересечение составляет 4 фронта. Параметр ε в критерии останова итераций был равен 10^{-7} . Рестарты итерационного процесса не производились, то есть значение параметра **nres** было выбрано достаточно большим.

3.1. Модельная задача для диффузионно-конвективного уравнения

В качестве первой модельной задачи была выбрана задача Дирихле для линейного диффузионно-конвективного уравнения

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} + p \frac{\partial u}{\partial x} + q \frac{\partial u}{\partial y} + r \frac{\partial u}{\partial z} = f(x, y, z), \quad (x, y, z) \in \Omega, \\ u|_{\Gamma} = g(x, y, z),$$

в единичном кубе $\Omega = [0; 1]^3$ на равномерной кубической сетке. Дискретизация задачи проводилась с помощью семиточечной схемы экспоненциального типа [7]. Функции f и g выбирались соответственно точному решению $u(x, y, z) = x^2 + y^2 + z^2$, а в качестве начального приближения была взята функция $u^0(x, y, z) = 0$. В таблицах 1–3 приведены результаты решения данной модельной задачи при различных значениях p , q и r .

Максимальная погрешность решения первой модельной задачи в равномерной норме составила $7,7 \cdot 10^{-7}$.

Таблица 1

Решение первой модельной задачи при $p = q = r = 0$

Сетка	N	N_{nz}		Количество узлов			
				2	4	8	16
64^3	262144	1810432	N_b	6104	16196	34688	71786
			n	12	16	21	32
			t_{fac}	3,31	0,94	0,46	0,30
			t_{tot}	4,65	1,74	1,18	1,26
128^3	2097152	14581760	N_b	24536	65508	139183	285485
			n	17	23	29	40
			t_{fac}	219	33,3	8,36	2,56
			t_{tot}	245	44,5	15,6	9,25

Таблица 2

Решение первой модельной задачи при $p = q = r = 16$

Сетка	N	N_{nz}		Количество узлов			
				2	4	8	16
64^3	262144	1810432	N_b	6104	16196	34688	71786
			n	10	11	14	24
			t_{fac}	3,29	0,95	0,50	0,30
			t_{tot}	4,44	1,67	1,11	1,08
128^3	2097152	14581760	N_b	24536	65508	139183	285485
			n	14	16	20	29
			t_{fac}	233	34,7	8,73	2,60
			t_{tot}	255	43,9	14,8	8,23

Таблица 3

Решение первой модельной задачи при $p = q = r = -16$

Сетка	N	N_{nz}		Количество узлов			
				2	4	8	16
64^3	262144	1810432	N_b	6104	16196	34688	71786
			n	10	11	15	25
			t_{fac}	3,33	1,00	0,47	0,44
			t_{tot}	4,48	1,97	1,14	1,33
128^3	2097152	14581760	N_b	24536	65508	139183	285485
			n	14	17	21	30
			t_{fac}	226	33,9	8,67	3,01
			t_{tot}	248	43,5	15,2	8,77

3.2. Модельная задача с волноводом

Вторым примером является решение трехмерной краевой задачи для уравнения Гельмгольца

$$\nabla \times \left(\frac{1}{\mu_r} \nabla \times \vec{E} \right) - k_0^2 \dot{\epsilon}_r \vec{E} = 0, \quad k_0 = \omega \sqrt{\epsilon_0 \mu_0},$$

дискретизация которой выполнялась с помощью конечно-элементных аппроксимаций с базисными функциями Неделека второго порядка [9, 10]. Расчетной областью является волновод с линейными размерами $a = 72$, $b = 34$, $c = 200$ мм. Параметры задачи: $\mu_r = 1$, $\dot{\epsilon}_r = 1 - 0,1i$, частота $\omega = 6\pi \cdot 10^9$ Гц. Граничные условия: на грани $z = 200$ мм задавалась касательная компонента поля

$$\vec{E}_0 \times \vec{n} = \vec{e}_y \sin(\pi x/a) \times \vec{n},$$

а на остальных гранях — условие металлической стенки ($\vec{E}_0 = 0$). Аналитическое решение в данном случае имеет вид

$$\vec{E} = \vec{e}_y \sin\left(\frac{\pi x}{a}\right) \frac{\sin \gamma z}{\sin \gamma c}.$$

В табл. 4 приводятся результаты решения указанной задачи при различных размерностях сетки.

Таблица 4

Решение второй модельной задачи

Сетка	N	N_{nz}		Количество узлов			
				2	4	8	16
$8 \times 4 \times 20$	21664	423392	N_b	1540	4678	10708	23052
			n	9	13	21	258
			t_{fac}	0,19	0,17	0,15	0,36
			t_{tot}	0,51	0,47	0,67	6,19
$15 \times 7 \times 40$	149874	3117779	N_b	5210	16646	37670	78446
			n	13	18	25	38
			t_{fac}	2,56	1,26	0,63	0,40
			t_{tot}	5,88	4,00	3,12	3,15
$29 \times 14 \times 80$	1196026	25795767	N_b	20562	67084	151032	318108
			n	18	26	34	49
			t_{fac}	108	39,9	14,9	5,29
			t_{tot}	173	85,6	43,3	32,94

Наличие точного решения позволяет проверить сходимость итерационного процесса. Данные тесты продемонстрировали ошибку порядка $O(h^2)$, что согласуется с теорией.

3.3. Примеры решения практических задач

В заключение данного пункта мы приведем результаты решения трех СЛАУ (обозначаемых ниже в табл. 5 как Г1, Г2, Г3), возникающих из актуальных практических краевых задач, описываемых трехмерными уравнениями гидродинамики, которые аппроксимируются с помощью неявных конечно-объемных схем на сложных неструктурированных сетках.

Таблица 5

Решение практических задач гидродинамики

Задача	N	N_{nz}		Количество узлов			
				2	4	8	16
Г1	4875172	26455278	N_b	180487	439384	895041	1831917
			n	40	58	85	162
			t_{fac}	101	64	49,8	19,7
			t_{tot}	166	117	119	201
Г2	1726272	11984539	N_b	6912	18533	42476	90000
			n	52	109	207	334
			t_{fac}	54,4	26,0	9,55	3,26
			t_{tot}	79,0	116	83,0	64,0
Г3	475000	13672497	N_b	2500	7500	17500	37110
			n	24	25	36	53
			t_{fac}	6,12	2,62	1,08	0,53
			t_{tot}	10,1	5,45	3,31	2,88

Следует отметить, что приведенные результаты параллельных вычислений дают значительный выигрыш в сравнении с однопроцессорными расчетами, в которых некоторые из рассмотренного типа СЛАУ с помощью «стандартных» предобусловленных крыловских алгоритмов решаются часами.

Заключение

Рассмотренные результаты позволяют сделать следующие выводы.

- С увеличением количества подобластей число итераций увеличивается, но (за одним исключением) не более чем линейно. В этом плане улучшение сходимости можно ожидать за счет включения других типов граничных условий на смежных границах, вместо используемого условия Дирихле.
- Применение прямого решателя PARDISO для подобластей является достаточно экономичным. В частности, с увеличением числа MPI-процессов относительные временные расходы на факторизацию вспомогательных подсистем значительно уменьшаются.
- Размерности СЛАУ в подпространствах следов (N_b) значительно меньше исходных порядков N . Это позволяет экономично реализовать с помощью GMRES итерации по подобластям.

- Описанная программная реализация позволяет проводить оптимизацию алгоритмов за счет выбора различных счетных параметров (величина перехлестов, варьирования числа ядер на подобласти, количества MPI-процессов на вычислительный узел и т.д.).
- При увеличении числа подобластей и используемых процессоров до нескольких тысяч целесообразен, по-видимому, переход от используемой 1-D декомпозиции к 2-D или даже 3-D декомпозиции.

Работа выполнена при поддержке грантов РФФИ № 11-01-00205 и Президиума РАН № 2.5.

Литература

1. Karypis, G. A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs / G. Karypis, V. Kumar // SIAM Journal on Scientific Computing. – 1999. – Vol. 20, № 1. – P. 359–392.
2. Saad, Y. Iterative Methods for Sparse Linear Systems, Second Edition. / Y. Saad. – SIAM, 2003.
3. Intel (R) Math Kernel Library from Intel: сайт. URL: <http://software.intel.com/en-us/articles/intel-mkl/> (дата обращения: 03.11.2012).
4. Eigen: сайт. URL: http://eigen.tuxfamily.org/index.php?title=Main_Page (дата обращения: 03.11.2012).
5. Ильин, В.П. Krylov: библиотека алгоритмов и программ для решения СЛАУ / В.П. Ильин, Д.С. Бутюгин, Е.А. Ицкович и др. // Современные проблемы математического моделирования. Математическое моделирование, численные методы и комплексы программ. Сборник трудов Всероссийских научных молодежных школ. – Ростов-на-Дону: Изд-во Южного федерального университета, 2009. – С. 110–128.
6. Кормен, Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест. – М: МЦНМО: БИНОМ. Лаборатория знаний, 2004.
7. Ильин, В.П. Методы и технологии конечных элементов. / В.П. Ильин. – Новосибирск: Изд-во ИВМиМГ СО РАН, 2007.
8. Кластер НКС-30Т: сайт. URL: <http://www2.sccc.ru/НКС-30Т/НКС-30Т.htm> (дата обращения: 03.11.2012).
9. Monk, P. Finite Element Methods for Maxwell's Equations. / P. Monk. – Oxford University Press, 2003.
10. Ingelstrom, P. A new set of H(curl)-conforming hierarchical basis functions for tetrahedral meshes / P. Ingelstrom // IEEE Transactions on Microwave Theory and Techniques. – 2006. – Vol. 54, № 1. – P. 160–114.

Дмитрий Сергеевич Бутюгин, младший научный сотрудник, Институт вычислительной математики и математической геофизики СО РАН, аспирант, Новосибирский Государственный Университет, dm.butyugin@gmail.com.

Валерий Павлович Ильин, доктор физико-математических наук, профессор, главный научный сотрудник, Институт вычислительной математики и математической геофизики СО РАН, ilin@sscc.ru.

Данил Валерьевич Перевозкин, младший научный сотрудник, Институт вычислительной математики и математической геофизики СО РАН, dperevozkin@mail.ru.

PARALLEL METHODS FOR SLAE SOLUTION ON THE SYSTEMS WITH DISTRIBUTED MEMORY IN KRYLOV LIBRARY

D.S. Butyugin, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russian Federation),

V.P. Il'in, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russian Federation),

D.V. Perevozkin, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russian Federation)

The paper presents an approach to creation of black-box parallel iterative solver, which is used in Krylov library for solving systems of linear algebraic equations (SLAEs) with large sparse matrices in CSR format arising from discretization of multidimensional boundary value problems. A variant of one-dimensional algebraic decomposition method is offered. The algorithm is based on breadth-first search of SLAE's adjacency graph that allows to reduce the matrix to block-tridiagonal form. The algebraic solver is based on additive Schwarz method which naturally suits distributed memory computer systems. The generalized minimal residual method is used to solve the SLAEs arising from relations on subdomains' boundaries. Auxiliary subdomain systems are solved with Intel MKL's multithreaded direct solver PARDISO. Implemented algorithms were tested on the numerical solution of the series of computational mathematics problems, such as problems of hydrodynamics, diffusion-convection equations, problems of electromagnetism and others. Added numerical experiments results show the effectiveness of the presented algorithms for multiprocessor computational systems with distributed memory.

Keywords: iterative algorithms, domain decomposition methods, parallel computing, algebraic systems, sparse matrices, numerical experiments, additive Schwarz method.

References

1. Karypis G., Kumar V. A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM Journal on Scientific Computing*. 1999. Vol. 20, № 1. P. 359–392.
2. Saad Y. *Iterative Methods for Sparse Linear Systems*, Second Edition. Society for Industrial and Applied Mathematics, 2003.
3. Intel (R) Math Kernel Library from Intel.
URL: <http://software.intel.com/en-us/articles/intel-mkl/>
4. Eigen. URL: http://eigen.tuxfamily.org/index.php?title=Main_Page
5. Il'in V.P., Butyugin D.S., Itskovich E.A et al. Krylov: biblioteka algoritmov i programm dlya resheniya SLAU [Krylov: a Library of Algorithms and

- Programs for Solving SLAE]. Trudy Vserossiyskikh nauchnykh molodezhnykh shkol "Sovremennye problemy matematicheskogo modelirovaniya. Matematicheskoye modelirovaniye, chislennyye metody i komplekсы programm" [Proceedings of All-Russian Youth Scientific Schools "Modern Problems of Mathematical Modeling. Mathematical Modeling, Numerical Methods and Program Complexes]. Rostov-on-Don, SFedU Publ., 2009, P. 110–128.
6. Cormen T., Leiserson C., Rivest R. Introduction to Algorithms. MIT Press, 2001.
 7. П'ин V.P. Metody i tekhnologii konechnykh elementov [Methods and Technologies of Finite Elements]. Novosibirsk, ICM&MG SBRAS Publ., 2007.
 8. NKS-30T Cluster URL: <http://www2.ssc.ru/НКС-30Т/НКС-30Т.htm>
 9. Monk P. Finite Element Methods for Maxwell's Equations. Oxford University Press, 2003.
 10. Ingelstrom P. A New Set of H(curl)-conforming Hierarchical Basis Functions for Tetrahedral Meshes IEEE Transactions on Microwave Theory and Techniques. 2006. Vol. 54, № 1. P. 160–114.

Поступила в редакцию 5 ноября 2012 г.

ИССЛЕДОВАНИЕ УДАРНОГО ВЗАИМОДЕЙСТВИЯ ИНДЕНТОРА С ТКАНЕВЫМИ БРОНЕПЛАСТИНАМИ, РАСПОЛОЖЕННЫМИ НА ПЛАСТИЛИНОВОМ ОСНОВАНИИ

Н.Ю. Долганина

Рассматривается суперкомпьютерное моделирование динамического взаимодействия индентора и тканевых бронепластин, состоящих из 1 – 5 слоев ткани размером 10x10 см с возможностью разрушения нитей, расположенных на пластилиновом основании. Определены механические характеристики регистрирующей среды (технического пластилина). Проведено сравнение расчетных и экспериментальных данных.

Ключевые слова: суперкомпьютерное моделирование, конечно-элементная модель, удар, регистрирующая среда, тканевая бронепластина.

Введение

Проверка тканевого бронезилята на прочность в стандартах разных стран – это обстрел бронепластин стандартными пулями контрольного оружия на основании, имитирующем тело человека (специальный пластилин) [1, 2]. Оценка последствий удара пули в бронезилят, надетый на человека, осуществляется путем сопоставления размеров вмятины, оставленной пластиной бронезилята на пластилине, с уровнем травмирования тела человека.

При ударе пули в тканевый бронезилят возникают сложные физические явления, связанные с большими перемещениями, скольжениями, фрикционными контактами, повышением температуры, пластическими деформациями, учесть которые в простых экспериментально-аналитических подходах ранее было невозможно [3]. Для более адекватного описания этого процесса в настоящее время используют численные методы, в частности, МКЭ-пакет LS-DYNA [4-7].

В данной работе проведено сопоставление расчетных (LS-DYNA) и экспериментальных данных по взаимодействию ударника с тканевыми бронепластинами, расположенными на пластилиновом основании.

Статья организована следующим образом. В разделе 1 приведена постановка задачи. В разделе 2 описываются методы исследования, и приводится описание задачи. В разделе 3 обсуждаются результаты проведенных исследований. В заключении суммируются основные результаты, полученные в данной работе.

1. Постановка задачи

Проведено суперкомпьютерное моделирование натуральных экспериментов: удар индентором в тканевую бронепластину, расположенную на пластилиновом основании. Определены механические свойства регистрирующей среды (технического пластилина). Проведено сравнение расчетных и экспериментальных данных.

Были рассмотрены бронепластины, состоящие из 1 – 5 слоев ткани полотняного переплетения размером 10x10 см. В работе использована ткань из арамидных нитей Русар. В расчетной модели нити имеют относительную свободу перемещения с возможностью

вытягивания с учетом сухого трения. Рассматривали нити, которые имеют прямоугольное поперечное сечение и были представлены одним оболочечным элементом по ширине с одной точкой интегрирования по толщине и выполнены из ортотропного материала с малыми поперечно – сдвиговыми свойствами. Нити в расчетной модели могли разрушаться. Тканевая бронепластина располагалась на неподвижном пластилиновом основании. В расчете индентор имел сферическую форму диаметром 4,5 мм и массой 0,5 г, с начальной скоростью 100 м/с и был выполнен из абсолютно жесткого материала.

2. Методы исследования

Чтобы снизить размерность задачи при расчете, геометрия нитей была предельно упрощена. Криволинейная ось нити была заменена ломаной, с прямолинейными горизонтальными участками и наклонными (34° к горизонтальному участку нити). Повторяющийся элемент ткани полотняного переплетения показан на рис. 1, где отмечены номера ключевых точек, координаты которых были введены в программу ANSYS, после чего по ключевым точкам были заданы соответствующие поверхности. Далее набор поверхностей был размножен до получения необходимых размеров модели.

Для решения задачи динамического взаимодействия тканевой бронепластины, расположенной на регистрирующей среде и жесткого индентора с помощью пакета программ LS-DYNA геометрия и сетка конечных элементов (рис. 2) (к-файл) были созданы в пакете программ ANSYS с последующей коррекцией.

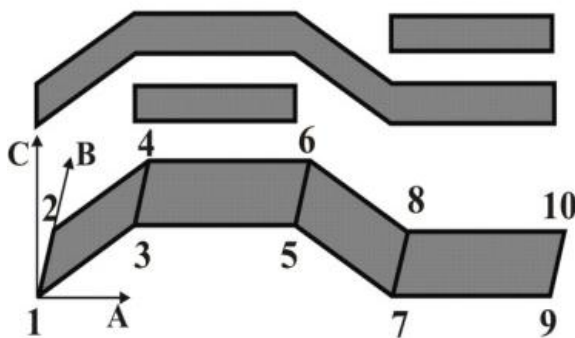


Рис. 1. Повторяющийся элемент

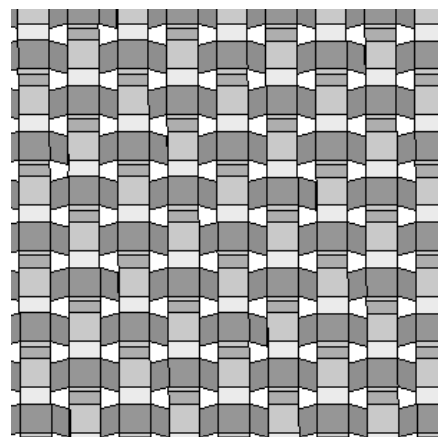


Рис. 2. Сетка конечных элементов

Реальная арамидная нить Русар номинальной линейной плотности 58,8 текс состоит из нескольких сотен элементарных волокон диаметром 12 – 15 мкм, которые имеют крутку 100 м^{-1} (зарубежные нити, как правило, некрученые). Для материала нити наиболее близкой является модель ортотропного упругого тела, в которой высокая жесткость имеет место в продольном направлении нити и на несколько порядков меньшая в поперечном и сдвиговом направлениях. Нити моделировали оболочечными элементами с одной точкой интегрирования по толщине Белычко-Вонга-Чанга (тип 10) [8].

Принято, что разрушение нити происходит при достижении в ней величины первого главного напряжения $\sigma_1 = 3 \text{ ГПа}$. Эта величина соответствует средней прочности исследованных арамидных нитей.

В пакете программ LS-DYNA представлено более сотни моделей материала [8], наиболее интересными для нас были малопараметрические модели, их анализ показал, что для нитей наиболее подходит материал *MAT_ENHANCED_COMPOSITE_DA-

MAGE, позволяющий задать ортотропные свойства нитей и учесть разрушение при достижении первого главного напряжения равным 3 ГПа (табл. 1).

Таблица 1

Механические характеристики нитей

Параметр	Обозначение	Величина
Толщина нити, мкм	T	100
Модуль упругости, МПа	E_x	$1,4 \cdot 10^5$
	E_y	$1,4 \cdot 10^3$
Плотность, кг/м ³	ρ	1440
Коэффициент Пуассона	μ_{xy}	0,3
Модуль сдвига, МПа	G_{xy}	$1,4 \cdot 10^3$
	G_{yz}	$1,4 \cdot 10^3$
	G_{zx}	$1,4 \cdot 10^3$

В конечно-элементной модели ткани нити имеют относительную свободу перемещения и возможность вытягивания с учетом сухого трения. Контакт объектов моделировали командой *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE с коэффициентом сухого трения 0,4, характерным для арамидных нитей [9].

Стальной индентор при взаимодействии с тканевой бронепластиной при исследуемых скоростях до 110 м/с не разрушается и не имеет пластических деформаций, поэтому для сокращения времени расчетов для него был выбран материал *MAT_RIGID (жесткое тело) с плотностью $\rho = 7800$ кг/м³ и модулем упругости $E = 2,1 \cdot 10^{11}$ Па.

Для того чтобы построить адекватную модель регистрирующей среды (технического пластилина), необходимо экспериментально определить ее механические свойства. Модуль упругости регистрирующей среды (технического пластилина) был определен из испытаний на квазистатическое сжатие цилиндрических образцов (диаметр 20 мм, высота 40 мм) на машине INSTRON 5882 со скоростями нагружения 5, 50, 250 и 500 мм/мин (рис. 3). Обработка начальных участков кривых деформирования дала значение $E = 0,7 \pm 0,1$ МПа. Плотность пластилина $\rho \approx 1800$ кг/м³.

Для определения зависимости предела текучести от скорости деформирования был поставлен эксперимент на низкоскоростной удар (рис. 4). Цилиндрический образец из пластилина, соединенный соосно со стальным грузом-цилиндром, бросали с разных высот $H = 0,5 - 2$ м на жесткое неподвижное основание. После удара была измерена остаточная длина образца. Скорость соударения V_0 была определена из равенства потенциальной и кинетической энергий (тепловые эффекты были незначительными) $mgH = mV_0^2/2$, где g – ускорение свободного падения, м/с²; m – масса груза и пластилина, кг.

Движение цилиндра (груза) считали равнозамедленным. С учетом введенных допущений среднюю скорость деформации образца из пластилина можно определить по зависимости: $\varepsilon \approx \frac{\Delta l}{l_0} \cdot \frac{1}{t} = \frac{\Delta l}{l_0} \cdot \frac{V_0}{2\Delta l} = \frac{V_0}{2l_0}$, где l_0 – начальная длина пластилинового цилиндра, м; t – время, с.

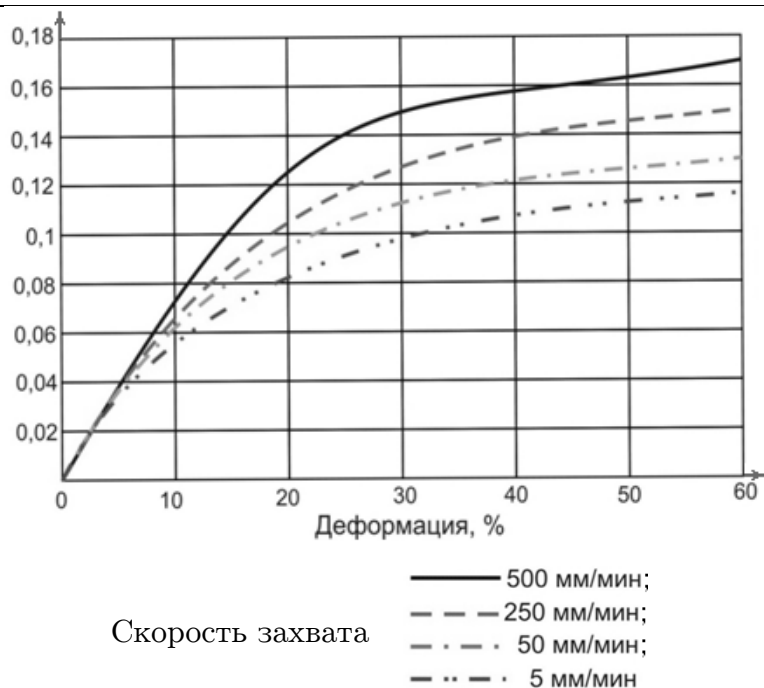


Рис. 3. Диаграммы деформирования цилиндрических образцов из пластилина с различными скоростями нагружения при сжатии

Так как в пластилиновом образце при ударе возникают большие пластические деформации, то для определения предела текучести упругим участком диаграммы деформирования пренебрегали (материал – жесткопластический). Предел текучести можно найти из равенства потенциальной энергии работе внутренних сил: $mgH = \sigma_T \Delta l S$, где σ_T – предел текучести; S – площадь поперечного сечения цилиндра из пластилина.

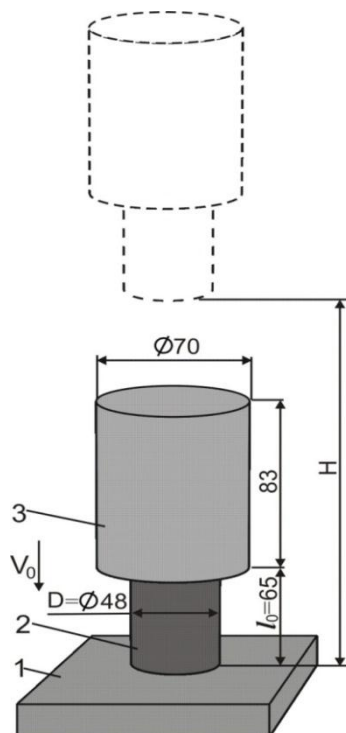


Рис. 4. Схема испытания на сжатие цилиндрических образцов из пластилина:
1 – жесткое неподвижное основание; 2 – цилиндр из пластилина; 3 – стальной цилиндр

По полученным в эксперименте значениям предела текучести σ_T была построена степенная функция зависимости σ_T (МПа) от скорости деформирования $\dot{\epsilon}$ (1/с) с показателем меньше единицы $\sigma_T(\dot{\epsilon}) = 0.94(\dot{\epsilon})^{0.15}$. Температура пластилина при испытаниях была (21 ± 1) °С.

Материал технического пластилина в модели считали упругопластическим с зависимостью предела текучести от скорости деформирования. Из списка материалов, заложенных в библиотеку пакета LS-DYNA, для технического пластилина был выбран *MAT_STRAIN_RATE_DEPENDENT_PLASTICITY, который позволяет учесть зависимость предела текучести от скорости деформирования в табличном виде [8].

3. Результаты исследований и их анализ

Расчеты динамического взаимодействия индентора с тканевыми бронепластинами (полотняное переплетение), состоящих из одного, двух, трех, четырех и пяти слоев ткани размером 10x10 см с возможностью разрушения нитей, расположенных на пластилиновом основании размером 10x10x5 см были проведены на суперкомпьютере «СКИФ – Аврора ЮУрГУ» [10], который используется для вычисления большого количества индустриальных задач [11].

В натурном эксперименте выстрелы производились из пневматического пистолета ИЖ-53 в центр бронепластины под прямым углом. Скорость индентора составляла (100 ± 5) м/с, ее измеряли хронографом S044 с погрешностью 1 м/с. Индентор имел сферическую форму диаметром 4,5 мм и массой 0,5 г, материал – сталь. По каждой бронепластине было сделано не менее 3-х выстрелов. После каждого выстрела были проведены измерения глубины отпечатков, оставленных бронепластиной на основании из пластилина (табл. 2). Различие расчетных и экспериментальных глубин вмятин не превышает 4 %.

Таблица 2

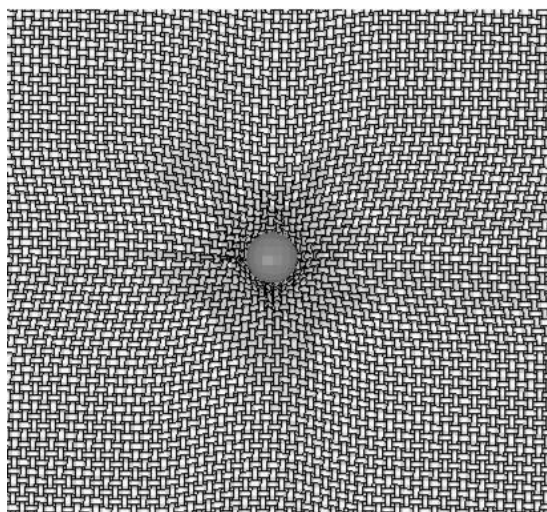
Сравнение расчетных и экспериментальных данных. Глубина вмятин в регистрирующей среде после выстрела

Количество слоев в бронепластине	Глубина отпечатка, мм		Погрешность, %
	Среднее экспериментальное значение	Расчетное значение	
1	10,0	9,6	4
2	6,0	5,8	3
3	5,0	4,9	2
4	4,4	4,4	0
5	5,2	5,2	0

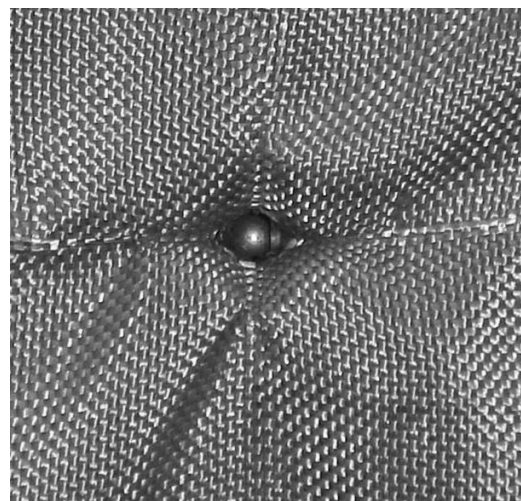
Сравнение расчетных и экспериментальных картин деформирования бронепластины из двух слоев ткани показано на рис. 5.

Как видно из рис. 5 характер деформирования тканевой бронепластины в расчетном и экспериментальном исследованиях одинаков. На рис. 5 видно, что в расчете и

эксперименте происходит вытягивание нитей, которые оказались расположенными под индентором.



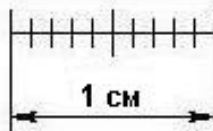
Расчет



Эксперимент

Рис. 5. Сравнение расчетных и экспериментальных картин деформирования тканевой бронепластины (два слоя ткани)

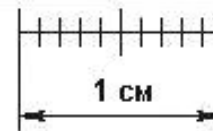
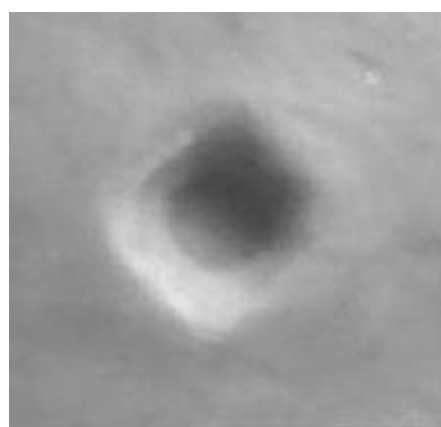
Сравнение расчетных и экспериментальных картин деформирования основания (формы и глубины вмятин) при деформировании бронепластины из двух слоев ткани показано на рис. 6.



Форма вмятины, расчет



Глубина вмятины, расчет



Форма вмятины, эксперимент



Глубина вмятины, эксперимент

Рис. 6. Сравнение расчетных и экспериментальных картин деформирования основания (формы и глубины вмятин) при деформировании бронепластины из двух слоев ткани

Заключение

Рассмотрено локальное динамическое деформирование многослойной тканевой бронепластины, расположенной на пластилиновом основании (МКЭ, пакет LS-DYNA) и проведено сопоставление с экспериментом. Показано, что форма и размеры вмятин, полученные расчетным методом, отлично согласуются с экспериментальными данными.

Показана возможность использования модели упругопластического тела с пределом текучести, зависящим от скорости нагружения для замены технического пластилина (регистрирующей среды, имитирующей реакцию тела человека на локальный удар).

Работа выполнена при поддержке грантов РФФИ № 12-08-31184 мол_а и № 10-07-96007-р_урал_а.

Литература

1. ГОСТ Р 50744 – 95. Бронеодежда. Классификация и общие технические требования. Прин. Постановлением Госстандарта России от 27.02.95 № 82. Введ. с изм. № 1, 2. Утвержден в сент. 1998 г., мае 2002. – М.: ИПК изд-во стандартов, 2003. – 8 с.
2. National Institute of Technology Standard, NIJ Standard 0101.06 Ballistic Resistance of Body Armor. – July, 2008.
3. Бова, В.Г. Концепция построения мягкого бронежилета для максимальной реализации свойств арамидных нитей / В.Г. Бова // Рабочая одежда и средства индивидуальной защиты. – 2001. – № 2(10). – С. 11 – 15.
4. Nilakantan, G. Using LS-DYNA[®] to Computationally Assess the V₀-V₁₀₀ Impact Response of Flexible Fabrics Through Probabilistic Methods / G. Nilakantan, M. Keefe, E.D. Wetzel, T.A. Bogetti, R. Adkinson, J.W. Gillespie // 11th International LS-DYNA[®] Users Conference: Proceedings of the International Users Conference (06-08 June 2010, Detroit). – 2010. – P. 5-1 – 5-12.
5. Долганина, Н.Ю. Оценка баллистического предела и прогиба многослойных тканевых пластин при ударе индентором / Н.Ю. Долганина // Вестник Южно-Уральского государственного университета. Серия "Машиностроение". – 2010. – Вып. 15. – № 10(186). – С. 17–23.
6. Долганина, Н.Ю. Моделирование ударных процессов в тканевых бронежилетах и теле человека на вычислительном кластере "СКИФ Урал" / Н.Ю. Долганина, С.Б. Сапожников, А.А. Маричева // Вычислительные методы и программирование: Новые вычислительные технологии. – 2010. – Т. 11. – С. 117-126.
7. Долганина, Н.Ю. Проектирование новых конструкций тканевых бронепанелей с использованием суперкомпьютерных вычислений / Н.Ю. Долганина, С.Б. Сапожников // Вестник Южно-Уральского государственного университета. Серия "Математическое моделирование и программирование". – 2011. – Вып. 10. – № 37(254). – С. 71–81.
8. Hallquist, J.O. LS-DYNA Keyword User's Manual v.970 / J.O. Hallquist – Livermore Software Technology Corporation, 2003. – 1564 p.

9. Martinez, M.A. Friction and wear behaviour of Kevlar fabrics / M.A. Martinez, C. Navarro, R. Cortes, J. Rodriguez, V. Sanchez-Galvez // Journal of materials science. – 1993. – Vol. 28. – P. 1305–1311.
10. Суперкомпьютер «СКИФ–Аврора ЮУрГУ»: URL: http://supercomputer.susu.ac.ru/computers/skif_avrora/ (дата обращения: 01.09.2012).
11. Исследование производительности суперкомпьютеров семейства "СКИФ Аврора" на промышленных задачах/ А.А. Московский, М.П. Перминов, Л.Б. Соколинский и др. // Вестник Южно-Уральского государственного университета. Серия "Математическое моделирование и программирование". – Вып. 6. – 2010. – № 35(211). – С. 66–78.

Наталья Юрьевна Долганина, кандидат технических наук, доцент кафедры системного программирования, начальник отдела поддержки и обучения пользователей Лаборатории суперкомпьютерного моделирования, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), dnu001@mail.ru.

Investigation of Impact Interaction of the Indenter With Fabric Armor Plate Which Located on the Clay Basis

N. Yu. Dolganina, South Ural State University (Chelyabinsk, Russian Federation)

Supercomputer simulation of the dynamic interaction between the indenter and textile armor plate is considered. Armor plate made of 1 - 5 layers of fabric 10x10 cm and located on the clay bottom. The fibers in textiles have the ability to break down. Mechanical characteristics of the backing material (technical clay) are defined. Comparison of the calculated and experimental results conducted.

Keywords: supercomputer simulation, FEA model, impact, backing material, textile armor plate.

References

1. GOST R 50744 – 95. Broneodezhda. Klassifikatsiya i obshchiye tekhnicheskiye trebovaniya [Armor Clothes. Classification and General Technical Requirements]. Prin. Postanovleniyem Gosstandarta Rossii ot 27.02.95 № 82. Vved. s izm. № 1, 2. Utverzhen v sent. 1998 g., maye 2002. Moscow: IPK izd-vo standartov, 2003. 8 p.
2. National Institute of Technology Standard, NIJ Standard 0101.06 Ballistic Resistance of Body Armor. July, 2008.
3. Bova V.G. Kontseptsiya postroyeniya myagkogo bronezhileta dlya maksimalnoy realizatsii svoystv aramidnykh nitay [The Concept of Building a Soft Body Armor to Maximize the Properties of Aramid Fibers] Rabochaya odezhda i sredstva individualnoy zashchity [Clothing and personal protective equipment]. 2001. No. 2(10). P. 11–15.
4. Nilakantan G., Keefe M., Wetzell E.D., Bogetti T.A., Adkinson R., Gillespie J.W. Using LS-DYNA[®] to Computationally Assess the V₀-V₁₀₀ Impact Response of Flexible Fabrics Through Probabilistic Methods // 11th International LS-DYNA[®] Users Conference: Proceedings of the International Users Conference (06-08 June 2010, Detroit). 2010. P. 5-1 – 5-12.

5. Dolganina N.Yu. Otsenka ballisticheskogo predela i progiba mnogosloynnykh tkanevykh plastin pri udare indentorom [Evaluation of Ballistic Limit and Multilayer Fabric Plate Deflection Under Indenter Impact] Vestnik Yuzhno-Uralskogo gosudarstvennogo universiteta. Seriya "Mashinostroyeniye" [Bulletin of South Ural State University. Series: Mechanical Engineering]. 2010. No. 10. P. 17-23.
6. Dolganina N.Yu., Sapozhnikov S.B., Maricheva A.A. Modelirovaniye udarnykh protsessov v tkanevykh bronezhiletakh i tele cheloveka na vychislitelnom klastere "SKIF Ural" [Modeling of Impact Processes in a Textile Armor and in a Human Body on a SKIF Ural Computational Cluster] Vychislitelnyye metody i programmirovaniye: Novyye vychislitelnyye tekhnologii [Numerical Methods and Programming: New Computing Technologies] 2010. Vol. 11. P. 117-126.
7. Dolganina N.Yu., Sapozhnikov S.B. Proyektirovaniye novykh konstruksiy tkanevykh bronepaneley s ispolzovaniyem superkompyuternykh vychisleniy [Design of new Constructions of Textile Armor Panel Using Supercomputing] Vestnik Yuzhno-Uralskogo gosudarstvennogo universiteta. Seriya "Matematicheskoye modelirovaniye i programmirovaniye" [Bulletin of South Ural State University. Series: Mathematical Modeling, Programming & Computer Software]. 2011. No. 37(254). Vol. 10. P. 71-81.
8. Hallquist J.O. LS-DYNA Keyword User's Manual v.970 / Livermore Software Technology Corporation, 2003. 1564 p.
9. Martinez M.A., Navarro C., Cortes R., Rodriguez J., Sanchez-Galvez V. Friction and Wear Behaviour of Kevlar Fabrics // Journal of materials science. 1993. Vol. 28. P. 1305 – 1311.
10. Supercomputer «SKIF-Avrora YuUrGU»: URL: http://supercomputer.susu.ac.ru/computers/skif_avrora/ (accessed: 01.09.2012).
11. Moskovskiy A.A., Perminov M.P., Sokolinskiy L.B., Cherepennikov V.V., Shamakina A.V. Issledovaniye proizvoditelnosti superkompyutеров semeystva "SKIF Avrora" na industrialnykh zadachakh [Performance Analysis of the Family of Supercomputers "SKIF Aurora" on Industrial Problems] Vestnik Yuzhno-Uralskogo gosudarstvennogo universiteta. Seriya "Matematicheskoye modelirovaniye i programmirovaniye" [Bulletin of South Ural State University. Series: Mathematical Modeling, Programming & Computer Software]. 2010. No. 35(211). P. 66–78.

Поступила в редакцию 5 ноября 2012 г.

КАЧЕСТВЕННОЕ ПОСТРОЕНИЕ РАСЧЕТНОЙ СЕТКИ ДЛЯ РЕШЕНИЯ ЗАДАЧ АЭРОДИНАМИКИ В ПРОГРАММНОМ КОМПЛЕКСЕ FLOWVISION

П.И. Карасев, А.С. Шишаева, А.А. Аксенов

В работе исследовано влияние адаптации сетки в различных областях течения газа около крылового профиля на результаты решения в программном комплексе FlowVision. На основе этого исследования даны рекомендации по выбору критериев адаптации расчетной сетки для эффективного решения задач аэродинамики. Выводы, сделанные в данной статье, могут быть распространены на другие задачи внешней аэродинамики, которые исследуют обтекание объектов более сложных, чем крыловой профиль.

Ключевые слова: вычислительная гидродинамика, расчетная сетка, адаптация к решению, профиль крыла.

Введение

Одним из важнейших этапов при численном моделировании движения жидкости или газа является построение расчетной сетки. Качественная расчетная сетка должна быть достаточно мелкой для того чтобы выявлять в процессе проведения расчета все основные особенности течения и, таким образом, обеспечивать достаточную точность результатов расчетов. В то же время она не должна содержать слишком большое количество ячеек, так как в этом случае увеличиваются требования к ресурсам компьютера. Необходимо также отслеживать сходимость по сетке. Проект считается сошедшимся по сетке, когда результат моделирования не изменяется при дальнейшем измельчении сетки. При этом наиболее экономично задавать подробную сетку в областях высоких градиентов переменных, и более грубую – в областях, где они малы.

Области, где необходимо задавать подробную сетку, можно задавать вручную или определять автоматически, исходя из значений переменных. Первый метод имеет ряд существенных недостатков. Во-первых, не всегда заранее известно, где будут расположены зоны высоких градиентов и насколько мелкая сетка потребуется для их разрешения. Во-вторых, при сложных трехмерных течениях ручное задание областей измельчения сетки либо потребует создания сложных геометрических объектов, точно воспроизводящих соответствующую область, либо, при использовании простых объектов, приведет к избыточному измельчению сетки.

Цель данного исследования - получение алгоритма, позволяющего создавать оптимальную расчетную сетку. Исследование проводится для задач дозвукового и околозвукового обтекания известных крыловых профилей в программном комплексе FlowVision, предназначенном для численного моделирования течений жидкости и газа.

1. Особенности расчетной сетки во FlowVision

Во FlowVision применяется неструктурированная локально-адаптивная сетка с подсеточным разрешением геометрии с преобладанием шестигранных ячеек [1]. Адаптация

подразумевает разбиение исходной ячейки пополам по каждому направлению. Адаптация может быть разного уровня. Уровень адаптации – это количество разбиений исходной ячейки (рис. 1). Деление ячеек на равные части позволяет избежать появления вырожденных и слишком вытянутых ячеек.

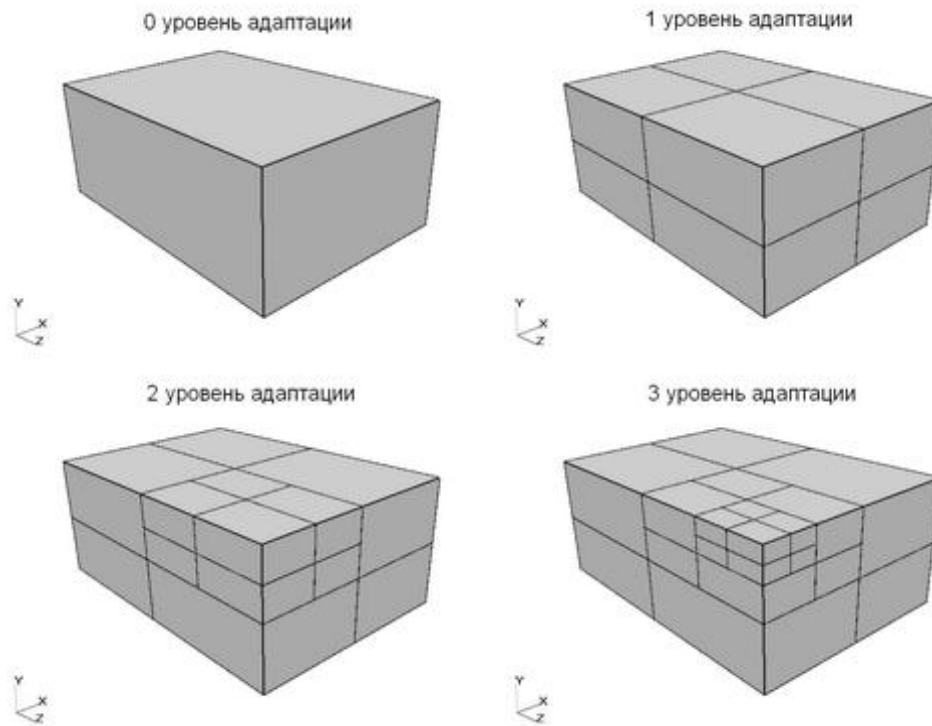


Рис. 1. Уровни адаптации в FlowVision

Вблизи границы расчетной области из ячеек начальной сетки вычитается часть геометрии, не принимающая участия в расчете. При этом расчетные ячейки принимают форму произвольных многогранников (рис. 2). Такая сетка называется сеткой «с подсеточным разрешением» [1].

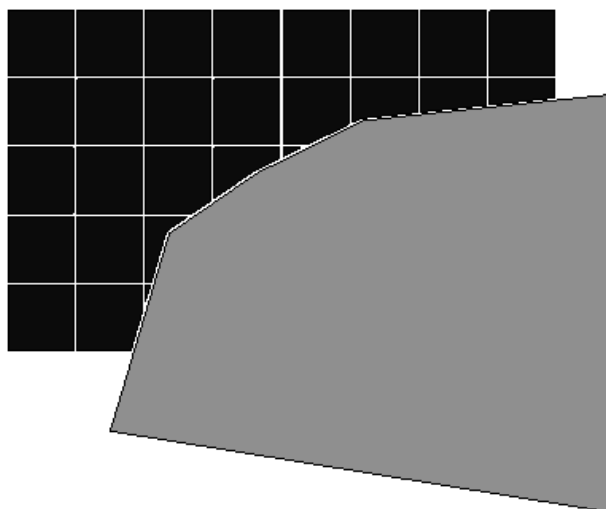


Рис. 2. Подсеточное разрешение геометрии

2. Выделение зон течения около крылового профиля для определения относительной важности их разрешения расчетной сеткой

Для оценки степени влияния разрешения различных областей течения расчетной сеткой условно выделим несколько зон рядом с профилем крыла (рис. 3).

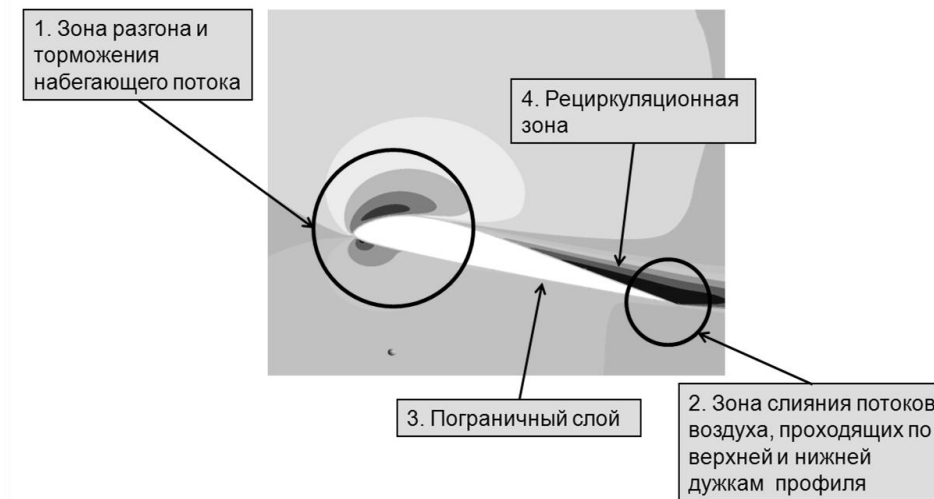


Рис. 3. Выделенные зоны около профиля

2.1. Зона разгона и торможения набегающего потока

В этой зоне поток, набегающий на профиль, тормозится в критической точке, расположенной рядом с передней кромкой. В критической точке поток разделяется. При удалении от нее скорость воздуха повышается из-за кривизны профиля, которая вызывает поджатие и, следовательно, ускорение потока воздуха. Так как поток, текущий по верхней части профиля испытывает большее сжатие, сильнее ускоряется и, соответственно, согласно закону Бернулли давление в нем падает сильнее, чем в потоке, текущем вдоль нижней части профиля. За счет этой зоны и возникает разность давлений, создающая подъемную силу.

2.2. Зона слияния потоков воздуха, проходящих по верхней и нижней дужкам профиля

Эта зона также характеризуется значительными градиентами скорости и давления, особенно заметными при наличии срыва потока. Поведение потока в этой области влияет на распределение давления по всему профилю вверх по потоку и на положение скачка уплотнения в случае трансзвукового обтекания профиля. Часто задняя кромка профиля бывает срезана и за ней образуется отрывная зона как показано ниже (рис. 4)

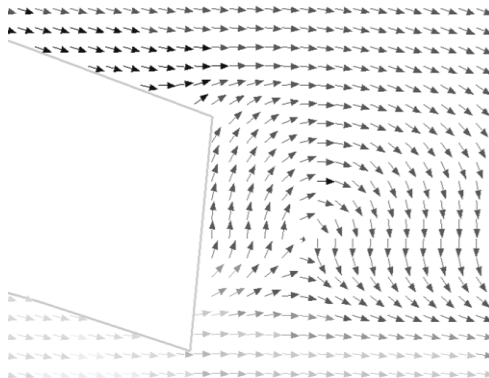


Рис. 4. Рециркуляционная зона при срезанной задней кромке профиля

2.3. Пограничный слой

Разрешение расчетной сеткой пограничного слоя рядом с обтекаемым объектом критически важно при решении аэродинамических задач.

2.4. Зона отрыва потока

Сильное влияние на характеристики профиля оказывает наличие отрыва потока и его положение. Вихри, образующиеся при отрывах потока очень чувствительны к точности разбиения расчетной сетки.

2.5. Скачок уплотнения

При околозвуковом обтекании профиля основное влияние на его характеристики оказывает положение скачков уплотнения рядом с верхней и нижней дужками профиля, поэтому область вокруг него можно выделить в отдельную зону (на рисунке данная зона не показана).

3. Исследование влияния на результат решения адаптации расчетной сетки в выделенных зонах около профиля NASA 0012

Для выявления важности адаптации в выделенных зонах (см. рис. 3) при безотрывном дозвуковом режиме обтекания профиля выбран профиль NASA 0012 – симметричный тонкий профиль, установленный под углом атаки 10 градусов.

Проведены расчеты для установления степени влияния на решение разрешения сеткой областей 1 и 2. Сравнение проводится с экспериментальными данными, взятыми из [2].

После предварительного расчета на начальной сетке были проведены расчеты с адаптацией в различных зонах (рис. 5).

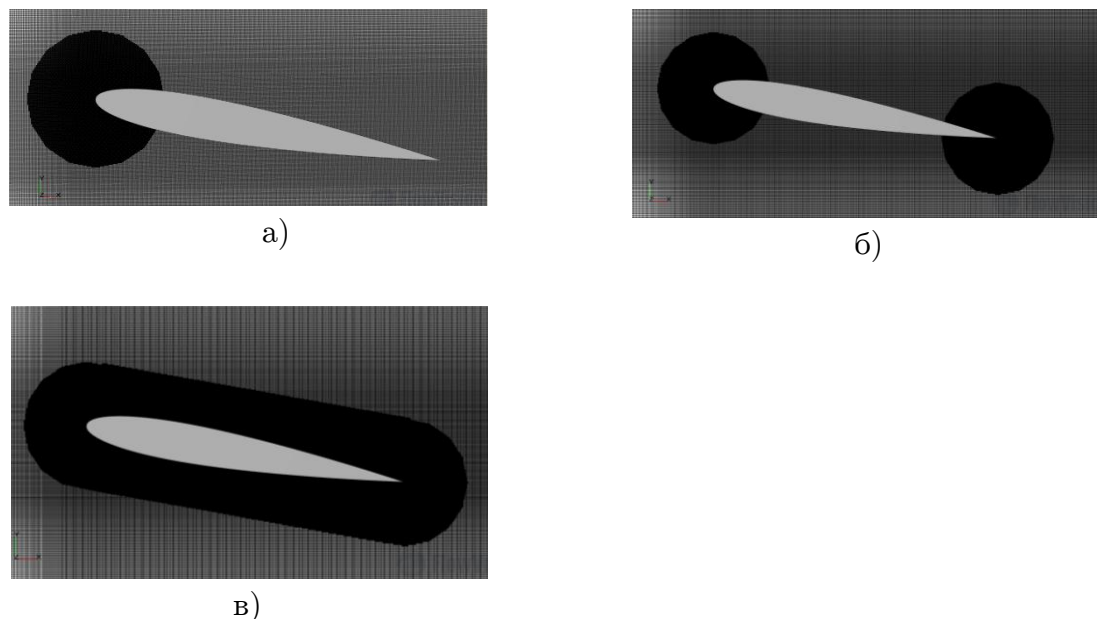


Рис. 5. Расчетные сетки для исследования влияния адаптации различных областей около крылового профиля NASA 0012 на результаты решения: с адаптацией рядом с передней кромкой профиля (а), с адаптацией рядом с передней и задней кромками (б), с адаптацией вокруг всего крылового профиля (в)

Результаты расчетов с показанной выше адаптацией сведены в таблицу (табл. 1).

Таблица 1

Сравнение результатов численного эксперимента для профиля NASA 0012

Способы адаптации	Значения			Погрешность относительно экспериментальных данных, %			Отличие погрешности от погрешности предыдущего расчета, %		
	C_x	C_y	m_z	ΔC_x	ΔC_y	Δm_z	ΔC_x	ΔC_y	Δm_z
без адаптации	0,0298	1,007	0,01175	54,5	-1,3	9,8			
рядом с передней кромкой	0,0259	1,043	0,00974	34,3	2,3	-8,9	13,10	-3,62	17,09
рядом с передней и задней кромками	0,0258	1,037	0,01128	33,7	1,7	5,4	0,46	0,61	-15,75
вокруг всего крылового профиля	0,0255	1,044	0,01070	31,9	2,3	0,0	1,31	-0,67	5,12

Получено, что на значение сил, действующих на профиль, наибольшее влияние оказывает измельчение сетки рядом с передней кромкой (зона 1), а на значение момента помимо этого сильно влияет адаптация рядом с задней кромкой профиля (зона 2).

4. Исследование влияния на результат решения адаптации расчетной сетки в выделенных зонах около профиля ЦАГИ Р-ША-15

Также был проведен численный эксперимент с адаптацией на профиле ЦАГИ Р-ША-15 под углом атаки 16° (рис. 6). На данном угле атаки образуется отрыв потока.

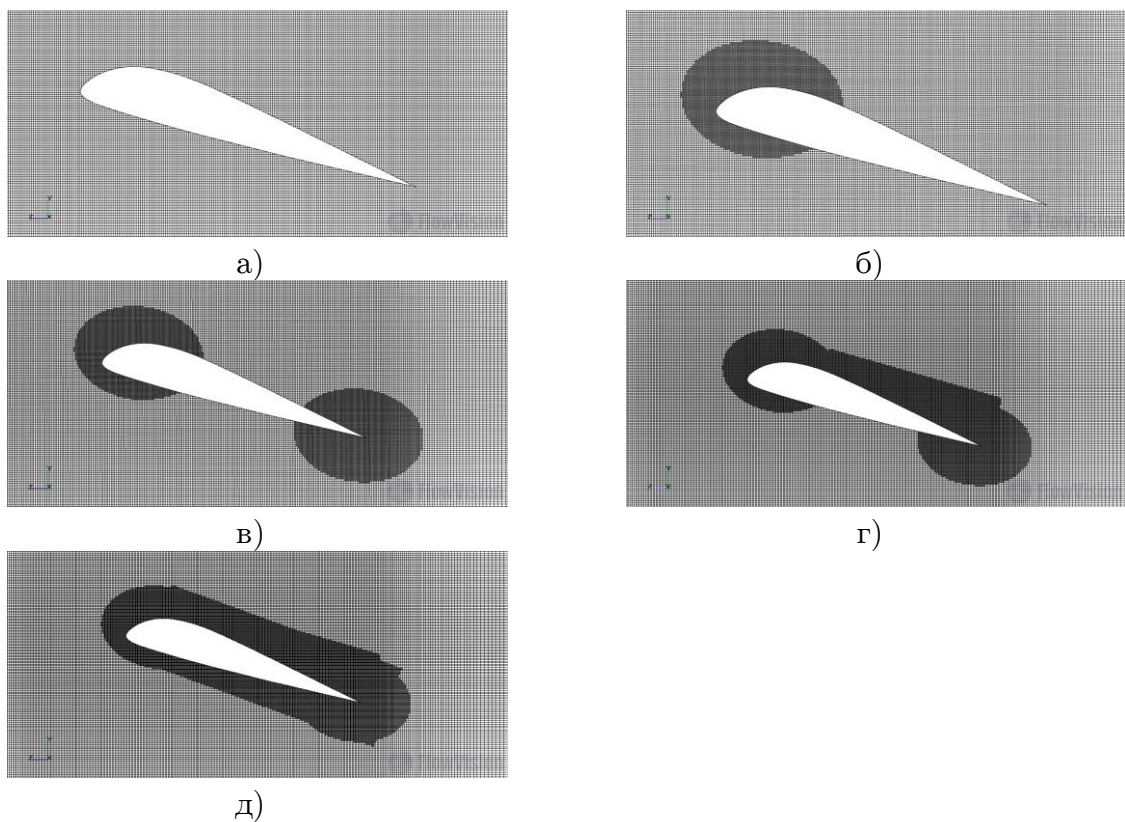


Рис. 6. Расчетные сетки при исследовании влияния адаптации различных областей около крылового профиля ЦАГИ Р-ША-15 на результаты решения: без адаптации (а), адаптация по передней кромке (б), адаптация по передней и задней кромке (в), адаптация по передней кромке, задней кромке и зоне отрыва потока (г), адаптация вокруг всего профиля (д)

Качественное построение расчетной сетки для решения задач аэродинамики...

Экспериментальные данные, сравнение с которыми производится в табл. 2, взяты из [3]. Получено, что разрешение зоны отрыва существенно влияет на результаты решения.

Таблица 2

Сравнение результатов численного эксперимента для профиля ЦАГИ Р-ША-15

Способы адаптации	Значения		Погрешность относительно эксперимента, %		Отличие погрешности от погрешности предыдущего расчета, %	
	Cx	Cy	Cx	Cy	Cx	Cy
без адаптации	0,07	1,50	52,61	-14,65		
адаптация рядом с передней кромкой	0,09	1,38	38,48	-5,54	14,13	-9,11
адаптация рядом передней и задней кромками	0,09	1,39	38,23	-5,85	0,26	0,31
адаптация по передней кромке, задней кромке и зоне отрыва потока	0,09	1,36	34,36	-3,51	3,87	-2,34
адаптация вокруг всего профиля	0,09	1,35	34,75	-3,15	-0,39	-0,36

5. Особенности адаптации к решению во FlowVision

Во FlowVision существует возможность автоматически адаптировать расчетную сетку в зависимости от значений и градиентов различных переменных в соответствующих ячейках. Она называется «адаптацией к решению». Уровень адаптации и максимальное количество ячеек расчетной сетки задается пользователем. Можно задавать разбиение по нескольким критериям. Вклад критерия в общее количество проадаптированных ячеек определяется весовым коэффициентом. При этом, все критерии адаптируют сетку только до одного определенного уровня.

Рассмотрим особенности адаптации во FlowVision на примере простейшей задачи, сверхзвуковом обтекании клина (рис. 7).

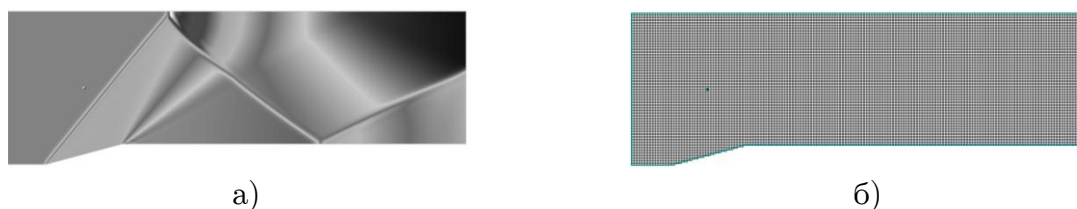


Рис. 7. а) Распределение давления и б) начальная сетка при сверхзвуковом обтекании клина

Проведем адаптацию по градиенту давления до первого уровня (рис. 8а).



Рис. 8. Расчетная сетка при адаптации по градиенту давления
а) первым и б) вторым уровнем

При изменении уровня адаптации до второго все ячейки первого уровня сливаются, и остается только адаптация второго уровня (рис. 8б). При увеличении уровня адаптации к решению и сохранении общего количества расчетных ячеек проадаптированные ячейки занимают меньший объем.

Если снова задать адаптацию первого уровня и то же количество ячеек расчетной сетки ячейки второго уровня слиты не будут. Сетка сохранит вид, показанный выше (рис. 8б). Если количество ячеек будет увеличено, области течения с меньшим градиентом будут проадаптированы до первого уровня (рис. 9а). При отсутствии градиента в какой-либо зоне расчетной области адаптация в ней проходить не будет.

Если уменьшить количество ячеек, выделенных под адаптацию к решению, часть проадаптированных ячеек в зонах с малыми градиентами или вдали от выбранного значения переменной сливается (рис. 9б). Важной особенностью является то, что создание адаптации к решению происходит за 1-2 шага, слитие же занимает значительно большее время.

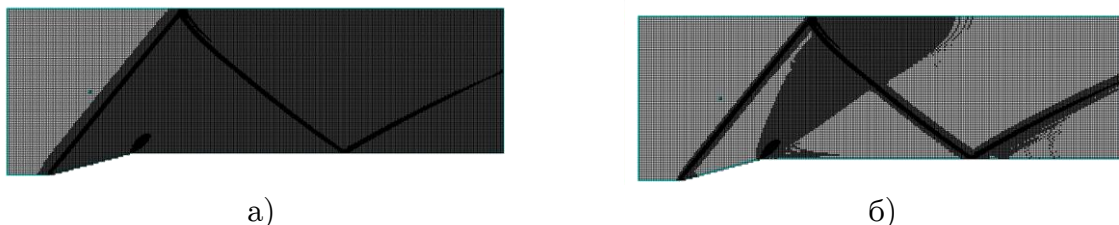


Рис. 9. Расчетная сетка а) при добавлении к предыдущей адаптации первого уровня и б) при уменьшении количества ячеек, использующихся при решении

6. Адаптация к решению для дозвукового обтекания профиля

Картина дозвукового обтекания профиля проще, чем картина трансзвукового обтекания. Поэтому разработку алгоритма применения адаптации к решению логично начать с дозвукового обтекания. Применение различных критериев адаптации при расчете дозвукового обтекания рассматривается на примере моделирования обтекания профилей ЦАГИ Р-ША-15 и НАСА 0012.

6.1. Адаптация к решению по градиенту давления

В результате расчетов получено, что адаптация по градиенту давления позволяет разрешать сеткой области разгона и торможения потока (рис. 10). Эти области оказывают наибольшее влияние на результаты при дозвуковом обтекании (см. табл. 1). Сле-

довательно, адаптацию по градиенту давления целесообразно использовать при моделировании дозвукового обтекания.

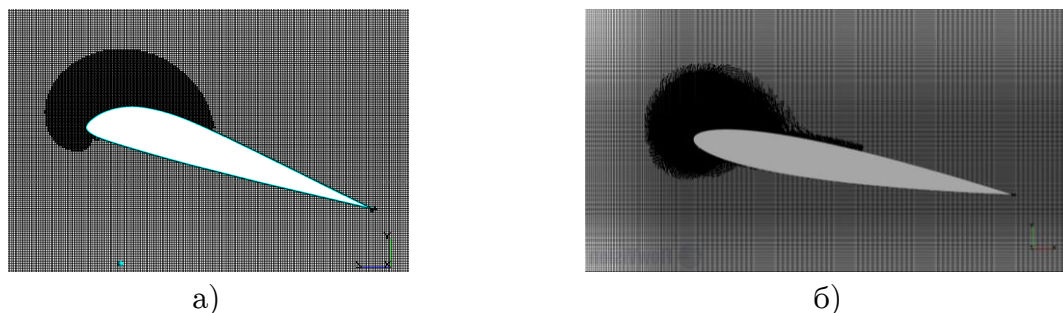


Рис. 10. Внешний вид сечения расчетной сетки при проведении адаптации к решению с использованием градиента давления а) для ЦАГИ Р-ША-15, б) для NASA 0012

6.2. Адаптация к решению по градиенту скорости

При использовании адаптации по градиенту скорости получено, что адаптация по этому критерию позволяет разрешать сеткой области разгона, торможения, пограничного слоя и слияния потока (рис. 11). Однако область слияния потока адаптируется по остаточному принципу. Пограничный слой может разрешаться частично, что может приводить к погрешностям в решении (рис. 12).

По вышеперечисленным причинам использование адаптации по градиенту скорости нецелесообразно.

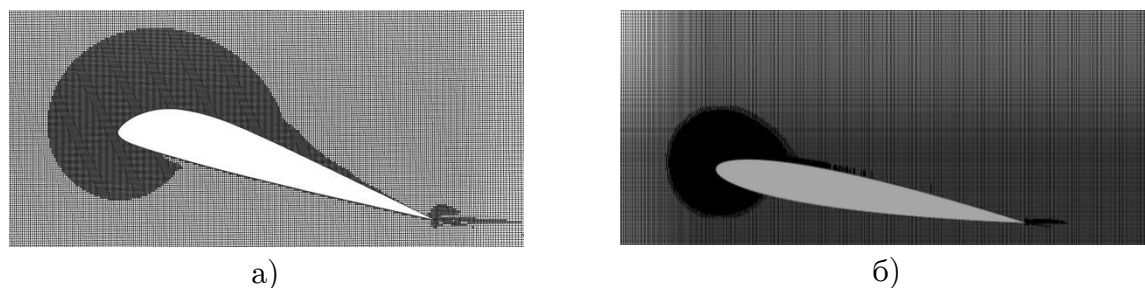


Рис. 11. Внешний вид сечения расчетной сетки при адаптации к решению с использованием градиента скорости а) для ЦАГИ Р-ША-15, б) для NASA 0012

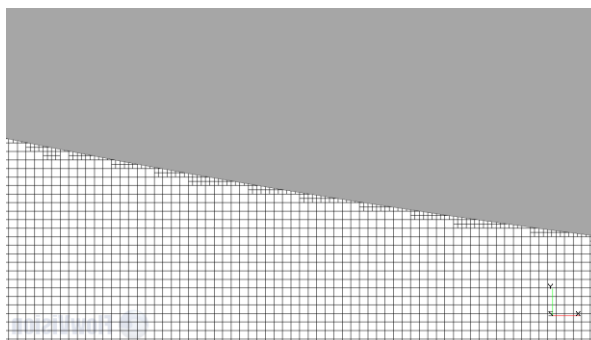


Рис. 12. Внешний вид сечения расчетной сетки рядом с профилем NASA 0012.

6.3. Адаптация к решению по значению скорости

В результате применения адаптации по нулевому значению скорости при моделировании обтекания профиля ЦАГИ Р-ША-15 преимущественно были проадаптированы ячейки в зоне отрыва потока. При моделировании обтекания профиля НАСА 0012, около критической точки и в области замедления потока за профилем (рис. 13). Получено, что адаптация по нулевому значению скорости позволяет разрешать сеткой области отрыва потока и критической точки. Следовательно, использование адаптации по значению скорости обосновано, если имеет место отрыв потока.

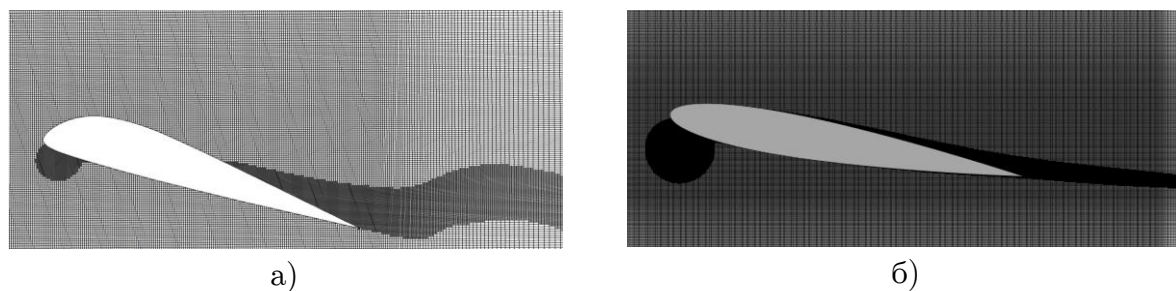


Рис. 13. Сечение расчетной сетки при адаптации по значению скорости а) для ЦАГИ Р-ША-15, б) для НАСА 0012

6.4. Адаптация к решению по расстоянию до стенки

Адаптация по расстоянию до стенки позволяет разрешать пограничный слой около профиля (рис. 14).

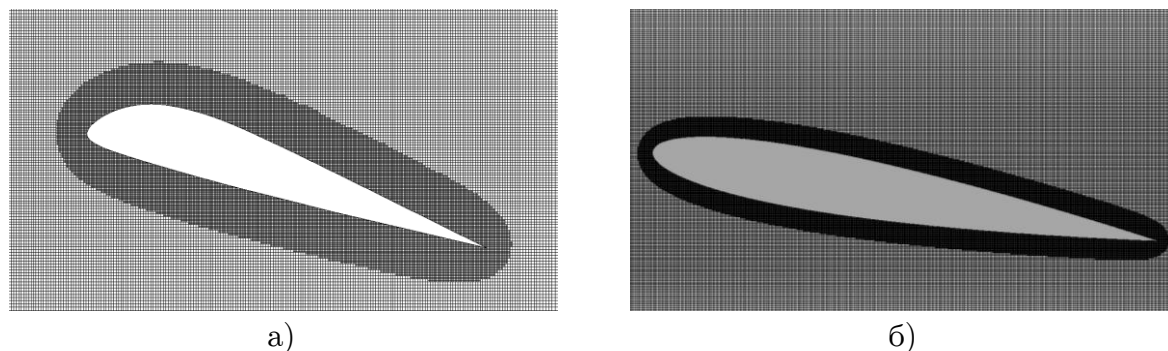


Рис. 14. Сечение расчетной сетки при адаптации по значению расстояния от стенки а) для ЦАГИ Р-ША-15, б) для НАСА 0012

7. Адаптация к решению для трансзвукового обтекания профиля

Течение при трансзвуковом обтекании профиля отличается от течения при дозвуковом обтекании наличием скачка уплотнения. Его положение определяет значения лобового сопротивления и подъемной силы для этих профилей. В остальном трансзвуковое течение вокруг профиля сходно с рассмотренным выше дозвуковым течением. Применение различных критериев адаптации расчете трансзвукового обтекания рассматривается на примере моделирования обтекания профиля НАСА 0012.

7.1. Адаптация к решению по градиенту давления

Было проведено моделирование с использованием адаптации по градиенту давления. Получено, что она позволяет разрешать области скачка, разгона и торможения потока (рис. 15). Следовательно, адаптацию по градиенту давления целесообразно использовать при моделировании трансзвукового обтекания.

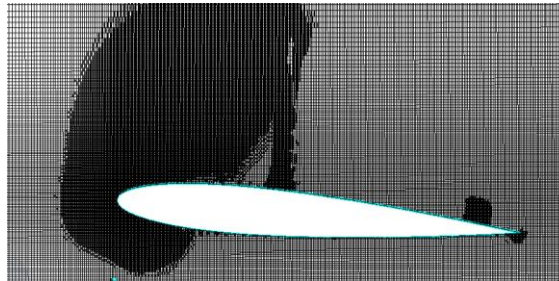


Рис. 15. Внешний вид сечения расчетной сетки при проведении адаптации к решению с использованием градиента давления для NASA 0012 при трансзвуковом режиме обтекания

7.2. Адаптация к решению использованием градиента скорости

При использовании адаптации по градиенту скорости, получено, что она также позволяет разрешать области скачка, разгона и торможения потока (рис. 16). Однако, поскольку эти области также хорошо разрешаются адаптацией по градиенту давления, совместное использование адаптации по градиенту давления и градиенту скорости нецелесообразно.

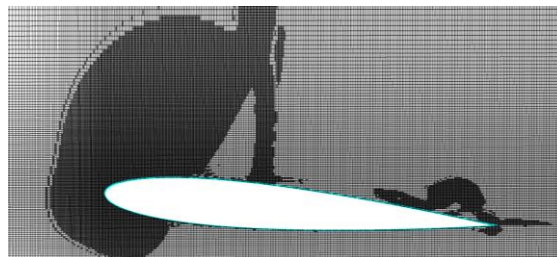


Рис. 16. Внешний вид сечения расчетной сетки при проведении адаптации к решению с использованием градиента скорости для NASA 0012 при трансзвуковом режиме обтекания

Необходимость добавлять для трансзвуковых профилей критерии адаптации, отличные от использовавшихся для дозвуковых профилей, нет.

8. Комбинация критериев адаптации

Из вышеизложенного следует, что для создания оптимальной расчетной сетки недостаточно использовать один критерий адаптации к решению. Необходима комбинация критериев адаптации. При безотрывном обтекании профиля целесообразно использовать адаптацию по градиенту скорости и адаптацию по расстоянию до стенки (рис. 17). При обтекании профиля с отрывом потока целесообразно добавлять к ним адаптацию по нулевому значению скорости. Для комбинации критериев адаптации предусмотрено задание весового коэффициента, который определяет, сколько ячеек от общего количества будет адаптировано с использованием этого критерия.

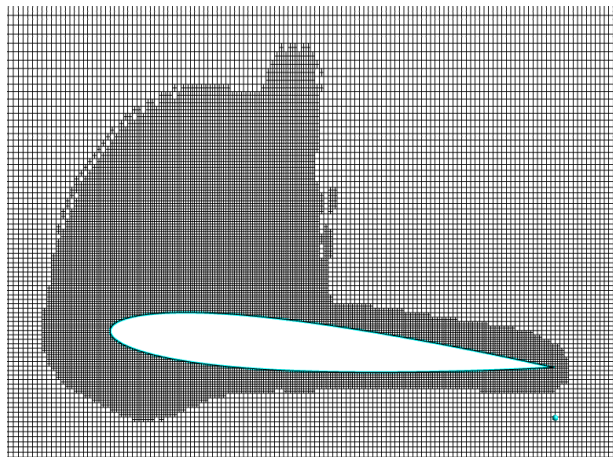


Рис. 17. Расчетная сетка при комбинации адаптации по градиенту давления и значению расстояния от стенки с весовыми коэффициентами 0.5 для каждого критерия.

9. Заключение

В результате проведенного исследования проанализирована применимость различных критериев адаптации расчетной сетки для получения количественного совпадения с экспериментальными данными при решении задач аэродинамики в программном комплексе FlowVision. Создан алгоритм адаптации расчетной сетки, позволяющий разрешать наиболее важные области течения, влияющие на точность моделирования аэродинамических характеристик. Сравнение расчетов с экспериментальными данными для профиля ЦАГИ Р-ША-15 и NACA 0012 показывает хорошую точность с практической точки зрения для коэффициентов лобового сопротивления, подъемной силы и момента крылового профиля.

Литература

1. Aksenov, A.A. Overcoming of Barrier between CAD and CFD by Modified Finite Volume Method / A.A. Aksenov, A.A. Dyadkin, V.I. Pokhilko // Proc. of "1998 ASME Pressure Vessels and Piping Division Conference" (San Diego, ASME PVP). – 1998. – Vol. 377-1.
2. Gregory, N. NPL 9615 and NACA 0012. A Comparison of Aerodynamic Data / N. Gregory, P.G. Wilby. Ministry of Defense, Aeronautical Research Council. – London: Her Majesty's Stationery office. – С.Р. No. 1261, 1973. – 56 p.
3. Кашафутдинов, С.Т. Атлас аэродинамических характеристик крыловых профилей / С.Т. Кашафутдинов, В.Н. Лушин. – Сибирский НИИ Авиации им. С.А. Чаплыгина, 1994. – 74 с.

ALGORITHM OF MESH ADAPTATION FOR ACCURATE AERODYNAMIC SIMULATION IN CFD CODE FLOWVISION

P.I.Karasev, Moscow Aviation Institute (National Research University) (Moscow, Russian Federation),

A.S.Shishaeva, Moscow Institute of Physics and Technology (State University) (Moscow, Russian Federation),

A.A.Aksenov, Moscow Institute of Physics and Technology (State University) (Moscow, Russian Federation),

Influence of mesh adaptation on getting accurate aerodynamic characteristics of airflow over wing profiles in CFD code FlowVision is investigated. Algorithm of adaptation criteria for accurate and fast solving aerodynamic problems is created on basis of this investigation. Conclusions done in this paper is possible to apply for solving different aerodynamic tasks more complex than airfoils.

Computational fluid dynamics, computational grid, adaptation to solution, airfoil, aerodynamic characteristics.

References

1. Aksenov A.A., Dyadkin A.A. Pokhilko V.I. Overcoming of Barrier between CAD and CFD by Modified Finite Volume Method. Proceedings of "1998 ASME Pressure Vessels and Piping Division Conference", San Diego, ASME PVP. 1998. Vol. 377-1.
2. Gregory N., Wilby P.G. NPL 9615 and NACA 0012. A Comparison of Aerodynamic Dat. Ministry of Defense, Aeronautical Research Council, London: Her Majesty's Stationery office. 1973. No. 1261. 56 p.
3. Kashafutdinov S.T., Lushin V.N. Atlas aerodinamicheskikh harakteristik krylovykh profilej [Atlas of the aerodynamic airfoils]. Sibirskij NII Aviacii im. S.A. Chaplygina [Siberian Research Institute of Aviation], 1994. 74 p.

Поступила в редакцию 5 ноября 2012 г.

ОБРАБОТКА ЗАПРОСОВ НА КЛАСТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ С МНОГОЯДЕРНЫМИ УСКОРИТЕЛЯМИ

П.С. Костенецкий

Работа посвящена вопросам моделирования процесса обработки запросов в мультипроцессорах параллельных систем баз данных. Предлагаются подходы для обработки запросов при помощи графических ускорителей, многоядерных сопроцессоров и центральных процессоров. На базе предложенных подходов реализован эмулятор параллельной СУБД. Приведены результаты вычислительных экспериментов и выполнен анализ эффективности предложенных подходов.

Ключевые слова: многоядерные ускорители, параллельная обработка запросов.

Введение

Важным классом приложений для многопроцессорных систем являются задачи, связанные с обработкой сверхбольших баз данных. Фактически единственным эффективным решением проблемы хранения и обработки сверхбольших баз данных является использование параллельных систем баз данных, обеспечивающих параллельную обработку запросов на многопроцессорных вычислительных системах. В области параллельных систем баз данных до сих пор остается ряд направлений, требующих дополнительных научных исследований [1]. Одно из них – дальнейшее развитие аппаратной архитектуры параллельных систем баз данных [19]. Разработка систем баз данных является одной из областей, где могут быть успешно применены многоядерные ускорители [3].

На сегодняшний день уже существуют работы, посвященные оптимизации процесса интеллектуального анализа данных [4, 17] и ускорению обработки запросов к базе данных с использованием графических ускорителей [2, 5]. Разработан алгоритм работы с индексом, учитывающий архитектурные особенности GPU [11]. Предложен метод ускорения операций над индексом, эффективно использующий большое число вычислительных ядер GPU [10]. Описаны алгоритмы сортировки [12]. В работе [9] описывается совместное использование GPU и CPU при обработке запросов, при котором запрос выполняется на GPU только в тех случаях, когда стоимость выполнения запроса на нем ниже, чем на центральном процессоре. Стоимость выполнения запроса вычисляется динамически на основе плана запроса. В работе [10] предлагаются алгоритмы группировки транзакций с целью их последующего выполнения на графическом ускорителе. Работа [6] предлагает использование GPU для оптимизации запросов. В работе [14] рассмотрена разработка прототипа СУБД, хранящей свои данные в памяти графического ускорителя. В некоторых работах проведена адаптация существующих СУБД для использования графических ускорителей и оценка эффективности такой адаптации. Так, в работе [8] поддержка вычислений с использованием GPU была интегрирована в СУБД Oracle 9, в работе [13] поддержка графических ускорителей была добавлена в СУБД WattDB, а в работе [5] использовалась СУБД SQLite.

Разработка параллельных систем баз данных [17], использующих многоядерные сопроцессоры и графические ускорители, осложнена наличием у данного оборудования ряда технических особенностей. В первую очередь, это ограничение скорости системной

шины, по которой данные передаются между центральным процессором и многоядерным ускорителем. Кроме того, ускорители имеют сравнительно небольшой объем оперативной памяти, что так же необходимо учитывать при составлении алгоритмов обработки баз данных. В связи с этим актуален вопрос оценки эффективности использования кластерных систем с многоядерными ускорителями и GPU для приложений баз данных. В свою очередь, для оценки эффективности подобных гибридных кластерных архитектур необходима разработка новых методов обработки запросов [16].

1. Методы обработки запросов

В ходе исследования были разработаны методы обработки наиболее распространенных типов запросов SELECT и JOIN. Методы были реализованы в виде программного эмулятора СУБД, позволяющего моделировать параллельную обработку запросов [18] к базе данных на многоядерном центральном процессоре и на графических ускорителях, поддерживающих технологию CUDA.

Реализация запроса SELECT. SELECT – оператор языка SQL, возвращающий набор данных (выборку) из базы данных, удовлетворяющих заданному условию. При формировании запроса SELECT описывается ожидаемый набор данных: его вид (набор атрибутов) и его содержимое (критерий попадания кортежа в набор, группировка значений, порядок вывода записей и др.). Запрос выполняется следующим образом: сначала извлекаются все записи из отношения, а затем для каждой записи набора проверяется ее соответствие заданному критерию. На рис. 1 приведена схема предлагаемого метода обработки запроса SELECT на графическом ускорителе.

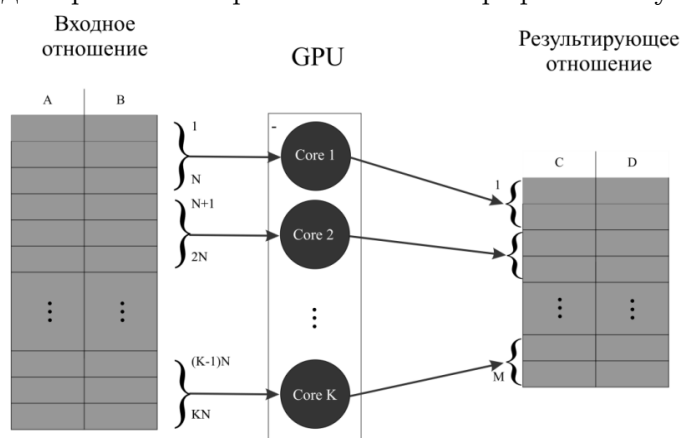


Рис. 1. Схема выполнения запроса SELECT на GPU

Алгоритм выполнения запроса SELECT на GPU состоит из следующих шагов:

- 1) генерация входного отношения R в оперативной памяти;
- 2) выделение памяти графического ускорителя под массив для результатов;
- 3) копирование отношения R из оперативной памяти в память графического ускорителя;
- 4) логическое разбиение отношения R на фрагменты, при котором количество фрагментов равно количеству нитей CUDA;
- 5) выполнение выборки из отношения R на графическом ускорителе, при котором каждая нить обрабатывает один фрагмент, а запись результатов проводится в массив созданный ранее в памяти графического ускорителя;

- б) копирование результата из памяти графического ускорителя в оперативную память.

Алгоритм выполнения запроса SELECT на многоядерном центральном процессоре состоит из следующих шагов:

- 1) генерация входного отношения R;
- 2) выполнение выборки из отношения R, при котором нити OpenMP в цикле обрабатывают кортежи и записывают результаты в результирующий динамический массив.

Реализация запроса JOIN. В работе был реализован запрос INNER JOIN, выполняющий внутреннее соединение двух отношений методом вложенных циклов. Заголовок результирующего отношения является объединением заголовков соединяемых отношений.

Результирующее отношение формируется следующим образом. Каждый кортеж одного отношения сопоставляется с каждым кортежем второго отношения, после чего проверяется условие соединения (вычисляется предикат соединения). Если условие истинно, кортежи добавляется в результирующее отношение. На рис. 2 приведена схема предлагаемого метода обработки запроса JOIN на графическом ускорителе.

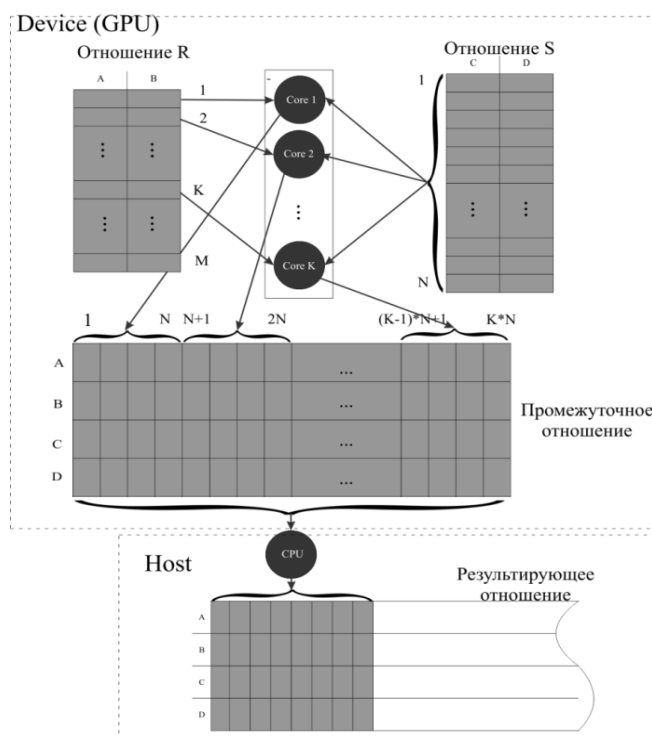


Рис. 2. Схема выполнения запроса JOIN

Алгоритм выполнения запроса INNER JOIN на графическом ускорителе состоит из следующих шагов:

- 1) генерация входных отношений R и S в оперативной памяти;
- 2) копирование отношений R и S из оперативной памяти в память графического ускорителя;
- 3) выделение памяти графического ускорителя под массив для результатов;
- 4) логическое разбиение отношения R на фрагменты, при котором количество кортежей во фрагменте равно количеству нитей CUDA;

- 5) выполнение соединения фрагмента отношения R и отношения S на графическом ускорителе, при котором каждая нить обрабатывает один кортеж R , а запись результатов проводится в массив, созданный ранее в памяти графического ускорителя;
- 6) копирование промежуточного результата из памяти графического ускорителя в оперативную память;
- 7) копирование промежуточного результата в результирующее отношение;
- 8) очистка массива для промежуточных результатов на GPU;
- 9) повторение шагов 5–9 для следующего фрагмента отношения R .

Алгоритм выполнения запроса JOIN на многоядерном центральном процессоре состоит из следующих шагов:

- 1) генерация входных отношений R и S ;
- 2) выполнение соединения отношений R и S , при котором нити OpenMP в цикле обрабатывают кортежи и записывают результаты в результирующий динамический массив.

2. Вычислительные эксперименты

Вычислительные эксперименты проводились с использованием разработанного эмулятора СУБД на оборудовании с характеристиками, приведенными в табл. 1.

Для проведения вычислительных экспериментов была сгенерирована тестовая база данных, состоящая из двух отношений R и S . Размеры отношений для операции соединения: $|R|=1\ 000$, $|S|=100\ 000$ кортежей. Для операции выборки использовалась база данных состоящая из одного отношения содержащего 75 000 000 кортежей. Отношения состоят из двух атрибутов и содержат данные типа *Integer*.

Таблица 1

Характеристики используемого оборудования

Оборудование	Характеристики	
Процессор	Intel Core i7-2600 (4 ядра по 3,8 ГГц)	
ОЗУ	8 Гб DDR3-1333 (пропускная способность 10,7 Гб/с)	
Жесткий диск	1 Тб SATA 2	
Системная шина	PCI Express 2.0 16X (пропускная способность 8 Гб/с)	
Графический ускоритель	Модель	NVIDIA GeForce GTX 550 Ti
	CUDA Cores	192
	Объем памяти	1024 Мб GDDR5 (4100 МГц)
	Разрядность памяти	192-bit
	Пропускная способность памяти	98,4 Гб/с

Исследование эффективности операции соединения для GPU. На рис. 3 приведена зависимость ускорения выполнения запроса INNER JOIN над тестовой базой данных на GPU при варьирующемся от 32 до 512 количестве нитей CUDA. Следует отметить, что используемый графический ускоритель имеет 192 физических ядра.

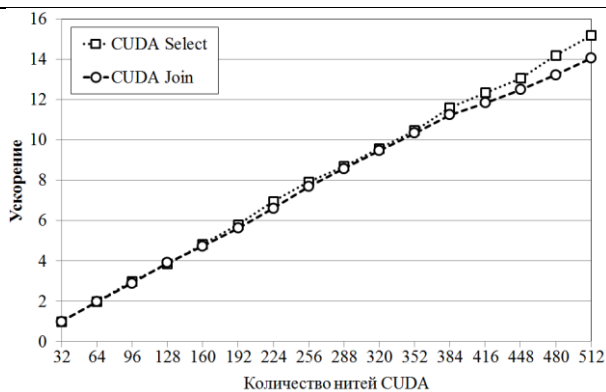


Рис. 3. Ускорение обработки запросов на GPU

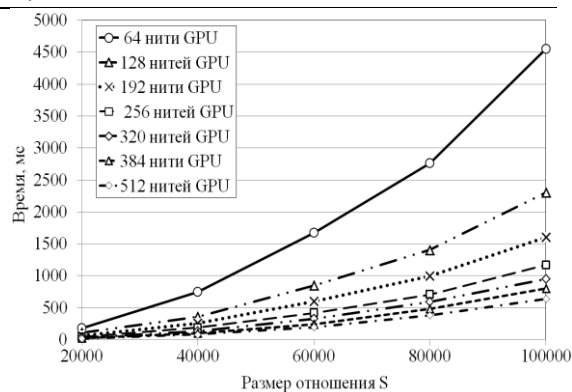


Рис. 4. Время выполнения запроса INNER JOIN на GPU

На рис. 4 приведено время выполнения запроса INNER JOIN на графическом ускорителе при варьирующемся количестве коротежей.

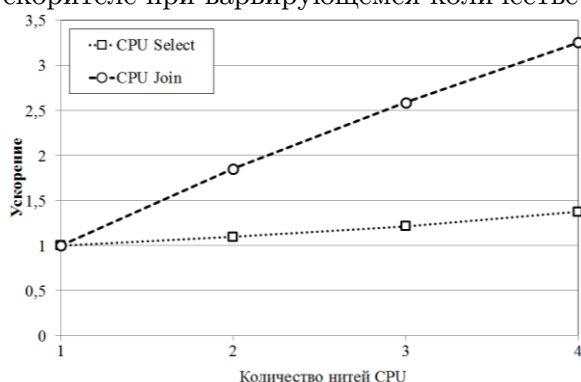


Рис. 5. Ускорение обработки запросов на CPU

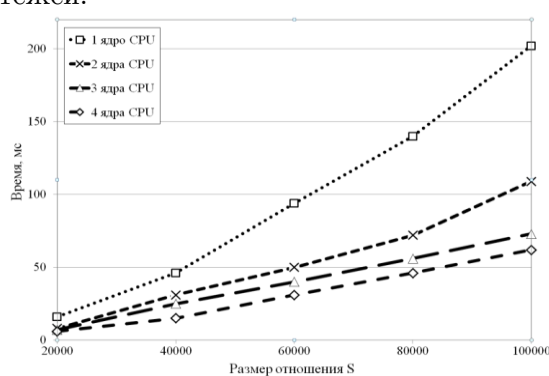


Рис. 6. Время выполнения запроса INNER JOIN на CPU

Исследование эффективности соединения для CPU. Для сравнения производительности различных аппаратных архитектур на рис. 5 и 6 приводятся графики ускорения и времени обработки запроса INNER JOIN на центральном процессоре при варьирующемся количестве нитей OpenMP. При реализации алгоритма JOIN на CPU использование количества нитей, превосходящего количество физических ядер, приводит к значительному снижению производительности, в отличие от реализации на CUDA. Из графиков видно, что использование CPU дает достаточно высокое ускорение. Например, использование четырех ядер CPU дает ускорение более чем в три раза. На основании данных экспериментов можно сделать вывод, что для операции JOIN при реализации вложенными циклами эффективнее использовать CPU, чем GPU. Это связано с низкой пропускной способностью шины PCI Express. Например, пропускная способность шины PCI Express 3.0, используемой в большинстве современных графических ускорителей, не превышает 32 ГБ/с, что накладывает серьезные ограничения на скорость обмена данными между GPU и оперативной памятью.

Исследование эффективности операции выборки для GPU. Как видно из графика ускорения обработки тестовой базы данных на GPU при варьирующемся количестве нитей CUDA (см. рис. 3), параллельная реализация операции SELECT на CUDA дает ускорение в 11,5 раза при использовании 512 нитей по сравнению с обработкой на 32 нитях.

Исследование эффективности операции выборки для CPU. Из графика ускорения обработки тестовой базы данных на CPU при варьирующемся количестве нитей OpenMP, представленного на рис. , видно, что для операции SELECT увеличение количества ядер приводит к небольшому приросту производительности. Например, использование четырех ядер создает ускорение в 1,3 раза по сравнению со временем обработки на одном ядре. После анализа графиков времени и ускорения можно сделать вывод, что использование CPU гораздо менее эффективно для данного типа запроса, чем использование GPU.

Если сравнивать производительность CPU и GPU, то для запроса SELECT использование GPU является более эффективным. При помощи технологии CUDA на бюджетном графическом ускорителе GeForce GTX 550 Ti достигнута эффективность ускорения работы запроса SELECT в 83 раза при обработке на 512 нитях GPU по сравнению с обработкой на четырех ядрах CPU Intel Core i7.

Заключение

В результате исследования определено, что с использованием многоядерных графических ускорителей можно добиться многократного ускорения обработки запроса SELECT или вычисления агрегатных функций. По сравнению с обработкой на 4 ядрах центрального процессора удалось добиться ускорения в 83 раза при обработке на 512 нитях GPU. В то же время значительно ускорить алгоритм JOIN с использованием GPU не удастся.

Исходя из результатов экспериментов, можно сделать вывод, что эффективным будет следующий комбинированный алгоритм работы системы баз данных. Обработка запроса JOIN выполняется на ядрах центрального процессора. В это же время на графическом ускорителе, установленном в вычислительном узле, выполняется обработка запросов SELECT или вычисление агрегатных функций, таких как MIN, MAX, SUM, AVG или COUNT. SELECT является наиболее распространенным запросом SQL, поэтому использование такой схемы работы может позволить значительно снизить нагрузку на центральные процессоры. Для реализации данного подхода может быть эффективным использование нескольких многоядерных ускорителей в рамках одного вычислительного узла кластера. Для реализации подобного комбинированного подхода к обработке запросов в сверхбольших базах данных потребуются новые аппаратные архитектуры систем баз данных, содержащие ускорители GPU либо MIC. Сборка прототипов подобных высокопроизводительных кластеров с гибридными узлами является чрезвычайно дорогостоящей. В связи с этим для выбора оптимальной конфигурации системы баз данных предварительно необходимо выполнять моделирование возможных аппаратных архитектур при помощи математических моделей [15].

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 12-07-31082.

Литература

1. Agrawal R., Ailamaki A., Bernstein P.A. et al. The Claremont Report on Database Research // Communications of the ACM, 2009. – Vol. 52, No. 6. – P. 56–65.

2. Bakkum P., Skadron K. Accelerating SQL Database Operations on a GPU with CUDA // The 3rd Workshop on General-Purpose Computation on Graphics Processing Units, Pittsburgh, USA, March 14, 2010, Proceedings. ACM, 2010. – P. 94–103.
3. Blas A.D., Kaldewey T. Data Monster // IEEE spectrum, 2009. – Vol. 46, No. 9.
4. Ding S., He J., Yan H., Suel T. Using Graphics Processors for High Performance IR Query Processing // The 18th international conference on World Wide Web, New York, USA, April 20– 24, 2009, Proceedings. ACM, 2009. – P. 421–430.
5. Govindaraju N., Lloyd B., Wang W., et al. Fast Computation of Database Operations Using Graphics Processors // ACM SIGGRAPH 2005 Courses, New York, USA. ACM, 2005. – P. 206.
6. Heimel M., Markl V. A First Step Towards GPU-assisted Query Optimization // The Third International Workshop on Accelerating Data Management Systems using Modern Processor and Storage Architectures, Istanbul, Turkey, August 27, 2012. – P. 1–12.
7. Bakkum P., Skadron K. Accelerating SQL Database Operations on a GPU with CUDA // 3rd Workshop on General Purpose Computation on Graphics Processing Units, New York, USA, March 14, 2010, Proceedings. ACM, 2010. – P. 94–103
8. Bandi N., Sun C., Agrawal D., Abbadi A.E. Hardware Acceleration in Commercial Databases: a Case Study of Spatial Operations. The 30th International Conference on Very Large Data Bases, August 31 – September 3, 2004, – Vol. 30, Proceedings. VLDB Endowment, 2004. – P. 1021–1032.
9. He B., Lu M., Yang K., Fang R, et.al. Relational query coprocessing on graphics processors. ACM Trans. Database Syst., – Vol. 34(4), ACM, 2009. – P. 21:1–21:39
10. He B., Xu Y.J. Highthroughput transaction executions on graphics processors // VLDB Endowment, Seattle, Washington, USA, August 29 – September 3, 2011, Proceedings. – Vol. 4, No. 5. VLDB Endowment, 2011. – P. 314–325.
11. Kim C., Chhugani J., Satish N. FAST: Fast Architecture Sensitive Tree Search on Modern CPUs and GPUs // ACM SIGMOD International Conference on Management of data, Indianapolis, USA, June 6–10, Proceedings. ACM, 2010. – P. 339–350.
12. Satish N., Kim C., Chhugani J, et. al. Fast Sort on CPUs and GPUs: a Case for Bandwidth Oblivious SIMD Sort. The 2010 ACM SIGMOD International Conference on Management of data, New York, USA, 2010, Proceedings. ACM. 2010. – P. 351–362
13. Vitor U.R. A GPU Operations Framework for Wattdb. Technical report. – Germany, Kaiserslautern: University of Kaiserslautern, 2012.
14. Hansen C.E. Christiansen M. CUDA DBMS. Technical report. – Denmark, Copenhagen: Aalborg University, 2009.
15. Костенецкий П.С., Лепихов А.В., Соколинский Л.Б. Технологии параллельных систем баз данных для иерархических многопроцессорных сред // Автоматика и телемеханика. – 2007. № 5. – С. 112–125.
16. Лепихов А.В., Соколинский Л.Б. Обработка запросов в СУБД для кластерных систем // Программирование. 2010. – № 4. – С. 25–39.
17. Пан К.С., Цымблер М.Л. Разработка параллельной СУБД на основе последовательной СУБД PostgreSQL с открытым исходным кодом // Вестник ЮУрГУ. Серия "Математическое моделирование и программирование". 2012. – Вып. 12. – № 18(277).– С. 112–120.

18. Соколинский Л.Б. Организация параллельного выполнения запросов в многопроцессорной машине баз данных с иерархической структурой // Программирование. 2001. – № 6. – С. 13–19.
19. Соколинский Л.Б. Обзор архитектур параллельных систем баз данных // Программирование. 2004. – № 6. – С. 49–63.

Костенецкий Павел Сергеевич, к.ф.-м.н., директор суперкомпьютерного центра, Южно-Уральский государственный университет, Kostenetskiy@gmail.com

Query Processing on Cluster Based Systems with Multicore Accelerators

P.S. Kostenetskiy, South Ural State University (Chelyabinsk, Russian Federation),

The paper is devoted to the problem of modeling query execution process in multiprocessors of parallel database systems. Original approaches to the query execution process on GPU, MIC and multicore CPU are presented. Based on this approach, a simulator of parallel DBMS is developed. Results of computational experiments are presented, and analysis of efficiency of the proposed approaches is performed.

Keywords: parallel query processing, database multiprocessor model.

References

1. Agrawal R., Ailamaki A., Bernstein P.A. et al. The Claremont Report on Database Research // Communications of the ACM, 2009. – Vol. 52, No. 6. – P. 56–65.
2. Bakkum P., Skadron K. Accelerating SQL Database Operations on a GPU with CUDA // The 3rd Workshop on General-Purpose Computation on Graphics Processing Units, Pittsburgh, USA, March 14, 2010, Proceedings. ACM, 2010. – P. 94–103.
3. Blas A.D., Kaldewey T. Data Monster // IEEE spectrum, 2009. – Vol. 46, No. 9.
4. Ding S., He J., Yan H., Suel T. Using Graphics Processors for High Performance IR Query Processing // The 18th international conference on World Wide Web, New York, USA, April 20–24, 2009, Proceedings. ACM, 2009. – P. 421–430.
5. Govindaraju N., Lloyd B., Wang W., et al. Fast Computation of Database Operations Using Graphics Processors // ACM SIGGRAPH 2005 Courses, New York, USA. ACM, 2005. – P. 206.
6. Heimel M., Markl V. A First Step Towards GPU-assisted Query Optimization // The Third International Workshop on Accelerating Data Management Systems using Modern Processor and Storage Architectures, Istanbul, Turkey, August 27, 2012. – P. 1–12
7. Bakkum P. and Skadron K. Accelerating SQL Database Operations on a GPU with CUDA // 3rd Workshop on General Purpose Computation on Graphics Processing Units, New York, USA, March 14, 2010, Proceedings. ACM, 2010. – P. 94–103
8. Bandi N., Sun C., Agrawal D., Abbadi A.E. Hardware Acceleration in Commercial Databases: a Case Study of Spatial Operations. The 30th International Conference on Very Large Data Bases, August 31 – September 3, 2004, – Vol. 30, Proceedings. VLDB Endowment, 2004. – P. 1021–1032.
9. He B., Lu M., Yang K., Fang R., et.al. Relational query coprocessing on graphics processors. ACM Trans. Database Syst., – Vol. 34(4), ACM, 2009. – P. 21:1–21:39

10. He B., Xu Y.J. Highthroughput transaction execution on graphics processors // VLDB Endowment, Seattle, Washington, USA, August 29 – September 3, 2011, Proceedings. – Vol. 4, No. 5. VLDB Endowment, 2011. – P. 314–325.
11. Kim C., Chhugani J., Satish N. FAST: Fast Architecture Sensitive Tree Search on Modern CPUs and GPUs // ACM SIGMOD International Conference on Management of data, Indianapolis, USA, June 6–10, Proceedings. ACM, 2010. – P. 339–350.
12. Satish N., Kim C., Chhugani J, et. al. Fast Sort on CPUs and GPUs: a Case for Bandwidth Oblivious SIMD Sort. The 2010 ACM SIGMOD International Conference on Management of data, New York, USA, 2010, Proceedings. ACM. 2010. – P. 351–362
13. Vitor U.R. A GPU Operations Framework for Wattdb. Technical report. – Germany, Kaiserslautern: University of Kaiserslautern, 2012.
14. Hansen C.E. Christiansen M. CUDA DBMS. Technical report. – Denmark, Copenhagen: Aalborg University, 2009.
15. Kostenetskii P.S., Lepikhov A.V., Sokolinskii L.B. Technologies of parallel database systems for hierarchical multiprocessor environments // Automation and Remote Control. 2007. – Vol. 68, No. 5. – P. 847-859.
16. Lepikhov A.V., Sokolinsky L.B. Query Processing in a DBMS for Cluster Systems // Programming and Computer Software. 2010. – Vol. 36. No. 4. – P. 205–215.
17. Pan C.S., Zymbler M.L. Razrabotka paralelnoj SUBD na osnove posledovatelnoj SUBD PostgreSQL s otkryтым ishodnym kodom [Development of a Parallel Database Management System on the Basis of Open-Source PostgreSQL DBMS]. Vestnik Yuzho-Uralskogo gosudarstvennogo universiteta. Seriya "Matematicheskoe modelirovanie i programmirovaniye" [Bulletin of South Ural State University. Series: Mathematical Modeling, Programming & Computer Software]. 2012. No. 18(277). Vol. 12. P. 112–120.
18. Sokolinsky L.B. Organization of Parallel Query Processing in Multiprocessor Database Machines with Hierarchical Architecture // Programming and Computer Software. 2001. – Vol. 27. No. 6. – P. 297–308.
19. Sokolinsky L.B. Survey of Architectures of Parallel Database Systems // Programming and Computer Software. 2004. – Vol. 30. № 6. – P. 337–346.

Поступила в редакцию 5 ноября 2012 г.

ПОДХОДЫ К ОПТИМИЗАЦИИ И РАСПАРАЛЛЕЛИВАНИЮ ВЫЧИСЛЕНИЙ В ЗАДАЧЕ ДЕТЕКТИРОВАНИЯ ОБЪЕКТОВ РАЗНЫХ КЛАССОВ НА ИЗОБРАЖЕНИИ¹

*Е.А. Козинев, В.Д. Кустикова, И.Б. Мееров, А.Н. Половинкин,
А.А. Сиднев*

Рассматривается задача детектирования объектов разных классов на статических изображениях: фотографиях или отдельных кадрах видеопотока. Описывается схема решения данной задачи с использованием алгоритма Latent SVM. Используется известный подход к ускорению вычислений — построение каскада классификаторов. Описывается вычислительная схема решения задачи детектирования с помощью каскадного Latent SVM. Обсуждаются проблемы распараллеливания и оптимизации времени поиска объектов одного класса на изображении. Проводится анализ вариантов решения указанных проблем. Выделяются наиболее трудоемкие участки реализаций, рассматриваются различные схемы распараллеливания, оцениваются их преимущества и недостатки. Приводятся результаты вычислительных экспериментов на базе изображений PASCAL Visual Object Challenge 2007, дается их анализ, а также формулируются выводы и планы по дальнейшему развитию.

Ключевые слова: детектирование объектов, алгоритм Latent SVM, каскадный классификатор, распараллеливание.

Введение

В настоящее время при решении многих практических задач используются системы компьютерного зрения (системы видеонаблюдения, управления процессами, организации информации и др.). Ядром вычислений в таких системах является обработка видеоданных. Поскольку основное применение результатов компьютерного зрения сосредоточено в области промышленной робототехники (автономные роботы, системы визуального контроля и измерений), то ключевым становится не только качество, но и скорость обработки видеоданных. Центральная задача, которая решается многими исследовательскими группами, — это анализ качественного состава сцены. При этом исследования ведутся как в направлении повышения точности распознавания объектов на кадрах видеопотока, так и в направлении снижения времени обработки видеoinформации.

В работе рассматривается задача детектирования объектов разных классов (автомобили, автобусы, люди и др.) на изображениях с использованием алгоритма поиска Latent SVM [4]. Приводится вычислительная схема решения данной задачи. Описывается один из наиболее общих алгоритмических подходов к ускорению вычислений в задачах компьютерного зрения — использование каскада классификаторов — на примере модификации Latent SVM [5]. Обсуждаются проблемы оптимизации и распараллеливания вычислений в системах с общей памятью. Выполняется анализ различных вариантов решения указанных проблем, предлагаются результаты вычислительных экспериментов на базе изображений PASCAL Visual Object Challenge

¹Статья рекомендована к публикации программным комитетом международной научной конференции «Параллельные вычислительные технологии 2012».

2007 (VOC2007, <http://pascallin.ecs.soton.ac.uk/challenges/VOC>). Разработка выполнена в рамках широко известной библиотеки компьютерного зрения OpenCV (<http://sourceforge.net/projects/opencvlibrary>).

1. Задача детектирования объектов на изображении с использованием методов, основанных на извлечении признаков

1.1. Постановка задачи

Задача детектирования объектов состоит в том, чтобы определить наличие объекта на изображении и найти его положение в системе координат пикселей исходного изображения. Положение объекта в зависимости от выбора алгоритма детектирования может определяться координатами прямоугольника, окаймляющего объект, либо контуром этого объекта, либо координатами набора точек, наиболее характерных для объекта. В данной работе считается, что положение определяется координатами окаймляющего прямоугольника.

Все вычисления должны выполняться в реальном времени, поэтому основная цель настоящей работы состоит в том, чтобы максимально уменьшить среднее время поиска за счет оптимизации и распараллеливания вычислений.

1.2. Общая схема решения

Один из возможных подходов к решению задачи детектирования состоит в том, чтобы использовать алгоритмы машинного обучения для построения моделей классов объектов и алгоритмы вывода для поиска объектов на изображении.

Построение модели состоит из двух этапов:

1. Извлечение признаков, характерных для объектов класса, – построение характеристических векторов-признаков для особых точек объекта (углов, линий, ребер [15] или контуров объектов [16]) или для всего объекта.
2. Построение модели объекта на полученных признаках для последующего распознавания объектов. На данном этапе выполняется тренировка какого-либо классификатора.

Методы, основанные на извлечении признаков, описывают объект с использованием векторов-признаков. Вектора строятся на основании цветовой информации (гистограмма ориентированных градиентов (*Histogram of Oriented Gradients* или *HOG*) – один из наиболее популярных способов). Также может быть использована контекстная информация (*context based*) [17, 18], а в некоторых случаях – данные о геометрии и взаимном расположении частей объекта (*part-based*) [4]. Тем не менее, все эти методы строят некоторую математическую модель объекта на каждом изображении тренировочной выборки, содержащем объект. Формально признак является числовой характеристикой. Таким образом, объект описывается набором векторов признаков в характерных точках. В результате тренировки строится модель, содержащая «усредненные» вектора признаков.

Алгоритм вывода (поиска) по существу включает два этапа:

1. Извлечение признаков объекта из тестового изображения согласно алгоритму, который использовался в процессе тренировки. При извлечении признаков возникает две основные проблемы:
 - на изображении может быть много объектов одного класса, а необходимо найти всех представителей. Поэтому необходимо просматривать все части изображения, проходя «бегущим» окном (*sliding window*) от левого верхнего до правого нижнего угла. При этом размер окна определяется размером изображений тренировочной выборки;
 - объекты на изображении могут иметь разный масштаб. Самое распространенное решение – масштабирование изображения и построение пирамиды изображений.
2. Поиск объектов на изображении. Входными данными этапа поиска являются формальное описание объекта – набор признаков, которые выделены из тестового изображения, – и модель класса объектов. На основании этой информации классификатор принимает решение о принадлежности объекта классу. Некоторые методы поиска также оценивают степень достоверности того, что объект принадлежит рассматриваемому классу.

Качество методов в основном зависит от того, насколько хорошо выбраны признаки, т.е. насколько хорошо эти признаки дифференцируют классы объектов. Существуют специализированные методы, основанные на извлечении признаков, например, для детектирования лиц [7, 8, 10], транспортных средств [11] и пешеходов [12–14].

2. Поиск объектов с использованием алгоритма Latent SVM

2.1. Модель объектов некоторого класса

При построении алгоритма Latent SVM модель объекта некоторого класса определяется набором компонент, каждая из которых соответствует одному из ракурсов этого объекта. Компонента включает совокупность фильтров [4]:

- грубый фильтр, определяющий набор векторов признаков, наиболее характерных для всего объекта;
- совокупность точных фильтров, описывающих отдельные части объекта.

2.2. Вычислительная схема поиска объектов

Алгоритм поиска Latent SVM состоит из двух этапов:

1. Построение пирамиды признаков.
2. Определение наиболее вероятных положений объектов заданного класса.

Построение пирамиды признаков включает масштабирование изображений – формирование пирамиды изображений [1, 4] – и построение матриц векторов-признаков для каждого изображения в пирамиде. Вектор признаков представляет гистограмму ориентации градиентов (*Histogram of Oriented Gradients*, [4]) для некоторого блока пикселей исходного изображения.

Смысл этапа поиска положения объекта состоит в том, чтобы некоторым образом оценить вероятность нахождения объекта во всех возможных положениях пирамиды

признаков и выбрать среди всего множества наиболее вероятное. Для определения положения объекта в алгоритмах [4, 5] вычисляется значение оценочной функции для каждого возможного положения. Расположение объекта определяется положением левого верхнего угла грубого фильтра в наборе векторов матрицы признаков какого-либо уровня. Оценочная функция строится как сумма скалярных произведений наборов векторов признаков грубого и точных фильтров модели с соответствующими наборами векторов признаков, построенных на изображении. В дальнейшем сумму скалярных произведений векторов фильтра и изображения будем называть сверткой. Дополнительно оценочная функция содержит слагаемое, которое определяет штраф за чрезмерное удаление частей объекта от самого объекта. Коэффициенты функции штрафа являются параметрами компоненты модели.

3. Каскад классификаторов

3.1. Схема построения каскада классификаторов

Построение и использование каскада классификаторов – один из наиболее распространенных подходов [2, 3] к ускорению вычислений при решении различных задач классификации. По существу идея построения аналогична AdaBoost-классификатору (*adaptive boost*, [6]). На каждой стадии тренировки строится некоторый классификатор, при этом учитывается информация об ошибках предшествующего классификатора. Итоговый классификатор представляется комбинацией построенных классификаторов.

При решении задач детектирования и классификации объектов на изображениях последовательность классификаторов, полученных на этапе тренировки, сортируется в зависимости от величины ошибки. Применение первых более слабых классификаторов позволяет последовательно отсекавать положения, в которых заведомо не может быть обнаружен объект. Таким образом, сильные классификаторы (как правило, вычислительно более трудоемкие) применяются не ко всем возможным положениям объекта, а только к некоторому небольшому набору.

Каскад классификаторов нашел применение при решении многих задач компьютерного зрения. Наиболее известным алгоритмом, в основе которого лежит каскад классификаторов, является алгоритм Виолы-Джонса для детектирования лиц [7, 8]. Идея построения каскада также используется при реализации каскадного преобразования Хафа [9], которое позволяет искать структурные объекты на изображении (линии, окружности и т.п.), и многих других алгоритмов. Далее рассмотрим каскадную схему алгоритма Latent SVM, предназначенного для детектирования объектов разных классов.

3.2. Детектирование объектов с использованием каскадного Latent SVM

3.2.1. Модель объектов некоторого класса

В каскадном Latent SVM [5] в модель объектов каждого класса включается дополнительная информация:

- PCA-проекции грубого и точных фильтров в пространство меньшей размерности;

- по два порога для каждого фильтра и его проекции:
 - порог идентификации присутствия объекта или его части, который определяет возможность присутствия объекта или его части в определенном месте изображения;
 - порог гипотетического отклонения определяет, насколько часть объекта может быть смещена от своего идеального расположения относительно грубого фильтра.

3.2.2. Вычислительная схема поиска объектов

Каскадный алгоритм Latent SVM [5] включает два этапа каскадной схемы, на которых выполняются следующие действия:

1. Построение множества возможных положений объекта. Для этого применяется преобразованная дополнительная пирамида признаков и соответствующие ей фильтры. На данном этапе осуществляется:
 - (a) Вычисление свертки грубого фильтра (РСА) в первом возможном положении (например, в левом верхнем углу изображения на одном из уровней). Если свертка меньше порога идентификации для грубого фильтра, то необходимо выполнить переход к рассмотрению следующего возможного положения грубого фильтра.
 - (b) Перебор сверток точных фильтров.
 - i. Выбор первого фильтра и вычисление его свертки с векторами матрицы признаков в некоторой окрестности грубого фильтра.
 - ii. Проверка условия: если сумма свертки точного фильтра и текущей оценки положения объекта меньше порога гипотетического отклонения, то положение грубого фильтра не рассматривается.
 - iii. Выбор положения с максимальной суммой текущей оценки и свертки среди всех подсчитанных положений. Сумма принимается за текущую оценку положения объекта.
 - iv. Проверка условия: если полученная оценка положения объекта меньше порога деформации для текущего точного фильтра, то рассматривается следующее положение грубого фильтра, иначе анализируется положение следующего точного фильтра.
 - (c) Если для всех точных фильтров было найдено расположение, то положение грубого фильтра считается возможным положением объекта на изображении текущего масштаба.
2. Определение наиболее вероятных положений объекта среди множества, построенного на предыдущем шаге. На данном этапе используется полная пирамида признаков. Как правило, количество возможных положений не очень большое. Процесс проверки существования объекта на полной пирамиде производится по той же схеме, что и для свернутой пирамиды признаков.

4. Этапы оптимизации и распараллеливания последовательной реализации алгоритма Latent SVM

4.1. Тестовая инфраструктура

На каждом этапе оптимизации и распараллеливания последовательной реализации Latent SVM в качестве тестового набора использовалась база данных конкурса PASCAL VOC 2007, содержащая различные изображения объектов двадцати классов (airplane, bicycle, bird, bottle и др.). Представленные фотографии различаются размером изображенных на них объектов, их положением на сцене, ракурсом и степенью освещенности. Указанные факторы оказывают значительное влияние на точность построенной модели.

Так как исходная задача состояла в реализации алгоритма поиска и не включала в себя реализацию алгоритма обучения, для проведения экспериментов были использованы модели авторов статей [4, 5], преобразованные в формат xml в соответствии с разработанной структурой.

Оценка качества детектирования объектов с помощью реализованного алгоритма вывода выполнялась средствами VOC Development Kit, в состав которого входит модуль, позволяющий вычислить среднюю точность предсказания (*average precision*). Указанная метрика определяется как математическое ожидание точностей следующим образом:

$$AP = \frac{1}{11} \sum_{r \in \{0; 0.1; \dots; 1\}} p(r), p(r) = \max_{\bar{r}, \bar{r} \geq r} p(\bar{r}),$$

где $p(\bar{r}) = \frac{a}{a+c}$ – точность, \bar{r} – процент перекрытия детектированного окаймляющего прямоугольника и прямоугольника, который был размечен на исходном изображении как окаймляющий $\bar{r} \in \{0; 0.1; \dots; 1\}$, a – количество объектов, для которых процент перекрытия не меньше, чем \bar{r} (т. е. считается, что объект детектирован правильно), c – количество объектов с процентом перекрытия, меньшим, чем \bar{r} (объект найден ошибочно).

Вычислительные эксперименты проводились с использованием следующей инфраструктуры:

- Язык разработки: C.
- Среда разработки: Microsoft Visual Studio 9.0.
- Компилятор: Microsoft C/C++ Compiler Version 15.00.30729 (x64).
- Процессор: 2 двухъядерных процессора Intel Xeon 5150 (2.66 GHz).
- Память: 4 Gb.
- Библиотеки: Intel Threading Building Blocks (TBB) 3.0 for Windows, v.3.0.2010.707.
- Технологии: OpenMP.
- Операционная система: Microsoft Windows Server 2008 Standard SP1 x64.

Далее в работе приводится последовательность шагов оптимизации и распараллеливания. На каждом этапе, который дает выигрыш по отношению к реализациям авторов [4, 5], приводятся результаты экспериментов по средней точности детектирования и среднему времени детектирования объектов на одном изображении. Объем тестовой выборки составляет примерно 5000 изображений.

4.2. «Горячие» точки последовательной реализации

Первый шаг при выполнении оптимизации заключается в локализации «горячих» точек программы. Для этого мы воспользовались инструментом Intel Parallel Amplifier в режиме «Hotspots». Результат анализа показал, что в разработанной вычислительной схеме метода Latent SVM основная операция – это вычисление сверток. Входными данными операции являются:

1. Двумерная матрица векторов признаков *featureMap* (матрица признаков на некотором уровне пирамиды признаков). Данная матрица может быть представлена в виде трехмерного прямоугольного параллелепипеда размерности $N_y^f N_x^f p$, где N_x^f – число столбцов матрицы векторов, N_y^f – количество строк матрицы векторов, $p = 31$ – размерность вектора признаков. Двумерную матрицу, полученную при каждом фиксированном p , будем называть каналом.
2. Двумерная матрица весовых векторов фильтра *filter* (это может быть точный или грубый фильтр). Данная матрица может быть представлена в виде трехмерного прямоугольного параллелепипеда размерности $N_y^g N_x^g p$, где N_x^g – число столбцов матрицы весовых векторов, N_y^g – количество строк матрицы весовых векторов, $p = 31$ – размерность вектора весов.

Результатом выполнения операции свертки является двумерная матрица свертки *conv* размерности $(N_y^f - N_y^g + 1)(N_x^f - N_x^g + 1)$.

Задача вычисления свертки матрицы векторов признаков и фильтра сводится к тому, чтобы определить значения элементов матрицы свертки в соответствии с формулой:

$$conv[i, j] = \sum_{di=0}^{N_y^g-1} \sum_{dj=0}^{N_x^g-1} \sum_{k=0}^{p-1} filter[di, dj, k] \cdot featureMap[i + di, j + dj, k]$$

По существу с точки зрения программной реализации получается цикл вложенности 5. Чтобы ускорить вычисление сверток, были выполнены некоторые компиляторные оптимизации (упрощение циклов, вынос инвариантов), но значительный выигрыш не был получен, т.к. очевидно, что компилятор выполнял эти оптимизации автоматически. Распараллеливание циклов в функции вычисления сверток не дает выигрыша, т.к. свертки вычисляются многократно, как следствие, возникают значительные накладные расходы на организацию параллелизма (возобновление и остановка потоков). Именно такие результаты были получены в процессе анализа степени параллелизма (Concurrency-анализ инструмента Intel Parallel Amplifier) разработанной реализации. Таким образом, распараллеливание необходимо выполнять на более высоком уровне.

4.3. Распараллеливание по уровням пирамиды признаков

Как выяснилось ранее, основное время занимает процедура поиска положения объектов (второй этап вычислительной схемы) и наиболее трудоемкой операцией является вычисление сверток. Заметим, что поиск объектов «бегущим» окном выполняется на каждом уровне пирамиды признаков независимо. Таким образом, можно выполнить распараллеливание разработанной последовательной реализации по уровням

пирамиды. Более того, на каждом уровне можно подсчитать количество выполняемых операций, как следствие, достаточно просто построить статическое расписание распределения нагрузки между потоками. Предлагается использовать следующую схему распределения – каждому потоку отдавать на обработку набор уровней так, чтобы суммарное количество операций было приближенно одинаковым. Такой подход обеспечит равномерность распределения нагрузки между потоками. Описанная схема распараллеливания была реализована с помощью средств библиотеки ТВВ.

Таблица 1

Средняя точность детектирования объектов на данных VOC 2007

Название класса объектов	Средняя точность			Время работы (с)		Ускорение (отношение времени последовательной к параллельной)
	Последовательная реализация	Реализация авторов [4]	Параллельная реализация в 4 потока	Последовательная реализация	Текущая реализация в 4 потока	
aeroplane	0,278	0,28	0,279	4,205	1,263	3,329
bicycle	0,525	0,544	0,525	4,412	1,303	3,386
bird	0,006	0,007	0,007	3,746	1,174	3,191
boat	0,123	0,145	0,124	3,623	1,157	3,131
bottle	0,26	0,262	0,261	3,438	1,125	3,056
bus	0,409	0,398	0,409	4,435	1,304	3,401
car	0,457	0,463	0,457	3,215	1,079	2,98
cat	0,163	0,16	0,164	5,032	1,402	3,589
chair	0,154	0,163	0,155	4,89	1,374	3,559
cow	0,167	0,167	0,167	4,696	1,355	3,466
diningtable	0,238	0,243	0,239	5,185	1,432	3,621
dog	0,07	0,05	0,07	4,85	1,372	3,535
horse	0,437	0,438	0,437	5,117	1,422	3,598
motorbike	0,384	0,382	0,385	5,12	1,42	3,606
person	0,338	0,342	0,338	3,747	1,188	3,154
pottedplant	0,101	0,079	0,101	3,417	1,117	3,059
sheep	0,166	0,172	0,166	3,13	1,061	2,95
sofa	0,214	0,221	0,215	4,998	1,399	3,573
train	0,331	0,34	0,331	4,208	1,362	3,089
tvmonitor	0,38	0,393	0,381	4,208	1,334	3,154
Средние значения:				4,283	1,282	3,321

Табл. 1 содержит результаты вычислительных экспериментов, проведенных над разработанной последовательной и параллельной реализациями. Очевидно, что средняя точность детектирования объектов с помощью подготовленных реализаций (второй и четвертый столбцы) приближенно совпадает с точностью, показанной авторами статьи [4] (третий столбец). Отметим, что, несмотря на практически идентичное качество детектирования, последовательная реализация в среднем обрабатывает одно изображение в 2–2,5 раза медленнее, чем реализация авторов алгоритма, которая решает задачу детектирования объектов одного класса на изображении примерно за 2 секунды [4]. Но дальнейшее распараллеливание последовательной версии по уровням пирамиды признаков в среднем дает ускорение в 3,3 раза на четырех потоках

(последний столбец таблицы), в результате чего время обработки было уменьшено в среднем до 1,28 секунды. Указанное ускорение — неплохой результат при условии, что распараллелены не все этапы алгоритма, а только поиск объектов, который занимает ориентировочно 80 % времени от общего времени поиска объектов заданного класса на изображении.

4.4. Реализация каскадного алгоритма Latent SVM

Следующим шагом оптимизации является реализация каскадной схемы, описанной в предыдущем разделе.

В табл. 2 приведены численные значения средней точности, полученные на каскадной реализации алгоритма вывода Latent SVM. Сравнение с результатами, приведенными в [5], показывает, что практически на всех классах объектов наблюдаются незначительные отклонения. Также, в таблице представлены результаты точности детектирования для выполненной последовательной реализации алгоритма [4] на моделях из [5], содержащих большее количество компонент. Из результатов сравнения можно видеть, что точность детектирования во всех трех реализациях практически совпадает. В табл. 2 полужирным начертанием выделены наилучшие точности детектирования для каждого класса объектов.

Таблица 2

Средняя точность детектирования объектов на данных VOC 2007

Название класса объектов	Средняя точность		Время работы (с)		Ускорение (отношение времени каскадной к предыдущей)
	Каскадная реализация	Реализация авторов [5]	Каскадная реализация	Предыдущая реализация	
aeroplane	0,28	0,23	1,37	10,37	7,57
bicycle	0,57	0,49	1,38	11,42	8,28
bird	0,09	0,11	1,72	10,76	6,26
boat	0,15	0,13	2,14	11,2	5,23
bottle	0,24	0,27	2	10,19	5,1
bus	0,47	0,47	1,46	11,05	7,57
car	0,54	0,50	1,78	9,45	5,31
cat	0,16	0,19	1,35	10,8	8
chair	0,20	0,16	2,28	10,94	4,8
cow	0,24	0,23	1,57	10,88	6,93
diningtable	0,22	0,11	1,61	11,14	6,92
dog	0,11	0,12	1,35	11,41	8,45
horse	0,56	0,36	1,54	10,88	7,06
motorbike	0,45	0,37	1,29	10,87	8,43
person	0,41	0,38	2,82	10,12	3,59
pottedplant	0,12	0,14	2,38	9,71	4,08
sheep	0,18	0,23	1,73	9,18	5,31
sofa	0,29	0,23	1,52	10,57	6,95
train	0,44	0,34	1,42	11,53	8,12
tvmonitor	0,40	0,40	1,66	11,35	6,84
Средние значения:			1,72	10,69	6,54

Наибольший интерес в данной работе представляет эффект от применения каскадной схемы. Поэтому обратим внимание на ускорение, которое получено по отношению к предшествующей реализации. Максимальное ускорение составляет 8,45 раза, среднее — примерно 6,5 раза. Очевидно, что применение каскадной схеме позволяет значительно уменьшить среднее время детектирования без потери в средней точности.

4.5. Оптимизация и распараллеливание каскадного алгоритма Latent SVM

На данный момент необходимо оценить среднее время обработки одного изображения с использованием разработанной каскадной реализации. Первый шаг состоит в том, чтобы определить «горячие» точки в процедуре поиска. Аналогично предыдущей реализации наиболее затратной операцией является операция вычисления сверток. Отметим, что при этом свертки вычисляются не во всех возможных положениях объекта (не на всей матрице признаков), а только на некотором наборе наиболее вероятных, которые остались после отсека. Соотношение времени построения пирамиды признаков и времени вычисления сверток близко к единице. На данном этапе операция вычисления сверток была переписана с использованием SSE. В результате в среднем выигрыш от такой оптимизации составил 0,1 секунды (четвертый и пятый столбцы таблицы). Далее было выполнено распараллеливание оптимизированной версии.

Обратимся к анализу каскадной схемы и возможности применения распараллеливания по уровням пирамиды признаков. На первом этапе каскадной схемы отсекаются наименее вероятные положения объектов. Как следствие, при переходе на следующий этап на каждом уровне пирамиды останутся достоверные положения объекта, количество которых заранее не известно. При этом невозможно спрогнозировать, в какой момент произойдет отсечение (отбрасывание) следующего положения, а потому нельзя вычислить количество выполняемых операций. Реализация такой схемы распараллеливания не даст выигрыша, т.к. невозможно достигнуть равномерного распределения нагрузки между потоками. Поэтому было принято решение параллелизовать каскадную схему на уровне компонент модели объектов класса. В табл. 3 показаны результаты экспериментов с параллельной реализацией каскадного Latent SVM, выполненной с использованием технологии OpenMP. Разработанная реализация не уступает по точности детектирования реализации авторов [5], а на некоторых классах объектов, напротив, показывает более высокие результаты (выделены полужирным начертанием во втором столбце). В среднем ускорение составило 2,66 раза на четырех потоках. Отметим, что указанная величина во многом определяется количеством возможных положений, которые попадают на вход второго этапа каскадной схемы в каждой компоненте. Таким образом, ускорение может принимать недетерминированные значения. В результате распараллеливания оптимизированной версии каскада в среднем время поиска уменьшилось в 18,4 раза по сравнению с исходной последовательной реализацией (последний столбец табл. 3).

5. Заключение

В ходе работы выполнены последовательная и параллельная реализации алгоритма вывода Latent SVM [4], позволяющие детектировать объекты с точностью, которая

Таблица 3

Средняя точность детектирования объектов на данных VOC 2007

Название класса объектов	Средняя точность		Время работы (с)				Ускорение	
	Параллельная каскадная реализация	Реализация авто-ров [5]	Исходная реализация	Последовательная каскадная реализация	Каскадная реализация после оптимизации	Параллельная реализация в 4 потока	Отношение параллельной каскадной к последовательной	Отношение предыдущей к параллельной
aeroplane	0,28	0,23	10,37	1,37	1,37	0,51	2,69	20,33
bicycle	0,57	0,49	11,42	1,38	1,30	0,50	2,60	22,84
bird	0,09	0,11	10,76	1,72	1,60	0,56	2,86	19,21
boat	0,15	0,13	11,2	2,14	1,97	0,77	2,56	14,55
bottle	0,24	0,27	10,19	2,00	1,81	0,66	2,74	15,44
bus	0,47	0,47	11,05	1,46	1,35	0,55	2,45	20,09
car	0,54	0,50	9,45	1,78	1,62	0,60	2,70	15,75
cat	0,16	0,19	10,8	1,35	1,29	0,53	2,43	20,38
chair	0,20	0,16	10,94	2,28	2,08	0,75	2,77	14,59
cow	0,24	0,23	10,88	1,57	1,47	0,55	2,67	19,78
diningtable	0,22	0,11	11,14	1,61	1,49	0,58	2,57	19,21
dog	0,11	0,12	11,41	1,35	1,31	0,50	2,62	22,82
horse	0,56	0,36	10,88	1,54	1,44	0,54	2,67	20,15
motorbike	0,45	0,37	10,87	1,29	1,23	0,49	2,51	22,18
person	0,41	0,38	10,12	2,82	2,55	0,86	2,97	11,77
pottedplant	0,12	0,14	9,71	2,38	2,17	0,72	3,01	13,49
sheep	0,18	0,23	9,18	1,73	1,57	0,59	2,66	15,56
sofa	0,29	0,23	10,57	1,52	1,43	0,55	2,60	19,22
train	0,44	0,34	11,53	1,42	1,33	0,53	2,51	21,75
tvmonitor	0,40	0,40	11,35	1,66	1,62	0,60	2,70	18,92
Средние значения:			10,69	1,72	10,69	0,597	2,66	18,40

не уступает реализации разработчиков алгоритма, о чем свидетельствуют результаты проведенных экспериментов. Реализации интегрированы в библиотеку с открытыми исходными кодами OpenCV, использование которой не требует дополнительного дорогостоящего программного обеспечения. Использование разработанных реализаций осложняется существенным временем поиска. Реализация каскадной схемы алгоритма Latent SVM [5] позволила уменьшить время поиска объектов на тестовой базе VOC 2007 в среднем в ~18 раз. В настоящее время выполняется интеграция каскадного Latent SVM в библиотеку OpenCV. Изучается возможность повышения качества указанного алгоритма для детектирования пешеходов и транспортных средств, а также исследуются направления дальнейшей оптимизации вычислений при решении задачи многоклассового детектирования объектов.

Работа выполнена в рамках программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы», государственный контракт № 11.519.11.4015.

Литература

1. Понс, Ж. Компьютерное зрение. Современный подход / Ж. Понс, Д. Форсайт. – М.: Изд. д. Вильямс, 2004. – 465 с.
2. Szeliski, R. Computer Vision: Algorithms and Applications / R. Szeliski. – Springer, 2010. – 979 p.
3. Sonka, M. Image Processing, Analysis and Machine Vision / M. Sonka, V. Hlavac, R. Boyle. – Thomson, 2008. – 866 p.
4. Felzenszwalb, P.F. Object Detection with Discriminatively Trained Part Based Models / P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2010. – Vol. 32, No. 9. – P. 1627–1645.
5. Felzenszwalb, P.F. Cascade object detection with deformable part model / P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan // Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR'10). – 2010. – P. 2241–2248.
6. Hastie, T. The elements of statistical learning. Data mining, inference and prediction / T. Hastie, R. Tibshirani, J. Freidman. – 2001. – 745 p.
7. Viola, P. Robust Real-Time Face Detection / P. Viola, M.J. Jones // International Journal of Computer Vision. – 2004. – No. 57(2) – P. 137–154.
8. Viola, P. Rapid object detection using a boosted cascade of simple features / P. Viola, M.J. Jones // Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'01). – 2001. – Vol. 1. – P. I-511–I-518.
9. Tuytelaars, T. The cascaded Hough transform / T. Tuytelaars, M. Proesmans, L. Van Gool // Proceedings of the IEEE International Conference on Image Processing (ICIP'97). – 1997. – Vol. II. – P. 736–739.
10. Pentland, A. Face Recognition for Smart Environments / A. Pentland, T. Choudhury // IEEE Computer Vision. – 2000. – P. 50–55.
11. Alonso, D. Robust Vehicle Detection through Multidimensional Classification for On Broad Video Based Systems / D. Alonso, L. Saldaro, M. Nieto // Proceedings of the IEEE International Conference on Image Processing (ICIP'07). – 2007. – P. IV-321–IV-324.
12. Dalal, N. Histograms of oriented gradients for human detection / N. Dalal, B. Triggs // Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR'05). – 2005. – Vol. 1. – P. 886–893
13. Viola, P. Detecting pedestrians using patterns of motion and appearance / P. Viola, M.J. Jones, D. Snow // Proceedings of the 9th International Conference on Computer Vision (ICCV'03). – 2003. – Vol. 1 – P. 734-741.
14. Gavrila, D.M. Vision-based pedestrian detection: the protector system / D.M. Gavrila, J. Giebel, S. Munder // Proceedings of the IEEE Intelligent Vehicles Symposium, Parma, Italy. – 2004. – P. 13–18.
15. Amit, Y. 2D Object Detection and Recognition: models, algorithms and networks / Y. Amit. – The MIT Press, 2002. – 325 p.

16. Shotton, J. Contour-based Learning for Object Detection / J. Shotton, A. Blake, R. Cipolla // Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV'05). – 2005. – Vol. 1. – P. 503–510.
17. Torralba, A. Context-based Vision System for Place and Object Recognition / A. Torralba, K.P. Murphy, W.T. Freeman, M.A. Rubin // Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV'03). – 2003. – Vol. 1. – P. 273–283.
18. Myung, J. Ch. Exploiting Hierarchical Context on a large database of object categories / J.Ch. Myung, J.J. Lim, A. Torralba, A.S. Willsky // Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'10). – 2010. – P. 129–136.

Евгений Александрович Козин, ассистент, кафедра «Математического обеспечения ЭВМ», Нижегородский государственный университет им. Н.И. Лобачевского, evgeniy.kozinov@gmail.com.

Валентина Дмитриевна Кустикова, ассистент, кафедра «Математического обеспечения ЭВМ», Нижегородский государственный университет им. Н.И. Лобачевского, valentina.kustikova@gmail.com.

Иосиф Борисович Мееров, к.т.н., доцент, кафедра «Математического обеспечения ЭВМ», Нижегородский государственный университет им. Н.И. Лобачевского, meerov@vmk.unn.ru.

Алексей Николаевич Половинкин, м.н.с., кафедра «Математического обеспечения ЭВМ», Нижегородский государственный университет им. Н.И. Лобачевского, alexey.polovinkin@gmail.com.

Алексей Александрович Сиднев, ассистент, кафедра «Математического обеспечения ЭВМ», Нижегородский государственный университет им. Н.И. Лобачевского, alexey.sidnev@gmail.com.

APPROACHES TO THE OPTIMIZATION AND PARALLELIZATION OF COMPUTATIONS IN THE PROBLEM OF DETECTING OBJECTS OF DIFFERENT CLASSES IN THE IMAGE

E.A. Kozinov, Nizhny Novgorod State University (Nizhny Novgorod, Russian Federation),

V.D. Kustikova, Nizhny Novgorod State University (Nizhny Novgorod, Russian Federation),

I.B. Meyerov, Nizhny Novgorod State University (Nizhny Novgorod, Russian Federation),

A.N. Polovinkin, Nizhny Novgorod State University (Nizhny Novgorod, Russian Federation),

A.A. Sidnev, Nizhny Novgorod State University (Nizhny Novgorod, Russian Federation)

This paper considers the problem of object detection in static images. We describe a state-of-the-art method based on Latent SVM algorithm. A well-known approach to speed up calculations, the construction of cascade classifiers, is used. We describe a computational scheme that uses cascade modification of the original Latent SVM algorithm. The issues of parallelization and performance optimization are discussed. We analyze the most time-consuming parts of implementation, consider several parallelization schemes and aspects of their performance. The results of numerical experiments on PASCAL Visual Object Challenge 2007 image dataset are given.

Keywords: object detection, algorithm Latent SVM, cascade classifier, parallelization.

References

1. Forsajt D., Pons Zh. *Komp'yuternoe zrenie. Sovremennyy podhod* [Computer Vision. A modern approach]. Moscow, Williams, 2004, 465 p.
2. Szeliski R. *Computer Vision: Algorithms and Applications*. Springer, 2010, 979 p.
3. Sonka M., Hlavac V., Boyle R. *Image Processing, Analysis and Machine Vision*. Thomson, 2008, 866 p.
4. Felzenszwalb P. F., Girshick R. B., McAllester D., Ramanan D. Object Detection with Discriminatively Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010, Vol. 32, No. 9, P. 1627–1645.
5. Felzenszwalb P. F., Girshick R. B., McAllester D., Ramanan D. Cascade object detection with deformable part model. *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR'10)*. 2010, P. 2241–2248.
6. Hastie T., Tibshirani R., Friedman J. *The elements of statistical learning. Data mining, inference and prediction*. 2001, 745 p.
7. Viola P., Jones M.J. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 2004, No. 57(2), P. 137–154.
8. Viola P., Jones M.J. Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'01)*. 2001, Vol. 1, P. I-511–I-518.
9. Tuytelaars T., Proesmans M., Van Gool L. The cascaded Hough transform. *Proceedings of the IEEE International Conference on Image Processing (ICIP'97)*. 1997, Vol. II, P. 736–739.
10. Pentland A., Choudhury T. Face Recognition for Smart Environments. *IEEE Computer Vision*, 2000, P. 50–55.
11. Alonso D., Saldaro L., Nieto M. Robust Vehicle Detection through Multidimensional Classification for On Broad Video Based Systems. *Proceedings of the IEEE International Conference on Image Processing (ICIP'07)*. 2007, P. IV-321–IV-324.
12. Dalal N., Triggs B. Histograms of oriented gradients for human detection. *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR'05)*. 2005, Vol. 1, P. 886–893.
13. Viola P., Jones M.J., Snow D. Detecting pedestrians using patterns of motion and appearance. *Proceedings of the 9th International Conference on Computer Vision (ICCV'03)*. 2003, Vol. 1, P. 734–741.

14. Gavrilu D. M., Giebel J., Munder S. Vision-based pedestrian detection: the protector system. *Proceedings of the IEEE Intelligent Vehicles Symposium, Parma, Italy*. 2004, P. 13–18.
15. Amit Y. *2D Object Detection and Recognition: models, algorithms and networks*. The MIT Press, 2002, 325 p.
16. Amit Y. Contour-based Learning for Object Detection. *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV'05)*. 2005, Vol. 1, P. 503–510.
17. Torralba A., Murphy K.P., Freeman W.T., Rubin M.A. Context-based Vision System for Place and Object Recognition. *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV'03)*. 2003, Vol. 1, P. 273–283.
18. Myung J. Ch., Lim J.J., Torralba A., Willsky, A.S. Exploiting Hierarchical Context on a large database of object categories. *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'10)*. 2010, P. 129–136.

Поступила в редакцию 5 ноября 2012 г.

ДВУХСЕТОЧНЫЕ ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ ДЛЯ РЕШЕНИЯ ДРОБНО-ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ АНОМАЛЬНОЙ ДИФФУЗИИ

С.Ю. Лукащук

Приводятся описание и анализ параллельных алгоритмов решения начально-краевых задач для уравнений аномальной диффузии, содержащих производные дробного порядка типа Римана-Лиувилля по пространственным и/или временным переменным. Параллельные алгоритмы построены на основе двухсеточного подхода. При этом грубая сетка используется для расчета эффектов пространственного и временного дальнего действия с использованием сплайн-аппроксимации, а мелкая сетка служит для конечно-разностной дискретизации решаемых уравнений. Рассматриваются алгоритмы с декомпозицией как по пространству, так и по времени. Для распараллеливания по времени используется подход, предложенный в известном алгоритме PARAREAL. Приводятся теоретические оценки параллельной эффективности предложенных алгоритмов. Показано, что алгоритмы имеют сверхлинейное ускорение по сравнению с классическим последовательным конечно-разностным алгоритмом и обеспечивают тот же порядок точности вычислений при условии согласованного выбора шагов точной и грубой сеток. Также приводятся некоторые результаты вычислительных экспериментов, подтверждающие эффективность предложенных алгоритмов.

Ключевые слова: двухсеточный параллельный алгоритм, аномальная диффузия, дифференциальное уравнение дробного порядка.

Введение

Во многих неупорядоченных сложных средах, таких как пористые и трещиновато-пористые среды, аморфные полупроводники, жидкие полимеры и стекла, перколяционные кластеры и самоподобные фрактальные среды, турбулентные потоки жидкости, газа и плазмы, кинетика протекания процессов диффузионного переноса часто отличается от классической, соответствующей нормальному закону распределения [1, 2]. Такие процессы принято называть процессами аномальной диффузии. Аномальный диффузионный перенос обычно обусловлен эффектами памяти среды, пространственной нелокальности и перемежаемости, имеет немарковскую стохастическую природу и поэтому не может быть описан классическими эволюционными уравнениями переноса [3].

Одним из эффективных и широко используемых подходов для описания процессов аномальной диффузии является использование аппарата интегро-дифференцирования дробного порядка [4–6]. В этом случае уравнение процесса является интегро-дифференциальным уравнением, содержащим производные дробного порядка по временной и/или пространственным переменным.

Основной проблемой численного моделирования процессов аномальной диффузии, описываемых уравнениями дробного порядка, является нелокальный характер последних. При использовании конечно-разностной дискретизации обычных диффузионных уравнений значение в текущем узле расчетной сетки зависит только от значений в соседних узлах. При конечно-разностной дискретизации уравнений дробного порядка значение в текущем узле будет зависеть от всех остальных пространственных узлов (при наличии в уравнении дробных производных по пространственным переменным) и всех предыдущих временных слоев (при наличии дробной производной

по времени) [5]. В результате при численных расчетах аномального переноса объем вычислений оказывается существенно (на порядки!) большим по сравнению с объемом вычислений в расчетах классических диффузионных процессов. При наличии дробной производной по времени объем вычислений также растет с увеличением номера временного слоя, поскольку увеличивается диапазон памяти системы, которую нужно учитывать. Существенно возрастает и объем необходимой оперативной памяти для хранения результатов расчета. Использование параллельных алгоритмов, построенных только на принципах пространственной декомпозиции расчетной области (domain decomposition), ситуацию не спасает, поскольку приводит к чрезвычайно большим объемам передачи данных между процессорами.

Указанные проблемы делают актуальной задачу разработки новых последовательных и параллельных численных алгоритмов решения уравнений аномальной диффузии с производными дробного порядка. В данной работе предлагается подход к построению таких алгоритмов, основанный на использовании идеи двухсеточных методов. Грубая сетка используется в этом случае как основа для расчета эффектов пространственного и временного дальнего действия, а мелкая сетка служит для непосредственных вычислений. По узлам грубой сетки может производиться пространственная и, при необходимости, временная декомпозиция расчетной области, что приводит к легко реализуемым параллельным алгоритмам. При распараллеливании по временной оси данный подход приводит к модификации алгоритма PARAREAL для уравнений с производными дробного порядка [7].

1. Постановка задачи

При описании процессов аномальной диффузии наиболее часто используются левосторонняя и правосторонняя производные дробного порядка α типа Римана-Лиувилля [4]:

$$({}_a D_x^\alpha y)(x) = \frac{1}{\Gamma(n - \alpha)} \left(\frac{d}{dx} \right)^n \int_a^x \frac{y(\xi)}{(x - \xi)^{\alpha - n + 1}} d\xi, \quad (1)$$

$$({}_x D_b^\alpha y)(x) = \frac{(-1)^n}{\Gamma(n - \alpha)} \left(\frac{d}{dx} \right)^n \int_x^b \frac{y(\xi)}{(\xi - x)^{\alpha - n + 1}} d\xi, \quad (2)$$

где $n = [\alpha] + 1$ и $[\alpha]$ — целая часть числа α .

В дальнейшем для иллюстрации предлагаемого подхода будем рассматривать уравнение аномальной диффузии дробного порядка следующего вида:

$$({}_0 D_t^\alpha u)(x, t) = \gamma ({}_0 D_x^{1+\beta} u)(x, t) + (1 - \gamma) ({}_x D_L^{1+\beta} u)(x, t) + f(x, t), \quad (3)$$

$$t \in (0, T], \quad x \in (0, L), \quad \alpha, \beta, \gamma \in (0, 1).$$

Для уравнения (3) поставим первую краевую задачу:

$$u(x, 0) = u^0(x), \quad u(0, t) = u_0(t), \quad u(L, t) = u_L(t), \quad (4)$$

где $u^0(x)$, $u_0(t)$, $u_L(t)$ — заданные функции. Отметим, что постановка таких краевых условий предъявляет определенные дополнительные требования к виду функции $f(x, t)$, которые однако не являются важными для рассматриваемых алгоритмов и поэтому здесь не приводятся.

При построении численных схем для дробно-дифференциальных уравнений используются различные конечно-разностные аппроксимации дробных производных. Для левосторонней дробной производной (1) в случае равномерной сетки все они могут быть представлены в следующем общем виде:

$$({}_a D_x^\alpha y)(x_k) \approx h^{-\alpha} \sum_{j=0}^{k+m} A_j y_{k+m-j}, \quad y_j = y(x_j), \quad x_j = a + jh \quad (5)$$

(здесь h — шаг сетки). Аналогичная формула имеет место и для правосторонней дробной производной (2). При $m = 0$ и $A_j = (-1)^j \binom{\alpha}{j}$ имеем наиболее часто используемую классическую аппроксимацию, получаемую из формулы Грюнвальда-Летникова [4], и обеспечивающую первый порядок точности аппроксимации дробной производной для аналитических функций [5]. Повысить порядок точности можно использованием полиномиальной интерполяции искомой функции между узлами расчетной сетки с последующим аналитическим вычислением соответствующих интегралов. Случай линейной межузельной интерполяции, обеспечивающий второй порядок точности, рассмотрен в [8].

2. Декомпозиция задачи на основе двухсеточного подхода

Введем в рассматриваемой пространственно-временной области $\bar{\Omega} = [0, L] \times [0, T]$ равномерную конечно-разностную сетку с шагом ΔX по пространственной и ΔT по временной переменным:

$$\omega_G = \{(X_i = i\Delta X, T_j = j\Delta T), i = 0, 1, \dots, N, j = 0, 1, \dots, M, \Delta X = L/N, \Delta T = T/M\}. \quad (6)$$

Сетку (6) в дальнейшем будем называть грубой сеткой. Каждую ячейку, ограниченную смежными узлами грубой сетки, будем рассматривать как отдельную расчетную подобласть:

$$\bar{\Omega}_{i,j} = [X_{i-1}, X_i] \times [T_{j-1}, T_j], \quad \bar{\Omega} = \bigcup_{i=1}^N \bigcup_{j=1}^M \bar{\Omega}_{i,j}.$$

Для $(t, x) \in \Omega_{i,j}$ производные дробного порядка, входящие в уравнение (3), могут быть записаны следующим образом:

$$\begin{aligned} {}_0 D_t^\alpha u &= {}_{T_{j-1}} D_t^\alpha u + {}_0 I_{T_{j-1}} u, \\ {}_0 D_x^{1+\beta} u &= {}_{X_{i-1}} D_x^{1+\beta} u + {}_0 J_{X_{i-1}} u, \quad {}_x D_L^{1+\beta} u = {}_x D_{X_i}^{1+\beta} u + {}_{X_i} J_L u, \end{aligned}$$

где введены следующие обозначения для интегралов:

$$\begin{aligned} {}_0 I_{T_{j-1}} u &= \frac{1}{\Gamma(-\alpha)} \int_0^{T_{j-1}} \frac{u(x, \tau)}{(t - \tau)^{\alpha+1}} d\tau, \\ {}_0 J_{X_{i-1}} u &= \frac{1}{\Gamma(-\beta - 1)} \int_0^{X_{i-1}} \frac{u(\xi, t)}{(x - \xi)^{\beta+2}} d\xi, \quad {}_{X_i} J_L u = \frac{1}{\Gamma(-\beta - 1)} \int_{X_i}^L \frac{u(\xi, t)}{(\xi - x)^{\beta+2}} d\xi. \end{aligned}$$

Уравнение (3) в области $\Omega_{i,j}$ может быть теперь переписано в следующем виде:

$${}_{T_{j-1}}D_t^\alpha u = \gamma {}_{X_{i-1}}D_x^{1+\beta} u + (1 - \gamma) {}_x D_{X_i}^{1+\beta} u + f(x, t) + Lu, \quad (7)$$

где

$$L = \gamma {}_0J_{X_{i-1}} + (1 - \gamma) {}_0J_{X_{i-1}} - {}_0I_{T_{j-1}}$$

— линейный интегральный оператор. Принципиально важным является тот факт, что вычисление Lu в области $\Omega_{i,j}$ производится по значениям функции $u(x, t)$ при $(x, t) \notin \Omega_{i,j}$.

Обозначим через $U = \{U_{i,j}, i = 0, 1, \dots, N, j = 0, 1, \dots, M\}$ сеточную функцию решения задачи на грубой сетке ω_G . В предлагаемом подходе ее значения используются для постановки начальных и граничных условий для уравнения (7), а также вычисления интегралов, входящих в Lu .

После того, как задача для уравнения (7) замыкается с помощью начальных и граничных условий и указан способ вычисления Lu по значениям U , численное решение задач в каждой подобласти $\Omega_{i,j}$ или соответствующим образом выбранной группе подобластей может выполняться параллельно. Для этого в каждой подобласти вводится точная сетка (для простоты считаем, что разбиение всех подобластей идентично):

$$\omega_F = \{(x_k = k\delta x, t_l = l\delta t), k = 0, 1, \dots, n, l = 0, 1, \dots, m, \delta x = \Delta X/n, \delta t = \Delta T/m\}. \quad (8)$$

С использованием аппроксимаций вида (5) на сетке ω_F строится конечно-разностная аппроксимация уравнений (7) требуемого уровня точности. Как и в случае классического уравнения диффузии, могут быть построены явные, неявные или полужавные схемы. Сеточную функцию задачи на точной сетке будем обозначать через $u = \{u_{k,l}, k = 0, 1, \dots, n, l = 0, 1, \dots, m\}$.

Таким образом, точная сетка ω_F используется для параллельного решения задач в отдельных подобластях, которые могут выполняться параллельно, а грубая сетка ω_G используется для постановки краевых условий и вычисления дополнительной функции Lu .

На основе такого двухсеточного подхода возможно построение алгоритмов двух видов. В алгоритмах первого вида счет ведется последовательно (например, по времени) и в узлы грубой сетки просто копируются значения, полученные при расчете на точной сетке. Затем эти значения рассылаются по всем подобластям, в которых они необходимы для вычисления соответствующих элементов. Второй вид алгоритмов в более полной мере реализует идею двухсеточного подхода и основан на схемах типа „предиктор-корректор“. В этом случае каким-либо образом рассчитываются начальные приближения в узлах грубой сетки, а затем проводятся расчеты (параллельные) на точных сетках и уточняются приближения на грубой сетке. К алгоритму второго типа относится, в частности, модификация алгоритма PARAREAL для распараллеливания по времени. Далее будут рассмотрены характерные представители алгоритмов обоих видов.

3. Описание параллельных алгоритмов

3.1. Параллельный алгоритм с декомпозицией по пространству

Данный алгоритм относится к алгоритмам первого вида. Расчет по времени ведется последовательно, поэтому шаги по времени грубой и точной сеток совпадают: ($\delta t = \Delta T$, $M = m$). Характерными особенностями алгоритма, обеспечивающими требуемый уровень точности вычислений, являются: 1) использование кубической сплайн-интерполяции по значениям в узлах грубой сетки на одном временном слое, 2) использование полиномов второй и третьей степени для аппроксимации в окрестности границ подобластей $\Omega_{i,j}$ производных дробного порядка по пространственной переменной.

Сохранение заданной точности вычислений производных дробного порядка по пространству после декомпозиции на подобласти $\Omega_{i,j}$ является главной проблемой при разработке соответствующих параллельных алгоритмов. Использование классических аппроксимаций вида (5) (типа Грюнвальда-Летникова) для смещенных дробных производных, входящих в уравнение (7), может приводить к катастрофической потере точности вычислений в окрестности границ областей $\Omega_{i,j}$ несмотря на сохранение общего порядка точности аппроксимации. Для подавления ошибки предлагается использовать следующие вычислительные приемы.

Во-первых, необходимо выполнить сдвиг вычисляемой функции на величину соответствующего граничного значения:

$$x_{i-1} D_x^{1+\beta} u = x_{i-1} D_x^{1+\beta} [u(x, t) - u(X_{i-1}, t)] + \Gamma^{-1}(-\beta)(x - X_{i-1})^{-\beta-1} u(X_{i-1}, t), \quad (9)$$

$$x D_{X_i}^{1+\beta} u = x D_{X_i}^{1+\beta} [u(x, t) - u(X_i, t)] + \Gamma^{-1}(-\beta)(X_i - x)^{-\beta-1} u(X_i, t). \quad (10)$$

В результате основная составляющая производной оказывается вычисленной аналитически, а по приближенным формулам рассчитываются уже относительно небольшие поправки.

Во-вторых, при расчете левосторонней производной из правой части (9) в точке x_1 и правосторонней производной из правой части (10) в точке x_{n-1} коэффициенты A_r соответствующих аппроксимаций вида (5) находятся на основе явного вычисления этих производных по трехузловой квадратичной интерполяции. Для следующих точек x_2 и x_{n-2} используется четырехузловая кубическая интерполяция. Для остальных точек может быть использована классическая аппроксимация первого или второго порядка точности.

В-третьих, вычисление интегралов ${}_0 J_{X_{i-1}} u$ и ${}_{X_i} J_L u$ производится аналитически для кубической сплайн-интерполяции, построенной по значениям на грубой сетке U . Известно [9], что оценка погрешности аппроксимации функции $f(x)$ кубическим сплайном $s(x)$ на равномерной сетке с шагом H имеет вид $|s(x) - f(x)| \leq 5AH^4/384$, где $|f^{IV}(x)| \leq A$. Тогда легко получается оценка точности вычисления интеграла ${}_0 J_{X_{i-1}} u$:

$$|{}_0 J_{X_{i-1}} u(x, t_l) - s(x, t_l)| \leq 5A(t)(\Delta X)^4(\delta x)^{-\beta-1}/384,$$

где $|u_{xxxx}(x, t)| \leq A(t)$. Аналогичная оценка справедлива для интеграла ${}_{X_i} J_L u$. Поскольку множитель $(\delta x)^{-1-\beta}$ является общим множителем при вычислении дробных производных на точной сетке по аппроксимациям вида (5), то точность вычисления интегралов по сплайн-интерполяции в разностной схеме решения уравнения (7) на

точной сетке имеет порядок $O(\Delta X^4)$. Если для аппроксимации дробных производных используется аппроксимация первого порядка точности по пространству, то для сохранения этого порядка у всей схемы для уравнения (7) необходимо выполнение следующего очевидного неравенства: $(\Delta X)^4 \leq \delta x$. При использовании аппроксимации второго порядка точности неравенство становится более жестким: $(\Delta X)^2 \leq \delta x$.

Далее представлены основные шаги алгоритма при использовании явной схемы для дискретизации уравнений (7).

1. По заданному начальному условию задачи $u^0(x)$ (4) каждый процессор параллельно вычисляет значения сеточной функции U на нулевом временном шаге. Следующие шаги алгоритма являются едиными для временных слоев $l = 1, 2, \dots, m$.
2. По значениям U для $(l - 1)$ -го временного слоя каждый процессор вычисляет коэффициенты соответствующего кубического интерполяционного сплайна.
3. На точной сетке все процессоры параллельно выполняют расчет $u_{k,l}$ ($k = 1, 2, \dots, n - 1$) на временном слое l . При этом производная по времени аппроксимируется формулой Грюнвальда-Летникова

$$({}_0D_t^\alpha u)(x_k, t_l) \approx (\delta t)^{-\alpha} \sum_{r=0}^l (-1)^r \binom{\alpha}{r} u_{k,l-r},$$

а производные по пространству аппроксимируются на $(l - 1)$ -м временном слое с учетом (9) и (10) по формулам

$$\begin{aligned} ({}_{X_{i-1}}D_x^{1+\beta} u)(x_k, t_{l-1}) &\approx \frac{1}{(\delta x)^{(1+\beta)}} \sum_{r=0}^{k+1} A_r (u_{k+1-r,l-1} - u_{0,l-1}) + \\ &\quad + \frac{u_{0,l-1}}{\Gamma(-\beta)(x_k - X_{i-1})^{1+\beta}}, \\ ({}_x D_{X_i}^{1+\beta} u)(x_k, t_{l-1}) &\approx \frac{1}{(\delta x)^{(1+\beta)}} \sum_{r=0}^{n-k+1} A_r (u_{k-1+r,l-1} - u_{0,l-1}) + \\ &\quad + \frac{u_{0,l-1}}{\Gamma(-\beta)(X_i - x_k)^{1+\beta}}, \end{aligned}$$

Коэффициенты A_r выбираются так, как описано выше. Интегралы ${}_0J_{X_{i-1}} u$ и ${}_x J_L u$ вычисляются по аналитическим формулам для сплайн-интерполяции, построенной на шаге 2.

4. Процессор, отвечающий за расчет подобласти $\Omega_{i+1,l}$ ($i = 1, 2, \dots, N - 1$), пересылает процессору, отвечающему за расчет соседней подобласти $\Omega_{i,l}$ значение приращения за временной шаг искомой функции в первом узле точной сетки: $u_{1,l}^{i+1} - u_{1,l-1}^{i+1}$ (здесь верхний индекс соответствует пространственному номеру области). Процессоры, получившие значения, вычисляют сеточную функцию в узлах грубой сетки на l -м временном слое по формуле

$$U_{i,l} = U_{i,l-1} + 0.5 (u_{1,l}^{i+1} - u_{1,l-1}^{i+1} + u_{n-1,l}^i - u_{n-1,l-1}^i).$$

5. Выполняется взаимный обмен рассчитанными значениями сеточной функции U таким образом, чтобы полный вектор этих значений находился на каждом процессоре.

Далее переход к следующему временному слою и повтор алгоритма с шага 2.

Отметим, что условие устойчивости использованной явной конечно-разностной схемы имеет вид

$$(1 + \beta)(\delta t)^\alpha \leq \alpha(\delta x)^{1+\beta} \quad (11)$$

(для случая $\alpha = 1$ это условие приведено в [8]). В предельном случае классического уравнения диффузии $\alpha = 1$, $\beta = 1$ это условие переходит в хорошо известное условие устойчивости явной разностной схемы для уравнения параболического типа.

3.2. Параллельный алгоритм с декомпозицией по времени

Известно, что параллельные алгоритмы для решения эволюционных задач могут быть построены не только на основе декомпозиции пространственной расчетной области, но и на основе декомпозиции по временной оси. Одним из наиболее широко используемых алгоритмов данного вида является алгоритм PARAREAL, впервые предложенный в работе [10]. В настоящее время данный алгоритм хорошо изучен, получены теоретические оценки его точности и сходимости [11–14], предложен целый ряд его модификаций для решения различных классов задач эволюционного типа [11, 15, 16]. В работе [7] была предложена модификация алгоритма PARAREAL для решения обыкновенных дифференциальных уравнений с производной дробного порядка $\alpha \in (0, 1)$. В данном разделе дается описание новой модификации алгоритма PARAREAL, совмещенной с приведенным в предыдущем разделе алгоритмом декомпозиции по пространству, и предназначенной для решения уравнений вида (3).

Алгоритм базируется на пространственно-временной декомпозиции расчетной области Ω на основе грубой сетки ω_G , представленной в разделе 3, и двумерной сплайн-интерполяции решения по значениям сеточной функции U в узлах грубой сетки.

Пусть с использованием некоторого сеточного метода G , который в дальнейшем будем называть „грубым“, построено приближенное решение U краевой задачи (3), (4) в узлах грубой сетки. Рассмотрим теперь задачу для уравнения (7) в области $\Omega_{i,j}$. Обозначим через $u^{i,j}$ функцию решения этого уравнения. С использованием сплайн-интерполяции по значениям U на $(j - 1)$ -м временном слое можно записать начальное условие для уравнения (7) в виде

$$u^{i,j}(x, T_{j-1}) = \phi(x, U_{0,j-1}, U_{1,j-1}, \dots, U_{N,j-1}). \quad (12)$$

Пусть интерполяция по временной переменной поля значений U выполняется таким образом, что значения интерполяционных коэффициентов для произвольного $(j - 1)$ -го слоя зависят только от значений U на этом и предыдущих временных слоях. Эта интерполяция используется для вычисления интегралов ${}_0I_{T_{j-1}}$. Тогда решение уравнения (7) с начальным условием (12) будет зависеть только от значений U на первых $(j - 1)$ -м временных слоях:

$$u^{i,j} = u^{i,j}(x, t, U_{k,l}), \quad k = 0, 1, \dots, N, \quad l = 0, 1, \dots, j - 1, \quad (x, t) \in \Omega_{i,j}. \quad (13)$$

Отметим, что для получения решений вида (13) необходимо, вообще говоря, на каждом временном слое решать систему уравнений вида (7) для всех подобластей, принадлежащих данному временному слою, с условиями сопряжения решений на границах подобластей. Однако, при численном решении этой задачи на мелкой сетке может быть эффективно использован алгоритм, приведенный в п. 4.1.

Зная решения (13), можно записать уравнения для нахождения значений в узлах грубой сетки:

$$U_{i,j} = u^{i,j}(X_i, T_j, U_{k,l}), \quad k = 0, 1, \dots, N, \quad l = 0, 1, \dots, j-1, \quad (14)$$

$$i = 1, 2, \dots, N, \quad j = 1, 2, \dots, M.$$

Система (14) является системой нелинейных уравнений относительно сеточной функции U и может быть решена методом Ньютона:

$$U_{i,j}^{q+1} = u^{i,j}(X_i, T_j, U^q) + \sum_{k=0}^N \sum_{l=0}^{j-1} \frac{\partial u^{i,j}(X_i, T_j, U^q)}{\partial U_{k,l}} (U_{k,l}^{q+1} - U_{k,l}^q), \quad (15)$$

$$i = 1, 2, \dots, N, \quad j = 1, 2, \dots, M$$

(здесь q — номер итерации метода Ньютона).

Ключевым моментом алгоритма PARAREAL является способ вычисления производных, входящих в (15):

$$\sum_{k=0}^N \sum_{l=0}^{j-1} \frac{\partial u^{i,j}(X_i, T_j, U^q)}{\partial U_{k,l}} (U_{k,l}^{q+1} - U_{k,l}^q) \approx G(X_i, T_j, U^{q+1}) - G(X_i, T_j, U^q), \quad (16)$$

где через G обозначено решение задачи на грубой сетке, получаемое с помощью „грубого“ метода. Таким образом, для вычисления производных на каждом следующем временном слое используется разность „грубых“ решений задачи, полученных по старым и новым, уже вычисленным на предыдущих временных слоях, значениям U .

Обозначим через $F(X_i, T_j, U)$ численное решение задачи на точной сетке в области $\Omega_{i,j}$. Полагая $u^{i,j}(X_i, T_j, U) \approx F(X_i, T_j, U)$, на основе (15), (16) получаем основную расчетную формулу алгоритма PARAREAL:

$$U_{i,j}^{q+1} = F(X_i, T_j, U_{k,l}^q) + G(X_i, T_j, U_{k,l}^{q+1}) - G(X_i, T_j, U_{k,l}^q), \quad (17)$$

$$k = 0, 1, \dots, N, \quad l = 0, 1, \dots, j-1, \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, M.$$

Данная вычислительная процедура позволяет параллельно по времени итерационно находить приближенные решения задачи в узлах грубой сетки, сопоставимые по точности со значениями, получаемыми при расчете на точной сетке во всей расчетной области.

В результате приходим к следующему параллельному алгоритму (полагаем, что имеется $N \times M$ -процессорная решетка $\{p_{i,j}\}$ и за вычисление каждой области $\Omega_{i,j}$ отвечает свой процессор).

1. С использованием классической явной конечно-разностной схемы G , построенной на основе аппроксимаций дробных производных вида (5), каждый процессор находит начальное приближение решения в узлах грубой сетки U^0 .
2. Организуется глобальная итерационная по q процедура метода Ньютона. На первом шаге $q = 0$.
3. По значениям $U_{k,l}^q$ ($k = 0, 1, \dots, N, \quad l = 0, 1, \dots, j-1$) каждый процессор $p_{i,j}$ строит двумерную кубическую по каждой переменной сплайн-интерполяцию поля грубого решения.

4. Все процессоры параллельно производят расчет каждый своей подобласти на точной сетке. При этом группы процессоров, отвечающие за расчет одинаковых временных слоев, реализуют алгоритм с декомпозицией по пространству, приведенный в п. 4.1. Интегралы, образующие Lu в уравнениях (3), рассчитываются по аналитическим формулам, полученным по принятому виду сплайн-интерполяции.
5. Последовательно (по временным слоям) рассчитывается следующее приближение в узлах грубой сетки U^{q+1} по формуле (17). При этом каждый процессор $p_{i,j}$ отвечает за вычисление одного значения $U_{i,j}^{q+1}$ и выполняет один дополнительный расчет „грубым“ методом $G(X_i, T_j, U_{k,l}^{q+1})$. После расчета каждого нового временного слоя полученные обновленные значения U^{q+1} рассылаются каждому процессору.
6. Проверяется условие остановки итерационного процесса: $\|U^{q+1} - U^q\| \leq \varepsilon$. Если условие не выполнено, то возврат к шагу 2. При этом временной слой $q + 1$ из расчета исключается (на нем уже получено решение, соответствующее решению на точной сетке, которое не может быть уточнено по (17)).

Описанный алгоритм допускает простую модификацию для случая, когда количество доступных процессоров достаточно для расчета только нескольких временных слоев одновременно. Кроме того, данный алгоритм существенно упрощается в следующих двух случаях. 1) При $\alpha = 1$, т.е. когда исходное уравнение (3) содержит обыкновенную (не дробную) производную по времени первого порядка — в этом случае отсутствует необходимость интерполяции по времени и алгоритм становится прямой комбинацией алгоритма п. 4.1 и классического алгоритма PARAREAL. 2) Если отсутствует пространственная декомпозиция — в этом случае отсутствует необходимость интерполяции по пространственной переменной и алгоритм становится прямой комбинацией модифицированного на уравнения дробного порядка алгоритма PARAREAL и классического последовательного алгоритма конечно-разностного решения таких уравнений.

4. Оценка параллельной эффективности алгоритмов

4.1. Оценка эффективности параллельного алгоритма с декомпозицией по пространству

Сначала приведем оценку трудоемкости последовательного алгоритма решения задачи (3), (4), построенного по явной схеме с аппроксимацией дробных производных по формуле Грюнвальда-Летникова. Шаг сетки по пространственной и временной переменным для последовательного алгоритма примем равным соответствующим шагам точной сетки параллельного алгоритма (в этом случае последовательное и параллельное решения будут иметь одинаковый порядок точности). Таким образом, имеем $Nn - 1$ расчетных узлов по пространству и M расчетных узлов по времени. Расчет l -го временного слоя по последовательному алгоритму требует $2(Nn - 1)(Nn + l + 4)$ операций. Принимая, как обычно при таких оценках, время выполнения одной операции за единицу, получаем время выполнения последовательного алгоритма равным $T_1 = M(Nn - 1)(2Nn + M + 9)$.

Теперь оценим время выполнения параллельного алгоритма, приведенного в п. 4.1. Будем полагать, что количество используемых для его выполнения процес-

соров равно количеству пространственных подобластей: $P = N$. Количество операций, выполняемых каждым процессором на l -м временном слое, складывается из четырех составляющих: 1) операций, необходимых для вычисления дробных производных: $2(n-1)(n+l+11)$, 2) операций, необходимых для вычисления интегралов ${}_0J_{x_{i-1}}u$ и x_iJ_Lu по явным аналитическим формулам через сплайн-интерполяцию: $228(n-1)(N-1)$, 3) операций, необходимых для расчета значений в узлах грубой сетки: $5(N-1)$ и 4) операций, необходимых для вычисления коэффициентов интерполяционного сплайна: $22(N-1)$ (для решения системы используется метод прогонки). Таким образом, трудоемкость расчета l -го временного слоя в параллельном алгоритме составляет $2(n-1)(n+l+11) + [228(n-1) + 27](N-1)$ операций, которые выполняются параллельно. Тогда время расчета всех M временных слоев в параллельном алгоритме составит $T_p = M[(n-1)(2n+M+228N-206) + 27(N-1)]$. Долей последовательного расчета, выполняемого на этапе инициализации алгоритма до начала основной вычислительной процедуры, в данной оценке пренебрегаем.

Тогда для ускорения параллельного алгоритма получаем следующую оценку:

$$S_p \equiv \frac{T_1}{T_p} = \frac{(Nn-1)(2Nn+M+9)}{(n-1)(2n+M+228N-206) + 27(N-1)} \approx \frac{N(2Nn+M)}{2n+M}$$

(последняя оценка становится справедливой при $N \ll n$). Заметим, что число временных слоев M при использовании явной схемы может оказаться достаточно большим.

При стремлении к нулю шага точной сетки, т.е. при $n \rightarrow \infty$, и неизменных остальных параметрах получаем следующую асимптотическую оценку ускорения $S_p \approx N^2 \equiv P^2$. Таким образом, по сравнению с классическим алгоритмом, предлагаемый параллельный алгоритм демонстрирует асимптотически квадратичное ускорение. Причина такого нелинейного ускорения очевидна: в то время как классический алгоритм использует для расчета все пространственные узлы, параллельный алгоритм рассчитывает далекодействующие нелокальные эффекты с использованием сплайн-аппроксимации, что приводит к существенному уменьшению количества операций в параллельном алгоритме по сравнению с последовательным.

Приведенная оценка не учитывает времени, затрачиваемого на обмен данными между процессорами, и является справедливой для систем с общей памятью. Для систем с распределенной памятью это время необходимо учитывать. Тем не менее, на величину асимптотической оценки это время влияния не оказывает, поскольку пропорционально $M(N-1)$.

4.2. Оценка эффективности параллельного алгоритма с декомпозицией по времени

Время выполнения последовательного алгоритма T_1 оценивается аналогично п. 5.1 и составляет $T_1 = Mt(Nn-1)(2Nn+Mt+9)$ (оценка изменилась, поскольку теперь имеем Mt слоев точной сетки по времени).

Оценим время исполнения параллельного алгоритма. Как и ранее, долей последовательного расчета, выполняемого на этапе инициализации алгоритма до начала основной вычислительной процедуры, пренебрегаем. Расчет выполняется на грубой ω_G и точной ω_F сетках, определенных в (6) и (8), соответственно.

Наибольшее число операций будут выполнять процессоры, отвечающие за расчет подобластей $\Omega_{i,M}$, соответствующих последнему M -му временному слою, поэтому

именно они и будут определять время реализации алгоритма. Для каждой итерации q количество операций, выполняемых каждым таким процессором, складывается из четырех составляющих: 1) операций, необходимых для вычисления дробных производных: $m(n-1)(2n+m+23)$, 2) операций, необходимых для вычисления интегралов ${}_0J_{X_{i-1}}u$, X_iJ_Lu и ${}_0I_{T_{M-1}}$ по явным аналитическим формулам через сплайн-интерполяцию: $114m(n-1)[2(N-1)+(M-1)]$, 3) операций, необходимых для расчета на грубой сетке по „грубому“ методу и уточнению значений U по формуле (17): $M(N-1)(2N+M+9)+2$ и 4) операций, необходимых для вычисления коэффициентов интерполяционного сплайна: $22(2MN-M-1)$. Сумма этих значений дает трудоемкость одной итерации метода Ньютона в параллельном алгоритме. Если для реализации алгоритма требуется Q итераций, то общее время выполнения параллельного алгоритма равно

$$T_p = Q[m(n-1)(2n+m+128N+114M-319) + M(N-1)(2N+M+53) + 22(M-1)].$$

По классической формуле $S_p = T_1/T_p$ может быть легко оценено ускорение параллельного алгоритма. Если $N \ll n$, $M \ll m$, то $T_1 \approx MNmn(2Nn+Mm)$, $T_p \approx Qmn(2n+m)$ и получаем следующую оценку ускорения:

$$S_p = MN(2Nn+Mm)Q^{-1}(2n+m)^{-1}.$$

Для случая квадратной грубой сетки ($M=N$) получаем $S_p = N^3/Q$. Учитывая, что общее количество процессоров, задействованных для выполнения параллельного алгоритма, равно $P = MN = N^2$, получаем следующую асимптотическую оценку ускорения: $S_p \approx P^{3/2}/Q$. Максимально возможное количество итераций $Q = M = \sqrt{P}$. Однако на практике количество итераций всегда оказывается существенно меньше, поэтому действительная асимптотическая оценка ускорения получается сверхлинейной.

5. Результаты тестовых расчетов

Точность и эффективность предлагаемых алгоритмов была проверена рядом тестовых расчетов. Ниже представлены некоторые результаты проведенных вычислительных экспериментов.

Решается уравнение (3) в единичном квадрате ($L=1$, $T=1$) при нулевых граничных условиях $u_0(t)=0$, $u_L(t)=0$, без функции источника $f(x,t)=0$, с начальным условием $u^0(x)=\sin(\pi x)$. На рис. 1 показано решение этой задачи в различные моменты времени при $\alpha=0,95$, $\beta=0,7$, $\gamma=0,5$.

На рис. 2 показаны ошибки приближенных решений Δu для двух моментов времени $t=0.1$ и $t=0.3$, полученных с помощью классического последовательного алгоритма при $m=10^5$ шагов по времени и количестве узлов сетки по пространству а) $n=200$ и б) $n=400$. Как и следовало ожидать, измельчение сетки приводит к уменьшению ошибки при неизменном характере ее распределения.

На рис. 3 показаны ошибки приближенных решений Δu для тех же моментов времени, полученных при использовании параллельного алгоритма с декомпозицией по пространству при $m=M=10^5$ и а) количестве процессоров $P=N=4$ и количестве узлов сетки по пространству в каждой подобласти $n=100$; б) количестве процессоров

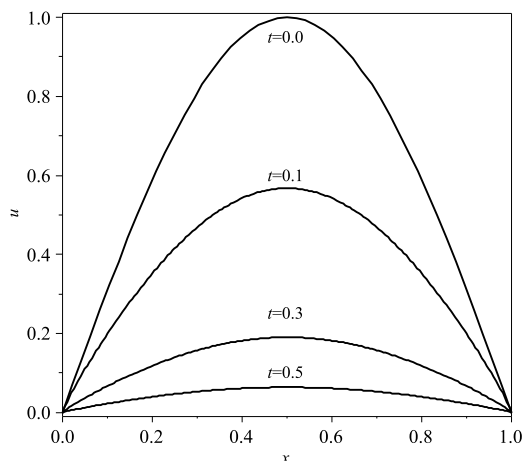
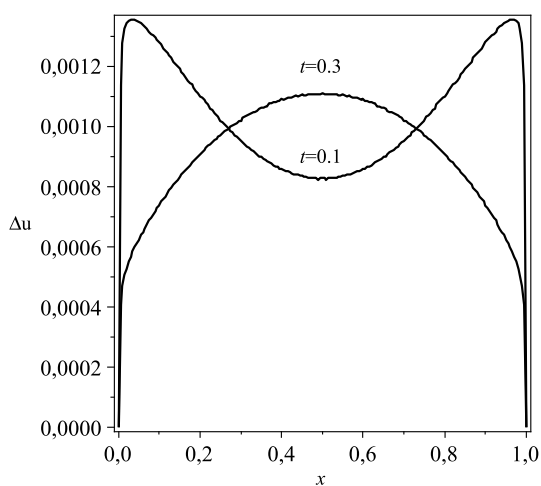
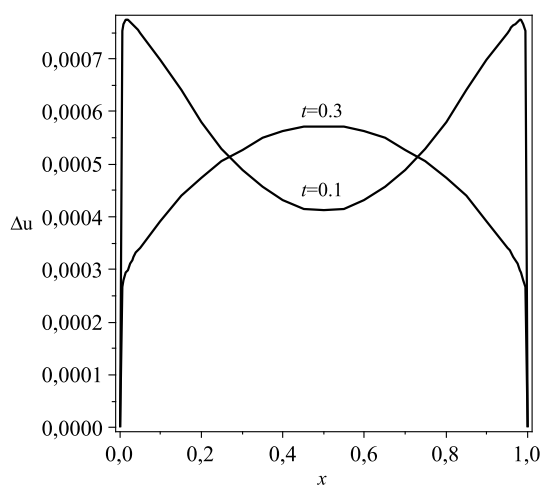


Рис. 1. Решение тестовой задачи для различных моментов времени

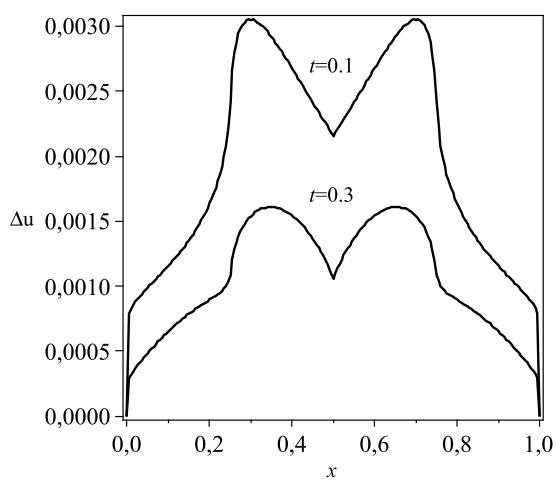


а)

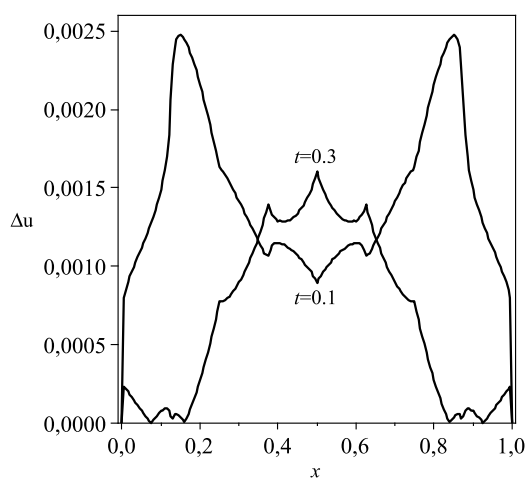


б)

Рис. 2. Ошибка решения Δu , полученного по классическому последовательному алгоритму: а) $n = 200$, б) $n = 400$



а)



б)

Рис. 3. Ошибка решения Δu , полученного с использованием параллельного алгоритма с декомпозицией по пространству: а) $P = N = 4$, $n = 100$, б) $P = N = 8$, $n = 50$

$P = N = 8$ и количестве узлов $n = 50$. Расчеты проводились на одном узле суперкомпьютера Уфимского государственного авиационного технического университета, объединяющего на общей памяти объемом 4 Гб два четырехядерных процессора Intel Xeon 5300. Из приведенных рисунков видно, что параллельный алгоритм показывает тот же порядок точности, что и последовательный.

Заключение

Разработанные двухсеточные параллельные алгоритмы решения уравнений аномальной диффузии дробного порядка показывают сверхлинейное ускорение по сравнению с классическим последовательным конечно-разностным алгоритмом и обеспечивают тот же порядок точности вычислений при условии согласованного выбора шагов точной и грубой сеток. Это свидетельствует о неоптимальности классического алгоритма. Поэтому представляется целесообразным и при последовательных расчетах использовать двухсеточные алгоритмы, получаемые очевидной редукцией предложенных параллельных алгоритмов на однопроцессорный случай.

Статья рекомендована к публикации программным комитетом международной научной конференции «Параллельные вычислительные технологии 2012».

Литература

1. Bouchaud, J.P. Anomalous diffusion in disordered media: statistical mechanisms, models and physical applications / J.P. Bouchaud, A. Georges // Phys. Rep. – 1990. – Vol. 195. – P. 127–293.
2. Metzler, R. The random walk's guide to anomalous diffusion: A fractional dynamic approach / R. Metzler, J. Klafter // Phys. Rep. – 2000. – Vol. 339. – P. 1–77.
3. Учайкин, В.В. Автомодельная аномальная диффузия и устойчивые законы / В.В. Учайкин // УФН. – 2003. – Т. 173, № 8. – С. 847–876.
4. Самко, С.Г. Интегралы и производные дробного порядка и некоторые их приложения / С.Г. Самко, А.А. Килбас, О.И. Маричев – Минск: Наука и техника, 1987.
5. Podlubny, I. Fractional differential equations / I. Podlubny – Academic press, San Diego, 1999.
6. Учайкин, В.В. Метод дробных производных / В.В. Учайкин – Ульяновск: Изд-во «Артишок», 2008.
7. Лукашук, С.Ю. Модификация алгоритма PARAREAL для решения дифференциальных уравнений дробного порядка. / С.Ю. Лукашук // Параллельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции. – Челябинск: Издательский центр ЮУрГУ, 2010. – С. 519–524.
8. Прямые задачи неклассического переноса радионуклидов в геологических формациях / В.М. Головизнин, В.П. Киселев, И.А. Короткин, Ю.И. Юрков // Известия Академии наук, серия «Энергетика». – 2004. – № 4. – С. 121–132.
9. Завьялов, Ю.С. Методы сплайн-функций / Ю.С. Завьялов, Б.И. Квасов, В.Л. Мирошниченко – М.: Наука, 1980.

10. Lions, J.-L. Resolution d'edp par un schema en temps parareal / J.-L. Lions, Y. Maday, G. Turinici // C.R. Acad Sci. Paris. Ser. I Math. – 2001. – Vol. 332. – P. 661–668.
11. Maday, Y. The Parareal in Time Iterative Solver: a Further Direction to Parallel Implementation / Y. Maday, G. Turinici // Lecture Notes in Computational Science and Engineering: Domain Decomposition Methods in Science and Engineering XVIII. – 2005. – Vol. 40. – P. 441–448.
12. Staff, G.A. Stability of the Parareal Algorithm / G.A. Staff, E.M. Ronquist // Lecture Notes in Computational Science and Engineering: Domain Decomposition Methods in Science and Engineering XVIII. – 2005. – Vol. 40. – P. 449–456.
13. Gander, M.J. On the Superlinear and Linear Convergence of the Parareal Algorithm / M.J. Gander, S. Vandewalle // Lecture Notes in Computational Science and Engineering: Domain Decomposition Methods in Science and Engineering XVI. – 2007. – Vol. 55. – P. 291–298.
14. Gander, M.J. Analysis of the parareal time-parallel time-integration method / M.J. Gander, S. Vandewalle // SIAM J. Sci. Comput. – 2007. – Vol. 29, No. 2. – P. 556–578.
15. Fischer, P.F. A Parareal in time semi-implicit approximation of the Navier-Stokes equations / P.F. Fischer, F. Hecht, Y. Maday // Lecture Notes in Computational Science and Engineering: Domain Decomposition Methods in Science and Engineering XVIII. – 2005. – Vol. 40. – P. 433–440.
16. Bal, G. Symplectic Parareal / G. Bal, Q. Wu // Lecture Notes in Computational Science and Engineering: Domain Decomposition Methods in Science and Engineering XVII. – 2008. – Vol. 60. – P. 401–408.

Станислав Юрьевич Лукащук, кандидат физико-математических наук, доцент, кафедра «Высокопроизводительных вычислительных технологий и систем», Уфимский государственный авиационный технический университет, lsu@mail.rb.ru.

PARALLEL TWO-GRIDS ALGORITHMS FOR SOLUTION OF ANOMALOUS DIFFUSION EQUATIONS OF FRACTIONAL ORDER

S. Yu. Lukashchuk, Ufa State Aviation Technical University (Ufa, Russian Federation)

New parallel algorithms are proposed for solving the initial-boundary value problems for anomalous diffusion equations with the Riemann-Liouville spatial- and/or time-fractional derivatives. A two-grid technique is employed to construct these algorithms. Spline-approximation on a coarse grid is used to compute the spatial and time long-range effects, and a fine grid is used for finite-difference discretization of the fractional diffusion equations. The parallel algorithms with a spatial and a time domain decomposition are discussed separately. The approach originally developed for the Parareal algorithm is used for time domain decomposition. The theoretical estimates of the speed-up and efficiency of the proposed algorithms are given. It has been shown that the algorithms have a superlinear speed-up in comparison with a classical sequential finite-difference algorithm, and have the same accuracy if the size of a fine grid is agreed with the size of a coarse grid. Some computational results are also presented to verify the efficiency of the proposed algorithms.

Keywords: parallel two-grid algorithm, anomalous diffusion, fractional differential equation.

References

1. Bouchaud J.P., Georges A. ANomalous Diffusion in Disordered Media: Statistical Mechanisms, Models and Physical APlications. *Phys. Rep.*, 1990, Vol. 195, P. 127–293.
2. Metzler R., Klafter J. The Random Walk's Guide to ANomalous Diffusion: a Fractional Dynamic AProach. *Phys. Rep.*, 2000, Vol. 339, P. 1–77.
3. Uchaikin V.V. Self-Similar ANomalous Diffusion and Levy-Stable Laws. *Phys. Usp.*, 2002, Vol. 46, No. 8, P. 821–849.
4. Samko S., Kilbas A., Marichev O. *Fractional Integrals and Derivatives. Theory and Applications*. Gordon & Breach Sci. Publishers, 1993. 1006 p.
5. Podlubny I. *Fractional Differential Equations*. Academic press, San Diego, 1999. 340 p.
6. Uchaikin V.V. *Metod drobnykh proizvodnykh* [The Method of Fractional Derivatives]. Ul'yanovsk, Artishok Publ., 2008. 512 p.
7. Lukashchuk S.Yu. *Modifikatsiya algoritma PARAREAL dlya resheniya differentsial'nykh uravneniy drobnogo poryadka* [Modification of the PARAREAL Algorithm for Solution of Fractional Differential Equations]. *Trudy MezhdunarodNoy NauchNoy Konferentsii "Parallel'nye Vichislitel'nye Tekhnologii (PaVT'2010)"* [Proc. Int. Conf. "Parallel Computational TechNologies (PCT'2010)"]. Chelyabinsk, YUrGU Publ., 2010, P. 519–524.
8. Goloviznin V.M., Kisilev V.P., Korotkin I.A., Yurkov Yu.I. Pryamye zadachi neklassicheskogo pereNosa radionuklidov v geologicheskikh formatsiyakh [Direct Problems of Nonclassical Radionuclide Transfer in Geological Formations]. *Izv. Ross. Akad. Nauk, Ser. Energetika*, 2004, No. 4, P. 121–132.
9. Zav'yalov Yu.S., Kvasov B.I., Miroshnichenko V.L. *Metody splain-funktsiy* [Methods of Spline-Functions]. Moscow, Nauka, 1980.
10. Lions J.-L., Maday Y., Turinici G. Resolution d'edp par un Schema en Temps Parareal. *C.R. Acad. Sci. Paris. Ser. I Math.*, 2001, Vol. 332, P. 661–668.
11. Maday Y., Turinici G. The Parareal in Time Iterative Solver: a Further Direction to Parallel Implementation *Lecture Notes in Computational Science and Engineering: Domain Decomposition Methods in Science and Engineering XVIII*, 2005, Vol. 40, P. 441–448.
12. Staff G.A., Ronquist E.M. Stability of the Parareal Algorithm *Lecture Notes in Computational Science and Engineering: Domain Decomposition Methods in Science and Engineering XVIII*, 2005, Vol. 40, P. 449–456.
13. Gander M.J., Vandewalle S. On the Superlinear and Linear Convergence of the Parareal Algorithm *Lecture Notes in Computational Science and Engineering: Domain Decomposition Methods in Science and Engineering XVI*, 2007, Vol. 55, P. 291–298.
14. Gander M.J., Vandewalle S. Analysis of the Parareal Time-Parallel Time-Integration Method *SIAM J. Sci. Comput.*, 2007, Vol. 29, No 2, P. 556–578.
15. Fischer P.F., Hecht F., Maday Y. A Parareal in Time Semi-implicit AProximation of the Navier-Stokes Equations *Lecture Notes in Computational Science and Engineering:*

Domain Decomposition Methods in Science and Engineering XVIII, 2005, Vol. 40, P. 433–440.

16. Bal G., Wu Q. Symplectic Parareal *Lecture Notes in Computational Science and Engineering: Domain Decomposition Methods in Science and Engineering XVII*, 2008, Vol. 60, P. 401–408.

Поступила в редакцию 4 ноября 2012 г.

АНАЛИЗ УСТОЙЧИВОСТИ ПОЛОЖЕНИЯ РАВНОВЕСИЯ МОДЕЛИ НЕЙМАНА ПРИ ИНТЕРВАЛЬНОЙ НЕОПРЕДЕЛЕННОСТИ

А.В. Панюков, А.Т. Латипова

Рассматривается анализ устойчивости положения равновесия при интервальных исходных данных. Доказано, что в случае мультипликативной неопределенности прямой и двойственный вектор Фробениуса определяются из точечной модели Неймана с матрицами центров интервалов. В случае интервальной неопределенности интервал для числа Фробениуса можно определить через нахождение положения равновесия для двух точечных моделей Неймана с матрицами, состоящими из верхних и нижних границ интервалов. Также в работе вводятся понятия слабого и сильного решений, которые используются для получения робастных оценок положения равновесия для интервальной модели Неймана.

Ключевые слова: продуктовая стратегия, линейное программирование, модель Неймана, интервальный анализ, теория игр, билинейные системы, программное обеспечение.

Введение

Модель многоотраслевой экономики Неймана оказала большое влияние на теорию экономического роста и накопления капитала и стимулировала развитие математической экономики [1, 2]. Тем не менее во многих работах модель Неймана описывается как нерешаемая чисто теоретическая модель. Практические результаты как правило получают с помощью модели Леонтьева, которая является частным случаем модели Неймана.

Модель Неймана можно применять не только для анализа многоотраслевой экономики, но и для решения других задач, в частности для оптимизации бюджета продаж при ценовой диверсификации (см. [3–5]). Поэтому необходимо разрабатывать численные методы для анализа модели Неймана и внедрять данные методы в пакеты популярного программного обеспечения, например, EXCEL, MATLAB, ПРОЕКТ ЭКСПЕРТ или 1С БУХГАЛТЕРИЯ.

Эффективные численные методы нахождения положения равновесия, использующие арифметику с плавающей точкой, представлены в [6]. В этих методах задача сводится к решению соответствующих матричных игр.

В реальной экономике численные значения для элементов матриц затрат и выпуска получают из данных статистики и экспертных оценок, поэтому может возникать неопределенность, которая зачастую является интервальной. Задача нахождения положения равновесия модели Неймана (A, B) , когда значения для элементов матриц для модели заданы только в виде интервалов, рассмотрена в [7, 8]. Показано, что в случае мультипликативной неопределенности прямой и двойственный векторы Фробениуса можно получить из точечной модели Неймана с матрицами центров интервалов, а в случае интервальной неопределенности интервал для числа Фробениуса можно получить, определив положение равновесия для двух точечных моделей Неймана с матрицами верхних и нижних границ интервалов.

В данной работе изложены результаты, анонсированные в [9].

1. Положение равновесия для точечной модели Неймана

Чтобы сохранить целостность повествования, введем ряд определений и рассмотрим кратко полученные ранее результаты.

Определение 1. *Общим* положением равновесия для модели Неймана (A, B) , где A и B – это матрицы $m \times n$ затрат и выпуска, элементы которых являются неотрицательными числами

$$a_{ij} \geq 0, b_{ij} \geq 0, i = 1, 2, \dots, n, j = 1, 2, \dots, m,$$

называют решение (λ, x, w) системы билинейных неравенств и уравнений

$$(A - \lambda B)x \leq 0, (x, e^m) = 1, x \geq 0, \quad (1)$$

$$(A - \lambda B)^T w \geq 0, (w, e^n) = 1, w \geq 0, \quad (2)$$

где $e^l \in \mathbf{R}^l$, $l \in \{m, n\}$, $(\forall i = 1, 2, \dots, l) e_i^l = 1$.

Из данного определения общего положения равновесия следует, что $\lambda \geq 0$.

Определение 2. *Невырожденным* положением равновесия в модели Неймана называется положением равновесия (λ, x, w) , при котором выполняется дополнительное условие (λ, x, w)

$$w^T A x > 0. \quad (3)$$

В данной статье мы рассматриваем алгоритмы для нахождения общего положения равновесия; т.е. мы хотим найти решение (λ, x, w) для системы (1)-(2).

В общем случае система (1)-(2) имеет не единственное решение. Поскольку число λ^{-1} интерпретируется как равновесный темп роста, который достигается за счет структуры интенсивностей процессов x и структуры равновесных цен на товары w , то для теории и практики наибольший интерес представляют частные положения равновесия с экстремальными значениями λ .

Экстремальные допустимые значения λ можно найти, решив следующие оптимизационные задачи.

$$\lambda^* = \min \{ \lambda : (A - \lambda B)x \leq 0, (x, e^m) = 1, x \geq 0 \}, \quad (4)$$

$$\lambda_n = \max \left\{ \lambda : (A - \lambda B)^T w \geq 0, (w, e^n) = 1, w \geq 0 \right\}. \quad (5)$$

Определение 3. Числа λ_n и λ^* для модели Неймана (A, B) называются *числами Неймана и Фробениуса* соответственно. Число Неймана λ_n определяет максимальный темп равновесного роста, а число Фробениуса λ^* определяет минимально возможный темп равновесного роста и продуктивность модели (см. [1, 2]).

Определение 4. Векторы x, w положения равновесия (λ, x, w) называются *прямым и двойственным лучами Неймана*, соответствующие значению λ .

Из ограничений (1), (2) и (4) следует, что для нахождения параметров продуктивности (числа Фробениуса λ^*) и устойчивого положения равновесия необходимо решить следующую задачу билинейного программирования

$$(\lambda^*, x^*, w^*)^T = \arg \max_{(\lambda, x, w)^T \in D(A, B)} \lambda, \quad (6)$$

$$D(A, B) = \left\{ \left(\begin{array}{c} \lambda \\ x \\ w \end{array} \right) \left| \begin{array}{l} (A - \lambda B)x \leq 0, \\ (A - \lambda B)^T w \geq 0, \\ (x, e^m) = 1, (w, e^n) = 1, \\ x \geq 0, w \geq 0, \lambda \geq 0 \end{array} \right. \right\}.$$

Далее под положением равновесия будут подразумеваться параметры продуктивности, если не будет уточнений.

Численные методы нахождения задачи (6) рассмотрены в [6]. Они основаны на нахождении корней монотонных функций

$$u(\lambda) = \min_{x: (x, e^m)=1, x \geq 0} \max_{i=1,2,\dots,n} \sum_{j=1}^m (a_{ij} - \lambda b_{ij}) x_j$$

или

$$v(\lambda) = \max_{w: (w, e^n)=1, w \geq 0} \min_{j=1,2,\dots,m} \sum_{i=1}^n (a_{ij} - \lambda b_{ij}) w_i.$$

При фиксированном λ значения функций $u(\lambda)$ и $v(\lambda)$ равны оптимальным значениям взаимодвойственных задач линейного программирования.

$$\min \{u : (A - \lambda B)x \leq u, (x, e^m) = 1, x \geq 0\}, \quad (7)$$

$$\max \left\{ v : (A - \lambda B)^T w \geq v, (w, e^n) = 1, w \geq 0 \right\}. \quad (8)$$

Когда значение λ близко к корням функций $u(\lambda)$, $v(\lambda) \rightarrow 0$ задачи (7) и (8) становятся вырожденными из-за близости к нулю базисных переменных u и v в оптимальном базисном решении этих задач. Поэтому задачи (7) и (8) не могут быть решены традиционными методами, использующими вычисления с плавающей точкой.

Далее рассмотрим наши результаты по нахождению корней функций $u(\lambda)$ и $v(\lambda)$ с помощью численных методов матричных игр. Первую теорему можно применять в случае отсутствия нулевых элементов в матрице B .

Теорема 1. [6]

Пусть (A, B) является моделью Неймана, и все элементы матрицы B положительны. Пусть Γ_C является матричной игрой с платежной матрицей $C = [a_{ij}/b_{ij}]$ ($i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$), и пусть $\hat{\lambda}$ является ценой игры Γ_C . Тогда число Неймана λ_n и число Фробениуса λ^* модели Неймана (A, B) равны $\hat{\lambda}$.

Если нулевые элементы отсутствуют в матрице A , то имеет место

Теорема 2. [7]

Пусть (A, B) – модель Неймана, и все элементы матрицы A положительны. Пусть Γ_C является матричной игрой с платежной матрицей $C = [b_{ij}/a_{ij}]$ ($i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$), и пусть $\hat{\mu}$ является ценой игры Γ_C . Тогда число Неймана λ_n и число Фробениуса λ_n модели Неймана $1/\hat{\mu}$.

Отметим, что теорема 1 (2) применима только для моделей Неймана для которых матрицы B (A) состоят из положительных элементов. Для следующего подхода некоторые элементы матриц A и B могут быть равны нулю. Поэтому он имеет более широкую область применения. Тем не менее теоремы 1 и 2 предполагают решение

только одной матричной игры, тогда как подход, представленный ниже, требует решения нескольких матричных игр.

Теорема 3. [6]

Пусть Γ является матричной игрой с платежной матрицей $(A - \lambda B)^T$, а \hat{x} , \hat{w} – первого и второго игроков, соответственно. Пусть \hat{u} – цена игры Γ . Тогда (\hat{u}, \hat{x}) и (\hat{u}, \hat{w}) являются оптимальными решениями задач (7) и (8), соответственно.

Доказательство теоремы является тривиальным. Оно следует из игровой интерпретации двойственности задач линейного программирования.

Существуют эффективные итеративные алгоритмы решения матричных игр. В частности, для решения задач (4)–(5) можно использовать алгоритм фиктивного разыгрывания, который является устойчивым к погрешностям округления, в отличие от симплекс-метода.

Симплекс-метод также можно применять при решении матричных игр, так как игра $\bar{\Gamma}$ с платежной матрицей $(A - \lambda B)^T + \gamma I$ имеет цену $\bar{u} = \hat{u} + \gamma$ и оптимальные решения $\bar{x} = \hat{x}$, $\bar{w} = \hat{w}$, где I – $(n \times m)$ -матрица, все элементы которой равны 1,

$$\gamma = \max\{a_{ij} - \lambda b_{ij} : i = 1, \dots, m, j = 1, \dots, n\},$$

при этом соответствующие задачи линейного программирования для игры с платежной матрицей $(A - \lambda B)^T$ имеют невырожденное оптимальное решение.

2. Положение равновесия для интервальной модели Неймана

В дальнейшем через \mathbf{A} и \mathbf{B} будем обозначать интервальные матрицы затрат и выпуска, т.е. элементы матриц \mathbf{A} и \mathbf{B} представляют собой числовые интервалы.

Определение 5. Интервальной назовем модель Неймана с интервальными матрицами затрат $\mathbf{A} = \{\mathbf{a}_{ij}\} = \left\{ \left[\underline{a}_{ij}, \overline{a}_{ij} \right] \right\}$ и выпуска $\mathbf{B} = \{\mathbf{b}_{ij}\} = \left\{ \left[\underline{b}_{ij}, \overline{b}_{ij} \right] \right\}$, $i = (\overline{1}, n)$, $j = (\overline{1}, m)$, $\text{mid}\mathbf{A} = (\underline{\mathbf{A}} + \overline{\mathbf{A}})/2$, $\text{mid}\mathbf{B} = (\underline{\mathbf{B}} + \overline{\mathbf{B}})/2$.

Обозначим с помощью $\text{mid}\mathbf{A}$ и $\text{mid}\mathbf{B}$ матрицы центров интервалов для элементов \mathbf{A} и \mathbf{B} соответственно.

Точечные матрицы верхних границ интервалов для элементов \mathbf{A} и \mathbf{B} обозначим $\overline{\mathbf{A}}$ и $\overline{\mathbf{B}}$ соответственно, а матрицы нижних границ интервалов для элементов матриц \mathbf{A} и \mathbf{B} $\underline{\mathbf{A}}$ и $\underline{\mathbf{B}}$ соответственно [10].

Теорема 4. Пусть (\mathbf{A}, \mathbf{B}) является интервальной моделью Неймана, и для точечных матриц $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}$ выполняется условие

$$\tilde{\mathbf{A}} \in \mathbf{A}, \quad \tilde{\mathbf{B}} \in \mathbf{B}. \tag{9}$$

Пусть также

$$(\tilde{\lambda}, \tilde{x}, \tilde{w})^T = \arg \max_{(\lambda, x, w)^T \in D(\tilde{\mathbf{A}}, \tilde{\mathbf{B}})} \lambda; \tag{10}$$

$$(\bar{\lambda}, \bar{x}, \bar{w})^T = \arg \max_{(\lambda, x, w)^T \in D(\overline{\mathbf{A}}, \overline{\mathbf{B}})} \lambda; \tag{11}$$

$$(\underline{\lambda}, \underline{x}, \underline{w})^T = \arg \max_{(\lambda, x, w)^T \in D(\underline{\mathbf{A}}, \underline{\mathbf{B}})} \lambda. \tag{12}$$

Тогда $\underline{\lambda} \leq \tilde{\lambda} \leq \bar{\lambda}$.

Доказательство. Из условия (10)

$$(\forall j = 1, 2, \dots, n \quad j = 1, 2, \dots, m) \quad (\bar{a}_{ij} \geq \tilde{a}_{ij}, \bar{b}_{ij} \geq \tilde{b}_{ij}, \underline{a}_{ij} \leq \tilde{a}_{ij}, \underline{b}_{ij} \leq \tilde{b}_{ij}). \quad (13)$$

Тогда точечные матрицы можно представить следующим образом.

$$\tilde{A} = \underline{A} + A' = \bar{A} - A'', \quad \tilde{B} = \underline{B} + B' = \bar{B} - B'',$$

где

$$A' = (a'_{ij}) = (\tilde{a}_{ij} - \underline{a}_{ij}), \quad B' = (b'_{ij}) = (\tilde{b}_{ij} - \underline{b}_{ij}),$$

$$A'' = (a''_{ij}) = (\bar{a}_{ij} - \tilde{a}_{ij}), \quad B'' = (b''_{ij}) = (\bar{b}_{ij} - \tilde{b}_{ij}),$$

при этом все элементы матриц A' , B' , A'' и B'' неотрицательны. Подставим $\tilde{A} = \bar{A} - A''$ и $\tilde{B} = \underline{B} + B'$ в задачу (10), получим

$$(\tilde{\lambda}, \tilde{x}, \tilde{w})^T = \arg \max_{(\lambda, x, w)^T \in D(\tilde{A}, \tilde{B})} \lambda, \quad (14)$$

$$D(\tilde{A}, \tilde{B}) = \left\{ \begin{pmatrix} \tilde{\lambda} \\ x \\ w \end{pmatrix} \left| \begin{array}{l} (\bar{A} - A'' - \lambda(\underline{B} + B'))x \leq 0, \\ (\bar{A} - A'' - \lambda(\underline{B} + B'))^T w \geq 0, \\ (x, e^m) = 1, (w, e^n) = 1, \\ x \geq 0, w \geq 0, \lambda \geq 0 \end{array} \right. \right\}.$$

Ввиду неотрицательности A'' и B'' во втором ограничении (14)

$$(\bar{A} - \tilde{\lambda}\underline{B})^T \tilde{w} \geq 0.$$

Таким образом, $(\forall j = 1, 2, \dots, m)$ выполняется

$$\tilde{\lambda} \leq \frac{\sum_{i=1}^n \bar{a}_{ij} \tilde{w}_i}{\sum_{i=1}^n \underline{b}_{ij} \tilde{w}_i} \leq \frac{\sum_{i=1}^n \bar{a}_{ij} \bar{w}_i}{\sum_{i=1}^n \underline{b}_{ij} \bar{w}_i}. \quad (15)$$

Последнее неравенство в данной последовательности следует из условия (11) теоремы, которое заключается в

$$\bar{w} = \arg \max_{w: (w, e^n) = 1, w \geq 0} \min_{j=1, 2, \dots, m} \frac{\sum_{i=1}^n \bar{a}_{ij} w_i}{\sum_{i=1}^n \underline{b}_{ij} w_i},$$

$$\bar{\lambda} = \max_{w: (w, e^n) = 1, w \geq 0} \min_{j=1, 2, \dots, m} \frac{\sum_{i=1}^n \bar{a}_{ij} w_i}{\sum_{i=1}^n \underline{b}_{ij} w_i},$$

отсюда $\tilde{\lambda} \leq \bar{\lambda}$.

Неравенство $\tilde{\lambda} \geq \underline{\lambda}$ доказывается аналогично. На самом деле, подставив $\tilde{A} = \underline{A} + A'$ и $\tilde{B} = \overline{B} - B''$ в задачу (10), получаем

$$(\tilde{\lambda}, \tilde{x}, \tilde{w})^T = \arg \max_{(\lambda, x, w)^T \in D(\tilde{A}, \tilde{B})} \lambda, \quad (16)$$

$$D(\tilde{A}, \tilde{B}) = \left\{ \begin{pmatrix} \lambda \\ x \\ w \end{pmatrix} \left| \begin{array}{l} (\underline{A} + A' - \lambda(\overline{B} - B''))x \leq 0, \\ (\underline{A} + A' - \lambda(\overline{B} - B''))^T w \geq 0, \\ (x, e^m) = 1, (w, e^n) = 1, x \geq 0, \\ w \geq 0, \lambda \geq 0 \end{array} \right. \right\}.$$

Из неотрицательности A' и B'' и первого неравенства (16) следует $(\underline{A} - \tilde{\lambda}\overline{B})\tilde{x} \leq 0$, поэтому $(\forall i = 1, 2, \dots, n)$ выполняется неравенство:

$$\tilde{\lambda} \geq \frac{\sum_{j=1}^m a_{ij}\tilde{x}_j}{\sum_{j=1}^m \bar{b}_{ij}\tilde{x}_j} \geq \frac{\sum_{j=1}^m a_{ij}\underline{x}_j}{\sum_{j=1}^m \bar{b}_{ij}\underline{x}_j}. \quad (17)$$

Последнее неравенство следует из (12)

$$\underline{x} = \arg \min_{x:(x, e^m)=1, x \geq 0} \max_{i=1,2,\dots,n} \frac{\sum_{j=1}^m a_{ij}x_j}{\sum_{j=1}^m \bar{b}_{ij}x_j},$$

$$\underline{\lambda} = \min_{x:(x, e^m)=1, x \geq 0} \max_{i=1,2,\dots,n} \frac{\sum_{j=1}^m a_{ij}x_j}{\sum_{j=1}^m \bar{b}_{ij}x_j},$$

отсюда $\tilde{\lambda} \geq \underline{\lambda}$.

Теорема 4 доказана. □

Теорема 4 показывает, что для интервальной модели Неймана (A, B) число Фробениуса принадлежит интервалу $\Lambda = [\underline{\lambda}, \bar{\lambda}]$, где $\bar{\lambda}$ – число Фробениуса для точечной модели Неймана $(\overline{A}, \overline{B})$, а $\underline{\lambda}$ – число Фробениуса для точечной модели Неймана $(\underline{A}, \underline{B})$. Тогда любая точечная модель Неймана $(\tilde{A}, \tilde{B}) \in (A, B)$: $\tilde{A} \in A$, $\tilde{B} \in B$ имеет число Фробениуса $\tilde{\lambda} \in \Lambda$.

Теорема 5. Пусть (A, B) является интервальной моделью Неймана, а числа β_A, β_B и матрицы \tilde{A}, \tilde{B} удовлетворяют ограничениям

$$\tilde{A} = \beta_A \cdot \text{mid}A \in A, \quad \tilde{B} = \beta_B \cdot \text{mid}B \in B.$$

Если

$$(\lambda^*, x^*, w^*)^T = \arg \max_{(\lambda, x, w)^T \in D(\text{mid}A, \text{mid}B)} \lambda,$$

то

$$\left(\frac{\lambda^* \beta_A}{\beta_B}, x^*, w^* \right)^T = \arg \max_{(\lambda, x, w)^T \in D(\tilde{A}, \tilde{B})} \lambda.$$

Доказательство. Задача нахождения положения равновесия для модели (\tilde{A}, \tilde{B}) является следующей.

$$(\tilde{\lambda}^*, \tilde{x}, \tilde{w})^T = \arg \max_{(\tilde{\lambda}, x, w)^T \in D(\tilde{A}, \tilde{B})} \tilde{\lambda}, \quad (18)$$

$$D(\tilde{A}, \tilde{B}) = \left\{ \left(\begin{array}{c} \tilde{\lambda} \\ x \\ w \end{array} \right) \left| \begin{array}{l} (\tilde{A} - \tilde{\lambda}\tilde{B})x \leq 0, \\ (\tilde{A} - \tilde{\lambda}\tilde{B})^T w \geq 0, \\ (x, e^m) = 1, (w, e^n) = 1, \\ x \geq 0, w \geq 0, \tilde{\lambda} \geq 0 \end{array} \right. \right\}.$$

Заменим переменную λ на $\tilde{\lambda} = \lambda\beta_A/\beta_B$ в задаче (18).

Получим

$$(\lambda^*, \tilde{x}, \tilde{w})^T = \arg \max_{(\lambda, x, w)^T \in D(\tilde{A}, \tilde{B})} \lambda, \quad (19)$$

$$D(\tilde{A}, \tilde{B}) = \left\{ \left(\begin{array}{c} \tilde{\lambda} \\ x \\ w \end{array} \right) \left| \begin{array}{l} (\tilde{A} - \lambda\beta_A/\beta_B\tilde{B})x \leq 0, \\ (\tilde{A} - \lambda\beta_A/\beta_B\tilde{B})^T w \geq 0, \\ (x, e^m) = 1, (w, e^n) = 1, \\ x \geq 0, w \geq 0, \lambda \geq 0 \end{array} \right. \right\}.$$

Используем матрицы $\text{mid}\mathbf{A}$ и $\text{mid}\mathbf{B}$ вместо \tilde{A} и \tilde{B} мы получаем следующую эквивалентную задачу.

$$(\lambda^*, x^*, w^*)^T = \arg \max_{(\lambda, x, w)^T \in D(\text{mid}\mathbf{A}, \text{mid}\mathbf{B})} \lambda, \quad (20)$$

$$D \left(\begin{array}{c} \text{mid}\mathbf{A}, \\ \text{mid}\mathbf{B} \end{array} \right) = \left\{ \left(\begin{array}{c} \lambda \\ x \\ w \end{array} \right) \left| \begin{array}{l} (\text{mid}\mathbf{A} - \lambda\text{mid}\mathbf{B})x \leq 0, \\ (\text{mid}\mathbf{A} - \lambda\text{mid}\mathbf{B})^T w \geq 0, \\ (x, e^m) = 1, (w, e^n) = 1, \\ x \geq 0, w \geq 0, \lambda \geq 0 \end{array} \right. \right\}.$$

Эта задача совпадает с задачей нахождения положения равновесия (λ^*, x^*, w^*) для точечной модели Неймана $(\text{mid}\mathbf{A}, \text{mid}\mathbf{B})$. Учитывая преобразование переменной λ , получаем, что кортеж $(\lambda^*\beta_A/\beta_B, x^*, w^*)$ является положением равновесия для модели (\tilde{A}, \tilde{B}) .

Теорема 5 доказана. □

Таким образом, в случае мультипликативной неопределенности (коэффициенты пропорциональности β_A и β_B неизвестны), несмотря на изменение числа Фробениуса, прямой и двойственный векторы Фробениуса не изменяются. Однако такая устойчивость может наблюдаться и при более общей интервальной неопределенности.

Пример 1. Модель (A, B) размера 2×2 с элементами, удовлетворяющими следующими ограничениями

$$a_{21} + a_{22} \leq \lambda^*(b_{21} + b_{22}); a_{11} = b_{11}; a_{12} = b_{12};$$

имеет положение равновесия

$$\{\lambda^*, (x^*)^T = (0, 5; 0, 5), (w^*)^T = (1; 0)\}.$$

Из этого примера видно, что существуют интервалы для элементов матриц A и B , для которых прямой и двойственный векторы Фробениуса (x^*, w^*) не меняются. Поэтому введем понятия *сильного решения* и *слабого решения*.

Определение 6. *Сильным* решением для интервальной модели Неймана (\mathbf{A}, \mathbf{B}) назовем пару векторов (x_s, w_s) такую, что для любой точечной модели Неймана (\tilde{A}, \tilde{B}) : $\tilde{A} \in \mathbf{A}$, $\tilde{B} \in \mathbf{B}$ существует положение равновесия $(\tilde{\lambda}, x_s, w_s)$.

Наличие слабого решения обеспечивает допустимость ограничений (1)-(2) для интервальной модели (\mathbf{A}, \mathbf{B}) .

Определение 7. *Слабым* решением для интервальной модели Неймана (\mathbf{A}, \mathbf{B}) назовем пару векторов (x', w') такую, что для любой точечной модели Неймана (\tilde{A}, \tilde{B}) : $\tilde{A} \in \mathbf{A}$, $\tilde{B} \in \mathbf{B}$ выполняются ограничения

$$\begin{cases} (\tilde{A} - \lambda \tilde{B})x' \leq 0; & (\tilde{A} - \lambda \tilde{B})^T w' \geq 0; \\ (x', e^m) = 1; & (w', e^n) = 1; \quad x', w', \lambda \geq 0. \end{cases}$$

Следующее утверждение можно использовать для проверки, является ли пара векторов (x, w) слабым решением.

Теорема 6. *Если система ограничений*

$$\begin{cases} (\underline{\mathbf{A}} - \lambda \underline{\mathbf{B}})x'' \leq 0; & (\underline{\mathbf{A}} - \lambda \underline{\mathbf{B}})^T w'' \geq 0; \\ (x'', e^m) = 1; & (w'', e^n) = 1; \quad x'', w'' > 0; \end{cases}$$

разрешима для пары векторов (x'', w'') , то (x'', w'') является слабым решением интервальной модели Неймана (\mathbf{A}, \mathbf{B}) .

Доказательство данной теоремы является по сути компиляцией доказательства теоремы 2.26, приведенной в [11].

Если модель (\mathbf{A}, \mathbf{B}) имеет слабое решение, то это решение можно использовать для оценки числа Фробениуса λ' .

Теорема 7. *Пусть (x', w') является слабым решением интервальной модели Неймана (\mathbf{A}, \mathbf{B}) . Пусть точечная модель Неймана (\tilde{A}, \tilde{B}) : $\tilde{A} \in \mathbf{A}$, $\tilde{B} \in \mathbf{B}$ имеет*

$$\lambda' = \max\{\lambda \mid (\tilde{A} - \lambda \tilde{B})x' \leq 0; (\tilde{A} - \lambda \tilde{B})^T w' \geq 0\},$$

тогда $\lambda' \in [\underline{\lambda}_n; \bar{\lambda}]$, где $\underline{\lambda}_n$ – число Неймана для модели $(\underline{\mathbf{A}}, \bar{\mathbf{B}})$.

Доказательство. Пусть $(\tilde{\lambda}, \tilde{x}, \tilde{w})$ является положением равновесия для модели (\tilde{A}, \tilde{B}) . Если $\lambda' = \max\{\lambda \mid (\tilde{A} - \lambda \tilde{B})x' \leq 0; (\tilde{A} - \lambda \tilde{B})^T w' \geq 0\}$, то

$$\lambda' = \min \frac{\sum_{i=1}^n \tilde{a}_{ij} w'_i}{\sum_{i=1}^n \tilde{b}_{ij} w'_i}.$$

Так как кортеж (λ', x', w') является допустимым, но не оптимальным решением (4), $\lambda' \leq \tilde{\lambda}$. Согласно (4) $\tilde{\lambda} \leq \bar{\lambda}$, откуда $\lambda' \leq \bar{\lambda}$.

Пара (x', w') является слабым решением модели Неймана (\mathbf{A}, \mathbf{B}) , значит, (x', w') являются допустимыми векторами для модели $(\underline{\mathbf{A}}, \bar{\mathbf{B}})$, поэтому ограничение

$$(\underline{\mathbf{A}} - \lambda \bar{\mathbf{B}})w' \geq 0.$$

выполняется.

Пусть $\underline{\lambda}' = \max\{\lambda \mid (\underline{\mathbf{A}} - \lambda \overline{\mathbf{B}})w' \geq 0\}$, тогда

$$\underline{\lambda}' = \min \frac{\sum_{i=1}^n a_{ij} w'_i}{\sum_{i=1}^n \bar{b}_{ij} w'_i}.$$

Кортеж $(\underline{\lambda}', x', w')$ является допустимым решением для модели $(\underline{\mathbf{A}}, \overline{\mathbf{B}})$, поэтому $\underline{\lambda} \geq \underline{\lambda}' \geq \underline{\lambda}_n$. Так как $(\check{\mathbf{A}}, \check{\mathbf{B}}) : \check{\mathbf{A}} \in \mathbf{A}, \check{\mathbf{B}} \in \mathbf{B}$, мы получаем $\underline{\mathbf{A}} = \check{\mathbf{A}} - \Delta \mathbf{A}, \overline{\mathbf{B}} = \check{\mathbf{B}} + \Delta \mathbf{B}, (\Delta a_{ij} \geq 0, \Delta b_{ij} \geq 0)$. Откуда

$$\underline{\lambda}_n \leq \underline{\lambda}' = \min \frac{\sum_{i=1}^n a_{ij} w'_i}{\sum_{i=1}^n \bar{b}_{ij} w'_i} \leq \lambda' = \min \frac{\sum_{i=1}^n \tilde{a}_{ij} w'_i}{\sum_{i=1}^n \tilde{b}_{ij} w'_i}.$$

Таким образом, $\lambda' \geq \underline{\lambda}_n$. □

Согласно теореме (7) оценка λ' числа Фробениуса может быть меньше $\underline{\lambda}$.

В следующей теореме применяется другой подход для оценки числа Фробениуса.

Теорема 8. Пусть (x^*, w^*) являются прямым и двойственным векторами Фробениуса для двух точечных моделей Неймана $(\check{\mathbf{A}}, \check{\mathbf{B}})$ и $(\hat{\mathbf{A}}, \hat{\mathbf{B}})$, для которых $\hat{a}_{ij} - \check{a}_{ij} = \Delta a_{ij} \geq 0, \hat{b}_{ij} - \check{b}_{ij} = \Delta b_{ij} \geq 0$. Пусть число Фробениуса модели $(\check{\mathbf{A}}, \check{\mathbf{B}})$ равно $\check{\lambda}$, а модель $(\hat{\mathbf{A}}, \hat{\mathbf{B}})$ имеет число Фробениуса $\hat{\lambda}$, $\Delta \lambda = \hat{\lambda} - \check{\lambda}$. Тогда если положение равновесия $(\check{\lambda}, x^*, w^*)$ не является вырожденным для модели $(\check{\mathbf{A}}, \check{\mathbf{B}})$, то

$$\Delta \lambda = \frac{(w^*)^T (\Delta \mathbf{A} - \check{\lambda} \Delta \mathbf{B}) x^*}{(w^*)^T (\check{\mathbf{B}} + \Delta \mathbf{B}) x^*}. \quad (21)$$

Доказательство. В соответствии с теоремой о дополняющей нежесткости должны выполняться следующие ограничения

$$(w^*)^T (\check{\mathbf{A}} - \check{\lambda} \check{\mathbf{B}}) x^* = 0; \quad (22)$$

$$(w^*)^T (\hat{\mathbf{A}} - \hat{\lambda} \hat{\mathbf{B}}) x^* = 0. \quad (23)$$

Если положение равновесия $(\check{\lambda}, x^*, w^*)$ не является вырожденным для модели $(\check{\mathbf{A}}, \check{\mathbf{B}})$, то в соответствии с (1)-(3) получаем

$$(w^*)^T \check{\mathbf{A}} x^* > 0; \quad (\check{\mathbf{A}} - \check{\lambda} \check{\mathbf{B}}) x^* \leq 0; \quad w^* \geq 0.$$

Из этого следует

$$(w^*)^T \check{\mathbf{B}} x^* > 0. \quad (24)$$

Отсюда

$$\check{\lambda} = \frac{(w^*)^T \check{\mathbf{A}} x^*}{(w^*)^T \check{\mathbf{B}} x^*} \quad (25)$$

Сделаем замену $\hat{\lambda} = \check{\lambda} + \Delta\lambda$, $\hat{A} = \check{A} + \Delta_A$ и $\hat{B} = \check{B} + \Delta_B$ в (25); откуда

$$(w^*)^T (\check{A} - \check{\lambda}\check{B} + \Delta_A - \check{\lambda}\Delta_B - \Delta\lambda(\check{B} + \Delta_B))x^* = 0,$$

тогда с учетом (22) и (24) получаем

$$\Delta\lambda = \frac{(w^*)^T (\Delta_A - \check{\lambda}\Delta_B)x^*}{(w^*)^T (\check{B} + \Delta_B)x^*}.$$

□

Если интервальная модель Неймана (\mathbf{A}, \mathbf{B}) имеет сильное решение (x_s, w_s) , мы можем оценить изменение числа Фробениуса для точечных моделей Неймана.

Теорема 9. Пусть (x_s, w_s) является сильным решением интервальной модели Неймана $([\check{A}, \hat{A}], [\check{B}, \hat{B}])$, обе модели Неймана (\check{A}, \check{B}) и (\hat{A}, \hat{B}) имеют невырожденные положения равновесия $(\check{\lambda}, x_s, w_s)$ и $(\hat{\lambda}, x_s, w_s)$ соответственно. Пусть также точечная модель Неймана (\tilde{A}, \tilde{B}) : $\tilde{A} = \hat{A} + k\Delta_A$, $\tilde{B} = \hat{B} + k\Delta_B$, $k \in [0; 1]$ имеет число Фробениуса $\tilde{\lambda}$. Тогда

$$\tilde{\lambda} - \check{\lambda} = k\Delta\lambda \frac{w_s^T (\check{B} + \Delta_B)x_s}{w_s^T (\tilde{B} + k\Delta_B)x_s}. \quad (26)$$

Доказательство теоремы (9) аналогично доказательству теоремы (8).

Заключение

Прямой и двойственный лучи Неймана для матрицы затрат A и матрицы выпуска B определяются моделью Неймана $(\text{mid}\mathbf{A}, \text{mid}\mathbf{B})$.

Число Фробениуса для модели Неймана (\mathbf{A}, \mathbf{B}) с интервальными матрицами затрат и выпуска принадлежит интервалу $\mathbf{\Lambda} = [\underline{\lambda}, \bar{\lambda}]$, где $\bar{\lambda}$ число Фробениуса для точечной модели Неймана $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$, а $\underline{\lambda}$ число Фробениуса для точечной модели Неймана $(\underline{\mathbf{A}}, \underline{\mathbf{B}})$, $\bar{\mathbf{A}}, \bar{\mathbf{B}}$ являются точечными матрицами верхних границ интервалов для матриц \mathbf{A} и \mathbf{B} , соответственно; $\underline{\mathbf{A}}, \underline{\mathbf{B}}$ являются точечными матрицами нижних границ интервалов для матриц. Это означает, что для любой точечной модели Неймана $(\tilde{A}, \tilde{B}) \in (\mathbf{A}, \mathbf{B})$ число Фробениуса $\tilde{\lambda} \in \mathbf{\Lambda}$.

Введенные понятия слабого и сильного решения для интервальной модели Неймана являются достаточно конструктивными и позволяют получать робастные оценки для положения равновесия. В дальнейшем предполагается разработать и изучить эффективные алгоритмы нахождения сильного и слабого решений для интервальной модели Неймана.

Литература

1. Ашманов, С.А. Введение в математическую экономику / А.С. Ашманов. – М.: Наука, 1984. – 296 с.
2. Альсевич, В.В. Введение в математическую экономику. Конструктивная теория / В.В. Альсевич. – М.: Либроком, 2005. – 256 с.

3. Латипова, А.Т. Модель оптимизации ценовой стратегии для задач бюджетирования / А.Т. Латипова; под ред. Ю.А. Кочетова // Труды Российской конференции «Дискретный анализ и исследование операций» (Новосибирск, 2004). – Новосибирск: Изд-во Ин-та математики СО РАН, 2004. – С. 206.
4. Латипова, А.Т. Ценовая диверсификация в бюджетировании / А.Т. Латипова; под ред. В.А. Кежаева // Труды Международной конференции «Экономика и управление: проблемы и перспективы» (Санкт-Петербург, 2005). – СПб.: Изд-во СПбГТУ, 2005. – С. 562–566.
5. Панюков, А.В. Оптимизация бюджета продаж / А.В. Панюков, А.Т. Латипова // Вестник Южно-Уральского государственного университета. Сер. «Рынок: Теория и практика». – 2006. – Вып. 4. – № 15(70). – С. 116–120.
6. Panyukov, A.V. Numerical Techniques for Finding Equilibrium in von Neumann's Model // Computational Mathematics and Mathematical Physics. – 2008. – Issue 14, Vol. 48. – P. 1999–2006.
7. Panyukov, A.V. Finding Equilibrium in von Neumann's Model / A.V. Panyukov, A.T. Latipova // Proceedings of 13th IFAC Symposium on Information Control Problems in Manufacturing. – 2010. – Vol. 13. Part 1. – URL: <http://www.ifac-papersonline.net/Detailed/40647.html> (Дата обращения 10.10.2012).
8. Панюков, А.В. Оценка положения равновесия в модели Неймана при интервальной неопределенности исходных данных / А.В. Панюков, А.Т. Латипова // Вестник УГАТУ, Сер. «Управление, вычислительная техника и информатика». – 2008. – Вып. 2(27), № 10. – С. 150–153.
9. Panyukov, A.V. Stability Analysis of Equilibrium Position of Von Neumann's Model under Interval Uncertainty / A.V. Panyukov, A.T. Latipova // Proceedings of 14th IFAC Symposium on Information Control Problems in Manufacturing. – 2012. – Vol. 14. Part 1. – URL: <http://www.ifac-papersonline.net/Detailed/53981.html> (Дата обращения 10.10.2012).
10. Jaulin, L. Applied Interval Analysis / L. Jaulin, M. Kieffer, O. Didrit, E. Walter. – Springer-Verlag, 2001. – 382 p.
11. Фидлер, М. Задачи линейной оптимизации с неточными данными / М. Фидлер, Й. Недома. – М. – Ижевск: НИЦ «Регулярная и хаотическая динамика». – 2008. – 288 с.

Анатолий Васильевич Панюков, доктор физико-математических наук, профессор, кафедра экономико-математических методов и статистики, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), a_panyukov@mail.ru.

Алина Таиховна Латипова, кандидат физико-математических наук, доцент, кафедра экономико-математических методов и статистики, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), latipova.alina@gmail.com.

STABILITY ANALYSIS OF EQUILIBRIUM POSITION OF VON NEUMANN'S MODEL UNDER INTERVAL UNCERTAINTY

Anatoly V. Panyukov. South Ural State University (Chelyabinsk, Russian Federation),

Alina T. Latipova, South Ural State University (Chelyabinsk, Russian Federation)

There is discussed the problem of stability analysis of equilibrium position under interval settings. In case of multiplicative uncertainty both primal and dual Frobenius vectors are obtained by exact von Neumann's model with matrices of interval centers. Interval of the Frobenius number in case of interval von Neumann's model are obtained by finding equilibrium for two exact von Neumann's models with exact matrices of interval upper and lower bounds. There are introduced definitions of weak and strong solutions, which are used to obtain robust estimates of equilibrium position of interval von Neumann's model.

Keywords: Product strategy, linear programming, von Neumann's model, interval analysis, game theory, bilinear systems, computer, software, interval uncertainty

References

1. Ashmanov S.A. Introduction to Mathematical Economics [Vvedenie v matematicheskuyu ekonomiku]. Nauka, Moscow, 1984. 296 p.
2. Al'sevich, V.V. Introduction to Mathematical Economics. Constructive Theory [Vvedenie v matematicheskuyu ekonomiku. Konstruktivnaya teoriya]. Moscow: Librocom, 2005. 256 p.
3. Latipova A.T. A Model for the Optimization of the Price Strategy in Budgeting [Model' optimizatsii tsenovroi strategii dlya zadach byudzhetrovaniya]. Trudy Rossiyskoy konferentsii "Diskretny analiz i issledovanie operatsiy" (Novosibirsk, 2004) [Discrete Analysis and Operations Research: Proc. of Russian Conf. (Novosibirsk, 2004)]. Novosibirsk, Institute of mathematics of Rus. Sci. Academy (Siberian Branch), 2004 . P. 206.
4. Latipova A.T. Price Diversification in Budgeting [Tsenovaya diversifikatsiya v byudzhetrovanii]. Trudy mezhdunarodnoy konferentsii Ekonomika i upravlenie: problemy i perspektivy [Economics and Management: Problems and Prospects: Int. Conf. Proc. (Sant Petersburg, 2005)]. Sant Petersburg: Izd-vo SPbGTU, 2005. P. 562 – 566.
5. Panyukov A.V., Latipova A.T. Optimization of Sales Budget [Optimizatsiya byudzheta prodazh]. // Vestnik Yuzhno-Ural'skogo Gosudarstvennogo Univ., Ser.: "Rynok: Teoriya i praktika" [Bulletin of South Ural State Univ., Ser. Markets: Theory and Practice]. Issue 4, Vol. 15(70), 2006. P. 116–120.
6. Panyukov A.V., Latipova A.T. Numerical Techniques for Finding Equilibrium in von Neumann's Model. // Computational Mathematics and Mathematical Physics, issue 14, Vol. 48, 2008. P. 1999–2006.
7. Panyukov A.V., Latipova A.T. Finding Equilibrium in von Neumann's Model. // Proceedings of 13th IFAC Symposium on Information Control Problems in

- Manufacturing. Vol. 13. Part 1. Elsevier, 2010. ISBN: 978-3-902661-43-2, available at: <http://www.ifac-papersonline.net/Detailed/40647.html>.
8. Panyukov A.V., Latipova A.T. Finding Equilibrium in von Neumann's Model in Case of Interval Uncertainty of Initial Data [Otsenka polozheniya ravnovesiya v modeli Neimana pri interval'noi neopredelennosti ishodnykh dannykh]. // Vestnik Ufimskogo Gosudarstvennogo Aviatsionnogo Tekhnicheskogo Univ., Ser. Upravlenie, vychislitel'naya tekhnika i informatika [Bulletin of Ufa State Aviation Univ., Ser. Control, Computer Engineering and Computer Science]. Issue 2(27), vol. 10, 2008. P. 150–153.
 9. Panyukov A.V. Latipova A.T. Stability Analysis of Equilibrium Position of Von Neumann's Model under Interval Uncertainty / A.V. Panyukov, A.T. Latipova // Proceedings of 14th IFAC Symposium on Information Control Problems in Manufacturing. Vol. 14. Part 1. Elsevier, 2010. ISBN: 978-3-902661-43-2, available at: www.ifac-papersonline.net/Detailed/53981.html.
 10. Jaulin, L., Kieffer M., Didrit O., Walter E. Applied Interval Analysis. – Springer-Verlag, 2001. 382 p.
 11. Fiedler M., Nedoma J., Ramik J., Rohn J., Zimmermann K. Linear Optimization Problems with Inexact Data. – Springer Science+Business Media, 2006. 288 p.

Поступила в редакцию 10 октября 2012 г.

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ КАТАЛИТИЧЕСКОЙ РЕАКЦИИ ($\text{CO} + \text{O}_2 \rightarrow \text{CO}_2$) С ПОМОЩЬЮ АСИНХРОННОГО КЛЕТОЧНОГО АВТОМАТА¹

А.Е. Шарифулина

Представлена параллельная реализация асинхронного клеточного автомата, моделирующего классическую реакцию гетерогенного катализа - окисление монооксида углерода на поверхности платины. В каталитических реакциях в неравновесных условиях могут возникать различные критические явления (автоколебания, хаос, гистерезис). Помимо фундаментального интереса изучение механизма протекания каталитических процессов на металлах платиновой группы имеет важное практическое применение, связанное с использованием в каталитических преобразователях для очистки выхлопных газов. Сложное поведение нелинейных каталитических систем наиболее эффективно может быть описано с помощью асинхронного клеточного автомата, который еще называют кинетическим методом Монте-Карло. КА-моделирование реакций гетерогенного катализа требует решения задач больших размеров, поэтому необходимо использовать эффективные алгоритмы распараллеливания. Распараллеливание асинхронных КА сопряжено с определёнными трудностями, которых можно избежать, преобразовав асинхронный КА в блочно-синхронный.

Блочно-синхронный режим работы уменьшает стохастичность моделируемого процесса, поэтому необходимо проверить эквивалентность эволюций асинхронного и блочно-синхронного КА. Для этого проводится статистический анализ основных характеристик моделирования реакции окисления: бифуркационных диаграмм, функций распределения концентраций реагентов, математических ожиданий и дисперсий концентраций, полученных с помощью асинхронного и блочно-синхронного КА. Вычисленные характеристики свидетельствуют о совпадении эволюций асинхронного и блочно-синхронного КА. Кроме того, выполнено сравнение эволюций асинхронного и блочно-синхронного КА для моделей «ZGB» и «наивная диффузия». На основе полученных результатов делается вывод о приемлемой точности аппроксимации асинхронного режима блочно-синхронным для класса задач «реакция - диффузия». В статье представлены результаты распараллеливания блочно-синхронного КА и приведены оценки эффективности параллельной реализации.

Ключевые слова: асинхронный клеточный автомат, блочно-синхронный режим функционирования, каталитическая реакция окисления, параллельная реализация

Введение

Сегодня компьютерное моделирование является основным средством изучения явлений в физике, химии, биологии, экономике, социологии и многих других науках. Традиционные методы моделирования, основанные на решении дифференциальных уравнений, недостаточно эффективны для описания распределенных динамических систем таких как популяции животных, человеческие сообщества, химические и физические процессы на микро-уровне. Эти явления существенно нелинейны и диссипативны, обладают способностью к самоорганизации и самовоспроизведению. Решение сложных систем дифференциальных уравнений с частными производными, использующихся для описания таких явлений, сопряжено со значительными математическими трудностями. Кроме того, пространственно распределенные, неоднородные системы,

¹Статья рекомендована к публикации программным комитетом международной научной конференции «Параллельные вычислительные технологии 2012»

далекие от состояния равновесия, невозможно описать в терминах дифференциальных уравнений. Сложное поведение таких нелинейных динамических систем в большинстве случаев можно определить простыми локальными правилами, описывающими явления на микро-уровне. Такие представления явлений легко выражаются в терминах клеточных автоматов.

Клеточный автомат (КА) представляет собой множество связанных по входам и выходам одинаковых конечных автоматов (клеток) с простыми детерминированными или вероятностными правилами переходов, вычисляющими новые состояния в зависимости от значений соседних клеток [1, 2]. КА наиболее эффективны для моделирования фазовых и бифуркационных переходов, нелинейных систем, где важно учитывать флуктуации и коллективное поведение определяется локальным поведением составляющих ее элементов [3].

Одним из примеров таких систем являются реакции гетерогенного катализа, использующиеся сегодня во многих областях человеческой деятельности. Классической реакцией гетерогенного катализа является окисление монооксида углерода (СО) на металлах платиновой группы (Pt, Pd). Изучение механизма протекания каталитических процессов на Pt и Pd важно с точки зрения фундаментальной науки, так как особенностью этих реакций является возникновение в неравновесных условиях таких критических явлений, как множественность стационарных состояний, кинетические фазовые переходы, автоколебания, хаос, гистерезис [4]. Кроме того, окислительно-восстановительные реакции на металлах платиновой группы имеют важное практическое применение, т.к. они являются основой каталитических преобразователей, использующихся для очистки выхлопных газов.

Динамика каталитических реакций определяется взаимодействием частиц на атомно-молекулярном уровне, влиянием процесса переноса вещества и температуры на скорость химического превращения. Асинхронный вероятностный клеточный автомат, известный еще как кинетический метод Монте – Карло, позволяет отобразить нелинейность каталитических процессов непосредственным образом, моделируя взаимодействие реальных атомов и молекул с помощью дискретных правил переходов.

Для изучения пространственно-временной динамики каталитических реакций необходимо моделировать взаимодействие большого количества молекул ($\approx 10^{20}$) в течение длительного периода времени ($\approx 10^{10}$ итераций). Следовательно, решение таких задач требует использования эффективных алгоритмов распараллеливания. Проблема эффективного распараллеливания асинхронных КА с вероятностными правилами переходов на сегодняшний день до конца не решена. В работах [5, 6] предложен метод достижения высокой эффективности распараллеливания путем аппроксимации асинхронного КА блочно-синхронным КА. Блочно-синхронный режим работы нарушает стохастичность моделируемого процесса, поэтому эквивалентность эволюций асинхронного и блочно-синхронного КА не может быть доказана в общем случае.

Целью работы является сравнение эволюций асинхронного и блочно-синхронного КА, моделирующего реакцию окисления СО на поверхности Pt₁₁₀, с помощью проведения вычислительных экспериментов; а также параллельная реализация блочно-синхронного КА и анализ эффективности распараллеливания.

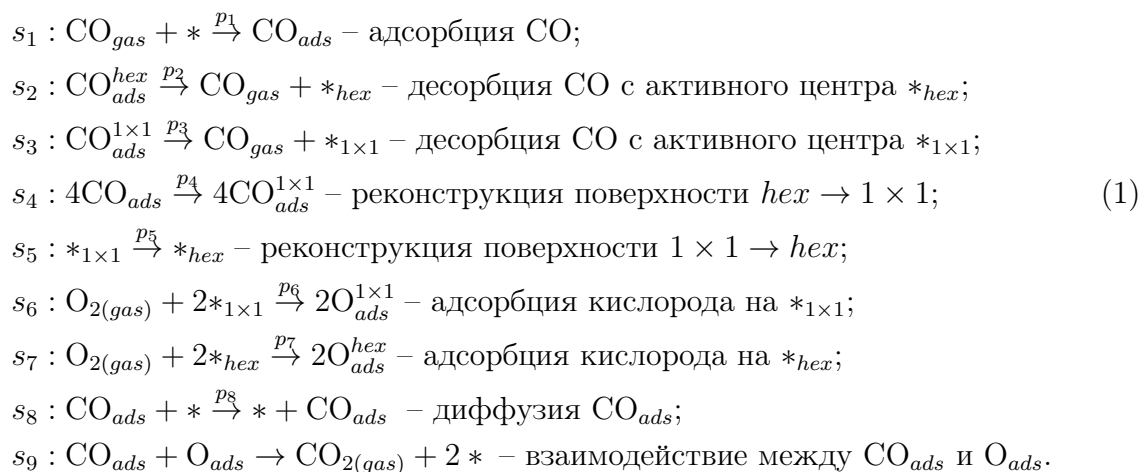
Во втором разделе статьи приведены уравнения реакции, описывающие каталитическое окисление СО на Pt₁₁₀, представлена клеточно-автоматная модель реакции

и результаты КА-моделирования. В третьем разделе описывается метод преобразования асинхронного режима работы КА в блочно-синхронный и проводится сравнительный анализ их эволюций. Четвертый раздел посвящен параллельной реализации блочно-синхронного КА, приводятся алгоритм и оценки эффективности распараллеливания.

1. Моделирование каталитической реакция окисления CO на поверхности платины

1.1. Описание механизма реакции

Экспериментальные и теоретические исследования, проведенные в [4, 7], показали, что в ходе реакции окисления происходит периодическая реконструкция структуры поверхности платины из гексагональной в кубическую ($hex \leftrightarrow 1 \times 1$). Реконструкция поверхности вызвана изменением каталитических свойств поверхности под воздействием адсорбированного CO. Детальный механизм реакции, учитывающий перестройку и изменение каталитических свойств поверхности, описан в [8]:



Символы $\langle\langle *_{hex} \rangle\rangle$ и $\langle\langle *_{1 \times 1} \rangle\rangle$ обозначают свободный активный центр поверхности с гексагональной и кубической структурой, соответственно. Символ $\langle\langle * \rangle\rangle$ – свободный активный центр с произвольной структурой hex или 1×1 . Активный центр, – это атом поверхности катализатора, на котором адсорбируются молекулы. На одном центре может адсорбироваться только одна молекула, и адсорбция на одном центре не влияет на возможность адсорбции на соседние.

Согласно алгоритму, предложенному в [8], активные центры поверхности катализатора выбираются случайным образом, для выбранного центра с вероятностью p_i выбирается одна из элементарных стадий s_i , $i = 1, \dots, 9$ приведенных выше (1). Вероятность выбора всех стадий, кроме s_9 , вычисляется по формуле:

$$p_i = \frac{k_i}{\sum_{l=1}^8 k_l}, \quad i = 1, \dots, 8, \tag{2}$$

где k_i – константа скорости i -й стадии, причем $k_8 = M_{diff} \cdot \sum_{l=1}^7 k_l$, где M_{diff} – параметр интенсивности диффузии. Стадия s_9 реализуется сразу же после выбора одной из

следующих стадий: s_1 – адсорбция CO, s_6 и s_7 – адсорбция O, s_8 – диффузия. Это связано с тем, что взаимодействие между молекулами CO_{ads} и O_{ads} , оказавшихся на соседних активных центрах, происходит мгновенно.

Известно, что в условиях, далеких от равновесных, реакция окисления CO на поверхности платиновых металлов может сопровождаться появлением таких критических явлений как автоколебания, подвижные волны, хаос. Колебательный характер реакции был обнаружен в вычислительных экспериментах [7, 8] с помощью метода Монте – Карло при следующем наборе констант скоростей элементарных стадий:

$$k_1 = 14,7, k_2 = 4, k_3 = 0,03, k_4 = 3, k_5 = 2, k_6 = 56, k_7 = 0,056, M_{diff} = 50. \quad (3)$$

Колебания концентраций реагентов обусловлены обратимым фазовым переходом поверхности платины из состояния hex в 1×1 , различная адсорбционная активность hex и 1×1 – поверхности является причиной периодической смены покрытий $\text{CO}_{ads} \leftrightarrow \text{O}_{ads}$.

1.2. Клеточно-автоматная модель реакции окисления CO

Механизм реакции, описанный в разделе 1.1, реализуется с помощью асинхронного КА. Поверхности катализатора соответствует клеточный массива, молекулы и атомы, участвующие в реакции, – это состояния клеток. Элементарные стадии s_i описываются вероятностными правилами переходов КА. Асинхронный режим работы соответствует случайному выбору активных центров поверхности катализатора.

Асинхронный клеточный автомат определяется тремя понятиями [9]: $\aleph_\alpha = \langle A, X, \Theta \rangle$,

A – это алфавит состояний клеток, X – множество имен клеток, Θ – локальный оператор, символ α обозначает асинхронный режим функционирования.

Алфавит состояний выбран в соответствии с реагентами, участвующими в реакции:

$$A = \{ *_{1 \times 1}, *_{hex}, \text{CO}_{ads}^{1 \times 1}, \text{CO}_{ads}^{hex}, \text{O}_{ads}^{1 \times 1}, \text{O}_{ads}^{hex} \}, \quad (4)$$

Символы $*_{1 \times 1}$ и $*_{hex}$ обозначают свободный активный центр поверхности с гексагональной и кубической структурой соответственно; $\text{CO}_{ads}^{1 \times 1}$, CO_{ads}^{hex} – это молекула монооксида углерода, адсорбированная на 1×1 и hex поверхности; $\text{O}_{ads}^{1 \times 1}$, O_{ads}^{hex} – молекула кислорода, адсорбированная на 1×1 и hex поверхности соответственно.

Множество имен $X = \{(i, j) : i = 1, \dots, M_i, j = 1, \dots, M_j\}$ определяется координатами клеток в дискретном пространстве, соответствующем поверхности катализатора. Клеткой называется пара $(u, (i, j))$, где $u \in A$ – это состояние клетки, $(i, j) \in X$ – имя клетки. На множестве имен вводятся именуемые функции $\varphi(i, j) : X \rightarrow X$, определяющие имена соседних клеток для клетки (i, j) . Конечное множество именуемых функций называется шаблоном соседства $T(i, j)$, ставящим в соответствие каждой клетке массива множество ее соседей [10].

В исследуемой модели используются следующие шаблоны соседства (рис. 1):

$$\begin{aligned}
 T_1(i, j) &= \{\varphi_0(i, j)\}, \\
 T_5(i, j) &= \{\varphi_0(i, j), \varphi_1(i, j), \dots, \varphi_4(i, j)\}, \\
 T_9(i, j) &= \{\varphi_0(i, j), \varphi_1(i, j), \dots, \varphi_8(i, j)\}, \\
 T_{13}(i, j) &= \{\varphi_0(i, j), \varphi_1(i, j), \dots, \varphi_{12}(i, j)\},
 \end{aligned} \tag{5}$$

где именуемые функции имеют следующий вид:

$$\begin{aligned}
 \{\varphi_0(i, j), \varphi_1(i, j), \dots, \varphi_{12}(i, j)\} &= \{(i, j), (i, j-1), (i+1, j), (i, j+1), (i-1, j), \\
 &(i-1, j-1), (i+1, j-1), (i+1, j+1), (i-1, j+1), \\
 &(i, j-2), (i+2, j), (i, j+2), (i-2, j)\}
 \end{aligned}$$

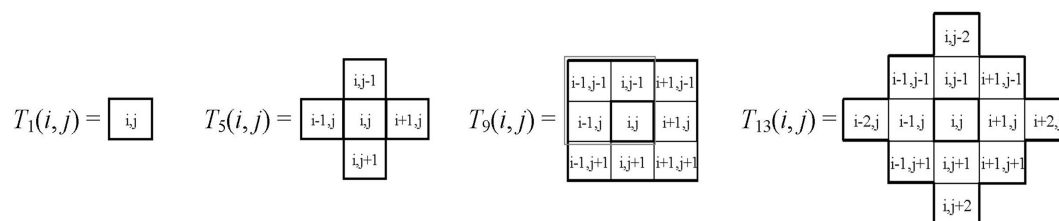


Рис. 1. Шаблоны соседства, используемые в КА-модели реакции окисления

Локальный оператор $\Theta(i, j)$ определяет правила изменения состояний клеток в соответствии с уравнениями реакции и является сложной композицией подстановок и их суперпозиций:

$$\Theta(i, j) = \{\theta_{(1,9)}, \theta_2, \theta_3, \theta_4, \theta_5, \theta_{(6,9)}, \theta_{(7,9)}, \theta_{(8,9)}\}. \tag{6}$$

где θ_l , $l \in \{2, 3, 4, 5\}$ – подстановки, соответствующие стадиям реакции окисления, а $\theta_{(l,9)} = \theta_9(\theta_l)$, $l \in \{1, 6, 7, 8\}$ – суперпозиции двух подстановок θ_l и θ_9 . Подстановки θ_l и суперпозиции $\theta_{(l,9)}$ выбираются с вероятностью p_l , вычисляемой по формуле (2).

В КА-модели реакции окисления каждой элементарной стадии s_l соответствует подстановка θ_l . Подстановки и суперпозиции изменяют состояние клеток в зависимости от состояний соседних клеток, принадлежащих соответствующему шаблону моделирования. Подстановки θ_l , $l \in 1, 2, 3, 5$ применяются к одной клетке, поэтому для них используется шаблон $T_1(i, j)$. Подстановка θ_4 применяется к блоку, состоящему из четырех соседних клеток, блок выбирается случайным образом по шаблону $T_9(i, j)$. Для применения остальных подстановок θ_l , $l \in 6, 7, 8, 9$ требуются состояния двух клеток: (i, j) и одной из четырех соседних клеток $\varphi_k(i, j)$, $k = 1, 2, 3, 4$. Соседняя для (i, j) клетка $\varphi_k(i, j)$ выбирается по шаблону $T_5(i, j)$ с вероятностью 0,25.

Суперпозиция подстановок $\theta_{(l,9)} = \theta_9(\theta_l)$, $l \in \{1, 6, 7, 8\}$ предполагает применение подстановки θ_9 к результату выполнения θ_l , $l = 1, 6, 7, 8$. Необходимость использования суперпозиции связана с особенностями реализации стадии s_9 . При применении суперпозиции $\theta_{(l,9)} = \theta_9(\theta_l)$, $l \in \{6, 7, 8\}$ сначала к клетке с именем (i, j) применяется подстановка θ_l , которая по шаблону $T_5(i, j)$ выбирает одну из четырех соседних клеток $\varphi_k(i, j)$, $k = 1, 2, 3, 4$. Сразу же после применения θ_l к выбранным клеткам (i, j) и $\varphi_k(i, j)$ применяется подстановка θ_9 , которая также по шаблону $T_5(i, j)$ выбирает одну

из четырех соседних клеток. В результате объединения шаблонов получаем, что при применении суперпозиции необходимо использовать шаблон $T_{13}(i, j) = \bigcup_{k=1}^4 T_5(\varphi_k(i, j))$. При применении суперпозиции $\theta_{(1,9)} = \theta_9(\theta_1)$ достаточно использования шаблона $T_5(i, j)$, т.к. подстановка θ_1 изменяет состояние одной клетки, и затем к этой же клетке применяется подстановка θ_9 , которая по шаблону $T_5(i, j)$ выбирает одну из четырех соседних клеток $\varphi_k(i, j)$.

Асинхронный режим функционирования КА предполагает, что локальный оператор $\Theta(i, j)$ применяется по очереди к случайно выбранным клеткам массива, сразу же изменяя их состояния. Далее асинхронный КА с определенными выше алфавитом A (4), множеством имен X и локальным оператором $\Theta(i, j)$ (6) будем обозначать символом « KA_α ».

Реагенты, адсорбированные на поверхности катализатора, постоянно диффундируют, тогда как остальные процессы (адсорбция, десорбция, реконструкция поверхности) происходят намного реже, поэтому константа скорости диффузии s_8 значительно выше, чем константы скорости остальных стадий. В КА-модели это реализуется с помощью увеличения вероятности выбора подстановки, моделирующей диффузию, в M_{diff} раз по сравнению с суммой вероятностей остальных подстановок. В соответствии с [7, 8] значение M_{diff} выбирается в диапазоне $50 \div 100$.

Время в КА-модели дискретно, весь процесс КА-моделирования разбивается на итерации. При моделировании реакции окисления итерация принимается равной $M_i \cdot M_j \cdot M_{diff}$ применениям локального оператора $\Theta(i, j)$ к случайно выбранным клеткам массива. За итерацию клеточный массив Ω переходит из одного глобального состояния $\Omega(t)$ в другое $\Omega(t + 1)$, где t – номер итерации. Последовательность $\sum(\Omega) = \Omega(0), \dots, \Omega(t), \Omega(t + 1), \dots, \Omega(t_{fin})$, полученная в результате итеративного функционирования КА, называется эволюцией, $\Omega(0)$ – исходное состояние клеточного массива, $\Omega(t)$ – состояние массива на t -й итерации, t_{fin} – число итераций [10].

1.3. Результаты КА-моделирования

Компьютерное моделирование реакции окисления с помощью последовательной реализации KA_α выполнялось на клеточном массиве размером $M_i \times M_j = 200 \times 200$ клеток с периодическими граничными условиями. Вероятности применения подстановок θ_l вычисляются по формуле (2) для констант скорости k_l (3): $p_1 = 3,61 \cdot 10^{-3}$, $p_2 = 9,83 \cdot 10^{-4}$, $p_3 = 7,37 \cdot 10^{-6}$, $p_4 = 7,37 \cdot 10^{-4}$, $p_5 = 4,92 \cdot 10^{-4}$, $p_6 = 1,38 \cdot 10^{-2}$, $p_7 = 1,38 \cdot 10^{-5}$, $p_8 = 9,80 \cdot 10^{-1}$ и $M_{diff} = 50$. В исходном состоянии все клетки массива находятся в состоянии $*_{hex}$.

В качестве характеристик КА-моделирования реакции окисления выбраны следующие величины:

- концентрации реагентов, адсорбированных на поверхности катализатора: $n(O_{ads}), n(CO_{ads})$;
- скорость образования CO_2 : $v(CO_2)$;
- доли поверхности с кубической и гексагональной структурой: $f(1 \times 1), f(hex)$.

Концентрация адсорбированных на поверхности катализатора веществ вычисляется после каждой итерации как отношение количества клеток (N), находящихся в со-

стоянии, соответствующем данному реагенту, к размеру клеточного массива ($M_i \cdot M_j$):

$$n(\text{CO}_{ads}) = \frac{N(\text{CO}_{ads}^{1 \times 1}) + N(\text{CO}_{ads}^{hex})}{M_i \cdot M_j}, \quad (7)$$

$$n(\text{O}_{ads}) = \frac{N(\text{O}_{ads}^{1 \times 1}) + N(\text{O}_{ads}^{hex})}{M_i \cdot M_j}. \quad (8)$$

Скорость образования CO_2 вычисляется как отношение числа применений подстановки θ_9 за итерацию к размеру клеточного массива:

$$v(\text{CO}_2) = \frac{N(\text{CO}_{ads} + \text{O}_{ads})}{M_i \cdot M_j}. \quad (9)$$

Доля 1×1 и hex поверхности вычисляется после каждой итерации по следующим формулам:

$$f(1 \times 1) = \frac{N(*_{1 \times 1}) + N(\text{CO}_{ads}^{1 \times 1}) + N(\text{O}_{ads}^{1 \times 1})}{M_i \cdot M_j}, \quad (10)$$

$$f(hex) = \frac{N(*_{hex}) + N(\text{CO}_{ads}^{hex}) + N(\text{O}_{ads}^{hex})}{M_i \cdot M_j}. \quad (11)$$

В результате КА-моделирования реакции окисления получены колебания концентраций реагентов $n(\text{O}_{ads})$, $n(\text{CO}_{ads})$, скорости образования CO_2 – $v(\text{CO}_2)$ и доли поверхности с кубической и гексагональной структурой $f(1 \times 1)$, $f(hex)$ (рис. 2).

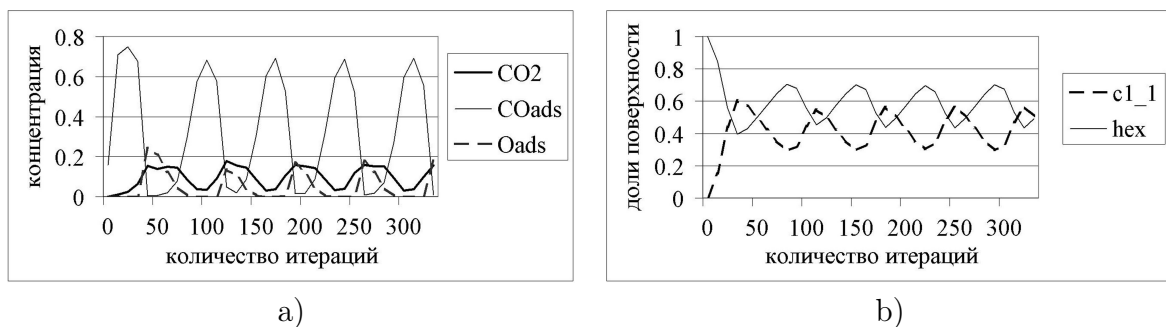


Рис. 2. Характер колебаний в реакции окисления CO на Pt : а) концентрация CO_{ads} , O_{ads} и скорость образования CO_2 б) доли 1×1 и hex поверхности

Колебания концентраций реагентов в реакции окисления CO наблюдаются только при определенных значениях констант скорости реакции. С помощью исследования эволюции KA_α построена бифуркационная диаграмма в пространстве констант скорости адсорбции кислорода $k_6 \in [0; 10^5]$ и монооксида углерода $k_1 \in [0; 200]$ (рис. 3). Точками обозначены значения, полученные в результате КА-моделирования, пунктирная линия построена с помощью линейной интерполяции.

В зависимости от значений констант k_6 и k_1 обнаружено наличие трех различных режимов протекания реакции:

- два равновесных состояния – поверхность катализатора покрывается $\text{CO}_{1 \times 1}$ (ниже границы 1) либо O_{hex} (выше границы 3);

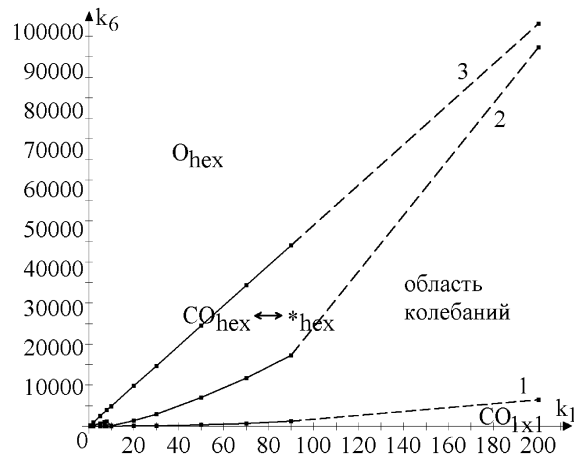


Рис. 3. Бифуркационная диаграмма реакции окисления CO на Pt

- режим колебаний концентраций реагентов и скорости образования CO_2 (область колебаний между кривыми 1 и 2);
- периодическая смена $\text{CO}_{ads}^{hex} \leftrightarrow *_{hex}$, при которой доля 1×1 поверхности мала, и адсорбция кислорода практически не происходит (область между кривыми 2 и 3).

2. Блочно-синхронный режим функционирования КА

2.1. Преобразование асинхронного КА в блочно-синхронный

Изучение пространственно-временной динамики реакции окисления CO требуют проведения вычислительных экспериментов с использованием клеточных массивов размером $M_i \times M_j = 8000 \times 8000$ клеток в течение 10^6 итераций. Решение таких задач на однопроцессорной машине займет несколько недель, поэтому необходимо использовать эффективные алгоритмы распараллеливания. Эффективное распараллеливание асинхронных КА является трудновыполнимой задачей, так как при асинхронном режиме межпроцессорный обмен данными приходится выполнять после изменения каждой граничной клетки. Поэтому асинхронный режим работы изменяется на блочно-синхронный, который вводит частичную синхронизацию режима функционирования, не нарушая при этом условия корректности, т.е. состояние клетки одновременно не могут изменить разные подстановки. Преобразование асинхронного КА $KA_\alpha = \langle A, X, \Theta \rangle$ в блочно-синхронный $KA_\beta = \langle A, X, \Theta \rangle$ выполняется следующим образом [5].

1. На множестве имен X определяется шаблон, называемый блоком $B(i, j)$. Для выполнения условия корректности блок должен включать в себя все шаблоны соседства (5):

$$T_1(i, j) \subset T_5(i, j) \subset T_9(i, j) \subset T_{13}(i, j) \subseteq B(i, j) \Rightarrow B(i, j) = T_{13}(i, j), \quad (12)$$

$$|B(i, j)| = 13.$$

Блок $B(i, j)$ определяет на множестве имен разбиение $\Pi = \{X_1, X_2, \dots, X_{13}\}$ такое, что для всех X_k , $k = 1, \dots, 13$, $|B(i, j)|$ выполняются соотношения (13):

$$|X_k| = \frac{|X|}{13}, \quad \bigcup_{(i,j) \in X_k} B(i, j) = X, \quad B(i, j) \cap B(g, h) = \emptyset, \quad (13)$$

$$(i, j), (g, h) \in X_k.$$

- Итерация разбивается на $|B(i, j)| = 13$ этапов. На каждом этапе локальный оператор $\Theta(i, j)$ применяется ко всем клеткам выбранного случайным образом множества X_k . Порядок выбора клеток внутри X_k не важен, т.к. условие (12) гарантирует, что локальный оператор применяется к разным блокам и шаблоны применения подстановок не пересекаются.

2.2. Сравнение асинхронного и блочно-синхронного режимов работы КА

При блочно-синхронном режиме работы на каждом этапе локальный оператор применяется только к клеткам из множества X_k , что существенно уменьшает стохастичность выбора клеток. Существуют классы задач, для которых эволюции асинхронного и блочно-синхронного КА не совпадают. Например, для КА со взвешенными шаблонами, вычисляющих новые состояния в зависимости от взвешенной суммы состояний соседних клеток, при асинхронном и блочно-синхронном режимах для различных последовательностей выбора клеток формируются совершенно разные устойчивые структуры (рис. 4).

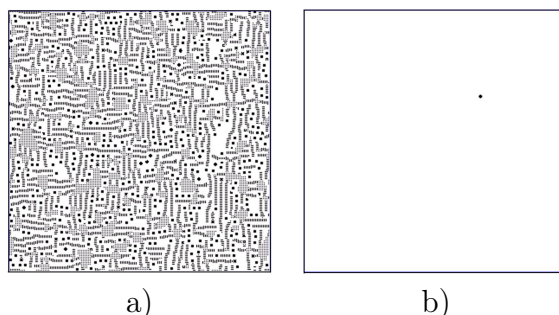


Рис. 4. Устойчивые состояния КА со взвешенными шаблонами при одинаковых начальных значениях и различных режимах функционирования: а) асинхронный режим; б) блочно-синхронный режим

В общем случае невозможно доказать эквивалентность эволюций асинхронного и блочно-синхронного КА, каждая задача требует проведения отдельного анализа результатов КА-моделирования. Вычислительные эксперименты для различных моделей класса «реакция – диффузия» подтверждают возможность использования блочно-синхронного преобразования для этого класса задач.

Например, значения скорости образования CO_2 , концентраций CO_{ads} и O_{ads} , полученные при реализации модели Ziff – Gulari – Varshad (ZGV) [11] с помощью асинхронного и блочно-синхронного КА, отличаются незначительно. На рис. 5а представлены изменения скорости образования CO_2 – $v(\text{CO}_2)$ при вероятности адсорбции CO равной 0,5. Уровень клеточно-автоматного шума для значений, получен-

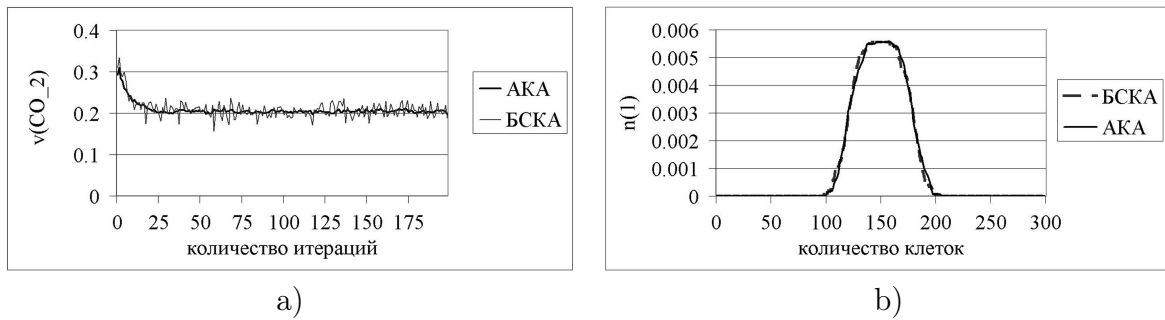


Рис. 5. Сравнение асинхронного и блочно-синхронного режимов при моделировании задач класса «реакция – диффузия»: а) скорость образования CO_2 в КА-модели ZGB; б) концентрация единиц в КА-модели диффузии

ных с помощью блочно-синхронного КА при $B(i, j) = T_9(i, j)$, находится в пределах допустимой величины и при увеличении размера блока уменьшается. Еще одним примером КА-модели, допускающей преобразование асинхронного режима в блочно-синхронный, является диффузия. При моделировании диффузии с помощью асинхронного и блочно-синхронного КА осредненные значения концентрации единиц $n(1)$ совпадают (рис. 5б).

Для проверки возможности применения блочно-синхронного преобразования в случае КА-моделирования реакции окисления CO на Pt (1), сравнивались результаты вычислительных экспериментов, выполненных при $M_i \times M_j = 200 \times 200$ клеток, $t_{fin} = 10^5$ итераций и для значений констант скорости (3). В качестве параметров сравнения выбраны следующие характеристики:

- распределение вероятностей $n(\text{O}_{ads}), n(\text{CO}_{ads}), v(\text{CO}_2), f(1 \times 1), f(hex)$ и периодов колебаний T ;
- математическое ожидание и дисперсия $n(\text{O}_{ads}), n(\text{CO}_{ads}), v(\text{CO}_2), f(1 \times 1), f(hex), T$ и доверительные интервалы для математического ожидания и дисперсии;
- бифуркационная диаграмма реакции окисления.

Анализ результатов моделирования показал, что значения распределений вероятностей $n(\text{O}_{ads}), n(\text{CO}_{ads}), v(\text{CO}_2), f(1 \times 1), f(hex), T$ полученные с помощью KA_α и KA_β , очень близки. На рис. 6 представлено распределение вероятностей $v(\text{CO}_2)$ и периодов колебаний. Среднеквадратичные разности распределений вероятностей этих характеристик для асинхронного и блочно-синхронного КА не превосходят 10^{-4} . Например, среднеквадратичные разности распределений вероятностей $v(\text{CO}_2)$ и периодов T составляют $E(v(\text{CO}_2)) = 5,9 \cdot 10^{-5}$ и $E(T) = 7,1 \cdot 10^{-5}$.

Математическое ожидание ($M\xi$), дисперсия ($D\xi$) и доверительные интервалы для математического ожидания $I_{M\xi}$ и дисперсии $I_{D\xi}$, вычисленные для скорости образования CO_2 и периодов колебаний, приведены в таблице 1. Доверительные интервалы рассчитывались с доверительной вероятностью $\gamma = 0,95$. Из таблицы видно, что значения числовых характеристик для концентраций адсорбированных реагентов, скорости образования CO_2 и доли 1×1 и hex поверхности, полученных в результате моделирования с помощью асинхронного и блочно-синхронного КА, отличаются незначительно.

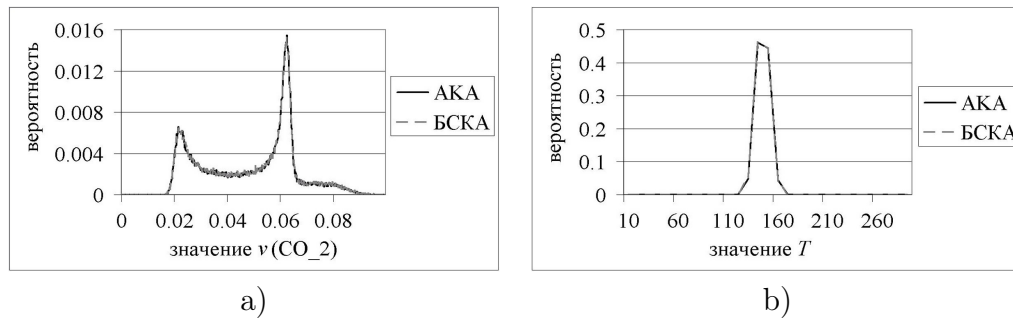


Рис. 6. Сравнение значений распределения вероятностей, вычисленных с помощью KA_α и KA_β : а) скорость образования CO_2 ; б) периоды колебаний

Бифуркационная диаграмма, построенная в результате анализа эволюции блочно-синхронного КА в пространстве констант скорости адсорбции кислорода k_6 и монооксида углерода k_1 , совпадает с диаграммой, вычисленной с помощью асинхронного КА (рис. 3). Это подтверждает, что KA_α и KA_β демонстрируют одинаковый характер поведения реакции окисления.

Таблица 1

Статистические характеристики $v(\text{CO}_2)$ и T , полученные с помощью KA_α и KA_β

Характеристики	$M\xi$	$D\xi$	$I_{M\xi}$	$I_{D\xi}$
$v(\text{CO}_2)_\alpha$	0,048841	0,000317	(0,048805; 0,048876)	(0,000316; 0,000318)
$v(\text{CO}_2)_\beta$	0,048826	0,000316	(0,048791; 0,048862)	(0,000315; 0,000317)
T_α	14,490580	0,461572	(14,485427; 14,495732)	(0,456621; 0,466523)
T_β	14,469609	0,538590	(14,464047; 14,475171)	(0,532817; 0,544363)

Полученные результаты свидетельствуют о совпадении эволюций асинхронного и блочно-синхронного КА.

3. Результаты распараллеливания блочно-синхронного КА

Распараллеливание блочно-синхронного КА заключается в разделении клеточного массива $\Omega(A, X)$ на домены $|Dom| = \frac{|X|}{n}$, которые распределяются между n процессами. Каждый процесс вычисляет новые значения клеток своего домена и пересылает граничные значения соседним процессам. Обмен граничными значениями выполняется в конце каждого этапа, т.к. условие (12) гарантирует, что при применении локального оператора $\Theta(i, j)$ к клеткам выбранного множества X_k не понадобятся значения состояний клеток, вычисленных на текущем этапе [5]. Объем пересылаемых данных составляет $2 \cdot P_{Dom}$ байт, где P_{Dom} – периметр домена.

Распараллеливание блочно-синхронного КА выполнялось на суперкомпьютере «МВС-100К» (МСЦ РАН) с использованием библиотеки MPI. При распараллеливании используется гибридная модель MPI+OpenMP. На каждом вычислительном узле запускается 4 MPI-процесса, в каждом из которых запускается по 2 потока. Для эффективного использования ресурсов вычислительного модуля потока явным образом

назначаются на ядра, объединенные общей кэш-памятью. Результаты распараллеливания клеточного массива размером $|X| = 8000 \times 8000$ клеток представлены в таблице 2. В качестве характеристик распараллеливания рассматриваются: T_n – время вычислений с использованием n процессов, $S(n) = \frac{T_1}{T_n}$ – ускорение и $Q(n) = \frac{T_1}{T_n \cdot n}$ – эффективность распараллеливания.

Таблица 2

Характеристики распараллеливания блочно-синхронного КА

n	1	4	16	32	64	128	256
$T_{n, s}$	867,15372	218,58	55,97	29,40	16,47	9,52	4,77
$S(n)$	1	3,97	15,49	29,49	52,64	91,12	181,92
$Q(n)$	1	0,99	0,97	0,92	0,82	0,72	0,71
$ Dom $	$2,56 \cdot 10^8$	$6,4 \cdot 10^7$	$1,6 \cdot 10^7$	$8 \cdot 10^6$	$4 \cdot 10^6$	$2 \cdot 10^6$	10^6

Данные приведенные в таблице показывают, что при использовании до 128 процессов эффективность $Q(n)$ выше 80%, при дальнейшем увеличении числа процессов эффективность распараллеливания падает. Это связано с возрастанием накладных расходов на обеспечение обмена данными между вычислительными узлами и недостаточной загрузкой ядер. Для достижения высокой эффективности размер домена $|Dom|$ должен превышать $2,04 \cdot 10^6$ клеток.

4. Заключение

В работе реализован асинхронный клеточный автомат, моделирующий реакцию окисления СО на поверхности Pt. Построенная КА-модель демонстрирует колебания скорости реакции, концентраций веществ, адсорбирующихся на поверхности катализатора, и долей 1×1 и *hex* поверхности. Колебания сопровождаются различными волновыми процессами на моделируемой поверхности.

Для достижения высокой эффективности распараллеливания выполнено преобразование асинхронного КА в блочно-синхронный. Для проверки эквивалентности эволюций KA_α и KA_β сравнивались значения распределения вероятностей концентраций и периодов колебаний, математическое ожидание и дисперсия распределения вероятностей, доверительные интервалы и бифуркационные диаграммы реакции окисления. Статистические характеристики, полученные в результате моделирования с помощью асинхронного и блочно-синхронного КА, отличаются незначительно, что свидетельствуют о применимости преобразования KA_α в KA_β для реакции окисления СО.

Выполнено распараллеливание блочно-синхронного КА, моделирующего реакцию окисления СО на Pt. Анализ характеристик распараллеливания при $|X| = 8000 \times 8000$ клеток показал, что для достижения высокой эффективности размер домена $|Dom|$ должен превышать $2,04 \cdot 10^6$ клеток.

Работа проводилась при финансовой поддержке 1) Междисциплинарного интеграционного проекта СО РАН № 47 (2012 год); 2) Гранта РФФИ 11-01-00567-а; 3) Программы Президиума РАН № 14-6 (2012 год).

Литература

1. Wolfram, S. New Kind of Science / S. Wolfram. - Wolfram Media, Inc. - 2002 / <http://www.wolframscience.com/>.
2. Тоффоли, Т. Машины клеточных автоматов / Т. Тоффоли, Н. Марголюс – М.: Мир, 1991. – 269 с.
3. Ванаг, В.К. Исследование пространственно распределенных динамических систем методами вероятностного клеточного автомата / В.К. Ванаг // Успехи физических наук. Обзоры актуальных проблем. – 1999. – Т. 169, № 5. – С. 481–505.
4. Imbihl, R. Oscillatory Kinetics in Heterogeneous Catalysis / Ronald Imbihl and Gerhard Ertl // Chemical Reviews. – 1995. – Vol. 95, № 3. – P. 697–733.
5. Bandman O. Parallel Simulation of Asynchronous Cellular Automata Evolution. / Bandman O. // Proceedings of ACRI-2006. In: Lecture Notes in Computer Science, Berlin: Springer. – 2007. –Vol. 4173. – P. 41–47.
6. Nedeя, S.V. Methods for parallel simulations of surface reactions / S.V. Nedeя, J.J. Lukkien, A.P.J. Jansen, P.A.J. Hilbers // arXiv:physics/0209017. – 2002. – Vol. 1.
7. Elokhin, V.I. Application of statistical lattice models to the analysis of oscillatory and autowave processes in the reaction of carbon monoxide oxidation over platinum and palladium surfaces / V.I. Elokhin, E.I. Latkin, A.V. Matveev, and V.V. Gorodetskii // Kinetics and Catalysis. – 2003. – Vol. 44, № 5. – P. 692–700.
8. Latkin, E.I. Monte Carlo model of oscillatory CO oxidation having regard to the change of catalytic properties due to the adsorbate-induced Pt(100) structural transformation / E.I. Latkin, V.I. Elokhin, V.V. Gorodetskii // Journal of Molecular Catalysis A: Chemical. – 2001. – Vol. 166. – P. 23–30.
9. Bandman O. Synchronous versus asynchronous cellular automata for simulating nanokinetics / Bandman O. // Bulletin of the Novosibirsk Computing Center, Comp. Science, Novosibirsk: NCC Publisher. – 2006. № 25. – P. 1–12.
10. Бандман, О.Л. Клеточно-автоматные модели пространственной динамики. / О.Л. Бандман // Системная информатика – Методы и модели современного программирования. – 2006. № 10. – С. 59–113.
11. Ziff, R.M. Kinetic phase transitions in irreversible surface-reaction model / R.M. Ziff, E. Gulari, Y. Bershad // Phys. Rev. Lett. – 1986. – Vol. 56, № 24. – P. 2553.

Шарифулина Анастасия Евгеньевна, аспирант, лаборатория «Параллельных алгоритмов и структур», Институт вычислительной математики и математической геофизики Сибирского отделения РАН, sharifulina@ssd.sccc.ru.

PARALLEL IMPLEMENTATION OF CATALYTIC REACTION ($\text{CO} + \text{O}_2 \rightarrow \text{CO}_2$) BY ASYNCHRONOUS CELLULAR AUTOMATA

A.E. Sharifulina, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russian Federation)

Parallel implementation of cellular automata (CA) model of heterogeneous catalysis classical reaction, namely carbon monoxide oxidation (CO) reaction over platinum surface is presented. Catalytic reactions when being far from equilibrium may be accompanied by such critical phenomena as oscillations, kinetic phase transitions and chaos. Besides fundamental interest studying of basic kinetic laws of physicochemical processes on metals of platinum group has the important practical application. These reactions are used for environmental cleaning of exhaust from CO . The asynchronous cellular automata being sometimes referred to as Monte-Carlo method are the most suitable for describing of complex behavior of nonlinear catalytic systems. CA simulation of heterogeneous catalysis reactions requires to solve problems of very large size, therefore it is necessary to use efficient algorithms of parallelization. For the asynchronous CA of efficient parallel implementation is stiff problem. Therefore to solve this problem the asynchronous CA is transformed in block-synchronous CA.

The block-synchronous mode of CA operation decreases the stochasticity of the process. Therefore it is necessary to check, whether block-synchronous CA conserves asynchronous CA evolution. This is done by comparative analysis of simulation characteristics such as probability distribution of reagents concentrations mathematical expectation and dispersion of concentrations and bifurcation diagrams of oxidation reaction obtained by CA simulation with asynchronous and block-synchronous operation modes. Obtained characteristics coincidence of asynchronous and block-synchronous CA evolutions is shown. In addition, comparison asynchronous and block-synchronous CA evolutions for models “ZGB” and “naive diffusion” are performed. Consequently, conclusion about acceptable accuracy of approximation of asynchronous mode to block-synchronous one for the class “reaction – diffusion” models is made. Parallel implementation of block-synchronous CA algorithm results and estimations of its efficiency are presented.

Keywords: asynchronous cellular automata, block-synchronous mode, catalytic oxidation reaction, parallel implementation.

References

1. Wolfram S. New Kind of Science. Wolfram Media, Inc, 2002. URL: <http://www.wolframscience.com/>
2. Toffoli T. and Margolus N. Cellular Automata Machines: A New Environment for Modeling. Cambridge, MA: MIT Press, 1987, Weisstein, P. 200.
3. Vanag V.K. Issledovanie prostranstvenno raspredelennykh dinamicheskikh sistem metodami veroyatnostnogo kelotochnogo avtomata [Study of spatially extended dynamical systems using probabilistic cellular automata]. Phys.-Usp., 1999, No. 42. P. 413.
4. Imbihl R. and Ertl G. Oscillatory Kinetics in Heterogeneous Catalysis. Chemical Reviews, 1995, Vol. 95, No. 3, P. 697–733.
5. Bandman O. Parallel Simulation of Asynchronous Cellular Automata Evolution. Proceedings of ACRI-2006. In: Lecture Notes in Computer Science, Berlin: Springer, 2007, Vol. 4173, P. 41–47.
6. Nedeя S.V., Lukkien J.J., Jansen A.P.J., Hilbers P.A.J. Methods for parallel simulations of surface reactions /arXiv:physics/0209017, Vol. 1, 2002.
7. Elokhin V.I., Latkin E.I., Matveev A.V., and Gorodetskii V.V. Application of statistical lattice models to the analysis of oscillatory and autowave processes in the reaction of carbon monoxide oxidation over platinum and palladium surfaces. Kinetics and Catalysis, 2003, Vol. 44, No. 5, P. 692–700.

8. Latkin E.I., Elokhin V.I., Gorodetskii V.V. Monte Carlo model of oscillatory CO oxidation having regard to the change of catalytic properties due to the adsorbate-induced Pt(100) structural transformation. Journal of Molecular Catalysis A: Chemical. 2001, Vol. 166, P. 23–30.
9. Bandman O. Synchronous versus asynchronous cellular automata for simulating nanokinetics. Bull. Nov. Comp. Center, Comp. Science, Novosibirsk: NCC Publisher, 2006, No. 25, P. 1–12.
10. Bandman O.L. Kletochno-avtomatnye modeli prostranstvennoy dinamiki. [Cellular-Automata models of spatial Dynamics]. Systemnaya informatika - Metody i modeli sovremennogo programirovaniya, 2006, No. 10, P. 59–113.
11. Ziff R.M., Gulari E., Bershad Y. Kinetic phase transitions in irreversible surface-reaction model. Phys. Rev. Lett, 1986, Vol. 56, No. 24, P. 2553.

Поступила в редакцию 5 ноября 2012 г.

Краткие сообщения

УДК 004.65, 004.272, 519.174.1

ПОДХОД К РАЗБИЕНИЮ СВЕРХБОЛЬШИХ ГРАФОВ С ПОМОЩЬЮ ПАРАЛЛЕЛЬНЫХ СУБД

К.С. Пан

Разбиение графов на подграфы представляет собой интересную задачу интеллектуального анализа графов, которая находит свое применение в ряде теоретических и практических задач (раскраска графа, проектирование БИС и ПЛИС, конечно-элементное моделирование и др.). Существующие последовательные и параллельные алгоритмы предполагают возможность размещения графов и промежуточных данных обработки в оперативной памяти и неприменимы для случая сверхбольших графов. Представлен подход к обработке сверхбольших графов на основе использования параллельной реляционной СУБД PostgreSQL, разработанной на базе свободной СУБД PostgreSQL.

Ключевые слова: интеллектуальный анализ, разбиение графов, параллельные СУБД.

Введение

В настоящее время одной из актуальных областей приложения технологий Data Mining [1–3] являются задачи интеллектуального анализа сверхбольших графов (имеющих сотни тысяч вершин и/или ребер), возникающие при моделировании сложных структур: химических соединений, белковых структур, биологических и социальных сетей, Web, потоков работ, XML документов и др.

Разбиение графов (graph partitioning) является одной из интересных задач интеллектуального анализа графов и определяется следующим образом. Пусть имеется граф $G = (N, E)$, где N — множество взвешенных вершин, E — множество взвешенных ребер, и целое положительное число p . Требуется найти непересекающиеся подграфы исходного графа N_1, N_2, \dots, N_p , такие, что

- $W(i) = W/p$, где $W(i)$ и W — суммы весов вершин N_i и N соответственно;
- сумма весов ребер, соединяющих подграфы, минимальна.

Далее в разделе 1 сформулирован многоуровневый подход к разбиению графов, в разделе 2 предложен способ реализации этого подхода с помощью параллельной СУБД.

1. Многоуровневое разбиение графов

Эффективное решение задачи разбиения графов имеет большое значение в ряде теоретических и практических задач. В качестве примера теоретических задач можно привести задачи раскраски графа, определения числа и состава компонент связности графа и представления графа в виде ярусно-параллельной формы. Примерами практических задач, в которых необходимо разбиение графа, являются проектирование сложных электронных схем, БИС (больших интегральных схем) и ПЛИС (программируемых логических интегральных схем), проектирование топологии локальной сети, конечно-элементное моделирование и др.

Для разбиения сверхбольших графов применяют так называемое *многоуровневое разбиение*, поскольку традиционные алгоритмы требуют недопустимо много времени и/или оперативной памяти. *Многоуровневое разбиение (multilevel partitioning)* предполагает три стадии этого процесса (рис. 1).

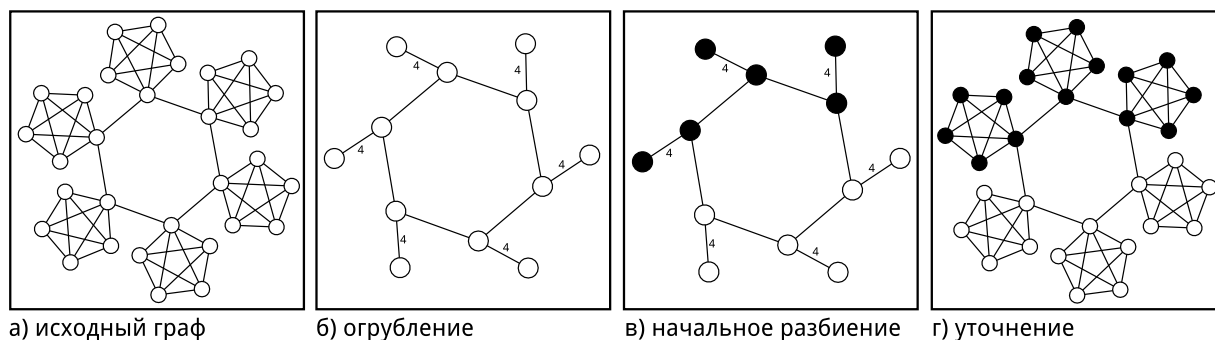


Рис. 1. Многоуровневое разбиение

На первой стадии выполняется «огрубление» (coarsening) графа, т.е. уменьшение количества вершин и ребер в нем путем «склеивания» вместе сильно связанных вершин и удаления возникающих при этом петель. На второй стадии выполняется начальное разбиение, когда огрубленный граф подвергают анализу обычными алгоритмами и получают грубое разбиение. Затем, на третьей стадии, выполняется «уточнение» (uncoarsening) грубого разбиения с помощью какой-либо эвристики.

Частным случаем задачи разбиения является бисекция графа ($p = 2$ — разбиение исходного графа на два подграфа). Разбиение на большее количество подграфов выполняется рекурсивно, т.е. каждый из найденных подграфов подвергается бисекции.

Процесс бисекции (рис. 2) начинается с присваивания вершинам произвольного «цвета», которым обозначается принадлежность вершины к тому или иному подграфу. После этого для каждой вершины производится подсчет показателя *выгоды (gain)* [4] для каждой вершины: насколько выгодно изменить цвет этой вершины на противоположный.

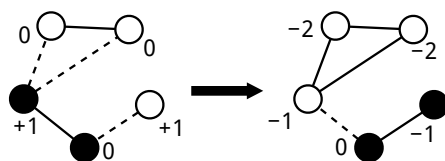


Рис. 2. Эвристика для уточнения

Выгода равна сумме весов всех ребер, соединяющих данную вершину с другими вершинами. Если ребро соединяет вершины разного цвета, то его вес увеличивает сумму, иначе — уменьшает. Затем вершины, цвет которых оказалось выгодно изменить, подвергаются «перекрашиванию». Процесс повторяется, пока такие вершины существуют.

2. Применение параллельной СУБД для разбиения графов

Существующие в настоящее время системы, использующие последовательные и параллельные алгоритмы разбиения графов [5], предполагают размещение графов и

промежуточных данных процесса обработки в оперативной памяти. Однако объем оперативной памяти современных компьютеров, хотя и велик, но не беспределен. При обработке сверхбольших графов (имеющих сотни тысяч вершин и/или ребер) упомянутые системы будут использовать жесткий диск для подкачки данных, что существенно замедлит решение задачи.

Реляционные системы управления базами данных (СУБД) на уровне программной архитектуры предполагают возможность эффективной реализации обработки данных, размер которых превышает объем доступной оперативной памяти [6, 7].

В соответствии с этим, реляционные СУБД могут быть применены для обработки графов: реализация алгоритмов выполняется в виде набора SQL запросов, а базовыми структурами хранения и обработки данных выступают реляционные таблицы и индексы. Для представления графа в виде реляционной таблицы используется список ребер.

Данный подход уже нашел свое применение в задачах поиска часто встречающихся подграфов [8] и поиска клики [9] в графах. Тем не менее, в случае обработки графов сверхбольших размеров, даже реляционная СУБД не может обеспечить надлежащую производительность.

Одним из решений данной проблемы является использование параллельной реляционной СУБД [10]. При обработке запросов к реляционной базе данных параллельная СУБД использует концепцию фрагментного параллелизма [11, 12].

Реляционные таблицы подвергаются горизонтальной фрагментации по дискам кластерной системы. На каждом узле кластера устанавливается модифицированное ядро последовательной СУБД, которое воспринимает базу данных как набор «своих» фрагментов таблиц. Запрос параллельно выполняется ядрами СУБД на всех узлах кластера, затем частичные результаты сливаются в результирующую таблицу. Модификация ядра последовательной СУБД заключается во внедрении в это ядро ряда механизмов параллельной обработки: автоматизированная фрагментация таблиц, распараллеливание запроса, координация действий ядер и обмена данными при выполнении запросов и др. Описанная идея реализована нами в параллельной СУБД PargreSQL [13], которая базируется на СУБД PostgreSQL, свободно распространяемой на уровне исходных кодов.

СУБД PargreSQL применена нами для решения задачи разбиения сверхбольших графов. Использование параллельной СУБД для решения данной задачи встречается, насколько нам известно, впервые. Схема решения задачи выглядит следующим образом (рис. 3). Сверхбольшой граф представляется в виде реляционной таблицы — списка ребер, распределяемой по узлам кластерной системы.

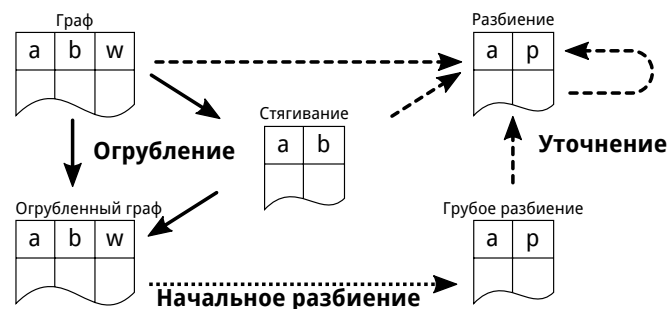


Рис. 3. Схема многоуровневого разбиения с помощью СУБД

На стадии огрубления параллельная СУБД выполняет запрос на языке SQL, результатом которого является реляционная таблица со списком несмежных ребер, имеющих наибольший суммарный вес. После этого СУБД выполняет запрос, который осуществляет стягивание найденных ребер. Получаемые в ходе стягивания петли уничтожаются, а кратные ребра преобразуются в одно ребро, имеющее вес, равный сумме весов кратных ребер. Процесс повторяется многократно, пока граф не перестанет быть сверхбольшим.

Полученный огрубленный граф может быть размещен целиком в оперативной памяти, что позволяет перейти к *стадии начального разбиения*. Огрубленный граф экспортируется из базы данных и подается на вход сторонней утилиты, которая выполняет начальное разбиение с помощью одного из традиционных алгоритмов. Результат начального разбиения импортируется в базу данных в виде реляционной таблицы, содержащей список вершин графа с указанием их цвета.

На стадии уточнения параллельная СУБД выполняет SQL-запрос, который отменяет стягивание ребер графа и «окрашивает» концы восстановленных ребер в цвет соответствующей вершины огрубленного графа. После этого СУБД выполняет запрос, который находит более выгодное разбиение путем «перекрашивания» вершин согласно некоторой эвристике (см. рис. 2). Процесс уточнения повторяется столько раз, сколько итераций производилось на стадии огрубления.

Итогом разбиения является реляционная таблица из двух столбцов: номер вершины графа и номер соответствующего подграфа.

Заключение

В статье предложен подход к использованию параллельной СУБД для реализации многоуровневого метода разбиения графов. В отличие от существующих последовательных и параллельных алгоритмов, предполагающих размещение данных в оперативной памяти, данный подход позволит разбивать сверхбольшие графы, в ходе анализа которых возникает очень много промежуточных данных.

Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта №12-07-31217 мол_а.

Литература

1. Ordonez, C. Bayesian Classifiers Programmed in SQL / C. Ordonez, S.K. Pitchaimalai // IEEE Transactions on Knowledge and Data Engineering. – 2010. – Vol. 22, № 1. – P. 139–144.
2. Ordonez, C. Integrating K-Means Clustering with a Relational DBMS Using SQL / C. Ordonez // IEEE Transactions on Knowledge and Data Engineering. – 2006. – Vol. 18, № 2. – P. 188–201.
3. Пан, К.С. Параллельный алгоритм решения задачи анализа рыночной корзины на процессорах Cell / К.С. Пан, М.Л. Цымблер // Вестник ЮУрГУ. Серия «Математическое моделирование и программирование». – 2010. – № 16(192). Вып. 5. – С. 48–57.
4. Kernighan, B.W. An Efficient Heuristic Procedure for Partitioning Graphs / B.W. Kernighan, S. Lin // The Bell system technical journal. – 1970. – Vol. 49, № 1. – P. 291–307.

5. Aggarwal, C.C. Managing and Mining Graph Data. / C.C. Aggarwal, H. Wang // Advances in Database Systems. – Springer, 2010. – Vol. 40. – 608 p.
6. Miniakhmetov, R.M. Integrating Fuzzy c-Means Clustering with PostgreSQL / R.M. Miniakhmetov // Труды Института системного программирования РАН. – 2011. – Т. 21. – С. 263–276.
7. Миниахметов, Р.М. Интеграция алгоритма кластеризации Fuzzy c-Means в PostgreSQL / Р.М. Миниахметов, М.Л. Цымблер // Вычислительные методы и программирование: Новые вычислительные технологии (Электронный научный журнал). – 2012. – Т. 13. – С. 46–52.
8. Chakravarthy, S. DB-FSG: An SQL-Based Approach for Frequent Subgraph Mining / S. Chakravarthy, S. Pradhan // Proceedings of the 19th International Conference on Database and Expert Systems Applications DEXA 2008 (Turin, Italy, September 1–5). – Springer, 2008. – P. 684–692.
9. Srihari, S. A Framework for SQL-Based Mining of Large Graphs on Relational Databases / S. Srihari, S. Chandrashekar, S. Parthasarathy // Advances in knowledge discovery and data mining. Lecture Notes in Computer Science. – 2010. – Vol. 6119. – P. 160–167.
10. Sokolinsky, L.B. Survey of Architectures of Parallel Database Systems / L.B. Sokolinsky // Programming and Computer Software – 2004. – Vol. 30. No. 6 – P. 337–346.
11. Лепихов, А.В. Обработка запросов в СУБД для кластерных систем / А.В. Лепихов, Л.Б. Соколинский // Программирование. – 2010. – № 4. – С. 25–39.
12. Lepikhov, A.V. Query Evaluation Techniques for Cluster Database Systems / A.V. Lepikhov, L.B. Sokolinsky // Proceedings of the 14th East European Conference on Advances in Databases and Information Systems (ADBIS 2010, Novi Sad, Serbia, September 20–24). Lecture Notes in Computer Science. – Springer, 2010. – Vol. 6295. – P. 351–362.
13. Пан, К.С. Разработка параллельной СУБД на основе последовательной СУБД PostgreSQL с открытым исходным кодом / К.С. Пан, М.Л. Цымблер // Вестник ЮУрГУ. Серия «Математическое моделирование и программирование». – 2012. – № 18(277). Вып. 12. – С. 112–120.

Пан Константин Сергеевич, аспирант, кафедра системного программирования, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), kvaren@gmail.com.

APPLYING PARALLEL DBMS FOR VERY LARGE GRAPH MINING

C.S. Pan, South Ural State University (Chelyabinsk, Russian Federation)

Graph partitioning is an interesting topic in graph mining, that comes into use for some theoretical and practical problems (graph coloring, integrated circuit desing, finite element modeling, etc.). The existing serial and parallel algorithms suppose that the graph being analyzed can fit into main memory along with all the intermediate data, so they cannot be

applied for very large graphs. We introduce a new way of partitioning - using the parallel relational DBMS PargreSQL that is based on open-source PostgreSQL DBMS.

Keywords: data mining, graph partitioning, parallel DBMS.

References

1. Ordonez C., Pitchaimalai S.K. Bayesian Classifiers Programmed in SQL. *IEEE Transactions on Knowledge and Data Engineering*. 2010, Vol. 22, No. 1, P. 139–144.
2. Ordonez C. Integrating K-Means Clustering with a Relational DBMS Using SQL. *IEEE Transactions on Knowledge and Data Engineering*. 2006, Vol. 18, No. 2, P. 188–201.
3. Pan C.S. A Parallel Algorithm for Market Basket Analysis on Cell Processor. *SUSU Bulletin "Mathematical Modeling and Programming"*. 2010, No. 5, P. 48–57.
4. Kernighan B.W., Lin S. An Efficient Heuristic Procedure for Partitioning Graphs. *The Bell system technical journal*. 1970, Vol. 49, № 1, P. 291–307.
5. Aggarwal C.C., Wang H. Managing and Mining Graph Data. *Advances in Database Systems*. Springer, 2010, Vol. 40, 608 p.
6. Miniakhmetov R.M. Integrating Fuzzy c-Means Clustering with PostgreSQL. *Proceedings of the Institute for System Programming of RAS*. 2011, Vol. 21, P. 263–276.
7. Miniakhmetov R.M., Zymbler M.L. Integracija algoritma klasterizacii Fuzzy c-Means v PostgreSQL [Embedding Fuzzy c-Means into PostgreSQL]. *Vychislitel'nye metody i programmirovaniye: Novye vychislitel'nye tehnologii (Jelektronnyj nauchnyj zhurnal)* [Computational Methods and Programming: New Computational Technologies (Digital Scientific Journal)]. 2012, Vol. 13, P. 46–52.
8. Chakravarthy S., Pradhan S. DB-FSG: An SQL-Based Approach for Frequent Subgraph Mining. *Proceedings of the 19th International Conference on Database and Expert Systems Applications (DEXA 2008, Turin, Italy, September 1–5)*. Springer, 2008, P. 684–692.
9. Srihari S., Chandrashekar S., Parthasarathy S. A Framework for SQL-Based Mining of Large Graphs on Relational Databases. *Advances in knowledge discovery and data mining. Lecture Notes in Computer Science*. 2010, Vol. 6119, P. 160–167.
10. Sokolinsky L.B. Survey of Architectures of Parallel Database Systems. *Programming and Computer Software*. 2004, Vol. 30, No. 6, P. 337–346.
11. Lepikhov A.V., Sokolinsky L.B. Query Processing in a DBMS for Cluster Systems. *Programming and Computer Software*. 2010, Vol. 36, No. 4, P. 205–215.
12. Lepikhov A.V., Sokolinsky L.B. Query Evaluation Techniques for Cluster Database Systems. *Proceedings of the 14th East European Conference on Advances in Databases and Information Systems (ADBIS 2010, Novi Sad, Serbia, September 20–24)*. *Lecture Notes in Computer Science*. Springer, 2010, Vol. 6295, P. 351–362.
13. Pan C.S., Zymbler M.L. Development of a Parallel Database Management System on the Basis of Open-Source PostgreSQL DBMS. *SUSU Bulletin "Mathematical Modeling and Programming"*. 2012, Vol. 18, No. 12, P. 112–120.

Поступила в редакцию 16 октября 2012 г.

СВЕДЕНИЯ ОБ ИЗДАНИИ

Серия основана в 2012 году.

Свидетельство о регистрации ПИ ФС77-26455 выдано 13 декабря 2006 г. Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия.

ПРАВИЛА ДЛЯ АВТОРОВ

1. Правила подготовки рукописей и пример оформления статей можно загрузить с сайта серии <http://vestnikvmi.susu.ru>. **Статьи, оформленные без соблюдения правил, к рассмотрению не принимаются и назад авторам не высылаются.**
2. Адрес редакции научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»:
Россия 454080, г. Челябинск, пр. им. В.И. Ленина, 76, Южно-Уральский государственный университет, факультет Вычислительной математики и информатики, кафедра СП, ответственному секретарю, доценту Цымблеру Михаилу Леонидовичу.
3. Адрес электронной почты редакции: vestnikvmi@gmail.com
4. **Плата с авторов за публикацию рукописей не взимается, и гонорары авторам не выплачиваются.**
5. Подписной индекс научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»: 10244, каталог «Пресса России». Периодичность выхода — 4 выпуска в год (февраль, май, август и ноябрь).