



# ВЕСТНИК

ЮЖНО-УРАЛЬСКОГО  
ГОСУДАРСТВЕННОГО  
УНИВЕРСИТЕТА

2013  
Т. 2, № 4

ISSN 2305-9052

СЕРИЯ

## «ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И ИНФОРМАТИКА»

ПОСВЯЩАЕТСЯ  
70-ЛЕТИЮ ЮЖНО-УРАЛЬСКОГО  
ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

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

**Л.Б. Соколинский**, д.ф.-м.н., проф., *отв. редактор*  
**В.П. Танана**, д.ф.-м.н., проф., *зам. отв. редактора*  
**М.Л. Цымблер**, к.ф.-м.н., доц., *отв. секретарь*  
**С.М. Абдуллаев**, д.г.н., проф.  
**А.В. Панюков**, д.ф.-м.н., проф.  
**К.С. Пан**, *техн. секретарь*

### Редакционный совет

**В.И. Бердышев**, д.ф.-м.н., акад. РАН, *председатель*  
**А. Андреяк**, PhD, профессор (Германия)  
**В.В. Воеводин**, д.ф.-м.н., чл.-кор. РАН  
**Дж. Донгарра**, PhD, профессор (США)

**И.И. Ерёмин**, д.ф.-м.н., акад. РАН  
**А.Б. Куржанский**, д.ф.-м.н., акад. РАН  
**В.Г. Романов**, д.ф.-м.н., чл.-кор. РАН  
**Д. Маллманн**, PhD, профессор (Германия)  
**А.Н. Томилин**, д.ф.-м.н., профессор  
**В.Е. Третьяков**, д.ф.-м.н., чл.-кор. РАН  
**А.М. Федотов**, д.ф.-м.н., чл.-кор. РАН  
**В.И. Ухоботов**, д.ф.-м.н., профессор  
**В.Н. Ушаков**, д.ф.-м.н., чл.-кор. РАН  
**М.Ю. Хачай**, д.ф.-м.н., профессор  
**П. Шумяцки**, PhD, профессор (Бразилия)  
**Е. Ямазаки**, PhD, профессор (Бразилия)



# BULLETIN

OF THE SOUTH URAL  
STATE UNIVERSITY

2013  
Vol. 2, no. 4

ISSN 2305-9052

---

SERIES

«COMPUTATIONAL MATHEMATICS  
AND SOFTWARE ENGINEERING»

---

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)  
**S.M. Abdullaev**, 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)  
**A. Andrzejak**, Heidelberg University (Germany)  
**V.V. Voevodin**, Lomonosov Moscow State University (Moscow, Russian Federation)  
**J. Dongarra**, University of Tennessee (USA)  
**I.I. Eremin**, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russian Federation)  
**A.B. Kurzhansky**, Lomonosov Moscow State University (Moscow, Russian Federation)  
**V.G. Romanov**, Sobolev Institute of Mathematics, Siberian Branch of the RAS (Novosibirsk, Russian Federation)  
**D. Mallmann**, Julich Supercomputing Centre (Germany)  
**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)  
**P. Shumyatsky**, University of Brasilia (Brazil)  
**Y. Yamazaki**, Federal University of Pelotas (Brazil)

## Содержание

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ СИСТЕМ НА ПРИМЕРЕ ДРЕВНЕГО ОБЩЕСТВА ЗЕМЛЕДЕЛЬЦЕВ В.А. Пьянков .....	5
УПРАВЛЕНИЕ РАВНОВЕСИЕМ ПО ШТАКЕЛЬБЕРГУ В ЗАДАЧАХ ГОСУДАРСТВЕННОГО РЕГУЛИРОВАНИЯ РЫНКА ЕСТЕСТВЕННОЙ МОНОПОЛИИ А.В. Панюков, Е.Д. Коновалова .....	17
KERNELGEN – ПРОТОТИП РАСПАРАЛЛЕЛИВАЮЩЕГО КОМПИЛЯТОРА С/FORTRAN ДЛЯ GPU NVIDIA НА ОСНОВЕ ТЕХНОЛОГИЙ LLVM Н.Н. Лихогруд, Д.Н. Микушин .....	28
ОТОБРАЖЕНИЕ НА КЛАСТЕРЫ С ГРАФИЧЕСКИМИ ПРОЦЕССОРАМИ DVMN-ПРОГРАММ С РЕГУЛЯРНЫМИ ЗАВИСИМОСТЯМИ ПО ДАННЫМ В.А. Бахтин, А.С. Колганов, В.А. Крюков, Н.В. Поддерюгина, М.Н. Притула .....	44
ASTROPNI: ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ МОДЕЛИРОВАНИЯ ДИНАМИКИ АСТРОФИЗИЧЕСКИХ ОБЪЕКТОВ НА ГИБРИДНЫХ СУПЕРЭВМ, ОСНАЩЕННЫХ УСКОРИТЕЛЯМИ INTEL XEON PHI И.М. Куликов, И.Г. Черных, Б.М. Глинский .....	57
ЭФФЕКТИВНОЕ ИСПОЛЬЗОВАНИЕ МНОГОЯДЕРНЫХ СОПРОЦЕССОРОВ ПРИ СУПЕРКОМПЬЮТЕРНОМ СТАТИСТИЧЕСКОМ МОДЕЛИРОВАНИИ ЭЛЕКТРОННЫХ ЛАВИН М.А. Марченко .....	80
<b>Краткие сообщения</b>	
ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ ПРОЦЕССОМ ПРИМЕНЕНИЯ АНТИВИРУСНЫХ ПРЕПАРАТОВ ПРИ ЛЕЧЕНИИ ВИЧ-ИНФЕКЦИИ И.П. Болодурина, Ю.П. Иванова .....	94
ПРЕДСТАВЛЕНИЕ ТОРГОВЫХ СИГНАЛОВ НА ОСНОВЕ АДАПТИВНОЙ СКОЛЬЗЯЩЕЙ СРЕДНЕЙ КАУФМАНА В ВИДЕ СИСТЕМЫ ЛИНЕЙНЫХ НЕРАВЕНСТВ М.М. Дышаев, И.М. Соколинская .....	103
РАЗРАБОТКА ВИРТУАЛЬНОГО ИСПЫТАТЕЛЬНОГО СТЕНДА ДЛЯ ЧИСЛЕННОГО МОДЕЛИРОВАНИЯ ГИДРОДИНАМИЧЕСКИХ ТЕЧЕНИЙ В БЕСФЛАНЦЕВЫХ ВИХРЕВЫХ РАСХОДОМЕРАХ С ИСПОЛЬЗОВАНИЕМ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ Е.В. Сафонов, К.А. Бромер, В.А. Дорохов .....	109

## Contents

ECONOMIC SYSTEM SIMULATION ON THE BASIS OF ANCIENT AGRICULTURAL SOCIETY V.A. Pyankov .....	5
THE CONTROL OF STACKELBERG EQUILIBRIUM IN PROBLEMS OF GOVERNMENT CONTROL OF NATURAL MONOPOLY A.V. Panyukov, E.D. Konovalova .....	17
KERNELGEN — A PROTOTYPE OF LLVM-BASED AUTO-PARALLELIZING C/FORTRAN COMPILER FOR NVIDIA GPUS N.N. Likhogrud, D.N. Mikushin .....	28
MAPPING DVMH-PROGRAMS WITH REGULAR DEPENDENCIES ONTO CLUSTERS WITH GPU V.A. Bakhtin, A.S. Kolganov, V.A. Krukov, N.V. Podderyugina, M.N. Pritula .....	44
ASTROPHI: A SOFTWARE PACKAGE FOR COMPLEX MODELLING OF ASTROPHYSICAL OBJECTS DYNAMICS WITH USING OF HYBRID ARCHITECTURE SUPERCOMPUTERS I.M. Kulikov, I.G. Chernykh, B.M. Glinsky .....	57
EFFECTIVE USE OF MULTICORE COPROCESSORS IN SUPERCOMPUTER STOCHASTIC SIMULATION OF ELECTRON AVALANCHES M.A. Marchenko .....	80
<b>Brief Reports</b>	
OPTIMUM CONTROL OF PROCESS OF APPLICATION OF ANTIVIRUS PREPARATIONS AT HIV INFECTION TREATMENT I.P. Bolodurina, Yu.P. Ivanova .....	94
REPRESENTATION OF TRADING SIGNALS BASED KAUFMAN ADAPTIVE MOVING AVERAGE AS A SYSTEM OF LINEAR INEQUALITIES M.M. Dyshaev, I.M. Sokolinskaya .....	103
DEVELOPMENT OF VIRTUAL TEST BENCH FOR CFD IN FLANGELESS VORTEX FLOWMETER WITH APPLICATION OF HIGH-PERFORMANCE COMPUTING E.V. Safonov, K.A. Bromer, V.A. Dorokhov .....	109

# ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ СИСТЕМ НА ПРИМЕРЕ ДРЕВНЕГО ОБЩЕСТВА ЗЕМЛЕДЕЛЬЦЕВ

*В.А. Пьянков*

Проводимое научное исследование заключается в моделировании социально-экономической системы с использованием положений теории физической экономики. Основным принципом данной теории состоит в оценке экономической деятельности человека с помощью физических параметров. Производительные силы должны выражаться в социальном времени (бюджет социального времени – величина доступного человечеству труда в год, измеренная в человеко-часах). Подзадачей исследования является проверка положений теории на симуляционных моделях. В ходе уже проведенного исследования был проверен принцип, гласящий, что по мере развития человеческого общества доля свободного (время, отличное от необходимого, расходуемого обществом для восполнения потраченных ресурсов) времени в общем фонде социального времени должна расти.

*Ключевые слова: физическая экономика, имитационное моделирование.*

## Введение

Проводимое научное исследование заключается в моделировании социально-экономической системы с использованием положений теории физической экономики. Объектом в данном научном исследовании является социально-экономическая система, включающая в себя человеческое общество и природную среду. Предметом в ходе данного исследования является энерго-трудовой цикл (простейшее древнее общество людей и среда). Цель исследования состоит в разработке имитационной модели энергетическо-трудового цикла.

Для достижения поставленной цели необходимо решить следующие задачи:

- создание модели взаимодействия древнего оседлого человеческого общества и среды;
- проведение имитационного эксперимента с полученной моделью;
- проверка положения теории (рост доли свободного времени в общем фонде социального времени).

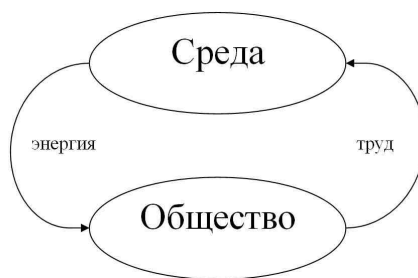
У различных научных групп (возглавляемых учеными, о которых будет упомянуто в обзоре), работающих в области физической экономики, имеется множество моделей современной социально-экономической системы, на основе которых они дают достаточно точные прогнозы экономического развития. Однако, несмотря на доступность общих положений теории физической экономики, автор не смог найти описания конкретных моделей. Это побудило к попытке создания своей модели. В виду сложности современной социально-экономической системы, было решено опробовать новый подход (основанный на теории физической экономики) к созданию модели социально-экономических систем на простом примере, которым является древняя человеческая община времен неолита.

В статье приводятся краткое описание модели, построенной автором на основе информации о обществе времен неолита и теории физической экономики, и результаты имитационного эксперимента с полученной моделью.

Статья организована следующим образом. В первом разделе приводится краткое описание теории физической экономики и новых работ в данной области. Во втором разделе приводится описание построенной математической модели. В третьем разделе описываются результаты имитационного эксперимента, проведенного с моделью. В заключении подведен краткий итог исследования и указано дальнейшее направление его развития.

## 1. Физическая экономика

Общей схемой модели социально-экономической системы, согласно теории физической экономики, является энергетическо-трудовой цикл, изображенный на рис. 1.



**Рис. 1.** Энергетическо-трудовой цикл

У начал данного течения стояли такие ученые, как Подолинский Сергей Андреевич, Побиск Георгиевич Кузнецов, Линдон Ларуш, Борис Евгеньевич Большаков. В дальнейшем идеи данного течения продолжали разрабатываться Кузнецовым О.Л., Шамилем Ш.Г., Алексеевым Г., Липенковым А.Д. и др. Предтечей физической экономики как научного течения был Сергей Андреевич Подолинский, являвшийся одним из основоположников русского космизма. Он был одним из первых, давших энергетическую трактовку человеческого труда [1]. Побиском Георгиевичем Кузнецовым были заложены основы данного течения, сформулирована его концепция [2]. Более того, анализ и описание трудового процесса Кузнецовым используется в данном исследовании для описания трудовых процессов внутри социально-экономической системы. Линдон Ларуш был основателем этого течения в США, имея свой особый взгляд на некоторые положения данного течения [4]. В дальнейшем это направление продолжало развиваться в трудах последователей Побиска Кузнецова.

Основной принцип данной научной концепции состоит в оценке экономической деятельности человека с помощью физических параметров. Производительные силы должны выражаться в социальном времени (бюджет социального времени – величина доступного человечеству труда в год, измеренная в человеко-часах). Такое представление намного упрощает анализ и понимание экономической системы. Все построения этой концепции лаконичны и в то же время, содержат в себе все необходимое, что должно быть в экономической теории. К тому же экономической системе, основанной на принципах физической экономики, изначально присуще управление, что делает ее поведение более предсказуемым, а значит, системе будет проще добиться поставленных перед ней целей.

Из современных исследований в данной области можно отметить работы Х. Майера, В.Е. Лихтенштейна и Г.В. Росса [10, 11]. В своей работе [10] Хельмут Майер для анализа текущих проблем экономики Евросоюза использует оригинальный подход, схожий с теорией физической экономики и основанный на оценке экономической деятельности с помощью

физических величин. В результате ему удается сделать нетривиальные выводы по поводу причин имеющихся проблем, а также указать возможные пути решения.

Совместная работа В.Е. Лихтенштейна и Г.В. Росса [11] представляет собой описание создаваемой ими теории развития, являющейся синтезом теории динамических систем и эволюционных алгоритмов. Теория позволяет анализировать развитие произвольной системы. В области моделирования социально-экономических систем теория развития имеет большой потенциал, что было показано авторами в их работе.

## 2. Модель социально-экономической системы

### 2.1. Описание модели

В качестве объекта моделирования выступает древняя человеческая община, занимающаяся земледелием.

Естественным будет положить в качестве элементарной временной единицы год, так как сама природа и человек вслед за ней живут годовыми циклами. Для упрощения моделирования вся деятельность общины разбивается на следующие категории:

- Земледелие, в качестве основной деятельности по обеспечению людей пропитанием. В данном случае община является земледельческой.
- Животноводство. Необходимо в земледельческой общине как вспомогательная деятельность.
- Приготовление пищи.
- Добыча ресурса и производство орудий труда и предметов потребления. Для осуществления работ общине необходим был инструмент, также для обеспечения человека одеждой и домом необходимы были достаточно заметные на фоне других работ трудовые усилия.

Также, отдельно будут составлены модели общества и системы распределения трудового ресурса.

Необходимо уточнить, что в дальнейшем под трудом, трудовым ресурсом будет пониматься то количество человеко-часов, которое затрачивается либо находится в распоряжении общины.

Земледелие в древней общине было простым. В начале посевной человек распахивал поле мотыгой или плугом с помощью тяглового скота, сажал потребляемую сельскохозяйственную культуру. При необходимости производились работы по орошению посевной территории, потом урожай собирался.

В модели полагается, что затрачиваемый на земледелие труд определяет рабочую площадь (площадь посева, возможная площадь обслуживания, доступная для сбора урожая площадь). Каждая из них суть линейная функция от затрачиваемого труда  $T$ , мощности орудий труда  $N$  (человек или тягловый скот; они образует работу (1)) и коэффициента преобразования работы в площадь.

Трудовые ресурсы, затрачиваемые на каждый вид деятельности внутри, относятся к разным временам внутри одного года, но поскольку в качестве элементарной временной единицы был выбран год, а земледельческая деятельность за весь год будет объединена, то необходимо определить тот принцип, согласно которому распределялся труд внутри земледелия.

Очевидно, что самое эффективное распределение будет в случае равенства всех трех площадей (посевов, обслуживаемая, сборов). Надо понимать, что данный принцип не применялся и не обдумывался древним человеком в явном виде, но рассуждения о том, что не стоит слишком много сеять, иначе община неспособна будет убрать, явно присутствовали.

Величина собранного урожая определялась плодородностью почв, на которых осуществлялась земледельческая деятельность. Согласно историческим изысканиям используемый древней общиной способ земледелия сильно истощал почвы, ухудшая их плодородие. Известно, что плодородие почв определяется тонким верхним слоем, который каждый год прирастает на малую постоянную величину. Очевидно, что при выращивании культур часть этого слоя поглощается урожаем. В таком случае, плодородность почвы можно определить как линейную функцию накопленной величины слоя за вычетом его использования для выращивания урожая.

Скот также был необходим для обеспечения тягловой силы в земледелии и служил источником таких видов ресурсов, как шерсть, шкуры.

Очевидно, что держать скот, дающий шерсть и работающий в качестве тягловой силы, сверх необходимого было не нужно древнему человеку. Исходя из потребностей в данных видах ресурсов, определялся необходимый размер стада.

С ресурсом, извлечение которого подразумевает забой скота, дело обстоит лишь немного сложнее. Исходя из естественной плодовитости можно определить доступную для забоя часть стада, а соответственно, по требуемой величине забоя можно определить необходимую величину стада.

Труд в животноводстве затрачивается на извлечение ресурса и на уход за скотом. Затрачивать труда на уход больше, чем того требует поголовье скота, равно как и тратить на извлечение больше, чем существующее поголовье может дать – неэффективно. В явном виде такие рассуждения не были присущи древнему человеку, но интуитивно, а также путем проб и ошибок, общество правильно определяло необходимые трудовые затраты.

Скот нуждается в пропитании, соответственно, человеку требовалось обеспечивать им свой стада. Поскольку времена года накладывали свой отпечаток, то пропитание достигалось двумя путями: выпас скота в теплое время года и покос луга для питания скота в холодное время года. Затраты труда на выпас и на покос существенно разные. Так же, как и в земледельческой деятельности, работа (человеческий труд, мощность человека) определяет рабочую площадь (выпаса и покоса).

Выпас скота и заготовка кормов для него, как это известно, истощают почву, на которой осуществляется данная хозяйственная деятельность. В данном случае связать рабочую площадь и объем потребленного скотом при выпасе и собранного корма с территории можно, как и в случае с земледелием, с помощью величины плодородия единицы рабочей площади.

Непосредственная работа со скотом и заготовка кормов для него, а также его выпас, могут быть объединены в одну деятельность – животноводство. Поскольку труд необходим для ухода, извлечения ресурса и заготовки кормов, то необходимо распределить его так, чтобы то поголовье, которое может прокормиться с заготовленного ресурса было равно тому поголовью, которое определяется трудовыми затратами на уход, также затрачиваемая на извлечение ресурса величина не должна превышать того количества, которое может быть полученного от поголовья, определенного выше.

Во время, к которому относится анализируемый этап развития человеческого общества, он уже активно и широко использовал орудия труда, равно как и пользовался результатами



своего труда в качестве предметов потребления (одежда, дома). При достаточности ресурсов человек, прикладывая трудовые усилия, изготавливал необходимый предмет. Если же для требуемого предмета не хватало ресурсов для изготовления, то они добывались специально для него. Таким образом, в модели для упрощения анализа можно положить, что еще на этапе добычи ресурса складывалась определенная пропорция, которая являла собой пропорции ресурса для изготовления определенного предмета. То есть, добывалось не просто требуемое количество ресурсов, а требуемое количество для каждого предмета, который производился.

При добыче, трудовые затраты автоматически и неявным для человека образом распределялись таким образом, чтобы было добыто как можно больше ресурсов для изготовления требуемого количества изделий с поддержанием пропорции распределения по видам ресурса. Между добычей и производством трудовой ресурс распределяется, исходя из потребности в результатах производства. Необходимо добывать столько ресурса (комплектов ресурса, каждый – для определенного предмета), сколько может быть с помощью труда обращено в конечный продукт. Необходимо будет учитывать как объемы ресурса, поступающего от животноводства, так и необходимость ремонта орудий труда.

Поскольку в ходе трудовой деятельности человек не затрачивает какое-то количество орудий труда, а лишь использует его ресурс, то при анализе в качестве затрат при использовании орудий труда удобно использовать оставшийся срок службы изделия, который при его использовании уменьшается на величину труда, затрачиваемого человеком при его использовании. Поэтому производить анализ деятельности по производству будет удобней с помощью фонда времени орудий труда и иных материальных ценностей, используемых человеком, который представляет собой доступное время использования соответствующего материального блага. Использование их будет уменьшать величину фонда, производство – увеличивать.

Моделирование деятельности по приготовлению пищи представляется наиболее простым, поскольку в ней просто необходимы определенные трудовые затраты для изготовления потребляемого продукта. Для унификации и удобства анализа, а также в соответствие с принципами физической экономики, его необходимо выразить в энергетических единицах, поскольку потребность человека в пище представляет лишь энергетическую потребность его организма (в данной модели не рассматривается важность определенного рациона питания, равно как и само понятие рациона).

Саму человеческую общину удобно моделировать в данном случае в связке с той внутренней энергией, которая ему присуща. Иными словами, у общества есть как бюджет социального времени, так и своеобразный энергетический бюджет, годовой объем которого определял среднегодовую численность общины. Такой подход удобен в моделировании, поскольку позволяет не задавать потребность человека в том или ином конкретном питательном ресурсе, а лишь задать определенную энергетическую потребность человека. Данный подход также нашел свое отражение в представлении деятельности по приготовлению пищи.

Каждая деятельность сама по себе присуща как животноводческой, так и земледельческой общине. Различия лежат лишь в масштабах каждой деятельности. Лишь взаимодействие этих видов деятельности древней общины и природы порождает определенный тип хозяйствования.

Необходимо описать, как именно распределялся годовой бюджет социального времени древней общины. Для сельскохозяйственной общины основой для распределения этого бюджета был урожай, собранный в прошлом году. Необходимо было определить, сколько зерна выделить на посев, а сколько – потребить. На данное решение влияло несколько факторов.

Первый существенный фактор – ограниченный срок проведения посевной. За отведенное время человек мог успеть распахать площадь гораздо меньшую, нежели теоретически мог за год. Второй фактор состоит в том, что прирост человеческой общины не мог превосходить определенной величины, потому потреблять лишнее, а значит, и сеять его было лишено смысла. С учетом этих факторов определялся размер посевов и объем потребления, а значит, и трудовые затраты, необходимые для них.

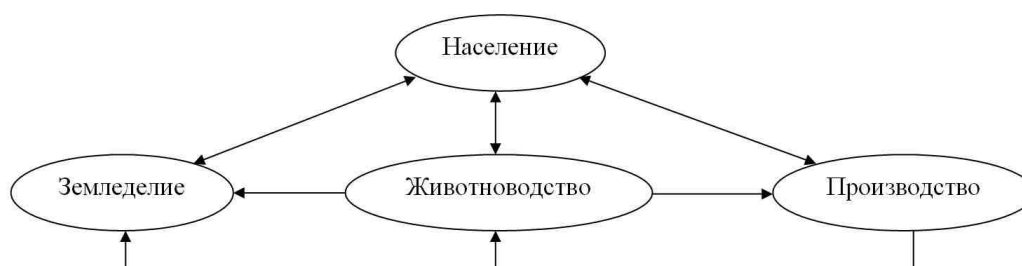
Для человека необходима одежда, поэтому, исходя из численности общины, определялись поголовья стад, с которых человек добывал себе шерсть и шкуры. Кожа, полученная из шкур, также необходима для производства орудий труда. Если же человек к тому времени уже применял тягловый скот, то он также держал и поголовье тяглового скота. Все это требовало определенных трудовых затрат, описанных для животноводческой деятельности выше.

Для различной деятельности, в частности, для земледелия необходимы были орудия труда (мотыги, сохи и прочее), также необходимо было изготавливать одежду, строить хижины для людей, загоны для скота. Необходимо было также чинить изношенные орудия труда. На это также были необходимы затраты социального времени.

Наконец, для поступающего объема питающего ресурса (зерновые культуры, а также побочный – мясо тех животных, которых забивали для иных целей) необходим был труд, чтобы приготовить его для употребления людьми.

Таким образом, исходя из начального решения об объемах потребления и посева зерновых культур, можно определить полные необходимые трудовые затраты древней общины.

В итоге, на основе исторических данных, также с использованием положений теории физической экономики, была составлена схема модели социально-экономической системы общества времен неолита, изображенная на рис. 2. Стрелками на рисунке обозначены различные потоки, как-то: труд, энергия, материалы.



**Рис. 2.** Общая схема модели

## **2.2. Математическая модель**

В качестве математического описания используются системы конечно-разностных уравнений. Они же использовались для создания органов управления.

Изложенный ниже подход для описания элементарной трудовой системы использовался в ходе составления математической модели социально-экономической системы общества времен неолита.

Модель является дискретной, поэтому необходимо определить величину шага моделирования. Естественным будет взять год, как величину шага для модели человеческого общества. Каждому шагу будет поставлены в соответствие различные величины, описывающие модель, например фонд социального времени – общая продолжительность трудовых процессов, которые могут быть выполнены обществом – выраженная в часах.

Основополагающим и базовым элементом рассмотрения социально-экономической системы являлся акт труда, в результате которого выполнялась работа и достигался тот или иной результат. В своих трудах [2] П.Г. Кузнецов показал, что для анализа трудовой деятельности применимо физическое понятие работы как произведения полезной мощности  $N$  на время работы  $T$ . Выполненная работа  $A$ , таким образом, будет равна:

$$A = N \cdot T. \quad (1)$$

В приложении к трудовой деятельности человеческого общества удобно сделать следующие изменения смысла величин. Поскольку в ходе выполнения одной и той же работы инструмент приходит в негодность, равно как его может быть больше одной единицы (в предыдущей формуле говорилось лишь о трудовом акте), то удобно говорить не просто о времени работы, но о затраченном фонде времени орудий труда  $T_G$ . Фонд орудий труда представляет собой общую продолжительность трудовой деятельности, которая может быть выполнена с помощью данного вида орудий. Но, поскольку орудиями манипулирует человек, то надо также учесть и фонд социального времени  $T_L$ , выделенный на данную работу. Фонд социального времени за определенный промежуток представляет собой произведение общего количества трудящихся людей или занятых в определенной деятельности на продолжительность анализируемого промежутка. В результате формула (1) примет вид:

$$A = N \cdot \min(T_G; T_L), \quad (2)$$

где  $N$  – мощность трудового процесса.

Всякая трудовая деятельность в модели дает материальный результат –  $P$ , который выражается в виде произведения работы на ее отдачу  $k$ :

$$P = k \cdot N \cdot \min(T_G; T_L), \quad (3)$$

Отдача от работы в модели может быть как постоянной величиной, так и переменной, которая зависит от давления на среду в виде человеческого труда  $A$ . В общем виде динамику изменения отдачи будет удобно выразить с помощью следующего уравнения:

$$\Delta k = \varepsilon - \theta A, \quad (4)$$

где  $\varepsilon, \theta$  – коэффициенты.

В результате трудового процесса фонд орудий труда  $\hat{T}_G$  динамически меняется, прирастая от производства на  $\tau r \alpha P$  и уменьшаясь от выполненной работы на  $T_G$ .

$$\Delta \hat{T}_G = \tau r \alpha P - T_G, \quad (5)$$

где  $\alpha$  – часть продукции, идущая на восстановление производственных фондов.

Годовой фонд социального времени для каждого шага  $T_L$  зависит от численности общины  $L$ , доли трудоспособного населения  $\gamma$  и длительности трудового  $\tau_L$ :

$$T_L = 365\tau_L\gamma L. \tag{6}$$

В модели полагается, что население растет с относительной скоростью  $K_L$ , равной естественной скорости прироста человеческой популяции, не ограниченной ничем. Тем не менее, на скорость прироста накладываются ограничения – недостаток еды и предметов потребления. Кратко это может быть выражено формулой:

$$\frac{\Delta L}{L} = \min \left( K_L; \frac{\beta_L(1 + K_L)L}{(1 - \alpha)P} - 1 \right), \tag{7}$$

где  $\beta_L$  – потребность одного человека в продукции  $P$ .

В результате можно составить следующую систему, которая иллюстрирует общий подход математического описания модели:

$$\begin{cases} P = k \cdot N \cdot \min(T_G; T_L); \\ \Delta k = \varepsilon - \theta A; \\ \Delta \hat{T}_G = \tau_P \alpha P - T_G; \\ T_L = 365\tau_L\gamma L; \\ \frac{\Delta L}{L} = \min \left( K_L; \frac{\beta_L(1+K_L)L}{(1-\alpha)P} - 1 \right). \end{cases} \tag{8}$$

### 3. Эксперимент

В виду большого количества уравнений, описывающих модель, и нелинейности многих из них было решено реализовать модель в виде имитационной модели в среде VisSim [12]. Начальные данные, а также различные коэффициенты модели соответствуют знаниям, полученным о людских общинах времен неолита [5, 7]. Была проведена симуляция модели земледельческой общины. Ниже на графиках будут приведены результаты симуляции. На рис. 3 представлен график численности  $L$  моделируемой человеческой общины.

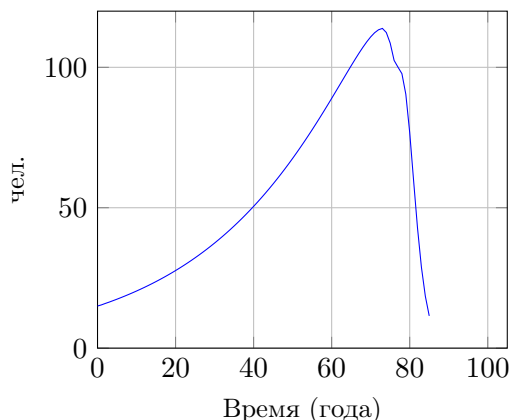
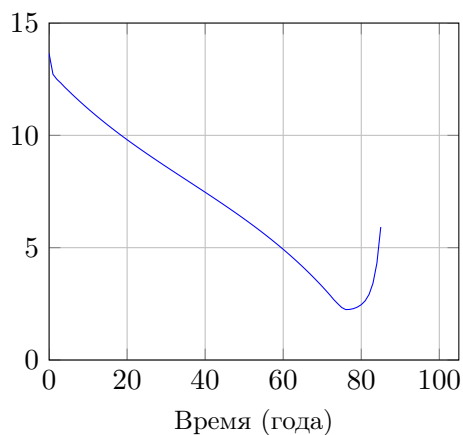
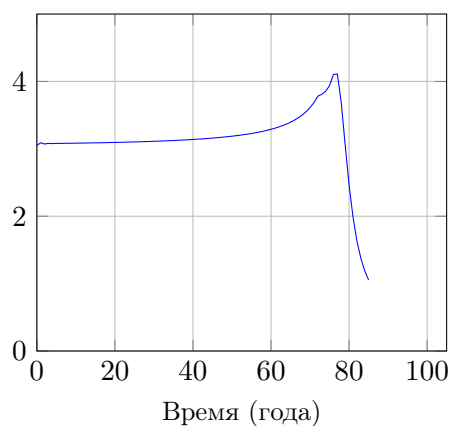


Рис. 3. Численность общины земледельцев

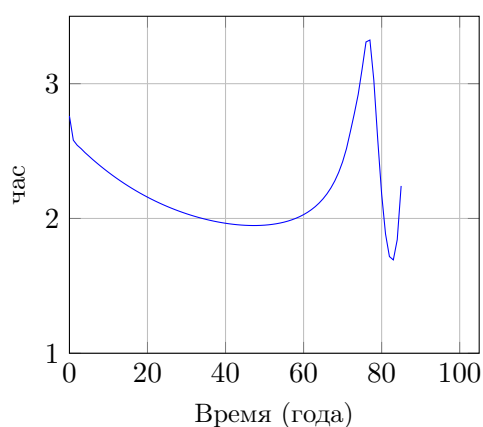
Через 73 года своего существования община достигает пика своей численности в 117 человек, затем численность падает, община вымирает. Это произошло вследствие падения



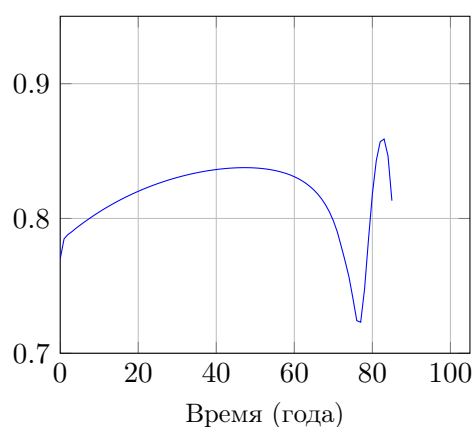
а) Нагрузка на одного рабочего



б) Продуктивность одного рабочего



в) Средняя продолжительность рабочего дня



г) Доля свободного времени

**Рис. 4.** Трудовые показатели общины

плодородности земли ниже того уровня, который может прокормить общину, вызванного эксплуатацией данного природного ресурса

На рис. 4а изображен график нагрузки на одного человека, занятого сельскохозяйственными работами. Иными словами – это отношение среднего числа людей не занятых в земледелии к среднему числу человек, занятых в нем.

Нагрузка на одного рабочего постепенно снижается, достигая минимума одновременно с пиком численности общины. На рис. 4б показан график выработки сельскохозяйственного ресурса одним работником, занятым в земледелии. Иными словами это – продуктивность одного рабочего в земледелии. Продуктивность постепенно растет, достигая своего пика одновременно с пиком численности, затем уменьшается.

Данные графики были приведены с целью описания динамики жизни моделируемой человеческой общины. Далее представлены средней продолжительности трудового дня  $\tau_L$  и доли свободного времени в общем фонде социального времени. График средней продолжительности трудового дня приведен на рис. 4в.

Из рисунка видно, что минимальная продолжительность трудового дня приходится вовсе не на период пика численности человеческой общины.

Важным результатом эксперимента является подтверждение теоретического постулата о росте доли свободного времени в общем фонде социального времени. Эта относительная величина представляет собой отношение величины фондов, необходимых для обеспечения

жизнедеятельности к максимально возможной величине фондов, его график показан на рис. 4г. Данное явление было описано в работе [2], однако в ней было дано лишь его словесное описание, тогда как в данной работе удалось воспроизвести его в ходе имитационного эксперимента.

Из графика видно, что доля свободного времени растет, имеет пик и его продолжительность аналогичные продолжительности трудового дня.

На основе графиков можно сделать следующие выводы относительно динамики развития человеческой общины. От начала отсчета и до минимума средней продолжительности трудового дня человеческая община росла интенсивно. Растущая численность была обеспечена самим этим ростом, благодаря чему трудовая нагрузка на одного рабочего постоянно падала (о чем говорят графики на рис. 4в и 4г). Далее, рост численности общины уже не мог поддержать сам себя и возникла необходимость в увеличении трудовой нагрузки на одного рабочего. В то же время, продуктивность одного рабочего продолжала расти, равно как и падала бремя не занятых в земледелии на одного рабочего в земледелии. Стоит подчеркнуть, что дальнейший рост общины, продуктивности и снижение бремени потребовали большей занятости каждого работника. В дальнейшем плодородность эксплуатируемых земель упала настолько, что ни рост занятых в земледелии, ни рост нагрузки не смогли компенсировать падение отдачи труда в земледелии. В результате чего жизнь общины на фиксированном участке территории прекращалась.

## **Заключение**

В статье приведено краткое описание теории физической экономики и новых работ в данной области, описана построенная имитационная математическая модель. Показаны результаты имитационных экспериментов с моделью, в ходе которых была показана адекватность модели историческим данным. Также было воспроизведено теоретически предсказанное явление возрастания доли свободного времени в ходе развития общества.

В дальнейшем необходимо провести более глубокий анализ трудовых процессов в социально-экономических системах разных масштабов и уровней развития с целью формулировки универсальной модели, описывающей сложные социально-экономические системы.

*Работа выполнена при финансовой поддержке Министерства образования и науки Российской Федерации, соглашение 8.4817.2011.*

## **Литература**

1. Подолинский, С.А. Труд человека и его отношение к распределению энергии / С.А. Подолинский. — М.: Ноосфера, 1991. — 82 с.
2. Кузнецов, П.Г. Система природа — общество — человек: Устойчивое развитие / О.Л. Кузнецов, П.Г. Кузнецов, Б.Е. Большаков. — Дубна: Международный университет природы, общества и человека «Дубна», 2000. — 353 с.
3. Большаков, Б.Е. Система универсальных мер-законов в науке устойчивого развития / Б.Е. Большаков // Устойчивое инновационное развитие: проектирование и управление. — 2011. — Т. 7, № 4. — С. 6–37.
4. LaRouche, L.H. The Science of Physical Economy as the Platonic Epistemological Basis for

- All Branches of Human Knowledge / L.H. LaRouche // Executive Intelligence Review. — 1994. — Vol. 21, No. 9. — P. 22–37.
5. Кравченко, А.И. Культурология: учебное пособие для вузов / А.И. Кравченко. — 3-е изд. — М.: Академический проект, 2001. — 496 с.
6. Кравченко, А.И. Первобытная культура / А.И. Кравченко // Культурология. — 2011. — № 3. — С. 65–67.
7. История мировой экономики // под ред. Г.Б. Поляка, А.Н. Марковой. — М.: 2002. — 727 с.
8. Пьянков, В.А. Имитационное моделирование экономических систем на основе принципов физической экономики / В.А. Пьянков, А.Д. Липенков // Математическое и статистическое исследование социально-экономических процессов. — 2011. — № 3. — С. 21–28.
9. Пьянков, В.А. Имитационное моделирование экономических систем на основе принципов физической экономики / В.А. Пьянков // Всероссийская конференция «Статистика. Моделирование. Оптимизация» (СМО-2011): Сборник трудов всероссийской конференции (Челябинск, декабрь, 2011 г.). — Челябинск: Издательский центр ЮУрГУ, 2011. — С. 315–319.
10. Майер, Х. Обоснование экономической и финансовой системы мира природы и ее влияние на человеческое общество и политику / Х. Майер. — СПб.: Нестор-История, 2010. — 178 с.
11. Лихтенштейн, В.Е. Введение в теорию развития / В.Е. Лихтенштейн, Г.В. Росс. — М.: Финансы и статистика, 2011. — 328 с.
12. VisSim User's Guide by Visual Solutions, Inc. URL: [http://vissim.com/downloads/doc/VisSim\\_UGv80.pdf](http://vissim.com/downloads/doc/VisSim_UGv80.pdf) (дата обращения: 03.07.2013)

Виталий Александрович Пьянков, аспирант, кафедра экономико-математических методов и статистики, Южно-Уральский государственный университет (Челябинск, Российская Федерация), cold.karadras@gmail.com.

---

## ECONOMIC SYSTEM SIMULATION ON THE BASIS OF ANCIENT AGRICULTURAL SOCIETY

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

The essence of ongoing research consists in economic system simulation using physical economics theory. The basic principle of this science conception can be expressed as follows: human economic activity should be evaluated via physical parameters. Production powers should be expressed in social time (social time budget is the amount of labour available to humanity within a year, expressed in hours). The research subtask is to verify the theory via simulation models. Existing research resulted in verification of the following principle: as the human society develops the fraction of free time (time that is different from necessary time, spent by society to replenish the resources used) in general budget of social time must grow.

*Keywords: simulation, economics, physical economics.*

## References

1. Podolinsky S.A. Trud cheloveka i ego otnosheniye k raspredeleniyu energii [Human labour and its relevance to energy distribution]. Moscow, Noosfera, 1991. 82 p.
2. Kuznetsov P.G., Kuznetsov O.L., Bolshakov B.E. Sistema priroda – obshestvo – chelovek [Nature – Society – Human System: sustainable development]. Dubna, Dubna International University of nature, society and human, 2000. 353 p.
3. LaRouche L. H. The Science of Physical Economy as the Platonic Epistemological Basis for All Branches of Human Knowledge // Executive Intelligence Review. 1994. Vol. 21, No. 9–11. P. 22–37.
4. Bolshakov B.E. Sistema universalnih mer-zakonov v nauke ustoichivogo razvitiya [Universal measure-law system in sustainable development science]. Ustoichivoye razvitiye: proectirovaniye i upravleniye [sustainable development: design and management]. 2011. Vol. 7, No. 4. P. 6–37.
5. Kravchenko A.I. Kulturologiya [Culturology]. Moscow, Academic Avenue, 2001. 496 p.
6. Kravchenko A.I. Pervobytnaya kultura [Primal culture]. Kulturologiya [Culturology], 2011. No. 3. P. 65–67.
7. Polyak G.B., Markova A.N. Istoriya Mirovoi Ekonomiki [World Economic History]. Moscow, Unity, 1999. 727 p.
8. Pyankov V.A., Lipenkov A.D. Imitacionnoye modelirovaniye ekonomicheskikh sistem na osnove principov fizicheskoi ekonomiki [Economic system simulation based on physical economics principles]. Matematicheskoye i statisticheskoye issledovaniye socialno-ekonomicheskikh processov [Mathematical and statistical analysis of socio-economic processes], 2011. No. 3. P. 21–28.
9. Pyankov V.A. Imitacionnoye modelirovaniye ekonomicheskikh sistem na osnove principov fizicheskoi ekonomiki [Economic system simulation based on physical economics principles]. Vserossiyskaya konferenciya «Statistika. Modelirovaniye. Optimizaciya» (SMO-2011): Sbornik trudov vs Rossiyskoi konferencii (Chelyabinsk, Dekabr, 2011) [Proceedings of the National Scientific Conference (Chelyabinsk, Russia, December, 2011)]. Chelyabinsk, Publishing of the South Ural State University, 2011. P. 315–319.
10. Mayer H. Obosnovaniye ekonomicheskoy i finansovoy sistemi mira prirodi i ee vliyaniye na chelovecheskoyr obshestvo i politiku [Nature’s economic and financial system statement and its influence on human society and politics]. St. Petersburg, Nestor-Istoriya, 2010. 178 P.
11. Lihtenshtein V.E., Ross G.V. Vvedeniye v teotiyu razvitiya [Introduction to development theory]. Moscow, Finansi i statistica, 2011. 328 p.
12. VisSim User’s Guide by Visual Solutions, Inc. URL: [http://vissim.com/downloads/doc/VisSim\\_UGv80.pdf](http://vissim.com/downloads/doc/VisSim_UGv80.pdf) (accessed: 03.07.2013)

*Поступила в редакцию 10 июня 2013 г.*



## УПРАВЛЕНИЕ РАВНОВЕСИЕМ ПО ШТАКЕЛЬБЕРГУ В ЗАДАЧАХ ГОСУДАРСТВЕННОГО РЕГУЛИРОВАНИЯ РЫНКА ЕСТЕСТВЕННОЙ МОНОПОЛИИ

*А.В. Панюков, Е.Д. Коновалова*

Данное исследование направлено на увеличение эффективности методик государственного регулирования рынка естественной монополии. В качестве методов регулирования рассматривается применение неценовых рычагов. Эффективность системы управления, скорость реакции на внешние изменения зависит от адаптивности. Решение проблемы адаптации государства к изменению ситуации на рынке происходит циклически. На каждом шаге решаются две задачи: построение модели взаимодействия монополиста и покупателя при заданных корректирующих государственных регуляторах (рынок рассматривается как замкнутая система); построение модели взаимодействия государства с рынком и оценка характеристик адаптивности государства. Решение задачи основано на определении равновесия по Штакельбергу в бескоалиционных играх двух лиц в смешанных стратегиях. При этом скорость, эффективность, оптимальность выбора государством корректирующего варианта и непосредственно момента его выбора характеризует адаптивность государства.

*Ключевые слова:* рынок естественной монополии, государственное регулирование, адаптивность, бескоалиционная игра, равновесие по Штакельбергу, смешанные стратегии.

### Введение

Необходимость государственного регулирования рынков с высокой степенью монополизации не вызывает споров, но методики такого регулирования остаются далеко не совершенными. Они являются одной из важнейших экономических проблем как на уровне государства, так и любого административно-территориального образования. При этом в качестве методов регулирования в условиях современной российской экономической системы приемлемо применение только неценовых рычагов регулирования таких рынков.

Наиболее эффективным способом регулирования монополизированных рынков является тарифное регулирование. На данный момент тарифы формируются исключительно затратным методом, что не позволяет монополиям получать сверхприбыли, но при этом никаким образом не регулируется конкурентная составляющая рынка. Особую сложность в связи с этим вызывает регулирование рынков естественных монополий. Сама суть тарифного регулирования монопольных рынков предполагает включение в тариф составляющей, которая отражает реакцию государства на естественную монополию. Это призвано сократить влияние монопольной власти и не затронуть существование таких компаний, так как они производят блага с высокой общественной полезностью.

Проблема гибкого реагирования на изменение ситуации на рынках естественных монополий поднята на законодательном уровне. В качестве одного из примеров можно привести регулирование вопросов транспортировки газа, нефти и нефтепродуктов. В 2010 году был внесен на рассмотрение Проект Федерального закона «О внесении изменений в ФЗ «О естественных монополиях» и отдельные законодательные акты РФ»

(от 17 декабря 2010 г.). Регулирование и (или) контроль деятельности субъектов естественных монополий предлагается осуществляется на основе иных принципов, в числе которых:

- соблюдение баланса интересов потребителей и субъектов естественных монополий, обеспечивающего доступность оказываемых услуг и надлежащий уровень их качества для потребителей, эффективное функционирование и развитие субъектов естественных монополий;

- применение гибкого тарифного регулирования субъектов естественных монополий с учетом отраслевых особенностей, масштабов их деятельности, рыночной конъюнктуры, среднесрочных (долгосрочных) макроэкономических и отраслевых прогнозов.

Данный проект имеет своими целями в числе прочих:

- дополнить существующий метод тарифного регулирования «затраты+» методами индексации тарифов, доходности на капитал, сопоставления тарифов между субъектами естественных монополий, а также учитывать конъюнктуру цен на сопряженных рынках;

- связать тарифные решения с показателями объемов оказываемых услуг, их надежности, качества и эффективности, и проводить постоянный мониторинг этих показателей.

В связи с тем, что рассматриваемый Проект не был принят, проблема остается острой и на сегодняшний день. Причем важно, что вопрос был поднят на высшем государственном уровне, но решение так и не было найдено.

Данное исследование направлено на увеличение эффективности методик государственного регулирования монополизированных рынков. Основную сложность при разработке способов регулирования монополизированных рынков, особенно рынков естественных монополий, создает нестабильность, связанная с противоборством интересов государства как защитника свободного конкурентного рынка, а с другой стороны, как гаранта снабжения населения жизнеобеспечивающими благами, поставщиком которых является естественный монополист. Эта ситуация требует динамического государственного вмешательства с целью обеспечения баланса интересов монополии и общества. Эффективность системы управления, скорость реакции на внешние изменения зависит от адаптивности.

Исследованию отдельных проблем адаптации участника рынка к условиям ускорения рыночных изменений посвящены труды многих экономистов, как российских, так и зарубежных. В их числе Р. Акофф, Дж. Баумоль, О. Харт, К. Гриффит, Дж. Гэлбрейт, Х. Демсец, Р. Коуз, Е. Маррис, Ф. Найт, Д. Порт, Э. Тоффлер, О. Уильямсон, С. Уинтер, С. Бир, Дж. Хикс, Э. Чемберлин. Применение этих разработок к условиям российского рынка, а также собственные подходы к решению этих проблем представили ведущие российские ученые по рассматриваемой тематике: В.С. Автономов, В.С. Катькало, В.М. Макаров, И.М. Розанова, С.В. Кожевников, В.И. Ширяев. Наиболее глубокое исследование в области адаптивности именно государства принадлежит В.А. Мау, который указывает на то, что проблемы оценки и повышения адаптивности целесообразно решать не только в сфере стратегического управления предприятиями, но и в сфере государственного управления [6]. Несмотря на внимание, уделяемое в настоящее время повышению адаптивности, наблюдается дефицит объективных количественных оценок адаптивности как характеристики производственной системы [1].

Данная работа направлена именно на решение вопроса построения количественной оценки указанной проблемы и опирается на анализ, предложенный С.Я. Чернавским и О.А. Эйсмонтом, а также модели, разработанные Л.В. Степановым.

Работа организована следующим образом. Раздел 1 содержит постановку задачи исследования, кратко описывает общий подход, выбранный для решения поставленной задачи. Раздел 2 посвящен вопросам построения модели рыночного окружения государства. В разделе 3 разработан алгоритм государственного управления состоянием монополизированного рынка и доказана теорема о результативности данного алгоритма. Работа представляет часть более обширного исследования по указанной проблеме, в заключении наряду с основными выводами приведен краткий перечень вопросов, рассмотренных авторами ранее в рамках поставленной проблемы.

## 1. Общий подход к решению задачи

Решение проблемы адаптации государства к изменению ситуации на рынке естественной монополии происходит пошагово. После установления государством определенных условий рынок рассматривается как замкнутая система, монополист и покупатели взаимодействуют без государственного влияния.

Для того, чтобы выбрать оптимальную стратегию, государство рассматривает замкнутый рынок, в результате чего у него есть возможность получить крайние варианты интересующих характеристик рынка и затем из вариантов определяется один, подходящий для государства и исходя из него избирается новая стратегия государства и новая матрица  $\Delta_{k+1}$  условий на рынке. После выбора способа влияния на конкретном шаге и изменения условий на рынке начинается новый шаг. При этом скорость, эффективность, оптимальность выбора корректирующего варианта и непосредственно момента его выбора характеризует адаптивность государства.

Исходя из этого на каждом шаге решается два вопроса:

1. Построение модели взаимодействия монополиста и покупателя при заданных корректирующих государственных регуляторах (налоги, акцизы, дотации, гранты и др.), представленных матрицей  $\Delta_k$ .
2. Построение модели взаимодействия государства с рынком и оценка характеристик адаптивности государства.

Поскольку изменение внешних условий работы объекта управления далеко не всегда поддается прогнозированию, для анализа процессов адаптации используют аппарат принятия решения в условиях риска и неопределенности. Процесс функционирования в нестабильной среде представляем в форме динамической стохастической игры. В качестве природы в данном случае выступает игрок, хотя и действующий сознательно, но имеющий цель не связанную с игрой и получающий результат независимый от решения игры. В рассматриваемой задаче природой выступает рыночное окружение государства, а стратегиями активного игрока являются принимаемые государством политические решения относительно правил функционирования рынка. Состояния природы в свою очередь можно рассматривать как результат игры продавцов с покупателями, с учетом установленных правил. Первым этапом исследования является решение задачи поиска этих состояний.

## 2. Построение модели рыночного окружения государства

Взаимодействие каждого конкретного покупателя с монополистом и всех покупателей с монополистом следует рассматривать как конфликты. Для определения ситуации на рынке, то есть для разрешения конфликта применяется аппарат теории игр.

Монополизированный рынок имеет характеристики, которые вносят особенности в формализацию игры и, следовательно, ее решение. Для того чтобы определиться со способом решения необходимо определить вид решаемой игры. Особенности рассматриваемого рынка заключаются в следующем: при монопольной конкуренции у монополиста и покупателей различные, но не противоположные цели; при монополии потребители не могут рассматриваться как коалиция, так как они действуют без координации стратегий участников; рассмотрение взаимодействия монополиста и множества отдельных потребителей при достаточно великом их множестве приводит к существенному усложнению расчетов. Анализ экономической сущности монопольного рынка показывает необходимость рассматривать монополизированные рынки в виде некооперативной неантагонистической (биматричной) игры двух лиц [8]: монополиста М и обобщенного потребителя П.

Игровой процесс заключается в следующем. Игрок М предлагает потребителям товар, пытаясь максимизировать прибыль (рис. 1) за счет увеличения ценовых характеристик товара. Игрок П, сравнивая неценовые характеристики и потребительскую выгоду от них с запрошенной ценой, формирует величину потребности в каждом из товаров. Уменьшение потребности игрока П в товаре влияет на величину получаемой прибыли от реализации этого товара игроком М. Так как игрок П имеет множественную структуру, то потребность в каждом товаре формируется независимо каждым из потребителей. Однако каждого из множества потребителей рассматривать неэффективно, при этом на рынке естественной монополии все потребители характеризуются сходными свойствами (невозможность единолично влиять на рынок, невозможность в чистом виде создавать значимые коалиции и др.) и одинаковой целью (максимизация потребительской полезности), поэтому будем рассматривать всю совокупность потребителей как единого игрока.

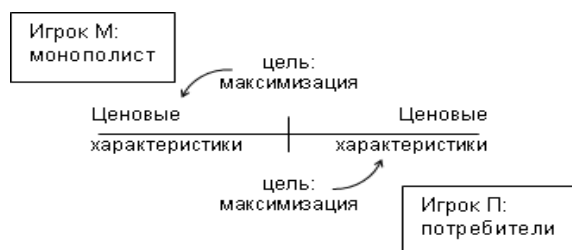


Рис. 1. Связь интересов игроков и характеристик товара

С позиции теоретико-игрового подхода игровой процесс на рынке в общей форме можно выразить кортежем

$$G = (\{M, P\}, \{X_M, X_P\}, \{H_M, H_P\}),$$

где

– М – предприятие-монополист,  $X_M = \{i = 1, 2, \dots, n\}$  – множество стратегий монополиста;  $S_i$  – предложение монополиста покупателям с использованием технологии  $i = \overline{1, n}$  (определяется товар и его цена  $P_i$ );

– П – множество предприятий-потребителей,  $X_\Pi = \{j = 1, 2, \dots, m\}$  – множество стратегий потребителей;  $D_j$  – рыночный спрос потребителя на товары с потребительскими свойствами  $j = \overline{1, m}$  (определяются группы потребителей и объемы их потребления  $V_j$ );

–  $H_M = \{PV^T - \Delta_M\}$  – выигрыш игры Г монополиста, где элементы матрицы  $H_{Mij} = P_i V_j - \Delta_{Mij}$  – прибыль монополиста при наступлении ситуации  $(i, j)$ ,  $P_i$  – цена, установленная монополистом,  $V_j$  – объем, который согласны купить потребители при объявленной цене,  $\Delta_M$  – матрица государственного воздействия на монополиста;

–  $H_\Pi = \{-PV^T + \Delta_\Pi\}$  – матрица выигрышей игры Г покупателей, элементы которой представляют совокупную полезность соответствующей ситуации для потребителей,  $\Delta_\Pi$  – матрица государственного воздействия на потребителей.

Воздействие на игроков со стороны государства не обязательно одинаковое  $\Delta_{\Pi ij} \neq \Delta_{Mij}$ , однако разумно предположить замкнутость системы, когда государство перераспределяет часть средств между монополистом и потребителями, т.е.

$$\sum_{i \in I, j \in J} \Delta_{Mij} = \sum_{i \in I, j \in J} \Delta_{\Pi ij}.$$

На данном этапе исследования важно сделать следующее допущение: государство действует только неценовыми рычагами, например налоговое регулирование, тарифное регулирование и др., но не непосредственное установление цены. Государство может воздействовать на платежные матрицы, корректировать их на выбранные  $\Delta_M$  и  $\Delta_\Pi$ , но не может полностью устранить конфликт между покупателем и монополистом путем задания цены на рынке. Рынок принимает вид двух платежных матриц (рис. 2).

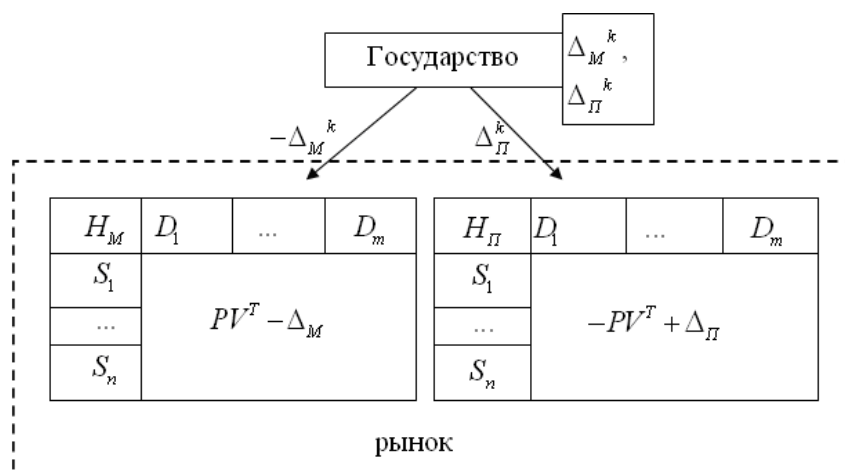


Рис. 2. Формализация рыночной ситуации

В реальных рыночных ситуациях, как правило, количество участников рынка достаточно большое, при этом существует огромное количество возможных стратегий каждого игрока. Применение модели адаптации государства к изменениям природы не имеет строгих ограничений, однако демонстрация использования модели была начата с идеализированной ситуации, чтобы количество возможных явлений природы было

наименьшим: рассматривался частный пример рынка с высокой монопольной властью – монопольно-монопсонический рынок [4], а затем произведено обобщение на классический монопольный рынок. В качестве примера монопольно-монопсонического рынка рассматривается рынок нефтяного попутного газа (НПГ) в Западносибирском регионе России [9]. Обобщение модели адаптивности государства на классический монопольный рынок производится на примере рынка односторонней монополии, образованный продавцом услуг по обеспечению газопроводов, в частности, Златоустовский филиал ООО «Челябинскрегионгаз».

### 3. Управление равновесием на монополизированном рынке

Для антагонистических игр принципы минимакса, максимина и равновесия совпадают (если они реализуемы). В таком случае они определяют единое понятие оптимальности и решения игры. В теории неантагонистических игр нет единого подхода к работе принципов оптимальности. Имеется множество принципов оптимальности, каждый из которых основывается на дополнительных предположениях о поведении игроков и структуре игры.

При моделировании монополизированного рынка эффективно использовать равновесие по Штакельбергу. Монополист – лидер, может объявлять цены, покупатель – ведомый. В чистых стратегиях поиск равновесия по Штакельбергу является простой задачей за исключением случаев, когда существует несколько вариантов с равными выигрышами для ведомого и разными для лидера. В таком случае прогнозирование поведения ведомого практически невозможно. Однако в рассматриваемом примере – исключить заведомо неэффективные для общества стратегии и варианты игры – является задачей государства и ограничение таких случаев входит в матрицу  $\Delta$ .

Ситуация  $(x_i, x_j) \in X_M \times X_\Pi$  называется М-равновесием по Штакельбергу в игре  $\Gamma$  двух лиц, а  $\overline{H}_M$  называется М-выигрышем, если

$$x_j(x_i) \in Z^M(x_i), \quad Z^M(x_i) = \left\{ \arg \max_{x_j} H_\Pi(x_i, x_j) \right\}$$

и выполняется равенство

$$\overline{H}_M = \max_{x_i} H_M(x_i, x_j(x_i)).$$

Понятие равновесия можно интерпретировать следующим образом. Игрок М (лидер) знает функции выигрыша обоих игроков, а следовательно и множество наилучших ответов игрока  $\Pi$  (ведомого) на любую стратегию игрока М. Тогда он, обладая этой информацией, максимизирует свой выигрыш, выбирает стратегию

$$x_{iM}^* = \arg \max_{x_i} H_M(x_i, x_j(x_i)).$$

Таким образом,  $\overline{H}_M$  – выигрыш М игрока, действующего оптимально в качестве лидера в игре  $\Gamma$ .

Необходимо отметить, что равновесие по Штакельбергу в чистых стратегиях не отражает некоторых особенностей моделирования монопольной рыночной ситуации, а именно не отражает доли мнения каждого отдельного покупателя в формировании совокупного спроса на товар и доли каждой технологии в общем технологическом процессе монополиста (монополист имеет возможность использовать в производстве не одну стратегию, а несколько одновременно в разных долях). Следовательно, наиболее близ-

кой к реальным условиям является нахождении равновесия по Штакельбергу в смешанных стратегиях.

В смешанных стратегиях выигрыш монополиста задается выражением  $q^T(PV^T - \Delta_M)s$ , а выигрыш потребителей  $q^T(-PV^T + \Delta_\Pi)s$ , где  $q$  – распределение общего объема производства (реализации) товара монополистом по различным технологиям, а  $s$  – распределение общего объема потребления товара потребителями по различным группам, обладающим различным спросом. Ситуация  $(q^*, s^*)$  называется ситуацией равновесия по Штакельбергу, если:

$$\begin{aligned} s^* &= \arg \max_s [q^{*T}(-PV^T + \Delta_\Pi)s], \\ s(q) &= \arg \max_s [q^T(-PV^T + \Delta_\Pi)s], \\ q^* &= \arg \max_q [q^T(PV^T - \Delta_M)s(q)]. \end{aligned}$$

Пусть на этапе  $k$  при игре монополиста и покупателей при заданной политике государства (т.е. матрицах  $\Delta_M^k, \Delta_\Pi^k$ ) на рынке в результате эволюции (т.е. достижения векторами  $P$  и  $V$  значений  $P^{(k)}$  и  $V^{(k)}$  соответственно) складывается ситуация равновесия  $(q^{(k)}, s^{(k)})$ , которая может перестать удовлетворять потребности государства по экономическим характеристикам или другим причинам. Стремление установить на рынке новое «справедливое» положение равновесия  $(q^{k*}, s^{k*})$  государство осуществляет за счет изменения матриц  $\Delta_M^k$  и  $\Delta_\Pi^k$  на  $\Delta_M^{k*}$  и  $\Delta_\Pi^{k*}$ .

Решение задачи построения матриц  $\Delta_M^{k*}$  и  $\Delta_\Pi^{k*}$  при которых реализуется равновесие  $(q^{k*}, s^{k*})$  в смешанных стратегиях нетривиальна. Построение таких матриц предлагается произвести следующим способом. Предварительно найдем в некоторых условных дискретных шкалах величин  $q$  и  $s$  корректирующие матрицы  $\Delta_M^{k*}(i^*, j^*)$  и  $\Delta_\Pi^{k*}(i^*, j^*)$ , приводящие ко всем возможным вариантам равновесия  $(i^*, j^*)$  в чистых стратегиях.

Вычисление скорректированных (желаемых) платежных матриц  $H_M = P^{(k)}V^{(k)T} - \Delta_M$  и  $H_\Pi = -P^{(k)}V^{(k)T} + \Delta_\Pi$ , при которых ситуация  $(i^*, j^*)$  есть равновесие по Штакельбергу в чистых стратегиях, сводится к построению таких матриц  $\Delta_M^{k*}(i^*, j^*)$  и  $\Delta_\Pi^{k*}(i^*, j^*)$ , что

$$\begin{aligned} (\forall i \in I, j \in J) \left\{ \begin{aligned} p(i)V(j) - \Delta_M(i, j) &\leq p(i^*)V(j) - \Delta_M(i^*, j), & (1) \\ -p(i)V(j) + \Delta_\Pi(i, j) &\leq -p(i)V(j^*) + \Delta_\Pi(i, j^*), & (2) \end{aligned} \right. \\ \sum_{i \in I, j \in J} \Delta_M(i, j) &= \sum_{i \in I, j \in J} \Delta_\Pi(i, j). & (3) \end{aligned}$$

Условия (1) и (2) отражают приемлемость стратегий  $i$  и  $j$  соответственно для монополиста и потребителей, условие (3) отражает замкнутость системы.

Разумной представляется политика, приводящая к минимальному перераспределению средств государством. В этом случае искомые матрицы можно найти как решение задачи линейного программирования

$$(\Delta_\Pi^{k*}, \Delta_M^{k*}) = \arg \min_{u, \Delta} \left\{ \sum_{i \in I, j \in J} (u_\Pi(i, j) + u_M(i, j)) \right\}, \quad (4)$$

где

$$(\forall i \in I, j \in J) (-u_M(i, j) \leq \Delta_M(i, j) \leq u_M(i, j), -u_\Pi(i, j) \leq \Delta_\Pi(i, j) \leq u_\Pi(i, j), u_\Pi, u_M \geq 0) \quad (5)$$

**Теорема 1.** Задача (1)–(5) имеет оптимальное решение.

*Доказательство.* Множество решений системы (1)–(3) не пусто, ее тривиальным решением является  $(\forall i \in I, j \in J)(\Delta_M(i, j) = \Delta_\Pi(i, j) = p(i)V(j))$ , что соответствует полному перераспределению средств государством.

Задача, двойственная задаче (1)–(5), имеет вид

$$\sum_{i \in I, j \in J} \left( t_\Pi(i, j) p(i) (v(j) - v(j^*)) + t_M(i, j) v(j) (-p(i) + p(i^*)) \right) \rightarrow \max_{r, s, t, u}, \quad (6)$$

$$(\forall i \in I, j \in J) \left\{ \left( -r_M(i, j) + s_M(i, j) - \sum_{i \in I, j \in J \setminus \{j^*\}} t_M(i, j) + u \leq 0 \right), r_M(i, j) - s_M(i, j) \leq 1, \right. \quad (7)$$

$$\left. \left( r_\Pi(i, j) - s_\Pi(i, j) - \sum_{i \in I \setminus \{i^*\}, j \in J} t_\Pi(i, j) - u \leq 0 \right), -r_\Pi(i, j) + s_\Pi(i, j) \leq 1, \right. \quad (8)$$

$$r, s, t, u \geq 0 \quad (9)$$

Легко заметить, что тривиальное нулевое решение  $r, s, t, u = 0$  является допустимым решением двойственной задачи. Так как задача является пряמודпустимой и двойственнодопустимой, то она имеет допустимое решение. Теорема доказана.

Матрицы  $\Delta_M^{k*}$  и  $\Delta_\Pi^{k*}$  можно построить как взвешенную сумму

$$\Delta_M^{k*} = \sum_{i \in I, j \in J} \Delta_M^{k*}(i, j) q_i^* s_j^*,$$

$$\Delta_\Pi^{k*} = \sum_{i \in I, j \in J} \Delta_\Pi^{k*}(i, j) q_i^* s_j^*.$$

Нужно отметить, что, несмотря на оптимальность матриц  $\Delta_M^{k*}$  и  $\Delta_\Pi^{k*}$ , переход к ним не всегда эффективен при конкретных внешних условиях. Например, в случае, если переход к новой стратегии требует слишком высоких финансовых затрат или длительной временной задержки, которая повлечет критическую ситуацию на рынке. Такие ситуации не исключаются, так как кроме замкнутости системы на  $\Delta_M^{k+1}$  и  $\Delta_\Pi^{k+1}$ , другие ограничения в условиях рассматриваемой задачи не накладываются. Именно поэтому возникает необходимость вводить пошаговый алгоритм решения задачи и оценивать адаптивность государства к изменению ситуаций на рынках с высокой степенью монополизации. Под адаптивностью понимается основное свойство, обеспечивающее устойчивость деятельности и способность системы управления не только реагировать на внешние изменения, но также предвидеть их.

## Заключение

Управление равновесием по Штакельбергу в модели монополизированной экономики является составляющей исследования по проблеме анализа эффективности адаптации инструментов государственного регулирования к изменениям ситуаций на рынках с высокой степенью монополизации.

На данном этапе наряду с рассмотренным в работе вопросом основными результатами исследования стало:

– построение системы показателей для оценки степени адаптивности государства [2, 4];



- разработка программа для автоматизированного их расчета [7];
- проведение вычислительного эксперимента;
- разработка статистического инструментария для определения ряда трудно оцениваемых показателей [3].

Работа над поднятой проблемой может дать значительные практические результаты в разработке налоговой и тарифной политики государства. Например, по результатам адаптивной оценки затрат временных ресурсов возможно дать рекомендации по оптимизации срока пересмотра количественных параметров тарифного регулирования по отраслям, налогового законодательства, антимонопольного, земельного законодательства и др. и указать общее направление этих изменений [5].

*Исследование выполнено при поддержке Министерства образования и науки Российской Федерации, соглашение 14.В37.21.0395.*

## Литература

1. Ключков, В.В. Методы анализа адаптивности производственных программ и организационных структур предприятий на примере воздушного транспорта и авиастроения / В.В. Ключков, Д.И. Сазонов // Экономика и математические методы. – М.: Наука, 2007. – Т. 43, № 2. – С. 44–56.
2. Коновалова, Е.Д. Анализ эффективности адаптации инструментов государственного регулирования к изменениям ситуаций на рынках с высокой степенью монополизации / Е.Д. Коновалова // Математическое и статистическое исследование социально-экономических процессов: сборник научных трудов / под ред. А.В. Панюкова. – Челябинск: Издательский центр ЮУрГУ, 2011. – Вып. 3. – С. 5–12.
3. Коновалова, Е.Д. Разработка статистического инструментария для проведения экспертного опроса / Е.Д. Коновалова // Статистика. Моделирование. Оптимизация: сборник трудов Всероссийской конференции (Челябинск, 28 ноября – 3 декабря 2011 г.). – Челябинск: Издательский центр ЮУрГУ, 2011. – С. 311–315.
4. Мальцева, Е.Д. Принципы адаптации государства к регулированию степени монополизации рынка в условиях несовершенной конкуренции / Е.Д. Мальцева // Математическое и статистическое исследование социально-экономических процессов: сборник научных трудов / под ред. А.В. Панюкова. – Челябинск: Издательский центр ЮУрГУ, 2009. – Вып. 2. – С. 36–51.
5. Мальцева, Е.Д. Методика прогнозирования налоговых поступлений в Федеральный бюджет РФ в условиях экономического кризиса / Е.Д. Мальцева // Современная статистика в диалоге с обществом: Сборник материалов интернет-конференции / Челябинскстат, Институт социально-экономических и региональных проблем Челябинского государственного университета. – Челябинск: Челстат, 2009. – С. 31–36.
6. Мау, В.А. Логика российской модернизации: исторические тезисы и современные вызовы. Газета.Ru. Комментарии. / В.А. Мау // URL: [http://www.gazeta.ru/comments/2005/07/04\\_a\\_309315.shtml](http://www.gazeta.ru/comments/2005/07/04_a_309315.shtml) (дата обращения: 5.07.2005).
7. Панюков, А.В. Анализ эффективности адаптивности государственного регулирования к изменениям ситуаций на рынках с высокой степенью монополизации / А.В. Панюков, Е.Д. Коновалова // Вестник Пермского университета. Серия «Экономика». – Пермь, 2012. – Специальный выпуск – С. 58–68.

8. Степанов, Л.В. Моделирование конкуренции в условиях рынка / Л.В. Степанов – Издательство «Академия Естествознания», 2009. – 115 с.
9. Чернавский, С.Я. Экономический анализ либерализации рынка природного газа в России / С.Я. Чернавский, О.А. Эйсмонт // Центральный экономико-математический институт РАН. Лаборатория экономических проблем энергетики. – 2008. – № 4(43). – С. 63–75.

Панюков Анатолий Васильевич, д.ф.м.н, профессор, заведующий кафедрой экономико-математических методов и статистики, Южно-Уральский государственный университет (Челябинск, Российская Федерация), a\_panyukov@mail.ru.

Коновалова Екатерина Дмитриевна, аспирант кафедры экономико-математических методов и статистики, Южно-Уральский государственный университет (Челябинск, Российская Федерация), konovalova\_ekaterina@bk.ru.

---

## THE CONTROL OF STACKELBERG EQUILIBRIUM IN PROBLEMS OF GOVERNMENT CONTROL OF NATURAL MONOPOLY

*A.V. Panyukov*, South Ural State University (Chelyabinsk, Russian Federation),  
*E.D. Konovalova*, South Ural State University (Chelyabinsk, Russian Federation)

Aim of this paper is the increasing of effectiveness of government control methodology of natural monopoly. Only non-price methods are considered as government control methods. The effectiveness of the control system and the rate of response to external changes depend on adaptability. Solution of the problem government adaptation to change market situation, is cyclical. Two problems: (1) to build a model interaction monopolist and buyer at the specified corrective government control methods (the market is seen as a loop system), (2) to build a model of interaction government and this market and estimate the government adaptability characteristics are solved for each step. It is demonstrated that Stackelberg equilibrium in non-cooperative two-person games in mixed strategies solves this problem. At the same time, rate, efficiency, optimal selection corrective government strategy and time selection define the government adaptability.

*Keywords: natural monopoly, adaptability, noncooperative game, Stackelberg equilibrium, mixed strategy.*

## References

1. Klochkov V.V., Sazonov D.I. Metodi analiza adaptivnosti proizvodstvennih program i organizatsionnih struktur predpriyatii na primere vozdušnogo transporta i aviastroeniya [Adaptability Analysis Methods of Production Programs and Business Structure Organizational on the Example of Air Transport and Aviation]. *Ekonomika i matematicheskie metodi* No. 2 [Economics and Mathematical Methods No. 2]. Moscow, Science, 2007. P. 44–56.
2. Konovalova E.D. Analiz effektivnosti adaptatsii instrumentov gosudarstvennogo regulirovaniya k izmeneniyam situatsii na rinkah s visokoi stepenyu monopolizatsii [Performance Analysis of Government Control Methods Adaptability at Temporal High-rate Monopolization Market State Changes]. *Matematicheskoe i statisticheskoe issledovanie socialno–*

- ekonomicheskikh processov: Sbornik nauchnih trudov No. 3 [Mathematical and Statistical Study of the Socio-economic Processes]. Chelyabinsk, Publishing of the South Ural State University, 2011. P. 5–12.
3. Konovalova E.D. Razrabotka statisticheskogo instrumentariya dlya provadeniya ekspertnogo oprosa [The Development of Statistical Tools for the Expert Survey]. Statistika. Modelirovanie. Optimizaciya: Sbornik trudov vserossiyskoi konferencii (Chelyabinsk, 28 noyabrya - 3 dekabrya 2011) [Statistics. Modeling. Optimization: Proceedings of the National Conference] Chelyabinsk, Publishing of the South Ural State University, 2011. P. 311–315.
  4. Maltseva E.D. Principi adaptacii gosudarstva k regulirovaniyu stepeni monopolizacii rinka v usloviyah nesovershennoy konkurencii [Principles of Government Adaptability to regulate the market monopolization degree in conditions of imperfect competition]. Matematicheskoe i statisticheskoe issledovanie socialno-ekonomicheskikh processov: Sbornik nauchnih trudov No. 2 [Mathematical and Statistical Study of the Socio-economic Processes]. Chelyabinsk, Publishing of the South Ural State University, 2009. P. 36–51.
  5. Maltseva E.D. Metodika prognozirovaniya nalogovih postuplenii v federalniy byudjet RF v usloviyah ekonomicheskogo krizisa [The Method of Predicting Tax Revenues to the Federal Budget of the Russian Federation in the Context of the Economic Crisis]. Sovremennaya statistika v dialogue s obshestvom: Sbornik materialov internet-konferencii [Modern Statistics in Dialogue with Society]. Chelyabinsk, Chelstat, 2009. P. 31–36.
  6. Mau V.A. Logika rossiiskoi modernizatsii: istoricheskie tezisi i sovremennye vizovi. Gazeta.Ru. Kommentarii [The Logic of Russia's Modernization: Historical Abstracts and Contemporary Challenges. Gazeta.Ru. Comments]. URL: [http://www.gazeta.ru/comments/2005/07/04\\_a\\_309315.shtml](http://www.gazeta.ru/comments/2005/07/04_a_309315.shtml) (accessed: 5.07.2005).
  7. Panyukov A.V., Konovalova E.D. Analiz effektivnosti adaptativnosti gosudarstvennogo regulirovaniya k izmeneniyam situatsii na rinkah s visokoi stepenyu monopolizatsii [Performance Analysis of Government Control Adaptability at Temporal High-rate Monopolization Market State Changes]. Vestnik Permskogo universiteta, seriya «Ekonomika» [Perm University Herald Special Issue Economy]. Perm, Publishing of the Perm University, 2012. P. 58–68.
  8. Stepanov L.V. Modelirovanie konkurencii v usloviyah rinka [Modeling of the Competition in the Market]. Publishing of the Academy of Natural Science, 2009. 115 p.
  9. Chernavskiy S.Y., Eysmont O.A. Ekonomicheskii analiz liberalizacii rinka prirodnogo gaza v Rossii [Economic Analysis of the Liberalization of Natural Gas market in Russia]. Laboratoriya ekonomicheskikh problem energetiki: Sbornik nauchnih trudov No. 4 [Laboratory of the Energetics' Economic Problems]. Central Economics and Mathematics Institute RAN, 2008. P. 63–75.

*Поступила в редакцию 27 марта 2013 г.*

# KERNELGEN — ПРОТОТИП РАСПАРАЛЛЕЛИВАЮЩЕГО КОМПИЛЯТОРА C/FORTRAN ДЛЯ GPU NVIDIA НА ОСНОВЕ ТЕХНОЛОГИЙ LLVM<sup>1</sup>

*Н.Н. Лихогруд, Д.Н. Микушин*

Проект KernelGen (<http://kernelgen.org/>) имеет цель создать на основе современных открытых технологий компилятор Fortran и C для автоматического портирования приложений на GPU без модификации их исходного кода. Анализ параллелизма в KernelGen основан на инфраструктуре LLVM/Polly и CLoG, модифицированной для генерации GPU-ядер и alias-анализе времени исполнения. PTX-ассемблер для GPU NVIDIA генерируется с помощью бекенда NVPTX. Благодаря интеграции LLVM-части с GCC с помощью плагина DragonEgg и модифицированного компоновщика, KernelGen способен, при полной совместимости с компилятором GCC, генерировать исполняемые модули, содержащие одновременно CPU- и GPU-варианты машинного кода. В сравнительных тестах с OpenACC-компилятором PGI KernelGen демонстрирует большую гибкость по ряду возможностей, обеспечивая при этом сравнимый или до 60 % более высокий уровень производительности.

*Ключевые слова:* GPU, LLVM, OpenACC, JIT-компиляция, выпуклый анализ.

## Введение

Широкое использование GPU в кластерных вычислительных системах требует массовой адаптации множества сложных приложений. Программные модели CUDA и OpenCL достаточно хорошо подходят для небольших программ с ярко выраженным вычислительным ядром. Однако для сложных приложений, состоящих из множества отдельных блоков, таких как математические модели, сложность настройки взаимодействия оригинального кода и кода для GPU многократно возрастает. Многие компании и научные группы по-прежнему откладывают портирование своих приложений, так как это приводит к возрастанию издержек на сопровождение и развитие нескольких различных версий одной и той же функциональности для CPU и GPU. Увеличить эффективность портирования призваны следующие типы технологий разработки:

- **Директивные расширения существующих языков высокого уровня с ручным управлением параллелизмом, по аналогии с OpenMP.** Данный тип технологий расширяет языки специальными директивами, с помощью которых пользователь может пометить части кода, предназначенные для выполнения на GPU. Основываясь на данной информации, компилятор автоматически генерирует гибридный исполняемый файл. Для стандартизации набора директив в языках C/C++/Fortran коммерческими разработчиками подобных решений созданы консорциумы OpenACC [7] и OpenHMP [8]. Аналогичный набор директив, но уже для ускорителей архитектуры Many Integrated Core (MIC) развивается компанией Intel [9]. В рамках систем F2C-ACC [10] и САПФОР [11] предложены наборы директив для преобразования исходного кода на языке Fortran в гибридную форму, причем САПФОР проводит распараллеливание как на уровне GPU, так и на уровне нескольких GPU-узлов MPI-кластера.

<sup>1</sup>Статья рекомендована к публикации программным комитетом Международной научной конференции «Параллельные вычислительные технологии - 2013».

В целом, несмотря на большую гибкость, директивные расширения все же требуют значительного участия программиста в организации корректных и эффективных вычислений. Компиляторы некоторых из приведенных директивных расширений реализуют проверку параллельности циклов и непротиворечивости директив, в других такая проверка отсутствует. Часто возникают ситуации, в которых компилятор слишком «осторожен» при принятии решений на основе внутреннего анализе циклов и производит распараллеливание только при наличии дополнительных указаний от пользователя. Большинство директивных расширений не поддерживают генерацию GPU-ядер для циклов, в которых присутствуют вызовы функций из других модулей компиляции или библиотек, что существенно ограничивает применимость подобных технологий в больших проектах.

- **Специализированные языки (domain-specific languages, DSL), со встроенными средствами параллелизма, ориентированные на определенный класс задач.** В последние годы было предложено множество различных DSL- и Embedded DSL-языков со встроенными средствами параллелизма. Их основная идея состоит в том, чтобы приблизить средства языка к характерным объектам и действиям задачи, в то же время исключив из языка конструкции, привязывающие реализацию к конкретной архитектуре. Обработка возникающего нового уровня абстракции производится компилятором или source-to-source процессором, генерирующим код для всех целевых архитектур. Так, в работе [1] предложен Си-подобный DSL-язык для выражения вычислений на сетках с учетом начальных и граничных условий, а также соответствующий компилятор с бекендами для различных CPU (SSE, AVX). В работе [4] аналогичная задача решается при помощи eDSL, основанного на шаблонах C++. Поддерживается генерация кода для CPU и GPU NVIDIA. Другой eDSL на основе C++ – Halide [5], с поддержкой x86-64/SSE, ARM v7/NEON и GPU NVIDIA, нацелен на эффективную реализацию методов обработки изображений. К классу DSL/eDSL можно отнести систему Nemerle Unified Device Architecture (NUDA) [6], позволяющую создавать новые расширения языка Nemerle и соответствующие плагины для компилятора. Тестирование DSL/eDSL как правило проводится в сравнении с программами, написанными вручную, что не позволяет судить о том, насколько существенен может быть выигрыш в эффективности DSL-языков по сравнению с директивными расширениями. Кроме того специализация ограничивает конкурентную среду, так как у каждого языка как правило существует только один разработчик. Глубокое сравнительное тестирование затруднено необходимостью реализации бенчмарков на каждом используемом языке.
- **Автоматический анализ параллельности кода с помощью эвристик или методов многогранного анализа.** Технологии данного типа предназначены для вычисления зависимостей данных и пространств итераций с помощью точных методов или эвристик. Эвристики в настоящее время являются частью большинства коммерческих компиляторов, когда как в составе открытых и экспериментальных решений можно найти более сложные методы, например, многогранный анализ (polyhedral analysis). В работе [12] для компилятора GCC реализовано расширение для автоматической идентификации параллельных циклов и генерации для них кода на OpenCL. Аналогичное расширение PPCG для компилятора Clang (LLVM) [13] способно преобразовывать код на C/C++ в CUDA-ядра. Обе технологии преобразуют вычислитель-

ные циклы из внутреннего представления компилятора в код на OpenCL или CUDA при помощи системы многогранного анализа Chunky Loop Generator (CLoopG) [14]. Source-to-source компилятор Par4all [15] преобразует код на языке C или Fortran в код CUDA, OpenCL или OpenMP с помощью системы многогранного анализа PIPS.

Явное программирование на CUDA, директивные расширения и DSL-языки в любом случае предполагают модификацию или переработку исходного кода программы. По этой причине портирование больших приложений на GPU с помощью этих технологий сильно затруднено. Если же приложение портировано лишь частично, то синхронизация данных между хостом и GPU может значительно влиять на общую производительность. Так, при портировании только одного блока WSM5 модели WRF с помощью директив PGI Accelerator, время обменов данными составляет 40–60 % общего времени [20].

На основе сопоставления свойств существующих технологий с требованиями, возникающими при портировании на GPU типичного вычислительного приложения, можно выделить ряд возможностей, имеющих потенциально наиболее важную роль при планировании и разработке программных систем следующего поколения:

- Поддержка широкого множества *существующих* языков программирования;
- Автоматическая оценка параллельности вычислительных циклов, не требующая внесения изменений в исходный код или каких-либо дополнительных действий со стороны пользователя;
- Генерация кода, полностью совместимая со стандартной хост-компиляцией;
- Минимизация обмена данными между памятью системы и GPU;
- Встраивание в существующие схемы распараллеливания, в первую очередь – MPI.

Целью проекта KernelGen является создание компилятора, удовлетворяющего всем перечисленным условиям и проработка стратегии развития необходимых для этого технологий. Очевидно, что подобная система не может быть построена ни на основе директивных расширений, ни на основе DSL, в то же время в ней вполне могли бы быть использованы наработки исследовательских решений по автоматическому анализу циклов.

Данная статья организована следующим образом. В разделе 1 предлагаются решения по организации процессов компиляции, компоновки и генерации кода, а также нестандартная модель исполнения, позволяющая естественным образом обеспечить более эффективное взаимодействие параллельных частей кода на GPU. В разделе 2 излагается способ модификации существующей технологии анализа параллельности циклов для генерации GPU-кода. Разделы 3 и 4 посвящены, соответственно, необходимым дополнительным подсистемам исполнения приложений и сравнительному анализу работы тестовых задач.

## 1. Этапы преобразования кода

При разработке системы компиляции на основе существующих наработок значительную роль играет выбор наиболее подходящей базовой инфраструктуры по большому числу критериев: наличие фронтендов для различных языков, полнота и гибкость внутреннего представления, существование базового набора оптимизирующих преобразований и эффективных бекендов для целевых архитектур, динамика развития и поддержка со стороны сообщества разработчиков. Наиболее развиты по этим критериям компиляторы GCC, LLVM и Open64. Компилятор GCC поддерживает наибольшее число языков программирования, но не имеет бекендов для GPU, тогда как LLVM и Open64 имеют бекенды для

NVIDIA PTX ISA. Компилятор Open64 имеет фронтенды для C, C++ и Fortran, генерирует качественный код, но при этом, к сожалению, имеет сильно сегментированное сообщество разработчиков, развивающих множество отдельных веток кода в интересах коммерческих компаний и исследовательских организаций. Компилятор LLVM не имеет собственного фронтенда для языка Fortran, но способен при помощи плагина DragonEgg [16] использовать фронтенды компилятора GCC. При этом он имеет собственный GPU-бекенд NVPTX, имеет простое внутреннее представление (LLVM IR – intermediate language) и развивается намного более интенсивно, чем GCC и Open64. Из этих соображений, за основу для KernelGen был выбран LLVM.

Компилятор KernelGen работает напрямую с оригинальным приложением, не требуя каких-либо изменений ни в исходном коде, ни в системе сборки. За счет использования фронтенда незначительно модифицированной версии GCC, он полностью совместим с его опциями, что гарантирует высокий уровень поддержки большого числа приложений. Чтобы обеспечить стандартный процесс сборки, в KernelGen используется схема, напоминающая LTO (link time optimization – инфраструктура компилятора для дополнительной оптимизации кода во время компоновки): код для GPU сначала добавляется в отдельную секцию объектных файлов, затем объединяется и снова разделяется на отдельные ядра на этапе компоновки. Окончательная компиляция GPU-ядер в ассемблер происходит при необходимости, уже во время работы приложения (JIT, just-in-time compilation). Схема основных этапов преобразования кода приведена на рис. 1.

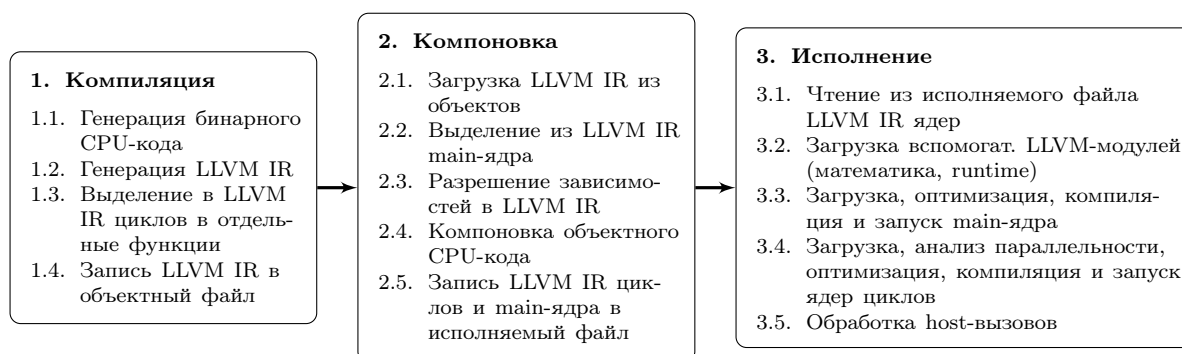


Рис. 1. Этапы преобразования кода компилятором KernelGen

В результате работы компилятора, исходное приложение преобразуется во множество GPU-ядер: одно или несколько *основных* ядер и множество *вычислительных* ядер. Основные ядра исполняются на GPU в одном потоке. Их задача – хранить данные, исполнять небольшие последовательные участки кода и производить вызовы вычислительных ядер и отдельных CPU-функций, которые невозможно или неэффективно переносить на GPU. Вычислительные ядра исполняются на GPU множеством параллельных нитей с полной загрузкой мультипроцессоров. Таким образом, максимальная доля кода выполняется на GPU, а CPU лишь координирует исполнение. В частности, при работе MPI-приложения каждый рабочий процесс в данном случае будет представлять собой GPU-ядро с небольшим числом CPU-вызовов MPI. Использование MPI дополнительно облегчается за счет поддержки GPU-адресов в командах обмена данными [18]. В целом, такая модель исполнения имеет много общего с native-режимом Intel MIC, но работает на GPU, где скалярные вычислительные блоки способны достигать высокой эффективности без необходимости векторизации.

## 1.1. Компиляция

При компиляции отдельных объектов, генерируется как x86-ассемблер (таким образом, приложение по-прежнему работоспособно при отсутствии GPU), так и представление LLVM IR. Для разбора исходного кода используется компилятор GCC, чье внутреннее представление *gimple* преобразуется в LLVM IR с помощью плагина DragonEgg. Затем в IR-коде производится выделение тел циклов в отдельные функции, вызываемые через универсальный интерфейс *kernelgen\_launch* (рис. 2), где *kernel* – имя или адрес функции (вместо имен в начале работы программы подставляются адреса), *data* – структура, агрегирующая аргументы вызова, *szdata* и *szdatai* – размер списка аргументов и списка целочисленных аргументов (последний используется для вычисления хеша функции и поиска ранее скомпилированных ядер во время исполнения).

```
__device__ int kernelgen_launch(
    unsigned char* kernel, unsigned long long szdata,
    unsigned long long szdatai, unsigned int* data);
```

Рис. 2. Интерфейс вызова функций-циклов

Стандартный механизм выделения каскадов вложенных циклов в функции LLVM *LoopExtractor* расширен, так чтобы цикл не заменялся, а дополнялся вызовом функции по условию, как показано на рис. 3. С помощью данного условия runtime-библиотека KernelGen может переключать выполнение между различными версиями цикла. Например, если цикл определен как непараллельный, то *kernelgen\_launch* возвращает -1, и код цикла начинает выполняться основным ядром в последовательном режиме. Тем не менее, данный цикл может содержать вложенные параллельные циклы, обработка которых будет проведена аналогичным образом. В конце концов, если весь каскад тесно вложенных циклов непараллелен, несовместим (например, содержит вызовы внешних CPU-функций) или оценен как неэффективный для GPU, то вся функция выгружается для работы на хосте с помощью вызова *kernelgen\_hostcall* (рис. 4), при котором GPU-приложение останавливает свою работу и передает данные и адрес функции для выполнения на CPU. Функции *kernelgen\_launch* и *kernelgen\_hostcall* работают в GPU-ядре и вызывают остановку его выполнения. После завершения работы другого ядра или CPU-функции, основное ядро продолжает работу. Хост-часть управляющих функций компилирует и выполняет заданную функцию с помощью интерфейса FFI (Foreign Function Interface).

```
if (kernelgen_launch(kernel, szdata, szdatai, data) == -1) {
    // Запустить оригинальный код цикла.
}
```

Рис. 3. Переключение версий кода цикла между функцией-ядром и оригинальным кодом

```
__device__ void kernelgen_hostcall(
    unsigned char* kernel, unsigned long long szdata,
    unsigned long long szdatai, unsigned int* data);
```

Рис. 4. Интерфейс хост-вызова



Одним из специфических свойств KernelGen является хранение всех данных приложения в памяти GPU. Для того чтобы обеспечить его совместимость с наличием CPU-вызовов, реализована простая система синхронизации памяти. При попытке CPU-функции обратиться к памяти по адресу из диапазона GPU возникающий сигнал сегментации обрабатывается дублированием страниц из памяти GPU в страницы CPU-памяти, расположенные по тем же адресам. После завершения работы CPU-функции, измененные CPU-страницы синхронизируют изменения с памятью GPU.

## 1.2. Компоновка

Во время компоновки отдельных объектов в приложение или библиотеку, LLVM IR также линкуется в один общий IR-модуль для main-ядра и по одному IR-модулю на каждый вычислительный цикл. IR-код погружается в исполняемый файл и в дальнейшем оптимизируется и компилируется в GPU код по мере необходимости во время работы приложения.

Специальной обработки требуют глобальные переменные. Синхронизация глобальных переменных между ядрами потребовала бы разработки для GPU динамического компоновщика. Вместо этого, в начале работы программы на CPU передаются адреса всех глобальных переменных. Во время выполнения, виртуальные глобальные переменные заменяются на соответствующие фактические адреса. Это корректно, т.к. в LLVM глобальная переменная реализована как указатель на память, содержащую ее логическое значение.

## 1.3. Модель исполнения

Основное ядро запускается в самом начале выполнения приложения и работает на GPU постоянно. Во время работы вычислительного ядра или CPU-функции основное ядро переходит в состояние активного ожидания и продолжает работу после завершения внешнего вызова. Для реализации данной схемы GPU должно поддерживать одновременное исполнение нескольких ядер (concurrent kernel execution) или временную выгрузку активного ядра (kernel preemption). Одновременное исполнение ядер доступно в GPU NVIDIA, начиная с Compute Capability 2.0, в GPU AMD такой возможности нет, но есть вероятность появления kernel preemption в одной из следующих версий OpenCL. По этой причине в данный момент KernelGen работает только с CUDA.

Вызовы *kernelgen\_launch* и *kernelgen\_hostcall* состоят из двух частей: device-функции на GPU и одноименного вызова в CPU-коде, который выполняет, соответственно, окончательную генерацию кода и запуск вычислительного ядра или загрузку данных с GPU и запуск CPU-функции средствами Foreign Function Interface (FFI). Взаимодействие между частями может быть организовано посредством глобальной памяти GPU или pinned-памяти хоста. Однако для гарантированной передачи корректного значения необходимо обеспечить *атомарный* режим операций чтения и записи, доступность которого является определяющим фактором. По этой причине был реализован метод, использующий глобальную память.

На GPU Kepler K20 запуск ранее скомпилированных ядер может быть проведен без взаимодействия с CPU, непосредственно из основного ядра, с помощью технологии динамического параллелизма.

Дополнительное препятствие взаимодействию GPU-ядра с другим ядром или CPU состоит в том, что данные нити (CUDA thread) хранятся в регистрах или локальной памяти. Это означает, что аргументы, переданные из основного GPU-ядра не могут быть использо-

ваны где-либо, кроме как в нем самом. Для преодоления этого ограничения, бекенд NVPTX изменен так, чтобы локальные переменные помещаются не в *.local*-секцию, а в *.global*, делая их доступными всем GPU-ядрам и хосту.

## 2. Генерация CUDA-ядер для параллельных циклов

Частью инфраструктуры LLVM является библиотека Polly [2] (от polyhedral analysis – многогранный анализ) – оптимизирующее преобразование циклов, основанное на CLooG. Оно способно распознавать параллельные циклы в IR-коде, оптимизировать кеширование за счет добавления блочности, оптимизировать доступ к памяти за счет перестановки циклов и генерировать код, использующий OpenMP. Для заданного кода CLooG строит *абстрактное синтаксическое дерево* (AST), а затем проводит расщепление циклов по некоторым измерениям. Благодаря возможности расщепления частично-параллельных измерений, для исходного цикла может быть найдено эквивалентное представление из одного или нескольких циклов, часть которых параллельна. Подобный подход используется довольно редко, большинство современных компиляторов ограничиваются проверкой параллельности измерений существующих циклов без глубокого анализа.

Polly работает с частями программы, для которых можно статически (без выполнения) предсказать поток управления и доступ в память в зависимости от фиксированного набора параметров. Такие части принято называть *статическими частями потока управления* (static control parts – SCoPs). Часть программы представляет собой SCoP при выполнении следующих условий:

1. Поток управления формируется условными операторами и циклами-счетчиками:
  - (a) каждый цикл-счетчик имеет одну индексную переменную с константным шагом изменения, верхняя и нижняя границы цикла заданы афинными выражениями, зависящими от параметров и индексных переменных внешних циклов;
  - (b) условные операторы сравнивают значения двух афинных выражений, зависящих от параметров SCoP и индексных переменных.
2. Обращения в память происходят со смещениями от указателей-параметров SCoP. Смещения задаются афинными выражениями от параметров SCoP и индексных переменных циклов.
3. Содержит вызовы только функций без побочных эффектов.

Первое условие означает структурированность потока управления: код можно логически разбить на иерархию вложенных блоков, имеющих один вход и один выход, каждый блок полностью вложен в объемлющий. Запрещены конструкции, нарушающие структуру потока управления (*break*, *goto*). Использование афинных выражений позволяет применять аппарат целочисленного программирования для расчета границ циклов и обращений в память в зависимости от параметров SCoP.

Если работать с высокоуровневым представлением программы (код на языке высокого уровня, абстрактное синтаксическое дерево), то множество приемов программирования (например, арифметика указателей, циклы *while*, операторы *goto*) будут нарушать описанные требования. Если же перейти к промежуточному, близкому к ассемблеру представлению, каковым является LLVM IR, то арифметика указателей будет реализована как набор арифметических операций с регистрами, и любой цикл, вне зависимости от типа (*for*, *while*),

будет реализован как обычный условный переход. Следствием работы Polly с LLVM IR являются две полезные возможности:

- распараллеливать *while*-циклы, когда как, например, стандартом OpenACC такая возможность не предусмотрена;
- распараллеливать циклы с адресной арифметикой, что, например, не поддерживается в PGI OpenACC.

При адаптации Polly для получения GPU-ядер был использован существующий генератор кода OpenMP, работающий следующим образом. Если внешний цикл является параллельным, то его содержимое перемещается в отдельную функцию, с добавлением вызовов функций библиотеки libgomp – GNU реализации OpenMP. При этом распределение итераций по ядрам производит среда исполнения, а распараллеливается только самый внешний цикл. Для KernelGen в эту логику были внесены следующие изменения:

1. Отображение пространства итераций на нити GPU, с учетом необходимости объединения запросов в память нитей варпа (coalescing transaction);
2. Рекурсивная обработка вложенных циклов с целью использования возможностей GPU по созданию многомерных сеток нитей.

Пусть в заданной группе циклов можно распараллелить  $N$  тесно-вложенных циклов. Тогда ядро может быть запущено на решетке с числом измерений  $N$  (для CUDA  $N \leq 3$ ). Для каждого измерения, распределяемого между нитями GPU, генерируется код, рассчитывающий положение нити в блоке и блока в сетке. Каждому параллельному циклу ставится во взаимно однозначное соответствие измерение решетки, причем в обратном порядке – внутреннему циклу соответствует измерение  $X$  (это позволяет объединять запросы в память). Для каждого параллельного цикла генерируется код, определяющий нижнюю и верхнюю границы части пространства итераций, которая должна быть выполнена нитью. Затем генерируется последовательный код цикла с измененными границами и шагом.

Схема этапов работы Polly, анализа и генерации кода приведена на рис. 5.

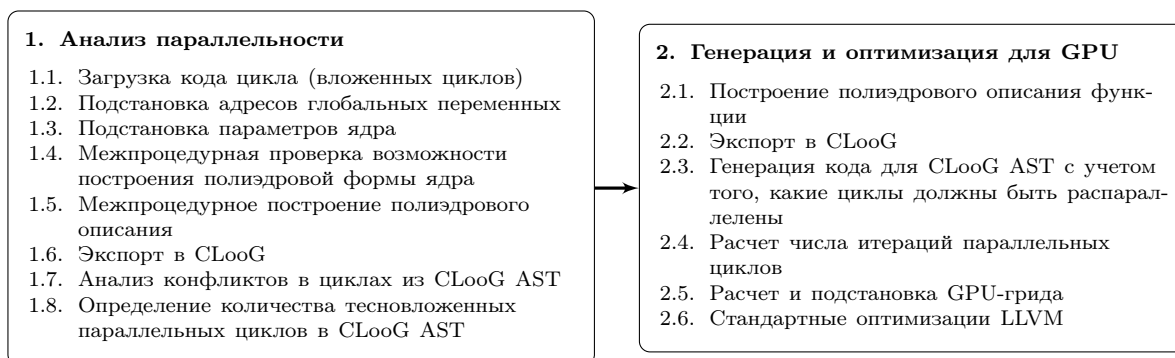


Рис. 5. Этапы генерации CUDA-ядер для параллельных циклов компилятором KernelGen. Оптимизация происходит сразу для всего SCoP, генерация – для каждой функции в отдельности

### 3. Дополнительные средства времени исполнения

Включение бекенда NVPTX для генерации GPU-кода в LLVM позиционировалось компанией NVIDIA как «открытие компилятора». Тем не менее, помимо того, что закрытым остается C/C++/CUDA-фронтенд, а clang обладает лишь минимальной поддержкой неко-

торых ключевых слов CUDA, недоступной также остается часть компилятора существенно важная для его применения в LLVM: библиотека математических функций C99. Поскольку в рамках закрытого компилятора CUDA эти функции реализованы в виде заголовочных файлов C/C++, их использование с другими языками на уровне LLVM невозможно, и пользователю NVPTX-бекенда доступны только функции, встроенные в аппаратуру (builtins), среди которых, например, нет точных версий функций `sin`, `cos`, `pow`. В KernelGen данная проблема решена путем конвертации заголовочных файлов в LLVM IR одним из двух способов: с помощью `clang` (требуется множество модификаций, полученный IR-код предположительно содержит некорректные части) или с помощью `siscc` (вызывается из `nvcc`). В последнем случае, IR-модуль можно получить, отправив на компиляцию пустой `.cu`-файл и выгрузив код IR-модуля из `siscc` с помощью отладчика. IR, сгенерированный `siscc` совместим с актуальной версией LLVM и позволяет производить компоновку математической библиотеки и использующего ее приложения на уровне IR-кода, вне зависимости от начального языка. В частности, благодаря этому KernelGen позволяет генерировать GPU-код для программ на языке Fortran, использующих любые стандартные математические функции.

Некоторые типы функций CUDA API, такие как выделение GPU-памяти и загрузка другого CUDA-модуля, всегда приводят к неявной синхронизации асинхронных операций. Поскольку схема работы KernelGen требует постоянного поддержания основного ядра в состоянии выполнения, необходимость выделения памяти или загрузки новых ядер в процессе его работы приведет к блокировке. В этом отношении существующие версии CUDA создают для развития KernelGen определенные препятствия, вынуждая реализовывать нестандартные эквиваленты базовой функциональности.

Если синхронность выделения памяти на GPU со стороны хоста еще можно считать разумным ограничением, то подтвержденная экспериментами синхронность вызовов `malloc` внутри GPU-ядер явно избыточна, так как память для индивидуальных потоков выделяется заранее. Так как оба стандартных варианта не могут быть использованы, KernelGen выполняет начальную преаллокацию памяти для собственного динамического пула и управляет его работой.

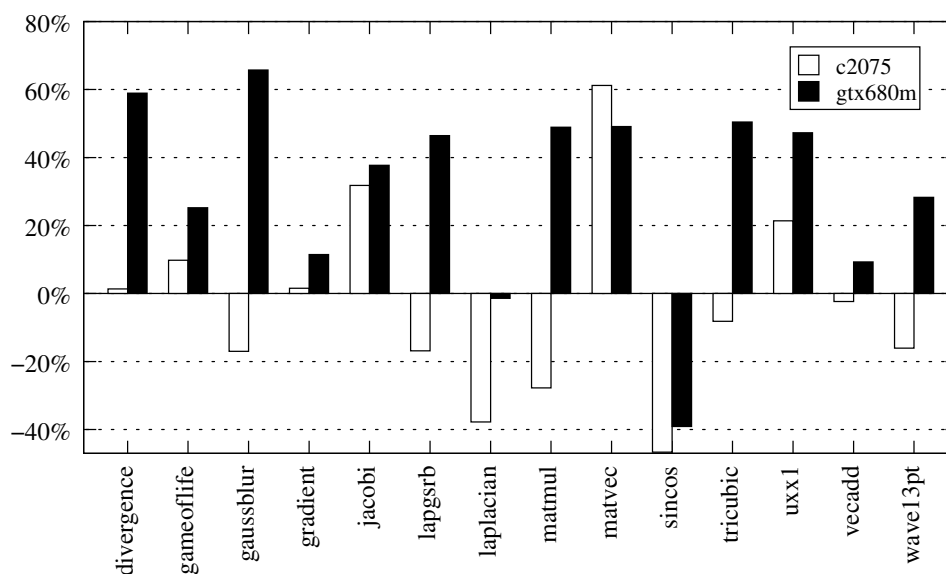
JIT-компиляция вычислительных ядер предполагает, что вновь скомпилированные GPU-ядра будут загружаться на GPU в фоне работающего основного ядра. Обычно динамическую загрузку ядер можно произвести с помощью стандартных функций `cuModuleLoad` и `cuModuleGetFunction` CUDA Driver API. Однако, обе эти функции являются синхронными, предположительно, из-за неявного выделения памяти для хранения кода и статических данных. В данной ситуации при разработке KernelGen не оставалось иного выбора, кроме как реализовать загрузчик кода новых ядер вручную, предварительно создав для них пустую функцию-контейнер. Загрузчик основан на технологиях проекта AsFermi [3] и действует следующим образом. В начале работы приложение на GPU загружается достаточно большое пустое ядро (содержащее инструкции NOP). По мере того, как в процессе работы приложения требуется запускать вновь скомпилированные ядра, их код копируется как данные в адресное пространство контейнера, которое известно благодаря инструкции LEPC (получить значение Effective Program Counter). Контейнер размещает код множества небольших ядер друг за другом, создавая своеобразный динамический пул памяти для кода. При этом необходимо учитывать, что различные ядра могут использовать различное число регистров. Для этого загрузчик создает 63 фиктивных ядра (точки входа), используя

ющих от 1 до 63 регистров с единственной инструкцией JMP для перехода по адресу начала требуемого ядра в контейнере.

Система синхронизации памяти между GPU и хостом использует вызов *ttar*, ограничивающий возможные диапазоны адресов величинами, кратными размеру страниц (4096 байт). Поэтому выравнивание всех данных GPU по границе 4096 было бы очень удобным упрощением на данном этапе. К сожалению, текущая реализация CUDA (5.0) учитывает настройки выравнивания данных при компиляции, но при этом игнорирует их во время исполнения. Обход этого дефекта реализован посредством выравнивания размеров всех данных по границе 4096 вручную с помощью функций библиотеки *libelf*.

## 4. Тестирование

KernelGen тестируется на трех типах приложений: тесты корректности, тесты производительности и работа на реальных задачах. Тесты корректности предназначены для контроля регрессивных изменений в генераторе кода, тесты производительности позволяют анализировать эффективность текущей версии KernelGen в сравнении с предыдущими сборками и другими компиляторами. При тестировании производительности предпочтение отдается сравнению с результатами других распараллеливающих компиляторов, поскольку в отличие от сравнения с кодом, оптимизированным вручную, это позволяет проанализировать достоинства и недостатки компилятора в своем классе систем.



**Рис. 6.** Сравнение производительности вычислительных ядер некоторых тестовых приложений, скомпилированных KernelGen r1780 и PGI 13.2 на GPU NVIDIA Tesla C2075 (Fermi sm\_20) и GTX 680M (Kepler sm\_30). Положительные величины – ядра, собранные KernelGen быстрее, чем у PGI, отрицательные величины – ядра, собранные KernelGen медленнее, чем у PGI

Для тестирования были выбраны приложения, реализующие различные типовые алгоритмы на двумерной или трехмерной регулярной сетке с одинарной точностью (часть тестов адаптировано из материалов работы [1], описание и исходный код приведены в [21]). На рис. 6 показано, насколько меняется скорость работы тестов, скомпилированных с помощью KernelGen, по сравнению с версией PGI OpenACC. Соответствующие абсолютные

времена и число регистров для каждой версии теста приведены в таблице. Тесты *jacobi*, *matmul* и *sincos* реализованы на языке Fortran, остальные тесты – на C (также были проверены реализации тестов *wave13pt* и *laplacian* на языке C++, однако для данного сравнения они не пригодны, поскольку компилятор PGI не поддерживает директивы OpenACC в C++). KernelGen автоматически распознает наличие вложенных параллельных циклов внутри непараллельного цикла по числу итераций, когда как PGI делает это только при соответствующей ручной расстановке директив OpenACC. Более низкая производительность KernelGen на тесте *matmul* обусловлена тем, что компилятор PGI реализует частичную раскрутку внутреннего цикла с редукцией на регистрах. Более низкая производительность ряда других тестов требует отдельного изучения.

Таблица

Сравнение времени исполнения (сек) и числа регистров (nregs) для вычислительных ядер некоторых тестовых приложений, скомпилированных KernelGen r1780 и PGI 13.2 на GPU NVIDIA Tesla C2075 (Fermi sm\_20) и GTX 680M (Kepler sm\_30)

Тест	NVIDIA Tesla C2075				NVIDIA GTX 680M			
	KernelGen		PGI		KernelGen		PGI	
	время	nregs	время	nregs	время	nregs	время	nregs
divergence	0,008462	18	0,008578	40	0,007924	20	0,012595	49
gameoflife	0,010180	21	0,011178	28	0,012555	21	0,015723	32
gaussblur	0,016009	56	0,013286	31	0,014991	51	0,024847	36
gradient	0,010582	21	0,010745	42	0,008855	22	0,009869	51
jacobi	0,007824	24	0,010314	23	0,006324	23	0,008711	30
lapgsrb	0,017307	55	0,014386	61	0,018310	40	0,026813	63
laplacian	0,008150	18	0,005070	39	0,007431	21	0,007325	48
matmul	0,000993	16	0,000717	23	0,000730	16	0,00108	33
matvec	0,016349	16	0,026357	22	0,028514	16	0,042523	25
sincos	0,010796	22	0,005757	26	0,008421	22	0,005129	29
tricubic	0,053462	63	0,049090	63	0,064770	61	0,097440	63
uxx1	0,016479	32	0,020002	59	0,015520	32	0,022861	63
vecadd	0,004837	12	0,004723	24	0,004454	12	0,004869	31
wave13pt	0,011778	34	0,009885	54	0,012461	34	0,015986	60

Тестирование на больших вычислительных приложениях COSMO [19] и WRF [20] показало, что KernelGen способен генерировать корректные исполняемые файлы с поддержкой GPU за разумное время.

## Заключение

В проекте KernelGen реализована оригинальная схема автоматического портирования кода на GPU, подходящая для сложных приложений. Не требуя никаких изменений в исходном коде, компилятор переносит на GPU максимально возможную часть кода, включая выделение памяти, тем самым создавая эффективную схему для преимущественно GPU-

вычислений. KernelGen реализует средства автоматического анализа параллелизма циклов, основанные на LLVM, Polly и других проектах, расширяя их поддержкой генерации кода для GPU. Генератор GPU-кода основан на NVPTX-бекенде для LLVM, совместно развиваемом компанией NVIDIA и силами сообщества LLVM. Тестирование показало, что GPU-код, генерируемый KernelGen по своей эффективности сравним с коммерческим компилятором PGI.

Для того чтобы начать использовать компилятор в прикладных задачах, остается реализовать некоторые функциональные элементы. В частности, в нынешней версии отсутствует механизм накопления статистики эффективности исполнения GPU-кода для принятия решений о переключении на CPU-реализацию в неэффективных случаях. В генераторе параллельных циклов желательно добавить возможность использования разделяемой памяти и распознавание в коде идиомы редукции. Запуск вычислительных ядер на архитектуре Kepler может быть организован более эффективно за счет использования динамического параллелизма.

Код KernelGen распространяется по лицензии University of Illinois/NCSA (за исключением плагина для GCC) и доступен на сайте проекта: <http://kernelgen.org/>.

*Работа поддержана грантом «Swiss Platform for High-Performance and High-Productivity Computing» (HP2C, [hp2c.ch](http://hp2c.ch)), а также контрактами Applied Parallel Computing LLC № 12-2011 и № 13-2011. Тестирование велось на оборудовании, предоставленном компаниями NVIDIA и HP, кластере «Ломоносов» МГУ им М.В. Ломоносова [22] и кластере «Tödi» Швейцарского национального суперкомпьютерного центра (CSCS).*

## Литература

1. Christen, M. PATUS for Convenient High-Performance Stencils: Evaluation in Earthquake Simulations / M. Christen, O. Schenk, Y. Cui // Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (Salt Lake City, USA, November 10–16, 2012).
2. Grosser, T. Polly – Polyhedral Optimization in LLVM / T. Grosser, H. Zheng, R. Aloor, A. Simbürger, A. Größlinger, L.-N. Pouchet // International Symposium on Code Generation and Optimization (Charmonix, France, April 2–6, 2011).
3. Hou, Y. AsFermi: An assembler for the NVIDIA Fermi Instruction Set / Y. Hou // URL: <http://code.google.com/p/asfermi/> (дата обращения: 03.12.2012).
4. Gysi, T. HP2C Dycore / T. Gysi // Workshop on COSMO dynamical core rewrite and HP2C project — March 2012, Offenbach, Germany — URL: [http://mail.cosmo-model.org/pipermail/pompa/attachments/20120306/079fad1/DWD\\_HP2C\\_Dycore\\_120305.pdf](http://mail.cosmo-model.org/pipermail/pompa/attachments/20120306/079fad1/DWD_HP2C_Dycore_120305.pdf) (дата обращения: 03.12.2012).
5. Ragan-Kelley, J. Decoupling Algorithms from Schedules for Easy Optimization of Image Processing Pipelines / J. Ragan-Kelley, A. Adams, S. Paris, M. Levoy, S. Amarasinghe, F. Durand. // ACM Trans. Graph. — 2012. — Vol. 31, No. 4. — P. 32:1–32:12.
6. Адинец, А.В. Программирование графических процессов при помощи расширяемых языков. // А.В. Адинец // Вестник Южно-Уральского государственного университета. Серия «Математическое моделирование и программирование». — 2011. — N. 25 (242). — Вып. 9. — С. 52-63.

7. The OpenACC™ Application Programming Interface. Version 1.0. URL: <http://www.openacc-standard.org> (дата обращения: 27.05.2012).
8. OpenHMPP, New HPC Open Standard for Many-Core. URL: <http://www.openhmpp.org/en/OpenHMPPConsortium.aspx> (дата обращения: 03.12.2012).
9. The Heterogeneous Offload Model for Intel® Many Integrated Core Architecture. URL: <http://software.intel.com/sites/default/files/article/326701/heterogeneous-programming-model.pdf> (дата обращения: 03.12.2012).
10. Govett, M. Development and Use of a Fortran → CUDA translator to run a NOAA Global Weather Model on a GPU cluster / M. Govett // Path to Petascale: Adapting GEO/CHEM/ASTRO Applications for Accelerators and Accelerator Clusters — 2009. — National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign. — URL: <http://gladiator.ncsa.uiuc.edu/PDFs/accelerators/day2/session3/govett.pdf> (дата обращения: 27.05.2012).
11. Бахтин, В.А. Автоматическое распараллеливание Фортран-программ на кластер с графическими ускорителями / В.А. Бахтин, Н.А. Катаев, М.С. Клинов, В.А. Крюков, Н.В. Поддерюгина, М.Н. Притула // Параллельные вычислительные технологии (ПаВТ'2012): Труды международной научной конференции (Новосибирск, 26 марта – 30 марта 2012 г.). — Челябинск: Издательский центр ЮУрГУ, 2012. — С. 373–379.
12. Kravets, A. GRAPHITE-OpenCL: Automatic parallelization of some loops in polyhedra representation / A. Kravets, A. Monakov, A. Belevantsev // GCC Developers' Summit (Ottawa, Canada, October 25–27, 2010).
13. Verdoolaege, S. Polyhedral parallel code generation for CUDA / S. Verdoolaege, J. Carlos Juega, A. Cohen, J. Ignacio Gómez, C. Tenllado, F. Catthoor // ACM Trans. Archit. Code Optim. — 2013. — Vol. 9, No. 4. — P. 54:1–54:23.
14. Bastoul, C. Code Generation in the Polyhedral Model Is Easier Than You Think / C. Bastoul // PACT'13 IEEE International Conference on Parallel Architecture and Compilation Techniques (Antibes Juan-les-Pins, France, September 29 - October 3, 2004).
15. Torquati, M. An innovative compilation tool-chain for embedded multi-core architectures / M. Torquati, M. Venneschi, M. Amini, S. Guelton, R. Keryell, V. Lanore, F.-X. Pasquier, M. Barreteau, R. Barrère, C.-T. Petrisor, É. Lenormand, C. Cantini, F. De Stefani. // Embedded World Conference (Nuremberg, Germany, February 2012).
16. Sands, D. Reimplementing llvm-gcc as a gcc plugin / D. Sands // Third Annual LLVM Developers' Meeting. — 2009. — Apple Inc. Campus, Cupertino, California. — URL: [http://llvm.org/devmtg/2009-10/Sands\\_LLVMGCCPlugin.pdf](http://llvm.org/devmtg/2009-10/Sands_LLVMGCCPlugin.pdf) (дата обращения: 03.12.2012).
17. Wolfe, M. The PGI Accelerator Programming Model on NVIDIA GPUs Part 3: Porting WRF / M. Wolfe, C. Toepfer // URL: <http://www.pgroup.com/lit/articles/insider/v1n3a1.htm> (дата обращения: 03.12.2012).
18. Squyres, J. Open MPI State of the Union / J. Squyres, G. Bosilca, S. Sumimoto, R. vandeVaart // Open MPI Community Meeting. — Supercomputing, 2011. — URL: <http://www.open-mpi.org/papers/sc-2011/Open-MPI-SC11-BOF-1up.pdf> (дата обращения: 03.12.2012).
19. Consortium for Small-scale Modeling. URL: <http://www.cosmo-model.org/> (дата обращения: 03.12.2012).



20. The Weather Research & Forecasting Model. URL: <http://www.wrf-model.org/index.php> (дата обращения: 03.12.2012).
21. KernelGen Performance Test Suite. URL: [https://hpcforge.org/plugins/mediawiki/wiki/kernelgen/index.php/Performance\\_Test\\_Suite](https://hpcforge.org/plugins/mediawiki/wiki/kernelgen/index.php/Performance_Test_Suite) (дата обращения: 27.01.2013).
22. Практика суперкомпьютера «Ломоносов» / Вл.В. Воеводин, С.А. Жуматий, С.И. Со-  
болев и др. // Открытые системы. — 2012. — No. 7. — С. 36–39.

Лихогруд Николай Николаевич, аспирант факультета вычислительной математики и кибернетики, МГУ им. М.В. Ломоносова (Москва, Российская Федерация), [nicolas@kernelgen.org](mailto:nicolas@kernelgen.org).

Микушин Дмитрий Николаевич, аспирант, ассистент, Институт информатики Университета Лугано (Швейцария), [dmitry@kernelgen.org](mailto:dmitry@kernelgen.org).

---

## KERNELGEN — A PROTOTYPE OF LLVM-BASED AUTO-PARALLELIZING C/FORTRAN COMPILER FOR NVIDIA GPUs

*N.N. Likhograd*, Lomonosov Moscow State University (Moscow, Russian Federation),  
*D.N. Mikushin*, Università della Svizzera italiana (Lugano, Switzerland)

The KernelGen project (<http://kernelgen.org/>) aims to develop Fortran and C compilers based on the state-of-art open-source technologies for automatic GPU kernels generation from unmodified CPU source code, significantly improving the code porting experiences. Parallelism detection is based on LLVM/Polly and CLoog, extended with mapping of loops onto GPU compute grid, and assisted with runtime alias analysis. PTX assembly code is generated with NVPTX backend. Thanks to integration with GCC frontend by means of DragonEgg plugin, and customized linker, KernelGen features full GCC compatibility, and is able to compile complex applications into hybrid binaries containing both CPU and GPU-enabled executables. In addition to more robust parallelism detection, test kernels produced by KernelGen are up to 60 % faster than generated by PGI compiler for kernels source with manually inserted OpenACC directives.

*Keywords: GPU, LLVM, OpenACC, JIT-compilation, polyhedral analysis.*

## References

1. Christen M., Schenk O., Cui Y. PATUS for Convenient High-Performance Stencils: Evaluation in Earthquake Simulations // Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (Salt Lake City, USA, November 10–16, 2012).
2. Grosser T., Zheng H., Aloor R., Simbürger A., Größlinger A., Pouchet L.-N. Polly – Polyhedral Optimization in LLVM // International Symposium on Code Generation and Optimization (Charmonix, France, April 2–6, 2011).
3. Hou Y. AsFermi: An assembler for the NVIDIA Fermi Instruction Set / Y. Hou // URL: <http://code.google.com/p/asfermi/> (accessed: 03.12.2012).
4. Gysi T. HP2C Dycore // Workshop on COSMO dynamical core rewrite and HP2C project. March 2012, Offenbach, Germany. URL: [http://mail.cosmo-model.org/pipermail/pompa/attachments/20120306/079fadc1/DWD\\_HP2C\\_Dycore\\_120305.pdf](http://mail.cosmo-model.org/pipermail/pompa/attachments/20120306/079fadc1/DWD_HP2C_Dycore_120305.pdf) (accessed: 03.12.2012).

5. Ragan-Kelley J., Adams A., Paris S., Levoy M., Amarasinghe S., Durand F. Decoupling Algorithms from Schedules for Easy Optimization of Image Processing Pipelines // ACM Trans. Graph. 2012. Vol. 31, No. 4. P. 32:1–32:12.
6. Adinetz A.V. Programming graphics processors with extensible languages // Bulletin of South Ural State University. Series “Mathematical Modelling, Programming and Computer Software”. 2011. No. 25(242). P. 52-63.
7. The OpenACC™ Application Programming Interface. Version 1.0. URL: <http://www.openacc-standard.org> (accessed: 27.05.2012).
8. OpenHMPP, New HPC Open Standard for Many-Core. URL: <http://www.openhmpp.org/en/OpenHMPPConsortium.aspx> (accessed: 03.12.2012).
9. The Heterogeneous Offload Model for Intel® Many Integrated Core Architecture. URL: <http://software.intel.com/sites/default/files/article/326701/heterogeneous-programming-model.pdf> (accessed: 03.12.2012).
10. Govett, M. Development and Use of a Fortran → CUDA translator to run a NOAA Global Weather Model on a GPU cluster / M. Govett // Path to Petascale: Adapting GEO/CHEM/ASTRO Applications for Accelerators and Accelerator Clusters — 2009. — National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign. — URL: <http://gladiator.ncsa.uiuc.edu/PDFs/accelerators/day2/session3/govett.pdf> (accessed: 27.05.2012).
11. Bakhtin V.A., Kataev N.A, Klinov M.S., Kryukov V.A., Podderiyugina N.V., Pritula M.N. Avtomaticheskoe rasparrallelivanie Fortran-programm na klaster s graficheskimi uskoritelyami [Automatic parallelization of Fortran programs for GPU-enabled clusters] // Parallel Computational Technologies (PAVT’2012): Proceedings of the International Scientific Conference (Novosibirsk, March 26 – 30, 2012). Chelyabinsk, Publishing of the South Ural State University, 2012. P. 373–379.
12. Kravets A., Monakov A., Belevantsev A. GRAPHITE-OpenCL: Automatic parallelization of some loops in polyhedra representation // GCC Developers’ Summit (Ottawa, Canada, October 25–27, 2010).
13. Verdoolaege S., Carlos Juega J., Cohen A., Ignacio Gómez J., Tenllado C., Catthoor F. Polyhedral parallel code generation for CUDA // ACM Trans. Archit. Code Optim. 2013. Vol. 9, No. 4. P. 54:1–54:23.
14. Bastoul C. Code Generation in the Polyhedral Model Is Easier Than You Think / C. Bastoul // PACT’13 IEEE International Conference on Parallel Architecture and Compilation Techniques (Antibes Juan-les-Pins, France, September 29 - October 3, 2004).
15. Torquati M., Venneschi M., Amini M., Guelton S., Keryell R., Lanore V., Pasquier F.-X., Barreateau M., Barrère R., Petrisor C.-T., Lenormand É., Cantini C., De Stefani F. An innovative compilation tool-chain for embedded multi-core architectures // Embedded World Conference (Nuremberg, Germany, February 2012).
16. Sands D. Reimplementing llvm-gcc as a gcc plugin / D. Sands // Third Annual LLVM Developers’ Meeting. — 2009. — Apple Inc. Campus, Cupertino, California. — URL: [http://llvm.org/devmtg/2009-10/Sands\\_LLVMGCCPlugin.pdf](http://llvm.org/devmtg/2009-10/Sands_LLVMGCCPlugin.pdf) (accessed: 03.12.2012).

17. Wolfe M., Toepfer C. The PGI Accelerator Programming Model on NVIDIA GPUs Part 3: Porting WRF // URL: <http://www.pgroup.com/lit/articles/insider/v1n3a1.htm> (accessed: 03.12.2012).
18. Squyres J., Bosilca G., Sumimoto S., vandeVaart R. Open MPI State of the Union // Open MPI Community Meeting. Supercomputing, 2011. URL: <http://www.open-mpi.org/papers/sc-2011/Open-MPI-SC11-BOF-1up.pdf> (accessed: 03.12.2012).
19. Consortium for Small-scale Modeling. URL: <http://www.cosmo-model.org/> (accessed: 03.12.2012).
20. The Weather Research & Forecasting Model. URL: <http://www.wrf-model.org/index.php> (accessed: 03.12.2012).
21. KernelGen Performance Test Suite. URL: [https://hpcforge.org/plugins/mediawiki/wiki/kernelgen/index.php/Performance\\_Test\\_Suite](https://hpcforge.org/plugins/mediawiki/wiki/kernelgen/index.php/Performance_Test_Suite) (accessed: 27.01.2013).
22. Voevodin V.I., Zhumatiy S.A., Sobolev S.I. et al. Practice of “Lomonosov” Supercomputer // Open Systems. 2012. No. 7. P. 36–39.

*Поступила в редакцию 8 июня 2013 г.*

# ОТОБРАЖЕНИЕ НА КЛАСТЕРЫ С ГРАФИЧЕСКИМИ ПРОЦЕССОРАМИ DVMH-ПРОГРАММ С РЕГУЛЯРНЫМИ ЗАВИСИМОСТЯМИ ПО ДАННЫМ<sup>1</sup>

*В.А. Бахтин, А.С. Колганов, В.А. Крюков, Н.В. Поддерюгина,  
М.Н. Притула*

В 2011 г. для новых гетерогенных и гибридных суперкомпьютерных систем в Институте прикладной математики им. М.В. Келдыша РАН была предложена модель DVMH (DVM for Heterogeneous systems), разработаны языки программирования высокого уровня, представляющие собой стандартные языки Фортран и Си, расширенные директивами отображения программы на параллельную машину, оформленными в виде специальных комментариев (или прагм). В статье описываются проблемы и методы отображения циклов с зависимостями на графические процессоры, демонстрируется эффективность разработанных на языке Fortran DVMH параллельных программ с регулярными зависимостями по данным.

*Ключевые слова:* DVM for Heterogeneous systems, Fortran DVMH, гибридные системы с ускорителями, графические процессоры, CUDA.

## Введение

В последнее время появляется много вычислительных кластеров с установленными в их узлах ускорителями. В основном, это графические процессоры компании NVIDIA. В 2012 г. начинают появляться кластеры с ускорителями другой архитектуры – Xeon Phi от компании Intel. Так, в списке Top500 [1] самых высокопроизводительных суперкомпьютеров мира, объявленном в ноябре 2012 г., 62 машины имеют в своем составе ускорители, из них 50 машин имеют ускорители NVIDIA, 7 – Intel, 3 – AMD/ATI, 2 – IBM. Данная тенденция заметно усложняет процесс программирования кластеров, так как требует освоения на достаточном уровне сразу нескольких моделей и языков программирования. Традиционным подходом можно назвать использование технологии MPI для разделения работы между узлами кластера, а затем технологий CUDA (или OpenCL) и OpenMP для загрузки всех ядер центрального и графического процессоров.

С целью упрощения программирования распределенных вычислительных систем были предложены высокоуровневые языки программирования, основанные на расширении директивами стандартных языков такие, как HPF [2], Fortran-DVM [3, 4], C-DVM [3, 5]. Также были предложены модели программирования и соответствующие основанные на директивах расширения языков для возможности использования ускорителей такие, как HMPP [6], PGI Accelerator Programming Model [7], OpenACC [8], hiCUDA [9].

Распараллеливание на GPU циклов без зависимостей, будь то ручное или с использованием высокоуровневых средств, обычно не вызывает больших идеологических трудностей, так как целевая массивно-параллельная архитектура хорошо подходит для их обработки. Циклы с зависимостями могут быть распараллелены с заметно большими трудностями, связанными в частности с моделью консистентности общей памяти, огра-

---

<sup>1</sup> Статья рекомендована к публикации программным комитетом Международной суперкомпьютерной конференции «Научный сервис в сети Интернет: все грани параллелизма – 2013».

ниченной поддержкой синхронизации потоков выполнения на GPU, большими потерями времени при невыравненном произвольном доступе к памяти.

Статья организована следующим образом. Разделы 1 и 2 статьи содержат описание способа исполнения циклов с регулярными зависимостями на GPU, примененном в языке Fortran-DVMH. Раздел 3 содержит наглядную демонстрацию новых возможностей языка Fortran-DVMH. Раздел 4 содержит описание применения языка Fortran-DVMH для распараллеливания нескольких тестов NAS на GPU. В заключении отмечены полученные результаты и направление для будущих исследований.

## 1. Язык Fortran-DVMH

В 2011 г. в Институте прикладной математики им. М.В. Келдыша РАН была расширена модель DVM для поддержки кластеров с ускорителями [10]. Это расширение названо DVMH и позволяет с небольшими изменениями перевести DVM-программу для кластера в DVMH-программу для кластера с ускорителями.

Язык Fortran-DVM был дополнен набором директив, обеспечивающих следующие возможности:

- задание вычислительных регионов — частей программы, для которых будет подготовлено исполнение на GPU;
- спецификация дополнительных свойств параллельных циклов;
- управление актуальным состоянием данных в памяти CPU.

Одним из важных аспектов функционирования такой программной модели, как DVMH, является вопрос отображения исходной программы на все уровни параллелизма и разнородные вычислительные устройства. Важными задачами механизма отображения является обеспечение корректного исполнения всех поддерживаемых языком конструкций на разнородных вычислительных устройствах, балансировка нагрузки между вычислительными устройствами, а также выбор оптимального способа исполнения каждого участка кода на том или ином устройстве.

В Fortran-DVM поддерживаются циклы с регулярными зависимостями по данным, для чего существует специальное указание ACROSS в директиве PARALLEL. Такие циклы могут корректно выполняться на кластере параллельно в режиме конвейера или в режиме обработки по гиперплоскостям.

В 2013 г. в язык Fortran-DVMH была добавлена поддержка работы циклов с регулярными зависимостями на GPU NVIDIA с использованием технологии программирования CUDA.

В модели DVM [3] имеется возможность спецификации циклов с регулярными зависимостями и их эффективного выполнения на кластерных архитектурах. При подключении возможности использовать графический ускоритель для таких циклов нужно решить следующий набор задач:

- подкачка данных между соседями, в том числе в режиме конвейера;
- эффективное отображение порции цикла с зависимостями на архитектуру CUDA;
- оптимизация обращений к глобальной памяти GPU.

В режиме конвейера часть цикла, попавшая на конкретный MPI-процесс, разбивается на части для того, чтобы соседние зависимые процессы как можно раньше начали

выполнять этот цикл. Так как стоит задача использовать GPU, то необходимо поддерживать такую подкачку во время обработки цикла.

Для циклов, имеющих более одного зависимого измерения, есть возможность не превращать эти измерения в последовательные, а использовать, например, метод гиперплоскостей для извлечения максимального параллелизма.

Так как обращения к глобальной памяти в случае невыравненного доступа очень медленные, то существует проблема эффективности выполнения циклов в случае изменения порядка обработки витков цикла, которая не решается простой перестановкой измерений массива или цикла на уровне исходного текста, как будет продемонстрировано на примере методов попеременных направлений и последовательной верхней релаксации. Данную проблему призван решать механизм динамического переупорядочивания массивов, реализованный в компиляторе Fortran-DVMH.

## 2. Алгоритм отображения циклов с зависимостями на GPU

Пусть есть программа, написанная на языке Fortran-DVMH, в которой присутствуют многомерные тесно-гнездовые циклы с регулярными зависимостями по данным. Один из известных алгоритмов, в котором есть зависимости по данным — SOR (Successful Over Relaxation) — метод последовательной верхней релаксации. Для простоты рассмотрим двумерный случай. Тогда для квадратной матрицы порядка  $N$  основной цикл выглядит так, как показано на рис. 1.

```

DO J = 2, N-1
  DO I = 2, N-1
    S=A(I, J)
    A(I, J)=(W/4) * (A(I-1, J)+A(I+1, J)+A(I, J-1)+A(I, J+1))+(1-W) * A(I, J)
    EPS=MAX(EPS, ABS(S-A(I, J)))
  ENDDO
ENDDO
    
```

Рис. 1. Основной цикл метода последовательной верхней релаксации

Пространством витков данного цикла назовем множество кортежей всех принимаемых значений индексных переменных цикла. В рассматриваемом цикле есть прямая и обратная зависимость по измерению  $I$  и  $J$ , следовательно, его пространство витков не может быть отображено на блок нитей GPU, так как все нити исполняются независимо. Следовательно, нужен другой метод отображения.

Одним из известных методов отображения подобных циклов является метод гиперплоскостей. Все элементы, лежащие на гиперплоскости, могут быть вычислены независимо друг от друга. Будем вычислять все пространство витков за несколько итераций: на первой итерации будут вычислены элементы первой гиперплоскости, на второй — элементы второй гиперплоскости и т.д., пока не будут вычислены все элементы. Если применить данный метод к рассматриваемому циклу, то будут вычисляться диагонали, параллельные побочной. Всего будет выполнено порядка  $2N$  итераций.

Обобщим предложенный алгоритм на многомерный случай. Пусть есть многомерный тесно-гнездовой цикл размерности  $k$  с регулярными зависимостями по данным по всем  $k$  измерениям. Тогда все элементы, лежащие на гиперплоскости ранга  $k-1$  могут быть вычислены независимо. Если из  $k$  измерений  $q$  не имеют зависимости по данным, а  $p$  — имеют, где  $k = p + q$ , то применим предложенный алгоритм к гиперплоскостям ранга  $p$ , а остальные  $q$  измерений вычислим как независимые.

Как уже упоминалось выше, при таком порядке выполнения витков цикла может возникать проблема эффективного доступа к глобальной памяти в силу того, что параллельно обрабатываются не соседние элементы массивов, что приводит к значительной потере производительности.

Для оптимизации доступа к глобальной памяти в систему поддержки выполнения Fortran-DVMH программ был внедрен механизм динамического переупорядочивания массивов. Данный механизм перед каждым циклом использует информацию о взаимном выравнивании цикла и массива, которая уже имеется в DVM-программе для отображения на кластер и распределения вычислений. Он устанавливает соответствие измерений цикла и измерений массива, после чего переупорядочивает массив таким образом, чтобы при отображении на архитектуру CUDA доступ к элементам осуществлялся наилучшим образом — соседние нити блока работают с соседними ячейками памяти.

Данный механизм осуществляет любую необходимую перестановку измерений массива, а также так называемую поддиагональную трансформацию, в результате которой соседние элементы на диагоналях (в плоскости необходимых двух измерений) располагаются в соседних ячейках памяти, что позволяет применять технику выполнения цикла с зависимостями по гиперплоскостям без значительной потери производительности на операциях доступа к глобальной памяти.

### 3. Примеры программ и характеристики их выполнения

Для иллюстрации описываемых новых возможностей Fortran-DVMH программ рассмотрим две простые программы — реализации метода попеременных направлений и метода последовательной верхней релаксации.

Рассмотрим Fortran-DVMH программу для метода попеременных направлений, показанную на рис. 2.

Как видно, в программе есть циклы с зависимостями, причем в каждом из трех циклов зависимость только по одному измерению. Данная особенность позволяет выделить достаточный уровень параллелизма из них, но существенно то, что при любом порядке измерений массива один из этих циклов будет работать значительно медленнее (примерно в 10 раз) остальных вследствие невозможности параллельной обработки в этом цикле элементов массива, лежащих в соседних ячейках памяти. В такой ситуации является чрезвычайно полезным механизм динамического переупорядочивания массивов. Для такой программы достаточно переупорядочивать массив 1 раз на три цикла (одну итерацию метода). Данная программа без дополнительных изменений, скомпилированная Fortran-DVMH компилятором и запущенная на выполнение автоматически обнаружит необходимость такой перестановки и ускорит ее выполнение по сравнению с более традиционным подходом, при котором расположение массива задано жестко и не может изменяться во время работы программы. Была произведена серия запусков на кластере К-100, установленном в ИПМ им. М.В. Келдыша РАН, в котором в вычислительном узле установлены 2 CPU Intel Xeon X5670 и 3 GPU NVIDIA Tesla C2050. Результаты запусков отображены в табл. 1.

```

PROGRAM ADI
PARAMETER (NX=400, NY=400, NZ=400, MAXEPS=0.01, ITMAX=100)
INTEGER NX, NY, NZ, ITMAX
DOUBLE PRECISION EPS, RELAX, A (NX, NY, NZ)
!DVM$ DISTRIBUTE (BLOCK, BLOCK, BLOCK) :: A
CALL INIT (A, NX, NY, NZ)
DO IT = 1, ITMAX
  EPS=0.00
!DVM$ ACTUAL (EPS)
!DVM$ REGION
!DVM$ PARALLEL (K, J, I) ON A (I, J, K) ,
!DVM$* ACROSS (A (1:1, 0:0, 0:0))
  DO K = 2, NZ-1
    DO J = 2, NY-1
      DO I = 2, NX-1
        A (I, J, K) = (A (I-1, J, K) + A (I+1, J, K)) / 2
      ENDDO
    ENDDO
  ENDDO
!DVM$ PARALLEL (K, J, I) ON A (I, J, K) ,
!DVM$* ACROSS (A (0:0, 1:1, 0:0))
  DO K = 2, NZ-1
    DO J = 2, NY-1
      DO I = 2, NX-1
        A (I, J, K) = (A (I, J-1, K) + A (I, J+1, K)) / 2
      ENDDO
    ENDDO
  ENDDO
!DVM$ PARALLEL (K, J, I) ON a (I, J, K) ,
!DVM$* REDUCTION (MAX (EPS)) , ACROSS (A (0:0, 0:0, 1:1))
  DO K = 2, NZ-1
    DO J = 2, NY-1
      DO I = 2, NX-1
        EPS = MAX (EPS, ABS (A (I, J, K) -
* (A (I, J, K-1)+A (I, J, K+1)) / 2))
        A (I, J, K) = (A (I, J, K-1)+A (I, J, K+1)) / 2
      ENDDO
    ENDDO
  ENDDO
!DVM$ END REGION
!DVM$ GET_ACTUAL (EPS)
  IF (EPS .LT. MAXEPS) GOTO 3
  ENDDO
3 CONTINUE
END

```

Рис. 2. Реализация метода попеременных направлений на Fortran-DVMH

На малых размерах видно небольшое замедление в силу недостаточной загруженности GPU, а значит и проигрыш от медленного доступа к памяти не играет такой важной роли.

Рассмотрим Fortran-DVMH программу для метода последовательной верхней релаксации, показанную на рис. 3.

Основной цикл данной программы имеет зависимости по всем своим измерениям, что приводит к значительным трудностям при распараллеливании даже в модели OpenMP. DVM-система позволяет исполнять такие циклы на GPU и получать ускорение за счет двух изложенных выше техник. Стоит отдельно подчеркнуть, что эта программа



не только может эффективно исполняться на GPU, но также она может быть исполнена и на кластере с GPU без каких-либо дополнительных изменений.

Таблица 1

Время работы для метода попеременных направлений (100 итераций)

Линейный размер ( $nx=ny=nz$ )	CPU, 1 ядро	GPU, без переупорядочивания массива		GPU, с переупорядочиванием массива	
	Время, с	Время, с	Ускорение	Время, с	Ускорение
64	0,17	0,13	1,31	0,18	0,94
128	1,52	0,92	1,65	0,52	2,92
256	12,66	6,84	1,85	3,16	4,01
400	48,69	26,17	1,86	11,34	4,29

В табл. 2 приведены характеристики эффективности применения алгоритма отображения как с применением переупорядочивания массива, так и без.

```

PROGRAM SOR
PARAMETER(N1=16000,N2=16000,ITMAX=100,MAXEPS=0.5E-6,W=0.5)
REAL A(N1,N2), EPS, W
INTEGER ITMAX
!DVM$ DISTRIBUTE A(BLOCK, BLOCK)
CALL INIT(A, N1, N2)
DO IT = 1,ITMAX
  EPS = 0.
!DVM$ ACTUAL(EPS)
!DVM$ REGION
!DVM$ PARALLEL (J, I) ON A(I, J), PRIVATE (S),
!DVM$* REDUCTION(MAX(EPS)), ACROSS (A(1:1,1:1))
  DO J = 2,N2-1
    DO I = 2,N1-1
      S = A(I, J)
      A(I, J) = (W/4)*(A(I-1, J)+A(I+1, J)+
*           A(I, J-1)+A(I, J+1)) + (1-W)*A(I, J)
      EPS = MAX(EPS, ABS(S - A(I, J)))
    ENDDO
  ENDDO
!DVM$ END REGION
!DVM$ GET_ACTUAL(EPS)
  IF (EPS .LT. MAXEPS) GOTO 4
ENDDO
4 CONTINUE
END
    
```

Рис. 3. Реализация метода последовательной верхней релаксации направлений на Fortran-DVMH

Таблица 2

Время работы для метода последовательной релаксации (100 итераций)

Линейный размер ( $N1=N2$ )	CPU, 1 ядро	GPU, без переупорядочивания массива		GPU, с переупорядочиванием массива	
	Время, с	Время, с	Ускорение	Время, с	Ускорение
2000	2,76	2,23	1,24	1,8	1,53
4000	11,10	5,6	1,98	3,9	2,85
8000	44,50	19,01	2,34	10,76	4,14
16000	178,02	80,24	2,22	32,91	5,41

#### 4. Применение на тестах NPB: BT, SP, LU

В пакете NAS Parallel Benchmarks присутствуют три теста, в алгоритмах которых есть циклы с зависимостью по данным: BT (Block Tridiagonal), SP (Scalar Pentadiagonal) и LU (Lower - Upper). Эти тесты решают синтетическую задачу дифференциальных уравнений в частных производных (трехмерная система уравнений Навье–Стокса для сжимаемой жидкости или газа), используя блочную трехдиагональную схему с методом переменных направлений (BT), скалярную пятидиагональную схему (SP), и метод симметричной последовательной верхней релаксации (SSOR, алгоритм LU при помощи симметричного метода Гаусса–Зейделя).

В тесте BT всего 57 тесно-гнездовых циклов, которые можно вычислить параллельно. Из них 6 циклов имеют зависимость по одному из трех отображаемых измерений, причем зависимое измерение в различных циклах соответствует различным измерениям обрабатываемых массивов. В полученной Fortran-программе 3850 строк в фиксированном формате, 76 из которых — директивы DVMH. Для лучшего доступа к глобальной памяти GPU в исходном тексте была произведена следующая оптимизация — во всех массивах были переупорядочены измерения так, чтобы самое внутреннее измерение цикла соответствовало первому измерению массива, в соответствии с принятым в Fortran способом расположения массива в памяти.

В тесте SP всего 56 тесно-гнездовых циклов, которые можно вычислить параллельно. Из них 12 циклов имеют зависимость по одному из трех отображаемых измерений, причем зависимое измерение в различных циклах соответствует различным измерениям обрабатываемых массивов. В полученной Fortran-программе 3500 строк в фиксированном формате, 215 из которых — директивы DVMH. Никаких оптимизаций на уровне исходных текстов по сравнению с исходной последовательной программой не производилось.

Эффективность распараллеливания тестов NPV: BT, SP, LU

Задача		CPU, 1 ядро	CPU, 12 ядер		Tesla C2050 (с ECC)		GeForce GTX 560 Ti (без ECC)	
Тест	Класс	Время, с	Время, с	Ускорение	Время, с	Ускорение	Время, с	Ускорение
BT	W	2,67	0,41	6,51	3,81	0,70	3,12	0,86
	A	67	10,7	6,26	27,9	2,40	19,7	3,40
	B	282,3	46	6,14	127	2,22	—	—
SP	W	8,45	1,3	6,50	3,83	2,21	3,02	2,80
	A	70	9,9	7,07	11,9	5,88	8,83	7,93
	B	290	40,8	7,11	45,5	6,37	30,6	9,48
	C	1246	159,5	7,81	161,5	7,72	—	—
LU	W	6,98	1,36	5,13	3,83	1,82	3,48	2,01
	A	48,1	10,02	4,80	7,43	6,47	6,6	7,29
	B	203	42	4,83	22,4	9,06	19,45	10,44
	C	819	163,9	5,00	73,8	11,10	59,51	13,76

В тесте LU всего 107 тесно-гнездовых циклов, которые можно вычислить параллельно. Из них 2 цикла имеют зависимость по трем отображаемым измерениям. В полученной Fortran-программе 4500 строк в фиксированном формате, 171 из которых — директивы DVMH. Для данного теста на уровне исходного текста была сделана такая же оптимизация, как и примененная в тесте BT. Также временные массивы, используемые в двух больших циклах с зависимостью, были удалены из цикла, а их выражения подставлены с целью уменьшения чтений из глобальной памяти GPU и сокращения объема занимаемой памяти.

Тестирование производилось на суперкомпьютере K100, имеющим процессоры Intel Xeon X5670 и GPU NVIDIA Tesla C2050 с включенным ECC и на локальной машине с процессором AMD Phenom II x4 и GPU NVIDIA GeForce GTX 560 Ti без ECC. Данные GPU имеют схожие характеристики, что позволит оценить влияние ECC на время вычисления. Последовательные версии программ были выполнены на суперкомпьютере K100. Также для сравнения были получены времена параллельных DVM программ, исполненных с использованием 12 процессорных ядер (один узел K100).

В табл. 3 и на рис. 4, рис. 5, рис. 6 приведены результаты тестирования производительности для этих тестов (для каждого варианта запуска бралась программа, показывающая лучший результат).

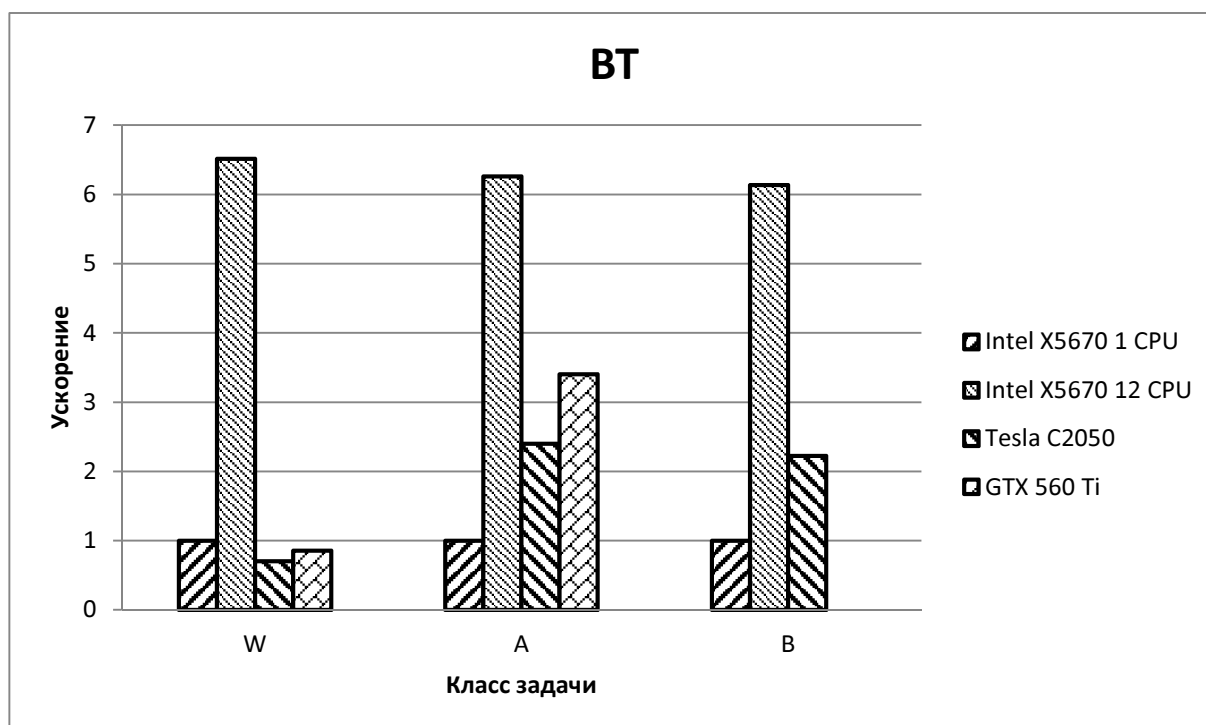


Рис. 4. Эффективность распараллеливания теста BT

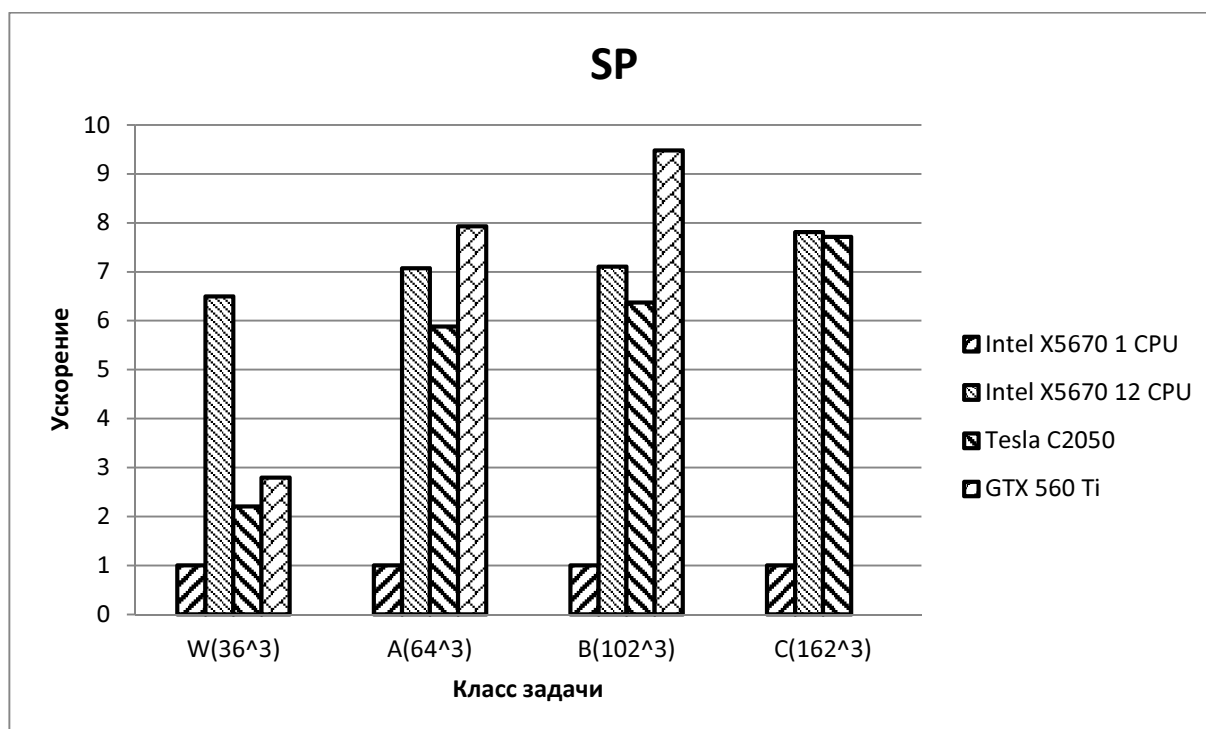


Рис. 5. Эффективность распараллеливания теста SP

## Заключение

В статье рассмотрена проблема отображения циклов с зависимостями на графические процессоры. Предложен метод выполнения DVMH-программ с такими циклами на GPU, в том числе механизм автоматического переупорядочивания массивов для ускорения доступа к его элементам в памяти GPU. DVM-система и язык Fortran-DVMH были расширены поддержкой циклов с зависимостями на GPU. Проведена апробация на те-

стах NAS, которая показывает результаты, близкие к результатам оптимизированных вручную вариантов данных тестов, описанным в [11], а также заметный выигрыш в сравнении с [12].

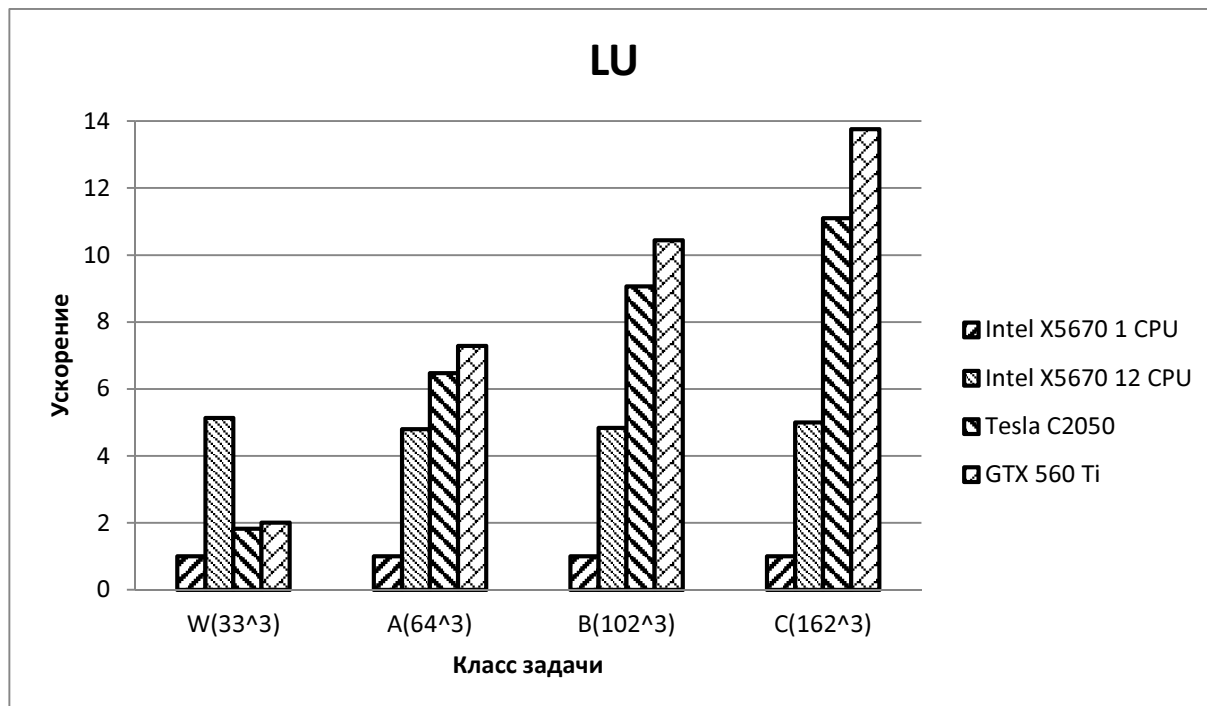


Рис. 6. Эффективность распараллеливания теста LU

В будущих исследованиях планируется повысить эффективность выполнения DVMH-программ с регулярными зависимостями по данным при использовании нескольких графических ускорителей.

*Исследование выполнено при финансовой поддержке грантов РФФИ № 11-01-00246, 12-01-33003 мол\_а\_вед, 12-07-31204-мол\_а и гранта Президента РФ НШ-4307.2012.9.*

## Литература

1. Top500 List – November 2012 TOP500 Supercomputer Sites. URL: <http://top500.org/list/2012/11/> (дата обращения 01.12.2012).
2. High Performance Fortran. URL: <http://hpff.rice.edu/> (дата обращения 01.12.2012).
3. Параллельное программирование в системе DVM. Языки Fortran-DVM и C-DVM / Н.А. Коновалов, В.А. Крюков, А.А. Погребцов и др. // Труды Международной конференции «Параллельные вычисления и задачи управления» (РАСО'2001). — Москва, 2001. — С. 140–154.
4. Fortran DVM – язык разработки мобильных параллельных программ / Н.А. Коновалов, В.А. Крюков, С.Н. Михайлов, А.А. Погребцов // Программирование. — 1995. — № 1. — С. 49–54.
5. Коновалов, Н.А. C-DVM – язык разработки мобильных параллельных программ / Н.А. Коновалов, В.А. Крюков, Ю.Л. Сазанов // Программирование. — 1999. — № 1. — С. 54–65.
6. Dolbeau, R. HMPP™: A Hybrid Multi-core Parallel Programming Environment / R. Dolbeau, S. Bihan, F. Bodin. URL: <http://www.caps-entreprise.com/wp->

- content/uploads/2012/08/caps-hmpp-gpgpu-Boston-Workshop-Oct-2007.pdf (дата обращения 02.12.2012).
7. The Portland Group. PGI Accelerator Programming Model for Fortran & C. URL: [http://www.pgroup.com/lit/whitepapers/pgi\\_accel\\_prog\\_model\\_1.3.pdf](http://www.pgroup.com/lit/whitepapers/pgi_accel_prog_model_1.3.pdf) (дата обращения 02.12.2012).
  8. OpenACC. URL: <http://www.openacc-standard.org/> (дата обращения 01.12.2012).
  9. Han, T.D. *hiCUDA: High-Level GPGPU Programming* / T.D. Han, T.S. Abdelrahman. // IEEE Transactions on Parallel and Distributed Systems. — 2011. — Vol. 22, No. 3 — P. 78–90.
  10. Расширение DVM-модели параллельного программирования для кластеров с гетерогенными узлами / В.А. Бахтин, М.С. Клинов, В.А. Крюков и др. // Вестник Южно-Уральского государственного университета, серия «Математическое моделирование и программирование». — Челябинск: Издательский центр ЮУрГУ, 2012. — Вып. 12 — № 18 (277). — С. 82–92.
  11. Pennycook, S.J. Performance Analysis of a Hybrid MPI/CUDA Implementation of the NAS-LU Benchmark / S.J. Pennycook, S.D. Hammond, S.A. Jarvis, G.R. Mudalige // ACM SIGMETRICS Performance Evaluation Review – Special issue on the 1st international workshop on performance modeling, benchmarking and simulation of high performance computing systems (PMBS 10). — 2011. — Vol. 38, Issue 4. — P. 23–29.
  12. Seo, S. Performance Characterization of the NAS Parallel Benchmarks in OpenCL / S. Seo, G. Jo, J. Lee // 2011 IEEE International Symposium on. Workload Characterization (IISWC). — 2011. — P. 137–148.

Бахтин Владимир Александрович, кандидат физико-математических наук, Институт прикладной математики им. М.В. Келдыша РАН (г. Москва, Российская Федерация), [bakhtin@keldysh.ru](mailto:bakhtin@keldysh.ru).

Колганов Александр Сергеевич, Институт прикладной математики им. М.В. Келдыша РАН (г. Москва, Российская Федерация), [alex-w900i@yandex.ru](mailto:alex-w900i@yandex.ru).

Крюков Виктор Алексеевич, доктор физико-математических наук, профессор, Институт прикладной математики им. М.В. Келдыша РАН (г. Москва, Российская Федерация), [krukov@keldysh.ru](mailto:krukov@keldysh.ru).

Поддериюгина Наталия Викторовна, кандидат физико-математических наук, Институт прикладной математики им. М.В. Келдыша РАН (г. Москва, Российская Федерация), [konov@keldysh.ru](mailto:konov@keldysh.ru).

Притула Михаил Николаевич, Институт прикладной математики им. М.В. Келдыша РАН (г. Москва, Российская Федерация), [pritmick@yandex.ru](mailto:pritmick@yandex.ru).

## MAPPING DVMH-PROGRAMS WITH REGULAR DEPENDENCIES ONTO CLUSTERS WITH GPU

**V.A. Bakhtin**, Keldysh Institute of Applied Mathematics Russian Academy of Sciences (Moscow, Russian Federation),

**A.S. Kolganov**, Keldysh Institute of Applied Mathematics Russian Academy of Sciences (Moscow, Russian Federation),

**V.A. Krukov**, Keldysh Institute of Applied Mathematics Russian Academy of Sciences (Moscow, Russian Federation),

**N.V. Podderugina**, Keldysh Institute of Applied Mathematics Russian Academy of Sciences (Moscow, Russian Federation),

**M.N. Pritula**, Keldysh Institute of Applied Mathematics Russian Academy of Sciences (Moscow, Russian Federation),

In the 2011 year DVMH programming model for new heterogeneous and hybrid supercomputer systems (or DVM for Heterogeneous systems) was introduced in the Keldysh Institute for Applied Mathematics of RAS. The developed high-level programming languages were based on standard Fortran and C programming languages, but extended with the directives for mapping the program onto a parallel computer. The directives are represented as special comments (or pragmas). The paper describes problems and methods for mapping loops, which have dependencies to the GPU. Efficiency of the developed Fortran DVMH parallel programs with regular dependencies is demonstrated.

*Keywords: DVM for Heterogeneous systems, Fortran DVMH, hybrid computational systems with accelerators, GPU, CUDA.*

### References

1. Top500 List – November 2012 TOP500 Supercomputer Sites. URL: <http://top500.org/list/2012/11/> (accessed 01.12.2012).
2. High Performance Fortran. URL: <http://hpff.rice.edu/> (accessed 01.12.2012).
3. Konovalov N.A., Krukov V.A., Pogrebtsov A.A., Podderugina N.V., Sazanov Y.L. Parallel'noe programmirovaniye v sisteme DVM. Yazyki Fortran-DVM i C-DVM [Parallel programming in the DVM system. Fortran-DVM and C-DVM languages]. Trudy Mezhdunarodnoy konferencii "Parallel'nye vychisleniya i zadachi upravleniya" (PACO'2001) [Proceedings of International conference "Parallel computations and control problems" (PACO'2001)]. Moscow, 2001. P. 140–154.
4. Konovalov N.A., Krukov V.A., Mihailov S.N., Pogrebtsov A.A. Fortran DVM – yazyk razrabotki mobil'nyh parallel'nyh programm [Fortran DVM – a language for mobile parallel programs development]. Programmirovaniye [Programming]. 1995. No. 1. P. 49–54.
5. Konovalov N.A., Krukov V.A., Sazanov Y.L. C-DVM – yazyk razrabotki mobil'nyh parallel'nyh programm [C-DVM – a language for mobile parallel programs development]. Programmirovaniye [Programming]. 1999. No. 1. P. 54–65.
6. Dolbeau R., Bihan S., Bodin F. HMPP™: A Hybrid Multi-core Parallel Programming Environment URL: <http://www.caps-entreprise.com/wp-content/uploads/2012/08/caps-hmpp-gpgpu-Boston-Workshop-Oct-2007.pdf> (accessed 02.12.2012).

7. The Portland Group. PGI Accelerator Programming Model for Fortran & C. URL: [http://www.pgroup.com/lit/whitepapers/pgi\\_accel\\_prog\\_model\\_1.3.pdf](http://www.pgroup.com/lit/whitepapers/pgi_accel_prog_model_1.3.pdf) (accessed 02.12.2012).
8. OpenACC. URL: <http://www.openacc-standard.org/> (accessed 01.12.2012).
9. Han T.D., Abdelrahman T.S. *hiCUDA: High-Level GPGPU Programming*. IEEE Transactions on Parallel and Distributed Systems. 2011. Vol. 22, No. 3. P. 78–90.
10. Bakhtin V.A., Klinov M.S., Krukov V.A., Podderiyugina N.V., Pritula M.N., Sazanov Y.L. *Rasshirenie DVM-modeli parallel'nogo programmirovaniya dlya klasterov s geterogennymi uzlami [Extension of the DVM parallel programming model for clusters with heterogeneous nodes]*. Vestnik Yuzhno-Ural'skogo gosudarstvennogo universiteta, seriya "Matematicheskoe modelirovanie i programmirovanie". Chelyabinsk, Publishing of the South Ural State University, 2012. No. 18 (277), Issue 12. P. 82–92.
11. Pennycook S.J., Hammond S.D., Jarvis S.A., Mudalige G.R. Performance Analysis of a Hybrid MPI/CUDA Implementation of the NAS-LU Benchmark. ACM SIGMETRICS Performance Evaluation Review – Special issue on the 1st international workshop on performance modeling, benchmarking and simulation of high performance computing systems (PMBS 10). 2011. Vol. 38, Issue 4. P. 23–29.
12. Seo S., Jo G., Lee J. Performance Characterization of the NAS Parallel Benchmarks in OpenCL. 2011 IEEE International Symposium on. Workload Characterization (IISWC). 2011. P. 137–148.

*Поступила в редакцию 18 октября 2013 г.*



# AstroPhi: ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ МОДЕЛИРОВАНИЯ ДИНАМИКИ АСТРОФИЗИЧЕСКИХ ОБЪЕКТОВ НА ГИБРИДНЫХ СУПЕРЭВМ, ОСНАЩЕННЫХ УСКОРИТЕЛЯМИ Intel Xeon Phi<sup>1</sup>

*И.М. Куликов, И.Г. Черных, Б.М. Глинский*

В статье представлен новый сверхмасштабируемый программный комплекс AstroPhi для моделирования динамики астрофизических объектов на гибридных суперЭВМ, оснащенных ускорителями Intel Xeon Phi. Численный метод решения газодинамических уравнений основан на специально адаптированной для реализации на множестве ускорителей комбинации метода крупных частиц и метода Годунова. Для решения уравнения Пуассона используется быстрое преобразование Фурье. Программная реализация была отдельно протестирована на газодинамических задачах, на задаче решения уравнения Пуассона и на классических задачах гравитационной газовой динамики. Показано ускорение программного комплекса при использовании ускорителей Intel Xeon Phi, уточнено понятие масштабируемости при использовании ускорителей. Представлены результаты моделирования коллапса астрофизических объектов.

*Ключевые слова:* математическое моделирование, параллельные вычисления, ускорители Intel Xeon Phi, астрофизика.

## Введение

Процессы коллапса астрофизических объектов в настоящее время активно исследуются теоретически в связи с появлением значительного числа наблюдательных данных. Явление коллапса имеет место как на начальной стадии звездной эволюции, так и на конечной стадии эволюции звезд (взрывы сверхновых с коллапсирующим ядром) [1]. Математическое моделирование играет более чем важную роль в теоретическом исследовании таких процессов. С каждым днем требования к астрофизическим моделям все возрастают и модели, которые несколько лет назад были актуальными, сейчас уже считаются устаревшими. Усложнение астрофизических моделей требует использования все больших вычислительных ресурсов, а, следовательно, модификации и создания новых вычислительных методов и параллельных алгоритмов для решения таких задач [2].

В последние два десятилетия из широкого диапазона газодинамических численных методов для решения нестационарных трехмерных астрофизических задач используются два основных подхода. Это лагранжев подход, в основном представленный SPH-методом [3, 4] (Smoothed Particle Hydrodynamics) и эйлеров подход с использованием адаптивных сеток или AMR [5, 6] (Adaptive Mesh Refinement). В последние пять лет появился ряд программных пакетов с использованием комбинации лагранжева и эйлерова подходов. Основной проблемой SPH метода является поиск соседей и организация их гравитационного взаимодействия. Для эффективного решения этой задачи были разработаны ряд алгоритмов. Например, particle-particle/particle-mesh или P<sup>3</sup>M метод [7], адаптация P<sup>3</sup>M метода с использованием

<sup>1</sup>Статья рекомендована к публикации программным комитетом Международной суперкомпьютерной конференции «Научный сервис в сети Интернет: все грани параллелизма — 2013».

иерархичности расчетной сетки AP<sup>3</sup>M [8], tree алгоритм [9], комбинация tree алгоритма и particle-mesh подхода Tree-PM метод [10]. Для решения уравнения Пуассона в сеточных методах используются в основном метод сопряженных градиентов (CGM), метод быстрого преобразования Фурье (FFT), метод последовательной верхней релаксации (SOR) и метод Федоренко [11] или многосеточный метод (MG).

Для численного решения газодинамических задач широкое применение получил метод Годунова [12], основным структурным элементом которого является задача о распаде произвольного разрыва (задача Римана) с параметрами газа в соседних ячейках разностной сетки. Как правило, параметры газа в соседних ячейках достаточно близки, что создает благоприятные условия для применения упрощенного алгоритма решения задачи о распаде разрыва. Различные алгоритмы получения приближенного решения задачи Римана дали большой класс методов [13, 14]. Основными методами являются методы типа Куранта–Изаксона–Риса [15] и Роу [16], которые строятся на основе использования различным образом линеаризованных гиперболических систем уравнений, Ошера [17], где решения задачи Римана строятся только с использованием волн Римана.

Основоположным подходом к оценке скоростей волн является двухволновой метод Хартена–Лакса–Ван Лиры (известный в литературе как HLL) [18], в котором учитываются левые и правые разрывы, без рассмотрения контактного разрыва. В схеме HLL использующей консервативные переменные, предложен простой, но эффективный способ выбора скоростей движения этих волн по максимальным наклонам характеристик в соседних ячейках разностной сетки. При этом веер волн разрежения заменяется скачком, но со скоростью распространения соответствующей максимальному наклону характеристик в этой волне разрежения. Поэтому этот выбор исключил проблему расчета «звуковой» точки при смене знака характеристик. Расчет ударных волн также проводится со скоростью превышающей точное значение. В этой связи схема HLL эффективна при расчете ударных волн и зон разрежения. Однако, в расчете энтропийных скачков методом установления принято допущение приводит к неприемлемому «размазыванию» контактного разрыва. Существуют также модификации HLL, такие как HLLC [19], где особо учитываются крайние собственные числа линеаризованной задачи, и HLLC [20], где производится дополнительный учет центрального разрыва, движущегося со скоростью равной центральному собственному значению линеаризованной задачи Римана. Также на основе метода Годунова были сделаны реализации высокого порядка – монотонная противопотоковая схема второго порядка точности MUSCL [21] и TVD схемы [25], третьего порядка кусочно-параболический метод PPM [5]. Однако, что понимается под высоким порядком точности в случае разрывных решений, не очень понятно [26].

В ходе адаптации метода SPH к различным космологическим задачам возникло множество модификаций алгоритма. Для определения гидродинамических величин очень важен выбор радиуса сглаживания, который определяет количество влияющих на частицу соседей. Одним из свойств метода является то, что для получения верных решений необходимо сохранение числа соседей почти равным для всех частиц расчетной области, что не выполняется в случае коллапса. Если число соседей у всех элементов различается на несколько частиц, то результаты расчетов нельзя считать достоверными. Для преодоления такого недостатка метода вводится адаптивный радиус сглаживания, определяемый по числу соседей. Такое определение радиуса сглаживания приводит к ряду вычислительных сложностей, поэтому многие реализации метода допускают большие отклонения в числе со-

седей от частицы к частице. Таким образом, определение радиуса сглаживания допускает возможность выбора, а значит, оказывает влияние на решение. Авторы программ, использующие AMR, обычно задают величину наиболее подробного шага сетки по пространству, хотя сложно оценить степень необходимого сгущения сетки особенно в случае коллапса, поскольку в зависимости от степени адаптации сетки к решению в случае возникновения особенностей могут оставаться проблемы, характерные для сеток с линиями координат, располагающимися вдоль границ областей решения. Несомненно, разработка эйлеровых сеточных методов, не допускающих влияния сеточных линий на решение, позволит отказаться от методики AMR, а, значит, избежать всех вышеперечисленных недостатков этого подхода. Создание таких методов представляет собой более сложную задачу, чем построение адаптивных сеток, но, тем не менее, возможно. Как известно, уравнения газовой динамики инвариантны относительно некоторой группы точечных преобразований в пространстве независимых и зависимых переменных. Такая инвариантность является следствием инвариантности законов сохранения, из которых вытекают уравнения газовой динамики. Использование расчетной сетки неизбежно вносит неинвариантность в алгоритм расчета, что может оказывать влияние, например, на расчеты особенностей потока (ударных волн, контактных границ, слабых разрывов), которые движутся под различными углами к линиям сетки. Построению инвариантных относительно поворота разностных схем посвящены работы одного из авторов пакета [48, 49]

В рамках лагранжева подхода на основе SPH метода были разработаны пакеты Hydra [8], Gasoline [22], GrapeSPH [23], GADGET [24]. В рамках эйлерова подхода (в том числе и с использованием AMR) были разработаны пакеты NIRVANA [27], FLASH [28], ZEUS-MP [29], ENZO [6], RAMSES [30], ART [31], Athena [32], Pencil Code [33], Heracles [39], Orion [40], Pluto [41], CASTRO [42]. Эйлеров подход с использованием AMR был впервые использован на гибридных суперкомпьютерах, оснащенных графическими ускорителями, в пакете GAMER [34]. Пакет BETH-*Hydro* [35], AREPO [36], CHIMERA [37] и авторский пакет PEGAS [38] основаны на комбинации лагранжева и эйлерова подходов.

Большое количество программных реализаций и численных методов говорит об актуальности исследований в области разработки новых методов и их программных реализаций для решения задач астрофизики. Кроме того, несмотря на развитие астрофизических пакетов в сторону петафлопсных вычислений, таких как PetaGADGET [43], Enzo-P [44], PetaART [45] нужно отметить фундаментальные ограничения в масштабируемости AMR и SPH подходов, которые используются в основном числе программных пакетов для решения задач астрофизики [46, 47].

Целью данной статьи является описание модификации численного метода и особенностей реализации пакета *AstroPhi* на гибридных суперкомпьютерах, оснащенных ускорителями Intel Xeon Phi, исследование его масштабируемости и исследование задач коллапса астрофизических объектов. Статья организована следующим образом. В разделе 1 описана численная схема решения уравнений гравитационной газовой динамики. Раздел 2 содержит описание параллельной реализации численной схемы. В разделе 3 приведена верификация программной реализации. Раздел 4 посвящен вычислительным экспериментам по моделированию процесса коллапса астрофизических объектов. В заключении приведены выводы по созданному программному пакету и обозначены направления дальнейшего его развития.

## 1. Описание численной схемы

Будем рассматривать трехмерную модель динамики самогравитирующего газа в декартовых координатах, включающих в себя расширенную систему уравнений газовой динамики в дивергентной форме, замкнутую уравнением состояния для идеального газа. Система уравнений газовой динамики дополнена уравнением Пуассона для гравитационного потенциала и вкладом в потенциал от центрального тела. Уравнения записаны в безразмерном виде

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \vec{v}) &= 0, \\ \frac{\partial \rho \vec{v}}{\partial t} + \operatorname{div}(\vec{v} \rho \vec{v}) &= -\operatorname{grad}(p) - \rho \operatorname{grad}(\Phi + \Phi_0), \\ \frac{\partial \rho E}{\partial t} + \operatorname{div}(\rho E \vec{v}) &= -\operatorname{div}(p \vec{v}) - (\rho \operatorname{grad}(\Phi + \Phi_0), \vec{v}), \\ \frac{\partial \rho \varepsilon}{\partial t} + \operatorname{div}(\rho \varepsilon \vec{v}) &= -(\gamma - 1) \rho \varepsilon \operatorname{div}(\vec{v}), \\ \Delta \Phi &= 4\pi \rho, \\ p &= (\gamma - 1) \rho \varepsilon,\end{aligned}$$

где  $p$  — давление,  $\rho$  — плотность,  $\vec{v}$  — вектор скорости,  $\rho E$  — плотность полной энергии,  $\Phi$  — собственный гравитационный потенциал,  $\Phi_0$  — вклад в гравитационный потенциал от центрального тела,  $\varepsilon$  — внутренняя энергия,  $\gamma$  — показатель адиабаты. В качестве основных характерных параметров выбраны радиус солнца  $L = R_\odot$ , масса солнца  $M_0 = M_\odot$ , гравитационная постоянная  $G = 6,67 \cdot 10^{-11}$  Н·м<sup>2</sup>/кг.

### 1.1. Метод решения уравнений газовой динамики

Введем в трехмерной области решения равномерную прямоугольную сетку с ячейками  $x_i = ih_x$ ,  $i = 1, \dots, I_{\max}$ ,  $y_k = kh_y$ ,  $k = 1, \dots, K_{\max}$ ,  $z_l = lh_z$ ,  $l = 1, \dots, L_{\max}$ , где  $h_x$ ,  $h_y$ ,  $h_z$  — шаги сетки,  $I_{\max}$ ,  $K_{\max}$ ,  $L_{\max}$  — количество узлов сетки по направлениям  $x$ ,  $y$ ,  $z$ :  $h_x = x_{\max}/I_{\max}$ ,  $h_y = y_{\max}/K_{\max}$ ,  $h_z = z_{\max}/L_{\max}$ . За основу метода решения системы уравнений газовой динамики выбран метод крупных частиц [38], уже хорошо зарекомендовавший себя в ходе решения астрофизических задач [2].

Исходная система газодинамических уравнений решается в два этапа. Система уравнений на первом, эйлеровом, этапе описывает процесс изменения параметров газа в произвольной области течения за счет работы сил давления, а также за счет разности потенциалов и охлаждения. Для исключения влияния направлений координатных линий использованы элементы операторного подхода [48]. В его основе определение плотности, давления, потенциала и импульса в ячейках, в узлах ячеек размещаются только вектор скорости. Для дискретных аналогов компонент скорости, определенных в узлах сетки, применяется функция осреднения в ячейку. Значения давления и скорости на всех границах ячеек  $P$  и  $V$  суть точное решение линеаризованной системы уравнений эйлерова этапа по каждому из направлений осей координат без учета вклада потенциала и охлаждения:

$$\begin{aligned}\frac{\partial v}{\partial t} &= -\frac{1}{\rho} \frac{\partial p}{\partial n}, \\ \frac{\partial p}{\partial t} &= -(\gamma - 1)p \frac{\partial v}{\partial n}.\end{aligned}$$

Эта система на каждой границе ячеек уже является линейной гиперболической системой и имеет аналитическое решение:

$$V = \frac{v_L + v_R}{2} + \frac{p_L - p_R}{2} \sqrt{\frac{\rho_L + \rho_R}{\rho_L \rho_R (\gamma - 1) (p_L + p_R)}},$$

$$P = \frac{p_L + p_R}{2} + \frac{v_L - v_R}{2} \sqrt{\frac{\rho_L \rho_R (\gamma - 1) (p_L + p_R)}{\rho_L + \rho_R}},$$

где  $f_L, f_R$  — значения соответствующих функций  $(\rho, v, p)$  справа и слева от границы ячеек. Эти значения и используются в схеме эйлерового этапа.

Система уравнений на втором, лагранжевом, этапе, содержит дивергентные слагаемые вида

$$\frac{\partial f}{\partial t} + \text{div}(f\vec{v}) = 0$$

и отвечает за процесс адвективного переноса всех газодинамических величин  $f$ . В исходном варианте численного метода использовался подход, связанный с вычислением вклада в соседние ячейки со схемной скоростью [49]. Однако, такой подход совершенно не годится в случае использования ускорителей. Для этого рассмотрим решение следующей одномерной постановки предыдущего уравнения:

$$\frac{f_{ikl}^{n+1} - f_{ikl}^n}{\tau} + \frac{F_{i+1/2,kl}^{n+1/2} - F_{i-1/2,kl}^{n+1/2}}{h} = 0,$$

где величина  $F_{i+1/2,kl}^{n+1/2}$  определяется по правилу полной деформации ячейки (см. рис. 1):

$$F_{i+1/2,kl}^{n+1/2} = \frac{\sum v_{i+1/2,k\pm 1,l\pm 1} f_{ikl}^+}{4},$$

$$f_{ikl}^+ = \begin{cases} f_{ikl}, v_{i+1/2,k\pm 1,l\pm 1} > 0, \\ f_{i+1,kl}, v_{i+1/2,k\pm 1,l\pm 1} \leq 0. \end{cases}$$

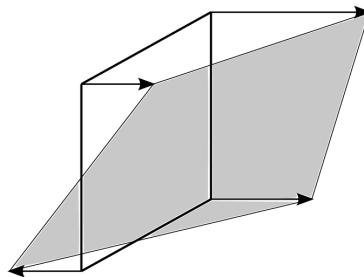


Рис. 1. Поток соответствующий газодинамической величины

На каждом временном шаге производится корректировка баланса энергий [50]. С этой целью осуществляется перенормировка схемных скоростей переноса массы, импульса и двух видов энергий на лагранжевом этапе метода таким образом, что происходит корректировка длины вектора скорости при неизменном направлении. Для этого все компоненты скорости в каждой ячейке области умножаются на множитель  $\kappa_{i,k,l}$ :

$$\kappa_{ikl} = \sqrt{\frac{2(E_{ikl} - \frac{1}{\rho_{ikl}} \frac{p_{ikl}}{\gamma-1})}{v_{x,ikl}^2 + v_{y,ikl}^2 + v_{z,ikl}^2}}.$$

Такая модификация метода обеспечивает справедливость детального баланса энергий. Заметим, что разностная схема не становится полностью консервативной, поскольку коррекция скорости вносит погрешность в закон сохранения импульса. Модификация схемы позволяет производить расчет одного шага по времени независимо для всех ячеек расчетной области. Таким образом, будет полностью отсутствовать взаимодействие между ядрами ускорителя, что должно привести к большому ускорению при использовании Intel Xeon Phi.

## 1.2. Метод решения уравнения Пуассона

После реализации газодинамической системы уравнений решается уравнение Пуассона для гравитационного потенциала. Для его решения используется 27-точечный шаблон. Потенциал и плотность представляется в виде суперпозиции по собственным функциям оператора Лапласа. Получим следующую схему решения уравнения Пуассона в пространстве гармоник:

$$\Phi_{jmn} = \frac{4\pi h^2 \rho_{jmn}}{6(1 - (1 - \frac{2}{3} \sin^2(\frac{\pi j}{T}))(1 - \frac{2}{3} \sin^2(\frac{\pi m}{K}))(1 - \frac{2}{3} \sin^2(\frac{\pi n}{L}))}.$$

Таким образом схема решения уравнения Пуассона примет следующий вид:

1. Преобразование в пространство гармоник выражения.
2. Решение в пространстве гармоник уравнения.
3. Обратное преобразование из пространства гармоник функции потенциала.

Для перехода в пространство гармоник и обратно, которое состоит в нахождении коэффициентов перехода, воспользуемся быстрым преобразованием Фурье.

Краевые условия уравнения Пуассона определяют решение задачи, поэтому их постановка является достаточно важной проблемой. Известно, что в бесконечном удалении от объекта гравитационный потенциал может считаться нулевым. Краевые условия приходится ставить на конечном расстоянии от газового объекта. Для решения данной задачи был предложен следующий вариант: считать, что масса тела сосредоточена в центре рассматриваемой области и модуль потенциала обратно пропорционален расстоянию от рассматриваемой границы до центра области.

## 2. Описание параллельной реализации

Трехмерность модели и нестационарность задачи выдвигают строгие требования к экономичности используемых методов решения. В последнее время бурное развитие вычислительной техники позволило производить ресурсоемкие расчеты и получать физически оправданные результаты для трехмерных программ. Использование суперкомпьютеров позволяет использовать большие объемы данных, на порядки повышать производительность вычислений, а как следствие, и точность. Основные вычислительные затраты приходятся на решение гидродинамических уравнений, решение которых занимает 90 процентов времени (рис. 2).

### 2.1. Параллельная реализация гидродинамических уравнений

В основе параллельной реализации решения гидродинамических уравнений лежит многоуровневая одномерная декомпозиция расчетной области. По одной координате внешнее

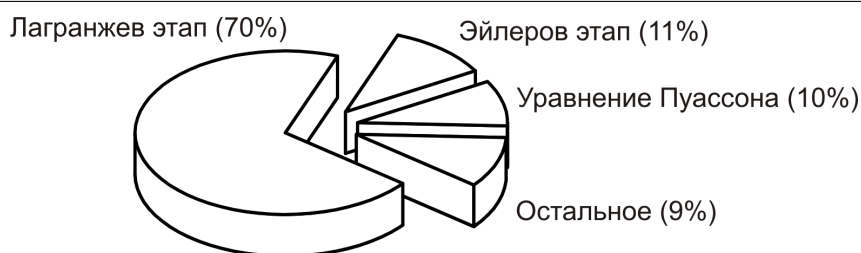


Рис. 2. Процентное соотношение вычислительных затрат на решение каждого из этапов

одномерное разрезание происходит средствами технологии MPI, внутри каждой подобласти разрезание происходит средствами OpenMP, адаптированного для MIC-архитектур (рис. 3).

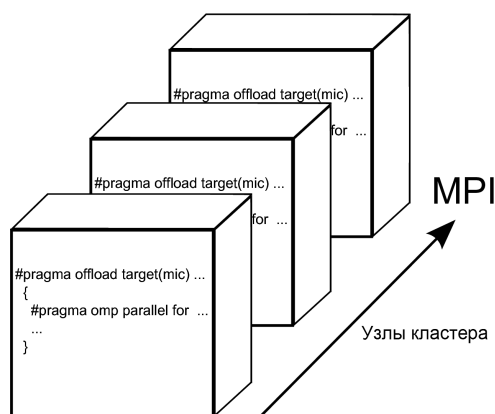


Рис. 3. Схема декомпозиции расчетной области для решения гидродинамических уравнений

Это связано с топологией и архитектурой гибридного суперЭВМ МВС-10П межведомственного суперкомпьютерного центра РАН, который был использован для вычислительных экспериментов. Модификация численного метода решения гидродинамических уравнений позволяет на каждом этапе численного метода независимо вычислять значения потоков через каждую ячейку. Декомпозиция области на каждом этапе осуществляется с перекрытием одного слоя граничных точек соседних областей.

## 2.2. Параллельная реализация уравнения Пуассона

Трехмерное параллельное быстрое преобразование Фурье выполняется с помощью процедуры из свободно распространяемой библиотеки FFTW. Способ распределения массивов также задается библиотекой. Перекрытие расчетных областей не требуется. В силу малых вычислительных затрат на решение уравнения Пуассона по сравнению с решением гидродинамических уравнений ускорители не использовались (на рис. 4 показано ускорение решения уравнения Пуассона для расчетной сетки  $1024^3$  в зависимости от числа процессов). Однако, в дальнейшем такая реализация, основанная на архитектуре библиотеки FFTW и адаптированная для решения задачи, обязательно будет сделана.

## 2.3. Эффективность параллельной реализации

Для определения эффективности гибридной реализации мы вводим следующие понятия масштабируемости. *SingleMIC performance* (сильная масштабируемость в рамках одного ускорителя Intel Xeon Phi) – уменьшение времени счета одного шага одной и той же

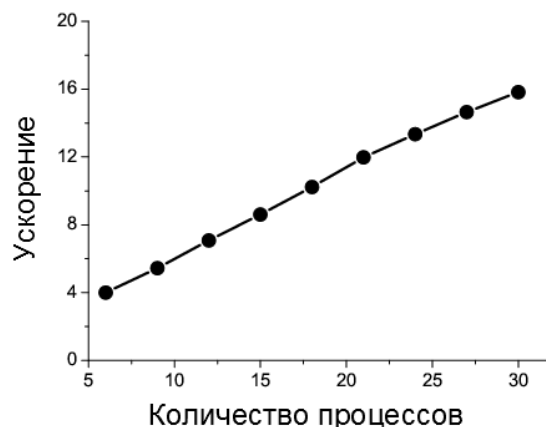


Рис. 4. Ускорение решения уравнения Пуассона

задачи при использовании большего числа ядер ускорителя. *MultiMIC performance* (слабая масштабируемость при использовании многих ускорителей Intel Xeon Phi) – сохранение времени счета одного шага одного и того же объема задачи при одновременном увеличении количества ускорителей. *FFTW performance* (сильная масштабируемость при использовании библиотеки FFTW) – уменьшение времени счета одного шага одной и той же задачи при использовании большего числа процессоров или ядер. Результаты эффективности программной реализации приведена на рис. 5 (на рисунке изображены эффективность параллельной реализации газодинамического этапа в зависимости от использованных ускорителей (а) и ускорение на одном Intel Xeon Phi для расчетной сетки  $128^3$  (б)).

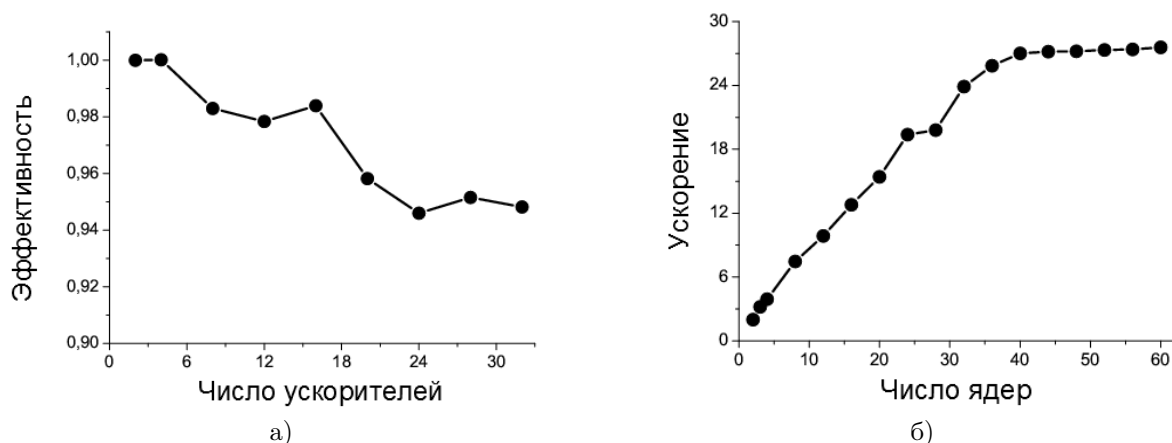


Рис. 5. Эффективность и ускорение решения газодинамических уравнений

Таким образом, для самого тяжелого – газодинамического этапа численного метода – было получено двадцатисемикратное ускорение при полном использовании ускорителя Intel Xeon Phi.

### 3. Верификация программной реализации

Так как в астрофизике математическое моделирование зачастую выступает единственной возможностью подтвердить или опровергнуть новые теории, то исследователи особен-



но нуждаются в применении надежных и заслуживающих доверия программ. Прежде чем представлять новые результаты моделирования, необходимо провести разнообразные тестовые расчеты для обоснования и верификации используемой программы. Верификация и обоснование – основные этапы развития для любой технологии, будь это пакет программ для математического моделирования или инструментарий для наблюдений. Для вычислительной технологии целью такого этапа тестирования является оценка правомерности и точности моделирования.

### 3.1. Тесты Годунова

В области вычислительной гидродинамики проделана большая работа по обоснованию и верификации. В процессе создания комплекса программ проводилась верификация численного алгоритма на тестах с решениями из специализированного банка данных. Одна из проблем – моделирование ударной волны. Известно, что различные методы по-разному моделируют эту область, а именно либо с осцилляциями, либо с диссипацией. В случае гравитационной газовой динамики осцилляции являются менее желательными, так как любая, в общем случае, случайная волна плотности будет стягивать на себя соседний газ, что приведет к образованию нефизичных флуктуаций газодинамических параметров. Также существует проблема моделирования существенной области разрежения. Известно, что многие методы дают нефизичный рост внутренней энергии в этом месте. Кроме этого стандартной проверкой робастности метода служит огромный начальный перепад давления (пять десятичных порядков), который должен выявить способность метода устойчиво моделировать сильные возмущения с возникновением быстро распространяющихся ударных волн. На всех этих тестах (см. табл. 1) была верифицирована данная программная реализация.

Таблица

Начальная конфигурация ударной  
трубы

Параметр	Тест 1	Тест 2	Тест 3
$\rho_L$	2	1	1
$v_L$	0	-2	0
$p_L$	2	0,4	1000
$\rho_R$	1	1	1
$v_R$	0	2	0
$p_R$	1	0,4	0,01
$x_0$	0,5	0,5	0,5
$t$	0,2	0,15	0,012

Целью первого теста является определение правильности описания контактного разрыва. Большинство методов решения газодинамических уравнений дают либо осцилляцию, либо диффузию («размазывание» ударных волн). Авторский метод дает размазывание решения в области контактного разрыва, которое уменьшается с дроблением сетки. В ходе второго теста, газ с одинаковыми термодинамическими параметрами разлетается в разные стороны, образуя в центре существенную область разрежения. Тест выявляет способность физически правдоподобно моделировать такую ситуацию. Из литературы известно, что

многие методы дают ошибочный (нефизический) рост температуры в области сильного разрежения и как следствие, получаемое решение искажается. Авторский метод успешно моделирует область разрежения. Основная задача третьего теста – проверка устойчивости численного метода. Огромный перепад давления (5 десятичных порядков) должен выявить способность метода устойчиво моделировать сильные возмущения с возникновением быстро распространяющихся ударных волн. Результаты показывают, что имеют место малые осцилляции решения в области контактного разрыва. Так называемая волна-предшественник (ступенька на графике внутренней энергии на правом фронте ударной волны) отражена корректно, без размазывания, что говорит в пользу метода (см. рис. 6).

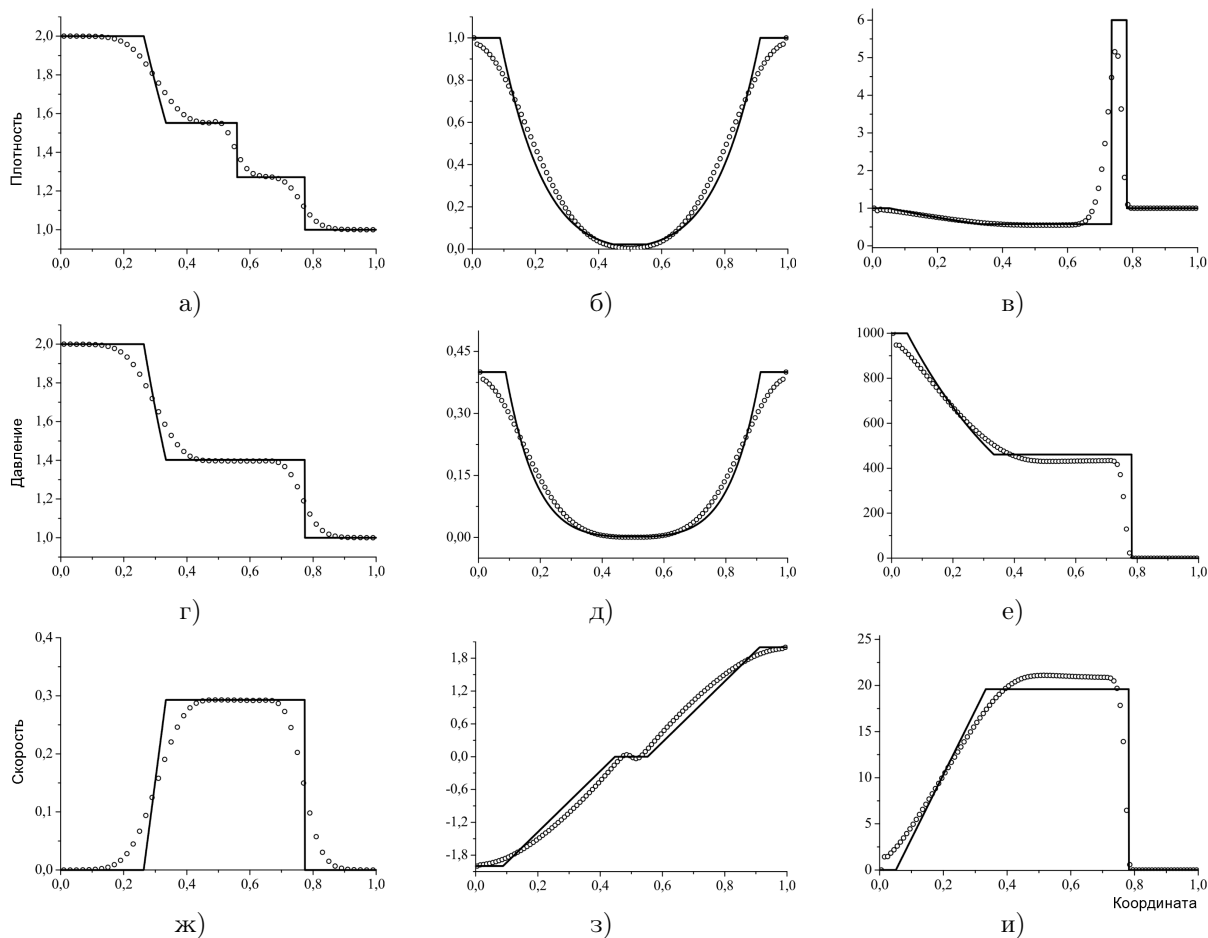


Рис. 6. Численное решение тестов Годунова

На рисунке 6 показаны распределения плотности, скорости и давления в результате моделирования первого теста (а, г, ж соответственно), второго теста (б, д, з соответственно), третьего теста (в, е, и соответственно), сплошной линией обозначено точное решение, точками обозначен результат расчета.

### 3.2. Тест Аксенова

Рассмотрим систему уравнений одномерной газовой динамики в размерном виде:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = -\frac{1}{\rho} \frac{\partial p}{\partial x},$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} = 0,$$

$$\frac{p}{p_0} = \left( \frac{\rho}{\rho_0} \right)^\gamma,$$

где  $p$  – давление,  $\rho$  – плотность,  $u$  – скорость,  $\gamma$  – показатель адиабаты.

В качестве характерных величин выберем  $l$  – характерная длина,  $\rho_0$  – характерная плотность,  $p_0$  – характерное давление. В этом случае характерная скорость  $u_0 = \sqrt{\gamma p_0 / \rho_0}$ , и характерное время  $t_0 = l / \sqrt{\gamma p_0 / \rho_0}$ . Выбрав в качестве размерных величин  $l = 1$ ,  $p_0 = 1$ ,  $\rho_0 = 1$ ,  $\gamma = 3$  и введя обозначения  $\lambda = 1/(\gamma - 1)$ ,  $r = \rho^{1/2\lambda}$ ,  $z = u/2\lambda$ , с использованием подхода, описанного в [51], можно взять в качестве начальных данных  $r = 1 + 0,5 \cos x$ ,  $z = 0$ . Тогда периодическое решение на интервале  $[0; 2\pi]$  записывается в виде:

$$r = 1 + 0,5 \cos(x - zt) \cos(rt),$$

$$z = 0,5 \sin(x - zt) \sin(rt).$$

Легко проверить, что такое решение с учетом обезразмеривания удовлетворяет исходной системе уравнений. Для сравнения численного результата, полученного авторским методом, с аналитическим решением, выберем момент времени, когда хотя бы одна из функций имеет простой (явный) вид. Выберем в качестве такого момента  $t = \pi/2$ . Тогда  $r(x) = 1$ , а уравнение для  $z$  имеет простой вид:

$$z = 0,5 \sin(x - zt).$$

Результаты вычислительного эксперимента представлены на рис. 7 (на рисунке изображены распределения плотности (а) и скорости (б) в момент времени  $t = \pi/2$ , сплошной линией обозначено точное решение, точками обозначен результат расчета).

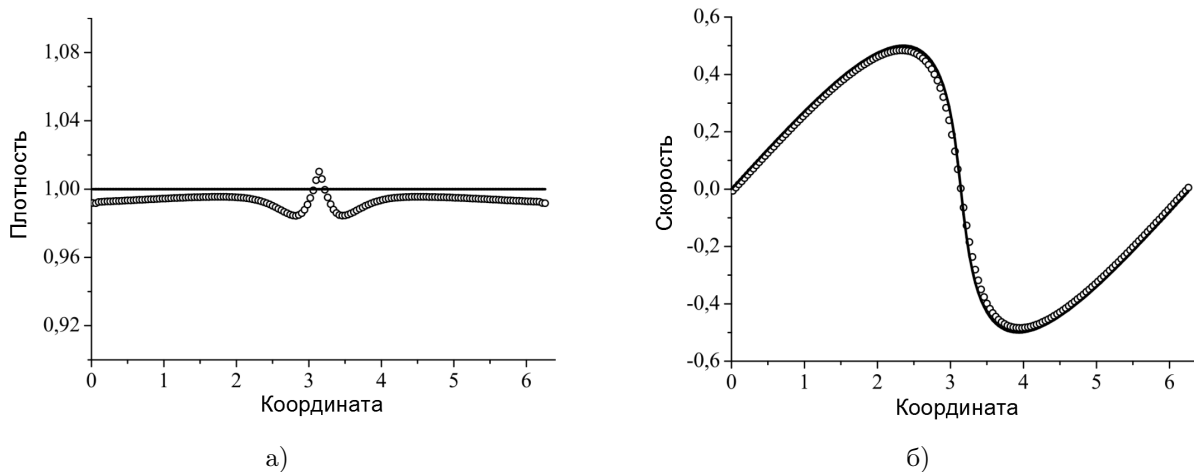


Рис. 7. Численное решение теста Аксенова

Из рисунков видно, что решение скорости ввиду его гладкости достаточно хорошо приближается численным решением, а график плотности имеет скачок в центре. Данный скачок имеет ту же природу, что и скачок температуры в третьем тесте Годунова, когда газ разлетается в разные стороны. Фактически в этом тесте имеет место энтропийный след, который образуется в результате «стекания» газа в эту область с нулевой скоростью. Полученную особенность имеет смысл исследовать для сравнения численных методов на таком тесте и возникающих особенностей при воспроизведении данной области. В любом случае стоит отметить, что данная особенность сосредоточена на конечном числе точек расчетной области и уменьшается при дроблении сетки.

### 3.3. Неустойчивость Кельвина–Гельмгольца и Релея–Тейлора

В основе физической постановки задачи лежит гравитационная неустойчивость, что приводит к математической некорректности по Адамару. Численный метод не может подавлять физическую неустойчивость. Для проверки корректного воспроизведения неустойчивых течений была сделана верификация численного метода на задачах о развитии неустойчивости Релея–Тейлора и Кельвина–Гельмгольца. В случае моделирования неустойчивости Релея–Тейлора проверяется возможность воспроизведения гравитационного термина. Неустойчивость Кельвина–Гельмгольца позволяет убедиться в возможности метода моделировать нелинейную гидродинамическую турбулентность.

Начальные условия для моделирования неустойчивости Релея–Тейлора:  $[-0,5; 0,5]^2$  – область моделирования,  $\gamma = 1,4$  – показатель адиабаты,

$$\rho_0(x) = \begin{cases} 1, & r \leq 0, \\ 2, & r > 0 \end{cases}$$

$p = 2,5 - \rho g y$  – равновесное давление,  $g$  – ускорение свободного падения,  $v_{y,0}(x, y) = A(y)[1 + \cos(2\pi x)][1 + \cos(2\pi y)]$ , где

$$A(y) = \begin{cases} 10^{-2}, & |y| \leq 0,01, \\ 0, & y > 0,01 \end{cases}$$

Начальные условия для моделирования неустойчивости Кельвина–Гельмгольца:  $[-0,5; 0,5]^2$  – область моделирования,  $\gamma = 1,4$  – показатель адиабаты,

$$\rho_0(x) = \begin{cases} 1, & r \leq 0, \\ 2, & r > 0 \end{cases}$$

$$v_x = \begin{cases} 0,5, & |y| \leq 0,25, \\ -0,5, & |y| > 0,25 \end{cases}$$

$p = 2,5$  – начальное давление,  $v_{y,0}(x, y) = A(y)[1 + \cos(8\pi x)][1 + \cos(8\pi y)]$ , где

$$A(y) = \begin{cases} 10^{-2}, & ||y| - 0,25| \leq 0,01, \\ 0, & ||y| - 0,25| > 0,01 \end{cases}$$

Результаты моделирования неустойчивости Кельвина–Гельмгольца и Релея–Тейлора представлены на рис. 8.

## 4. Моделирование процесса коллапса астрофизических объектов

Процессы коллапса астрофизических объектов в настоящее время активно исследуются теоретически в связи с появлением значительного числа наблюдательных данных. Явление коллапса имеет место как на начальной стадии звездной эволюции, так и на конечной стадии эволюции звезд (взрывы сверхновых с коллапсирующим ядром).

### 4.1. Сжатие невращающегося облака

Приведем результаты вычислительного эксперимента моделирования коллапса. Сначала необходимо оценить точность моделирования коллапса. Основным критерий правильно-

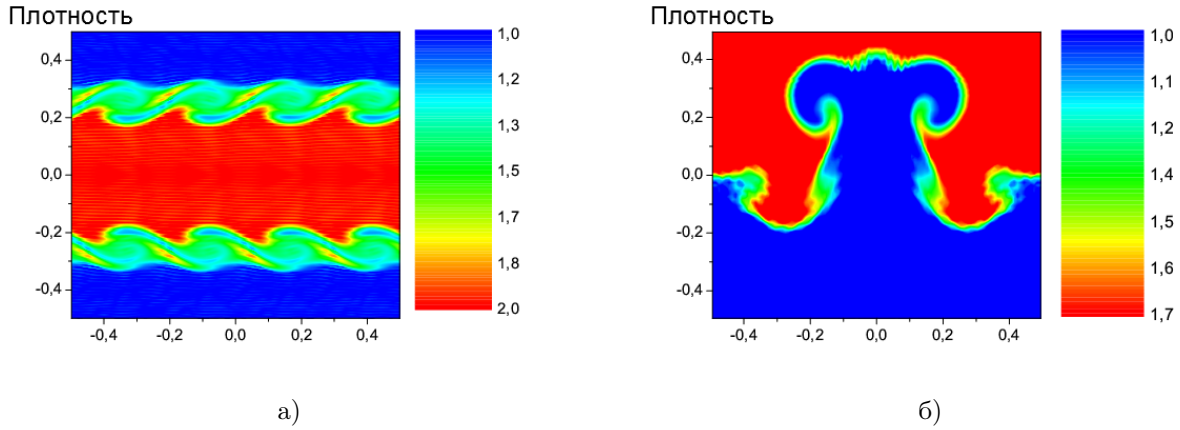


Рис. 8. Моделирование неустойчивости Кельвина–Гельмгольца (а) и Релея–Тейлора (б)

сти – поведение полной энергии системы. Затем необходимо провести сравнение профилей плотности, полученных с помощью реализаций методов FLIC (авторская реализация метода) и SPH. Профиль начального распределения плотности  $\rho(r) \simeq 1/r$ . Начальное распределение давления  $p = 0,1 \cdot \rho^\gamma$ , показатель адиабаты  $\gamma = 5/3$ .

Целью исследования точности моделирования коллапса является поведение полной энергии при дроблении сетки. Источником ошибки в законе сохранения полной энергии является конечная стадия коллапса, когда плотность и другие газодинамические величины увеличиваются в 10 или 100 раз. Подавляющая масса газа находится в шаре с радиусом  $R_{collaps} = 0,1 \cdot R_0$  (10 % от начального радиуса газового шара). Поэтому для моделирования этого процесса нам нужно иметь на радиусе  $R_{collaps}$  достаточное число ячеек для удовлетворительного моделирования процесса. Проведем сравнение поведения закона сохранения полной энергии при дроблении сетки (см. рис. 9).

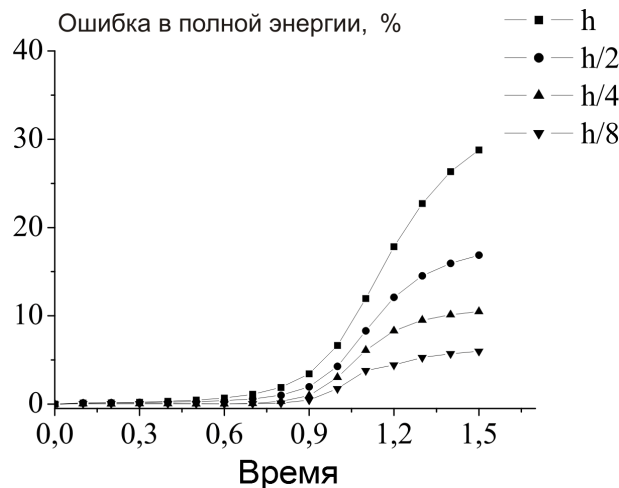


Рис. 9. Относительная погрешность полной энергии

Из рисунка видно, что при дроблении сетки относительная погрешность полной энергии уменьшается, а, значит, дробление сетки приведет к моделированию коллапса с наперед заданной точностью. На сетке  $512 \times 512 \times 512$  погрешность на уровне 5 %, что является удовлетворительным для качественного сравнения решения полученного FLIC методом с решением, полученным SPH методом.

## 4.2. Сжатие быстро вращающегося облака

В рамках исследования возможности моделирования коллапса вращающихся протозвездных облаков будем моделировать газовое облако, ограниченное сферой радиуса  $R_0 = 3,81 \cdot 10^{14}$  м, с массой  $M_g = 3,457 \cdot 10^{30}$  кг, с равномерным распределением плотности  $\rho = 1,492 \cdot 10^{-14}$  кг/м<sup>3</sup> и давления  $p = 0,1548 \cdot 10^{-10}$  Н/м<sup>2</sup>, вращающийся с угловой скоростью  $\omega = 2,008 \cdot 10^{-12}$  рад/с. Показатель адиабаты соответствует водороду  $\gamma = 5/3$ . Масса центрального тела  $M_\odot = 1,998 \cdot 10^{30}$  кг. В качестве размерных величин выберем следующие значения:  $L_0 = 3,81 \cdot 10^{14}$  м,  $\rho_0 = 1,492 \cdot 10^{-14}$  кг/м<sup>3</sup>,  $p = 0,1548 \cdot 10^{-7}$  Н/м<sup>2</sup>,  $v_0 = 1010$  м/с,  $t = 3,7 \cdot 10^{11}$  с,  $\omega_0 = 0,27 \cdot 10^{-11}$  рад/с. Тогда в безразмерных величинах задача ставится следующим образом:  $\rho = 1,0$  – плотность газового облака,  $p = 10^{-3}$  – давление в газовом облаке,  $\omega = 0,744$  – угловая скорость вращения,  $m_\odot = 2,42$  – масса центрального тела,  $\gamma = 5/3$  – показатель адиабаты,  $[0; 6,4]^3$  – расчетная область.

В ряде работ, посвященных коллапсированию протозвездных облаков, проводился поиск ответа, каким будет распределение плотности в экваториальной плоскости облака после коллапсирования. Четкого ответа так и не было получено. Основным результатом в данной задаче является поведение энергий (см. рис. 10) и торообразный профиль плотности в экваториальной плоскости.

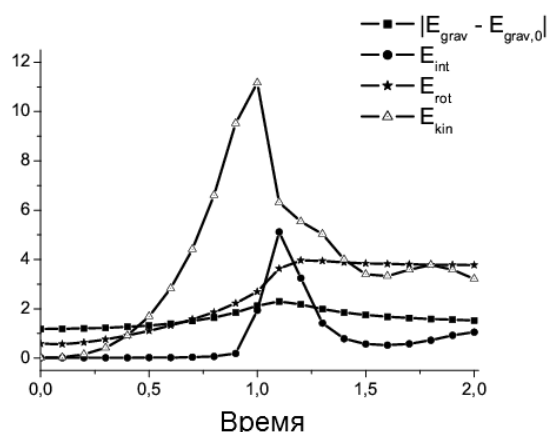


Рис. 10. Поведение различных видов энергии в задаче сжатия быстро вращающегося облака

Стоит отметить, что поведение энергий качественно, а до момента коллапса – количественно, – совпадают с результатами других авторов [1].

## 4.3. Сжатие вращающегося молекулярного облака

В рамках исследования возможности моделирования коллапса вращающихся молекулярных облаков будем моделировать газовое облако, ограниченное сферой радиуса  $R_0 = 100$  парсек, с массой  $M_g = 10^7 M_\odot$ , с распределением плотности  $\rho(r) \simeq 1/r$  и температуры  $T \approx 2000$  К, вращающийся с угловой скоростью  $\omega = 21$  км/с. Показатель адиабаты соответствует водороду  $\gamma = 5/3$ . Скорость звука  $c \approx 3,8$  км/с. В качестве размерных величин выберем следующие значения:  $L_0 = 100$  парсек,  $\rho_0 = 1,2 \cdot 10^{-18}$  кг/м<sup>3</sup>,  $v_0 = 21$  км/с. Тогда в безразмерных величинах задача ставится следующим образом:  $\rho = 1,0$  – плотность газового облака в центре,  $p = 2 \times 10^{-2}$  – давление в газовом облаке в центре,  $\omega = 1$  – угловая скорость вращения,  $\gamma = 5/3$  – показатель адиабаты,  $[0; 6,4]^3$  – расчетная область.

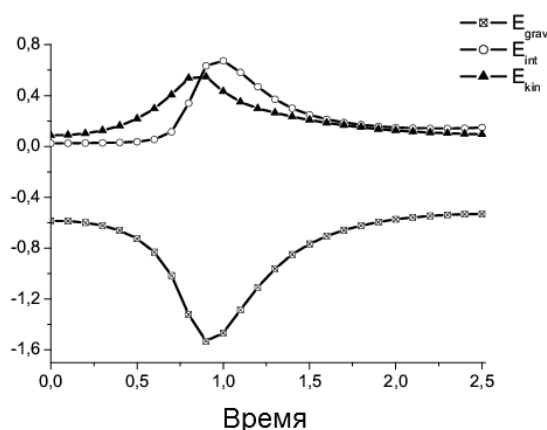


Рис. 11. Поведение различных видов энергии в задаче сжатия молекулярного облака

В рамках данного исследования количественно поведение энергий (см. рис. 11) совпало с результатом других авторов [53].

## Заключение

Статья посвящена новому сверхмасштабируемому программному комплексу AstroPhi для моделирования динамики астрофизических объектов на гибридных суперЭВМ, оснащенных ускорителями Intel Xeon Phi. В статье описан новый численный метод решения газодинамических уравнений, основанный на специально адаптированной для реализации на множестве ускорителей комбинации метода крупных частиц и метода Годунова. Для решения уравнения Пуассона используется быстрое преобразование Фурье. Программная реализация была отдельно протестирована на газодинамических задачах, на задаче решения уравнения Пуассона и на классических задачах гравитационной газовой динамики. Показано ускорение программного комплекса при использовании ускорителей Intel Xeon Phi, уточнено понятие масштабируемости при использовании ускорителей. Представлены результаты моделирования коллапса астрофизических объектов. В текущей реализации AstroPhi реализована только модель гравитационной газовой динамики, в дальнейшем планируется добавить магнитную составляющую и бесстолкновительную компоненту, основанную на первых моментах уравнения Больцмана.

*Исследование выполнено при частичной финансовой поддержке грантов РФФИ № 12-01-31352 мол-а, 13-07-00589-а, 12-07-00065-а, 13-01-00231-а, 11-05-00937; МИП № 39 СО РАН, МИП № 130 СО РАН; гранта Президента РФ МК-4183.2013.9; Программы Президиума РАН Проект № 15.9; гранта Мэрии города Новосибирска.*

## Литература

1. Ardeljan, N.V. An Implicit Lagrangian Code for the Treatment of Nonstationary Problems in Rotating Astrophysical Bodies / N.V. Ardeljan, G.S. Bisnovatyi-Kogan, S.G. Moiseenko // *Astronomy & Astrophysics*. — 1996. — Vol. 115. — P. 573–594.
2. Tutukov, A. Gas Dynamics of a Central Collision of Two Galaxies: Merger, Disruption, Passage, and the Formation of a New Galaxy / A. Tutukov, G. Lazareva, I. Kulikov //

- Astronomy Reports. — 2011. — Vol. 55, I. 9. — P. 770–783.
3. Gingold, R.A. Smoothed Particle Hydrodynamics: Theory and Application to Non-spherical Stars / R.A. Gingold, J.J. Monaghan // Monthly Notices of the Royal Astronomical Society. — 1977. — Vol. 181. — P. 375–389.
  4. Luci, L.B. A Numerical Approach to the Testing of the Fission Hypothesis / L.B. Luci // The Astrophysical Journal. — 1977. — Vol. 82, № 12. — P. 1013–1024.
  5. Collela, P. The Piecewise Parabolic Method (PPM) for Gas-dynamical Simulations / P. Collela, P.R. Woodward // Journal of Computational Physics. — 1984. — Vol. 54. — P. 174–201.
  6. Norman, M. The Impact of AMR in Numerical Astrophysics and Cosmology / M. Norman // Lecture Notes in Computational Science and Engineering. — 2005. — Vol. 41. — P. 413–430.
  7. Hockney, R.W. Computer Simulation Using Particles / R.W. Hockney, J.W. Eastwood — New York: McGraw-Hill, 1981. — 540 p.
  8. Couchman, H. Hydra Code Release / H. Couchman, F. Pearce, P. Thomas. URL: <http://arxiv.org/abs/astro-ph/9603116> (дата обращения: 25.07.2013).
  9. Barnes, J. A Hierarchical  $O(N \log N)$  Force-calculation Algorithm / J. Barnes, P. Hut // Nature. — 1986. — Vol. 324. — P. 446–449.
  10. Dubinski, J. GOTPM: A Parallel Hybrid Particle-Mesh Treecode / J. Dubinski, J. Kim, C. Park, R. Humble // New Astronomy. — 2004. — Vol. 9. — P. 111–126.
  11. Fedorenko, R. A Relaxation Method for Solving Elliptic Difference Equations / R. Fedorenko // U.S.S.R. Computational Mathematics and Mathematical Physics. — 1961. — Vol. 1. — P. 1092–1096.
  12. Godunov, S.K. A Difference Scheme for Numerical Solution of Discontinuous Solution of Hydrodynamic Equations / S.K. Godunov // Matematicheskii Sbornik. — 1959. — Vol. 47. — P. 271–306.
  13. Kulikovskii, A.G. Mathematical Aspects of Numerical Solution of Hyperbolic Systems / A. Kulikovskii, N. Pogorelov, A. Semenov. — Moscow:Fizmatlit, 2001. — 608 p.
  14. Toro, E.F. Riemann Solvers and Numerical Methods for Fluid Dynamics / E. Toro. — Heidelberg:Springer-Verlag, 1999. — 686 p.
  15. Courant, R. On the Solution of Nonlinear Hyperbolic Differential Equations by Finite Difference / R. Courant, E. Isaacson, M. Rees // Communications on Pure and Applied Mathematics. — 1952. — Vol. 5, № 3. — P. 243–255.
  16. Roe, P. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes / P. Roe // Journal of Computational Physics. — 1997. — Vol. 135, I. 2. — P. 250–258.
  17. Engquist, B. One-sided Difference Approximations for Nonlinear Conservation Laws / B. Engquist, S.J. Osher // Mathematics of Computation. — 1981. — Vol. 36, № 154. — P. 321–351.
  18. Harten, A. On Upstream Differencing and Godunov-type Schemes for Hyperbolic Conservation Laws / A. Harten, P.D. Lax, B. Van Leer // Society for Industrial and Applied Mathematics Review. — 1983. — Vol. 25, № 1. — P. 35–61.



19. Einfeld, B. On Godunov-type Methods for Gas Dynamics / B. Einfeld // Society for Industrial and Applied Mathematics Journal on Numerical Analysis. — 1988. — Vol. 25, № 2. — P. 294–318.
20. Batten, P. On the Choice of Savespeeds for the HLLC Riemann Solver / P. Batten, N. Clarke, C. Lambert, D.M. Causon // Society for Industrial and Applied Mathematics Journal on Computing. — 1997. — Vol. 18, № 6. — P. 1553–1570.
21. Van Leer, B. Towards the Ultimate Conservative Difference Scheme. V - A Second-order Sequel to Godunov's Method / B. Van Leer // Journal of Computational Physics. — 1979. — Vol. 32. — P. 101–136.
22. Wadsley, J.W. Gasoline: a Flexible, Parallel Implementation of TreeSPH / J.W. Wadsley, J. Stadel, T. Quinn // New Astronomy. — 2004. — Vol. 9, I. 2. — P. 137–158.
23. Matthias, S. GRAPESPH: Cosmological Smoothed Particle Hydrodynamics Simulations with the Special-purpose Hardware GRAPE / S. Matthias // Monthly Notices of the Royal Astronomical Society. — 1996. — Vol. 278, I. 4. — P. 1005–1017.
24. Springel, V. The Cosmological Simulation Code GADGET-2 / V. Springel // Monthly Notices of the Royal Astronomical Society. — 2005. — Vol. 364, I. 4. — P. 1105–1134.
25. Jin, S. The Relaxation Schemes for Systems of Conservation Laws in Arbitrary Space Dimensions / S. Jin, Z. Xin // Communications on Pure and Applied Mathematics. — 1995. — Vol. 48. — P. 235–276.
26. Godunov, S.K. Experimental Analysis of Convergence of the Numerical Solution to a Generalized Solution in Fluid Dynamics / S.K. Godunov, Yu.D. Manuzina, M.A. Nazareva // Computational Mathematics and Mathematical Physics. — 2011. — Vol. 51. — P. 88–95.
27. Ziegler, U. Self-gravitational Adaptive Mesh Magnetohydrodynamics with the NIRVANA Code / U. Ziegler // Astronomy & Astrophysics. — 2005. — Vol. 435. — P. 385–395.
28. Mignone, A. The Piecewise Parabolic Method for Multidimensional Relativistic Fluid Dynamics / A. Mignone, T. Plewa, G. Bodo // The Astrophysical Journal. — 2005. — Vol. 160. — P. 199–219.
29. Hayes, J. Simulating Radiating and Magnetized Flows in Multiple Dimensions with ZEUS-MP / J. Hayes, et al. // The Astrophysical Journal Supplement Series. — 2006. — Vol. 165. — P. 188–228.
30. Teyssier, R. Cosmological Hydrodynamics with Adaptive Mesh Refinement. A New High Resolution Code Called RAMSES / R. Teyssier // Astronomy & Astrophysics. — 2002. — Vol. 385. — P. 337–364.
31. Kravtsov, A. Constrained Simulations of the Real Universe. II. Observational Signatures of Intergalactic Gas in the Local Supercluster Region / A. Kravtsov, A. Klypin, Y. Hoffman // The Astrophysical Journal. — 2002. — Vol. 571. — P. 563–575.
32. Stone, J. Athena: A New Code for Astrophysical MHD / J. Stone, et al. // The Astrophysical Journal Supplement Series. — 2008. — Vol. 178. — P. 137–177.
33. Brandenburg, A. Hydromagnetic Turbulence in Computer Simulations / A. Brandenburg, W. Dobler // Computer Physics Communications. — 2002. — Vol. 147. — P. 471–475.

34. Schive, H. GAMER: a GPU-accelerated Adaptive-Mesh-Refinement Code for Astrophysics / H. Schive, Y. Tsai, T. Chiueh // *The Astrophysical Journal*. — 2010. — Vol. 186. — P. 457–484.
35. Murphy, J. BETHE-Hydro: An Arbitrary Lagrangian-Eulerian Multidimensional Hydrodynamics Code for Astrophysical Simulations / J. Murphy, A. Burrows // *The Astrophysical Journal Supplement Series*. — 2008. — Vol. 179. — P. 209–241.
36. Springel, V. E Pur Si Muove: Galilean-invariant Cosmological Hydrodynamical Simulations on a Moving Mesh / V. Springel // *Monthly Notices of the Royal Astronomical Society*. — 2010. — Vol. 401. — P. 791–851.
37. Bruenn, S. 2D and 3D Core-collapse Supernovae Simulation Results Obtained with the CHIMERA Code / S. Bruenn, et al. // *Journal of Physics*. — 2009. — Vol. 180. — P. 1–5.
38. Vshivkov, V. Hydrodynamical Code for Numerical Simulation of the Gas Components of Colliding Galaxies / V. Vshivkov, G. Lazareva, A. Snytnikov, I. Kulikov, A. Tutukov // *The Astrophysical Journal Supplement Series*. — 2011. — Vol. 194, I. 47. — P. 1–12.
39. Gonzalez, M. HERACLES: a Three-dimensional Radiation Hydrodynamics Code / M. Gonzalez, E. Audit, P. Huynh // *Astronomy & Astrophysics*. — 2007. — Vol. 464. — P. 429–435.
40. Krumholz, M.R. Radiation-hydrodynamic Simulations of the Formation of Orion-like Star Clusters. I. Implications for the Origin of the Initial Mass Function / M.R. Krumholz, R.I. Klein, C.F. McKee, J. Bolstad // *The Astrophysical Journal*. — 2007. — Vol. 667, I. 74. — P. 1–16.
41. Mignone A. PLUTO: A Numerical Code for Computational Astrophysics / A. Mignone, et al. // *The Astrophysical Journal Supplement Series*. — 2007. — Vol. 170. — P. 228–242.
42. Almgren A. CASTRO: A New Compressible Astrophysical Solver. I. Hydrodynamics and Self-gravity / A. Almgren, et al. // *The Astrophysical Journal*. — 2010. — Vol. 715. — P. 1221–1238.
43. Feng Y. Terapixel Imaging of Cosmological Simulations / Y. Feng, et al. // *The Astrophysical Journal Supplement Series*. — 2011. — Vol. 197, I. 18. — P. 1–8.
44. Enzo-P: Petascale Enzo and the Cello Project / URL: <http://mnggrid.ucsd.edu/projects/cello/> (дата обращения: 25.07.2013).
45. PetaART: Toward Petascale Cosmological Simulations Using the Adaptive Refinement Tree (ART) Code / URL: <http://www.cs.iit.edu/~zlan/petaart.html> (дата обращения: 25.07.2013).
46. Ferrari, A. A New Parallel SPH Method for 3D Free Surface Flows / A. Ferrari, et al. // *High performance computing on vector systems 2009*. — 2010. — Part 4. — P. 179–188.
47. Van Straalen, B. Scalability Challenges for Massively Parallel AMR Applications / B. Van Straalen, J. Shalf, T. Ligocki, N. Keen, W. Yang // In *IPDPS '09: Proceedings of the 2009 IEEE International Symposium on Parallel & Distributed Processing*, Washington, DC, USA. — P. 1–12.
48. Vshivkov, V. A Operator Approach for Numerical Simulation of Self-gravitation Gasdynamic Problem / V. Vshivkov, G. Lazareva, I. Kulikov // *Computational Technologies*. — 2006. — Vol. 11, № 3. — P. 27–35.

49. Vshivkov, V. A Modified Fluids-in-cell Method for Problems of Gravitational Gas Dynamics / V. Vshivkov, G. Lazareva, I. Kulikov // Optoelectronics, Instrumentation and Data Processing. — 2007. — Vol. 43, I. 6. — P. 530–537.
50. Vshivkov, V. Computational Methods for Ill-posed Problems of Gravitational Gasodynamics / V. Vshivkov, G. Lazareva, A. Snytnikov, I. Kulikov, A. Tutukov // Journal of Inverse and Ill-posed Problems. — 2011. — Vol. 19, I. 1. — P. 151–166.
51. Aksenov, A.V. Symmetries and Relations Between Solutions of a Class of Euler-Poisson-Darboux Equations / A.V. Aksenov // Reports of RAS. — 2001. — Vol. 381, I. 2. — P. 176–179.
52. Vshivkov, V. Supercomputer Simulation of an Astrophysical Object Collapse by the Fluids-in-Cell Method / V. Vshivkov, G. Lazareva, A. Snytnikov, I. Kulikov // Lecture Notes in Computational Science. — 2009. — Vol. 5698. — P.414–422.
53. Petrov, M.I. Simulation of the Gravitational Collapse and Fragmentation of Rotating Molecular Clouds / M.I. Petrov, P.P. Berczik // Astronomische Nachrichten. — 2005. — Vol. 326. — P. 505–513.

Куликов Игорь Михайлович, к.ф.-м.н., младший научный сотрудник, Институт вычислительной математики и математической геофизики Сибирского отделения Российской академии наук (Новосибирск, Российская Федерация), kulikov@ssd.sccc.ru.

Черных Игорь Геннадьевич, к.ф.-м.н., научный сотрудник, Институт вычислительной математики и математической геофизики Сибирского отделения Российской академии наук (Новосибирск, Российская Федерация), chernykh@parbz.sccc.ru.

Глинский Борис Михайлович, д.т.н., заведующий лабораторией, Институт вычислительной математики и математической геофизики Сибирского отделения Российской академии наук (Новосибирск, Российская Федерация), gbm@org.sccc.ru.

---

## **AstroPhi: A HYDRODYNAMICAL CODE FOR COMPLEX MODELLING OF ASTROPHYSICAL OBJECTS DYNAMICS BY MEANS OF HYBRID ARCHITECTURE SUPERCOMPUTERS ON Intel Xenon Phi BASE**

*I.M. Kulikov*, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russian Federation),

*I.G. Chernykh*, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russian Federation),

*B.M. Glinsky*, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russian Federation)

There is a new hydrodynamical code AstroPhi for modeling of astrophysical objects dynamics on hybrid supercomputer proposed. This software package is optimized for using with Intel Xeon Phi calculations accelerators. AstroPhi code is based on combination of Godunov and author's FLIC numerical methods for solving of gas dynamics equations. Fast Fourier Transform was used for Poisson equation solution. AstroPhi was tested on gas dynamics problems, Poisson equation solution and classical gravitational gas dynamics problems. The results of this tests and results

of gravitational collapse of astrophysical objects modeling proposed. The results of AstroPhi scalability based on Intel Xeon Phi runs are shown.

*Keywords: numerical simulation, parallel computing, Intel Xeon Phi accelerated, computational astrophysics.*

## References

1. Ardeljan N.V., Bisnovatyi-Kogan G.S., Moiseenko S.G. An Implicit Lagrangian Code for the Treatment of Nonstationary Problems in Rotating Astrophysical Bodies // *Astronomy & Astrophysics*. 1996. Vol. 115. P. 573–594.
2. Tutukov A., Lazareva G., Kulikov I. Gas Dynamics of a Central Collision of Two Galaxies: Merger, Disruption, Passage, and the Formation of a New Galaxy // *Astronomy Reports*. 2011. Vol. 55, I. 9. P. 770–783.
3. Gingold R.A., Monaghan J.J., Smoothed Particle Hydrodynamics: Theory and Application to Non-spherical Stars // *Monthly Notices of the Royal Astronomical Society*. 1977. Vol. 181. P. 375–389.
4. Luci L.B. A Numerical Approach to the Testing of the Fission Hypothesis // *The Astrophysical Journal*. 1977. Vol. 82, № 12. P. 1013–1024.
5. Collela P., Woodward P.R. The Piecewise Parabolic Method (PPM) for Gas-dynamical Simulations // *Journal of Computational Physics*. 1984. Vol. 54. P. 174–201.
6. Norman M. The Impact of AMR in Numerical Astrophysics and Cosmology // *Lecture Notes in Computational Science and Engineering*. 2005. Vol. 41. P. 413–430.
7. Hockney R.W., Eastwood J.W. *Computer Simulation Using Particles* New York: McGraw-Hill, 1981. 540 p.
8. Couchman H., Pearce F., Thomas P. Hydra Code Release / H. Couchman, F. Pearce, P. Thomas URL: <http://arxiv.org/abs/astro-ph/9603116> (accessed: 25.07.2013).
9. Barnes J., Hut P. A Hierarchical O(Nlog N) Force-calculation Algorithm // *Nature*. 1986. Vol. 324. P. 446–449.
10. Dubinski J., Kim J., Park C., Humble R. GOTPM: A Parallel Hybrid Particle-Mesh Treecode // *New Astronomy*. 2004. Vol. 9. P. 111–126.
11. Fedorenko R. A Relaxation Method for Solving Elliptic Difference Equations // *U.S.S.R. Computational Mathematics and Mathematical Physics*. 1961. Vol. 1. P. 1092–1096.
12. Godunov S.K. A Difference Scheme for Numerical Solution of Discontinuous Solution of Hydrodynamic Equations // *Matematicheskii Sbornik*. 1959. Vol. 47. P. 271–306.
13. Kulikovskii A.G., Pogorelov N.V., Semenov A.Yu. *Mathematical Aspects of Numerical Solution of Hyperbolic Systems* Moscow:Fizmatlit, 2001. 608 p.
14. Toro E.F. *Riemann Solvers and Numerical Methods for Fluid Dynamics* Heidelberg:Springer-Verlag, 1999. 686 p.
15. Courant R., Isaacson E., Rees M. On the Solution of Nonlinear Hyperbolic Differential Equations by Finite Difference // *Communications on Pure and Applied Mathematics*. 1952. Vol. 5, № 3. P. 243–255.
16. Roe P. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes // *Journal of Computational Physics*. 1997. Vol. 135, I. 2. P. 250–258.

17. Engquist B., Osher S.J. One-sided Difference Approximations for Nonlinear Conservation Laws // *Mathematics of Computation*. 1981. Vol. 36, № 154. P. 321–351.
18. Harten A., Lax P.D., Van Leer B. On Upstream Differencing and Godunov-type Schemes for Hyperbolic Conservation Laws // *Society for Industrial and Applied Mathematics Review*. 1983. Vol. 25, № 1. P. 35–61.
19. Einfeld B. On Godunov-type Methods for Gas Dynamics // *Society for Industrial and Applied Mathematics Journal on Numerical Analysis*. 1988. Vol. 25, № 2. P. 294–318.
20. Batten P., Clarke N., Lambert C., Causon D.M. On the Choice of Savespeeds for the HLLC Riemann Solver // *Society for Industrial and Applied Mathematics Journal on Computing*. 1997. Vol. 18, № 6. P. 1553–1570.
21. Van Leer B. Towards the Ultimate Conservative Difference Scheme. V - A Second-order Sequel to Godunov's Method // *Journal of Computational Physics*. 1979. Vol. 32. P. 101–136.
22. Wadsley J.W., Stadel J., Quinn T. Gasoline: a Flexible, Parallel Implementation of TreeSPH // *New Astronomy*. 2004. Vol. 9, I. 2. P. 137–158.
23. Matthias S. GRAPESPH: Cosmological Smoothed Particle Hydrodynamics Simulations with the Special-purpose Hardware GRAPE // *Monthly Notices of the Royal Astronomical Society*. 1996. Vol. 278, I. 4. P. 1005–1017.
24. Springel V. The Cosmological Simulation Code GADGET-2 // *Monthly Notices of the Royal Astronomical Society*. 2005. Vol. 364, I. 4. P. 1105–1134.
25. Jin S., Xin Z. The Relaxation Schemes for Systems of Conservation Laws in Arbitrary Space Dimensions // *Communications on Pure and Applied Mathematics*. 1995. Vol. 48. P. 235–276.
26. Godunov S.K., Manuzina Yu.D., Nazareva M.A. Experimental Analysis of Convergence of the Numerical Solution to a Generalized Solution in Fluid Dynamics // *Computational Mathematics and Mathematical Physics*. 2011. Vol. 51. P. 88–95.
27. Ziegler U. Self-gravitational Adaptive Mesh Magnetohydrodynamics with the NIRVANA Code // *Astronomy & Astrophysics*. 2005. Vol. 435. P. 385–395.
28. Mignone A., Plewa T., Bodo G. The Piecewise Parabolic Method for Multidimensional Relativistic Fluid Dynamics // *The Astrophysical Journal*. 2005. Vol. 160. P. 199–219.
29. Hayes J. et al. Simulating Radiating and Magnetized Flows in Multiple Dimensions with ZEUS-MP // *The Astrophysical Journal Supplement Series*. 2006. Vol. 165. P. 188–228.
30. Teyssier R. Cosmological Hydrodynamics with Adaptive Mesh Refinement. A New High Resolution Code Called RAMSES // *Astronomy & Astrophysics*. 2002. Vol. 385. P. 337–364.
31. Kravtsov A., Klypin A., Hoffman Y. Constrained Simulations of the Real Universe. II. Observational Signatures of Intergalactic Gas in the Local Supercluster Region // *The Astrophysical Journal*. 2002. Vol. 571. P. 563–575.
32. Stone J. et al. Athena: A New Code for Astrophysical MHD // *The Astrophysical Journal Supplement Series*. 2008. Vol. 178. P. 137–177.
33. Brandenburg A., Dobler W. Hydromagnetic Turbulence in Computer Simulations // *Computer Physics Communications*. 2002. Vol. 147. P. 471–475.

34. Schive H., Tsai Y., Chiueh T. GAMER: a GPU-accelerated Adaptive-Mesh-Refinement Code for Astrophysics // *The Astrophysical Journal*. 2010. Vol. 186. P. 457–484.
35. Murphy J., Burrows A. BETHE-Hydro: An Arbitrary Lagrangian-Eulerian Multidimensional Hydrodynamics Code for Astrophysical Simulations // *The Astrophysical Journal Supplement Series*. 2008. Vol. 179. P. 209–241.
36. Springel V. E Pur Si Muove: Galilean-invariant Cosmological Hydrodynamical Simulations on a Moving Mesh // *Monthly Notices of the Royal Astronomical Society*. 2010. Vol. 401. P. 791–851.
37. Bruenn S. et al. 2D and 3D Core-collapse Supernovae Simulation Results Obtained with the CHIMERA Code // *Journal of Physics*. 2009. Vol. 180. P. 1–5.
38. Vshivkov V., Lazareva G., Snytnikov A., Kulikov I., Tutukov A. Hydrodynamical Code for Numerical Simulation of the Gas Components of Colliding Galaxies // *The Astrophysical Journal Supplement Series*. 2011. Vol. 194, I. 47. P. 1–12.
39. Gonzalez M., Audit E., Huynh P. HERACLES: a Three-dimensional Radiation Hydrodynamics Code // *Astronomy & Astrophysics*. 2007. Vol. 464. P. 429–435.
40. Krumholz M.R., Klein R.I., McKee C.F., Bolstad J. Radiation-hydrodynamic Simulations of the Formation of Orion-like Star Clusters. I. Implications for the Origin of the Initial Mass Function // *The Astrophysical Journal*. 2007. Vol. 667, I. 74. P. 1–16.
41. Mignone A. et al. PLUTO: A Numerical Code for Computational Astrophysics // *The Astrophysical Journal Supplement Series*. 2007. Vol. 170. P. 228–242.
42. Almgren A. et al. CASTRO: A New Compressible Astrophysical Solver. I. Hydrodynamics and Self-gravity // *The Astrophysical Journal*. 2010. Vol. 715. P. 1221–1238.
43. Feng Y. et al. Terapixel Imaging of Cosmological Simulations // *The Astrophysical Journal Supplement Series*. 2011. Vol. 197, I. 18. P. 1–8.
44. Enzo-P: Petascale Enzo and the Cello Project / URL: <http://mnggrid.ucsd.edu/projects/cello/> (accessed: 25.07.2013).
45. PetaART: Toward Petascale Cosmological Simulations Using the Adaptive Refinement Tree (ART) Code / URL: <http://www.cs.iit.edu/~zlan/petaart.html> (accessed: 25.07.2013).
46. Ferrari A., et al. A New Parallel SPH Method for 3D Free Surface Flows // *High performance computing on vector systems* 2009. 2010. Part 4. P. 179–188.
47. Van Straalen B., Shalf J., Ligocki T., Keen N., Yang W. Scalability Challenges for Massively Parallel AMR Applications // In *IPDPS '09: Proceedings of the 2009 IEEE International Symposium on Parallel & Distributed Processing*, Washington, DC, USA. P. 1–12.
48. Vshivkov V., Lazareva G., Kulikov I. A Operator Approach for Numerical Simulation of Self-gravitation Gasdynamic Problem // *Computational Technologies*. 2006. Vol. 11, № 3. P. 27–35.
49. Vshivkov V., Lazareva G., Kulikov I. A Modified Fluids-in-cell Method for Problems of Gravitational Gas Dynamics // *Optoelectronics, Instrumentation and Data Processing*. 2007. Vol. 43, I. 6. P. 530–537.
50. Vshivkov V., Lazareva G., Snytnikov A., Kulikov I., Tutukov A. Computational Methods for Ill-posed Problems of Gravitational Gasdynamics // *Journal of Inverse and Ill-posed Problems*. 2011. Vol. 19, I. 1. P. 151–166.

51. Aksenov A.V. Symmetries and Relations Between Solutions of a Class of Euler-Poisson-Darboux Equations // Reports of RAS. 2001. Vol. 381, I. 2. P. 176–179.
52. Vshivkov V., Lazareva G., Snytnikov A., Kulikov I. Supercomputer Simulation of an Astrophysical Object Collapse by the Fluids-in-Cell Method // Lecture Notes in Computational Science. 2009. Vol. 5698. P.414–422.
53. Petrov M.I., Berczik P.P. Simulation of the Gravitational Collapse and Fragmentation of Rotating Molecular Clouds // Astronomische Nachrichten. 2005. Vol. 326. P. 505–513.

*Поступила в редакцию 18 октября 2013 г.*

# ЭФФЕКТИВНОЕ ИСПОЛЬЗОВАНИЕ МНОГОЯДЕРНЫХ СОПРОЦЕССОРОВ ПРИ СУПЕРКОМПЬЮТЕРНОМ СТАТИСТИЧЕСКОМ МОДЕЛИРОВАНИИ ЭЛЕКТРОННЫХ ЛАВИН<sup>1</sup>

*М.А. Марченко*

Для моделирования развития электронных лавин в газе разработаны трехмерный параллельный алгоритм метода Монте-Карло и программа ELSHOW, реализованная с использованием комбинирования принципов крупно- и мелкозернистого параллелизма. Для реализации параллельных вычислений на высокопроизводительных гибридных вычислительных системах с сопроцессорами Intel Xeon Phi используется хорошо зарекомендовавшая себя библиотека PARMONC. Применение разработанной технологии распараллеливания существенно уменьшает вычислительную трудоемкость оценки таких интегральных характеристик, как число частиц в лавине, коэффициент ударной ионизации, скорость дрейфа и др.

*Ключевые слова:* электронная лавина, метод Монте-Карло, распараллеливание, суперкомпьютер.

## Введение

Для моделирования электронной лавины в газе необходимо решать уравнение Больцмана, с этой целью нами используется метод Монте-Карло [1, 2]. Он позволяет учесть влияние «маловероятных процессов», что практически невозможно для других моделей (например, BOLSIG+ [3]). При всех достоинствах данного метода нужно обратить особое внимание на то, что приходится держать в памяти ЭВМ координаты в шестимерном фазовом пространстве ( $x, y, z, V_x, V_y, V_z$ ) всех электронов лавины, количество которых растет экспоненциально со временем [4]. Частично эту проблему решает известная лексикографическая схема «ветвления» траекторий. Практически достаточный выигрыш во времени расчетов позволяет здесь получить использование технологии распараллеливания, что и было реализовано в представляемой программе ELSHOW (ELectron SHOWer). При моделировании влияние собственного электрического поля электронов и ионов лавины на внешнее электрическое поле не учитывалось, т.е. исследовалась только начальная стадия развития лавины (до формирования критической лавины, когда собственное поле лавины уравнивается с внешним электрическим полем).

Основной проблемой использования метода Монте-Карло для моделирования ионизационного размножения электронов является лавинообразное нарастание количества электронов и ионов (моделируемых частиц, для каждой из которых необходимо решать уравнения движения). Для решения этой проблемы можно использовать различные методы укрупнения (например, когда моделируемая частица представляется в виде облака, содержащего в себе несколько элементарных частиц), что приводит к ухудшению точности моделирования. Целесообразно, однако, учитывать траектории каждого отдельно взятого

---

<sup>1</sup> Статья рекомендована к публикации программным комитетом Международной суперкомпьютерной конференции «Научный сервис в сети Интернет: все грани параллелизма – 2013».



электрона и использовать для моделирования эффективные технологии распараллеливания на многопроцессорных высокопроизводительных вычислительных системах [2, 4].

Алгоритм метода Монте-Карло для моделирования развития электронных лавин подробно описан в работе [5]. В настоящей работе делается акцент на технологии распараллеливания статистического моделирования на высокопроизводительных вычислительных системах с разными архитектурами.

Статья организована следующим образом. В разделе 1 описан алгоритм статистического моделирования развития электронных лавин в газе. В разделе 2 для разработанного алгоритма статистического моделирования представлены технологии распараллеливания для разных видов многопроцессорных вычислительных систем. В разделе 3 представлены результаты статистического моделирования электронных лавин и показан эффект от распараллеливания. В заключении сформулированы направления дальнейших исследований по численному статистическому моделированию электронных лавин.

## 1. Алгоритм статистического моделирования развития электронных лавин

### 1.1. Общее описание алгоритма

Параллельный трехмерный алгоритм статистического моделирования, реализованный в программе ELSHOW, учитывает ускорение электронов в электрическом поле, процессы упругого рассеяния электронов на молекулах газа и ионизации, а также возбуждения. Для этого используются сечения 24-х типов взаимодействий для азота, приведенные в [3, 6].

Рассматривается открытая система с внешним электрическим полем, напряженность которого  $\mathbf{E} = (0, 0, -E_z)$  считается постоянной. В ходе моделирования с катода из точки  $\mathbf{r} = (x, y, z) = (0, 0, 0)$  в момент времени  $t = 0$  происходит выброс  $n_0$  электронов с нулевыми энергиями. Затем прослеживаются траектории движения каждого из электронов, а также всех вторичных электронов, образовавшихся в результате ионизации, до достижения времени  $t_{\max}$ . С этой целью делаются одинаковые шаги  $\Delta t$  по времени такие, чтобы при этом «прямолинейный» пробег  $\Delta l$  был меньше 0,4 % длины свободного пробега для любых энергий (см. далее). За время  $\Delta t$  электрон с энергией  $T_{i-1}$  движется из точки  $\mathbf{r}_{i-1}$  до точки  $\mathbf{r}_i$ , где  $i$  – номер шага. При этом координаты и скорости частицы с зарядом  $e$  и массой  $m_0$  изменяются следующим образом [2]:

$$\mathbf{r}_i = \mathbf{r}_{i-1} + \mathbf{V}_{i-1} \Delta t - e \mathbf{E} \Delta t^2 / (2 m_0), \quad \mathbf{V}_i = \mathbf{V}_{i-1} - e \mathbf{E} \Delta t / m_0.$$

В конце каждого временного шага разыгрывается столкновение с вероятностью

$$P = 1 - \exp(-\mathbf{b}_t(T_i) N \Delta l),$$

где  $\mathbf{b}_t$  – полное микроскопическое сечение взаимодействий,  $N$  – концентрация частиц газа. Затем разыгрывается тип взаимодействия в соответствии с сечениями упругого рассеяния, возбуждения и ионизации.

Упругое рассеяние не меняет энергию частицы, а направление движения определяется согласно дифференциальным сечениям [7, 8]. При этом азимутальный угол выбирается равномерно на отрезке  $[0, 2\pi]$ .

Для всех видов возбуждения энергия уменьшается на величину потенциала возбуждения, а направление движения разыгрывается также, как и для упругого рассеяния.

При ионизации энергия налетающего электрона  $T$  сначала уменьшается на величину потенциала ионизации  $T_{bd}$ , а затем остаток делится [2] между двумя вылетающими электронами:  $T^{(1)} = T_1 + T_2$ ,  $T_1 = T - \omega T_{bd}$ ,  $T_2 = (\omega - 1) T_{bd}$ , где  $\omega$  – доля переданной энергии. Из простых кинематических уравнений (законов сохранения импульса и энергии) можно получить направления движения образующихся в результате ионизации электронов. Они задаются углами вылета  $\theta_1$  и  $\theta_2$ :

$$\sin^2 \theta_1 = T_2 / T^{(1)}, \sin^2 \theta_2 = T_1 / T^{(1)}.$$

Азимутальные углы связаны между собой соотношением  $\varphi_1 = \varphi_2 + \pi$ , причем один из них выбирается равновероятно на отрезке  $[0, 2\pi]$ .

## 1.2. Вычисление сечений

Для ускорения расчетов имеющаяся эмпирическая таблица сечений была дополнена значениями с помощью линейной интерполяции. Область энергий была поделена на несколько частей, в каждой из которых выбран свой достаточно маленький шаг, общий для всех типов взаимодействия. Значения сечений с таким шагом по энергии занесены в новую таблицу. Этот подход позволяет сделать поиск необходимого интервала в массиве сечений с помощью операции взятия целой части без длительного перебора. Окончательно требуемые значения получаются линейной интерполяцией в этом интервале.

## 1.3. Моделирование ионизации

Рассмотрим подробнее процедуру моделирования  $\omega$  согласно плотности [2]:

$$\psi_\omega(u, T) = A [(3u + 4) \arctg b + b / (1 + b^2) + 2b(u - 4) / (1 + b^2)^2] / (\zeta u^3),$$

$$u \in [1, \zeta].$$

Здесь  $A$  – константа,  $b^2 = \zeta - u$ ,  $\zeta = T / T_{bd}$  – энергия налетающего электрона в единицах  $T_{bd}$ . Поскольку интервал, на котором функция  $\psi_\omega(u)$  не равна нулю, зависит от энергии, перейдем к моделированию другой случайной величины

$$\gamma = (\omega - 1) / (\zeta - 1)$$

с плотностью распределения

$$\psi_\gamma(w, T) = A (\zeta - 1) [(3u + 4) \arctg b + b / (1 + b^2) + 2b(u - 4) / (1 + b^2)^2] / (\zeta u^3),$$

$$w \in [0, 1],$$

где  $u = w(\zeta - 1) + 1$ ,  $b^2 = (\zeta - 1)(1 - u)$ . Эта функция была затабулирована для набора энергий  $\{T_j\}$ .

Для промежуточных значений  $T \in (T_{j-1}, T_j)$  плотность можно определить с помощью линейной интерполяции по параметру

$$\psi_\gamma(w, T) = \psi_\gamma(w, T_{j-1}) (T_j - T) / (T_j - T_{j-1}) + \psi_\gamma(w, T_j) (T - T_{j-1}) / (T_j - T_{j-1}).$$

При этом моделирование величины  $\gamma$  осуществляется методом суперпозиции [1] с вероятностью  $P_1 = (T_j - T) / (T_j - T_{j-1})$  согласно плотности  $\psi_\gamma(w, T_{j-1})$ , а с вероятностью  $P_2 = (T - T_{j-1}) / (T_j - T_{j-1})$  – соответственно  $\psi_\gamma(w, T_j)$ . Затем обратным преобразованием получаем  $\omega = \gamma(\zeta - 1) + 1$ .

Аналогичная процедура использовалась при моделировании углов рассеяния по таблицам дифференциальных сечений для упругого и неупругого рассеяния.

## 1.4. Лексикографическая схема

В программе ELSHOW моделирование индивидуальных соударений построено таким образом, что прослеживается путь частицы от момента вылета с катода до времени  $t_{\max}$ .

При этом в массив фактически записываются лишь вторичные частицы, которые образовались в результате ионизации. Затем моделируется путь одной из вторичных частиц до достижения времени  $t_{\max}$ , и так далее до тех пор, пока все вторичные частицы не будут рассмотрены. Такая схема расчета называется лексикографической. Она не требует больших объемов памяти для хранения всех частиц. Кроме того, предусмотрена возможность использования метода укрупненных соударений, в котором на определенном шаге по времени производится процедура удаления электронов с вероятностью  $P_u$  из рассмотрения («русская рулетка»). Для оставшихся частиц статистический «вес» умножается на величину  $1/(1 - P_u)$ . Получаемые при этом весовые оценки являются статистически несмещенными с конечной дисперсией, которая также оценивается.

### 1.5. Функционалы, погрешность, построение гистограммы

Программа ELSHOW в результате своей работы выдает для момента времени  $t_{\max}$  значения числа частиц  $n$ , положения центра масс  $\mathbf{r}_c = (\langle x \rangle, \langle y \rangle, \langle z \rangle)$ , средней скорости  $\langle V_z \rangle$ , средней энергии  $\langle \epsilon \rangle$ , гистограммы плотности частиц, а также их среднеквадратические погрешности. Это позволяет вычислить различные характеристики лавины такие, как скорость центра масс  $V_c = \langle z \rangle / t_{\max}$  и скорость дрейфа  $V_{dr} = \langle V_z \rangle$ . Коэффициенты поперечной  $D_T$  и продольной диффузии  $D_L$  находятся путем определения по гистограмме соответствующих диффузионных радиусов [3]. Коэффициент ударной ионизации  $\alpha$  определялся с помощью формулы [9]:

$$\alpha = (V_c - (V_c^2 - 4 \nu_i)^{1/2}) / (2 D_L),$$

где  $\nu_i = \ln(n/n_0) / t_{\max}$  — частота ионизации. Более того, все числовые характеристики можно получить и для нескольких промежуточных значений времени, что дает возможность исследовать их динамику.

Гистограммой оценивалась плотность частиц, как функция расстояния  $r$  от оси  $OZ$ . При выборе шага гистограммы учитывалось, что в оптимальном варианте детерминированная и статистическая погрешности функциональной оценки должны быть приближенно равными. Для наиболее важной «осевой» ячейки было получено соотношение для шага  $\Delta r = (8 D_T t)^{1/2} (2 n)^{-1/6}$ .

### 1.6. Выбор шага

Шаг  $\Delta t$  был выбран пропорциональным  $0.004 / \max(\sigma_t(T) (2T/m)^{1/2})$ . Это соответствует тому, что  $\Delta l$  меньше случайного пробега с вероятностью  $P_{pr} = 0,996$ . Такое жесткое условие продиктовано тем, что требуется получать результаты с высокой точностью. С помощью метода зависимых траекторий была проведена серия расчетов, в которых использовалось два разных датчика случайных чисел. Один из них применялся только при розыгрыше столкновения, а другой – во всех остальных формулах распределенным способом. Это позволило скоррелировать траектории электронов для различных значений  $\Delta t$ . Наши расчеты показали, что при  $P_{pr} = 0,996$  основные транспортные характеристики практически не отличаются для шагов  $\Delta t$  и  $\Delta t/2$ , т.е. указанный шаг  $\Delta t$  является удовлетворительным.

Указанный выше способ оценки подходящего шага  $\Delta t$  был успешно проверен с помощью тестового альтернативного расчета для  $E = 0$  и среднего ненулевого значения начальной энергии, в котором моделировалась экспоненциально распределенная длина  $\Delta l$  свободного пробега с соответствующим пересчетом времени. Разности искомых величин

для основного («пошагового») и вспомогательного («экспоненциального») вариантов расчета оказались статистически незначимыми. При этом также был реализован распределительный способ использования псевдослучайных чисел.

## 2. Технологии распараллеливания статистического моделирования

Вероятностные модели, описывающие развитие электронных лавин, обладают большой вычислительной сложностью. Тем не менее, такого рода модели могут быть успешно реализованы с использованием распараллеливания на многопроцессорных вычислительных системах, например, с графическими ускорителями [10]. В программе ELSHOW распараллеливание осуществлялось в рамках единого подхода к реализации распределенного статистического моделирования, предложенного в работе [11]. Применяемая методология позволяет добиться высокой эффективности в использовании гибридных вычислительных систем и существенного уменьшения трудоемкости вычислений: в наших расчетах моделирование развития лавин электронов проводилось вплоть до  $10^9$  частиц без ухудшения точности оценки функционалов.

### 2.1. Метод распределенного статистического моделирования

Метод распределенного статистического моделирования состоит в распределении моделирования независимых реализаций по вычислительным ядрам с периодическим осреднением полученных выборочных значений по статистически эффективной формуле

$$\bar{\zeta} = \left[ \sum_{m=1}^M l_m \right]^{-1} \sum_{m=1}^M l_m \bar{\zeta}_m. \quad (1)$$

Здесь  $\bar{\zeta}$  — искомое выборочное среднее значение,  $M$  — общее число ядер,  $l_m$  — объем выборки, полученной на  $m$ -м ядре,  $\bar{\zeta}_m$  — соответствующее  $m$ -му ядру выборочное среднее значение.

Очевидно, что главным критерием осуществимости такой параллельной реализации является возможность «поместить» данные вычислительной программы для моделирования реализаций в оперативную память каждого ядра. Подчеркнем, что в целях распределенного статистического моделирования допустимо использовать вычислительные ядра с разной производительностью. Важно, чтобы пересылка данных на центральное ядро и соответствующий прием данных осуществлялись в асинхронном режиме.

### 2.2. Параллельный генератор псевдослучайных чисел

Как правило, при параллельной реализации необходимый объем выборки базовых случайных чисел очень велик, поэтому целесообразно использовать длиннопериодные псевдослучайные последовательности. А именно, для решения «больших» задач по методу Монте-Карло предлагается использовать генератор следующего вида:

$$u_0 = 1, u_n \equiv u_{n-1} A \pmod{2^{128}}, \alpha_n = u_n 2^{-128}, n = 1, 2, \dots \quad (2)$$

$$A \equiv 5^{100109} \pmod{2^{128}}.$$

Получаемая последовательность псевдослучайных чисел является периодической, длина ее периода равна  $2^{126}$ .

Для реализации статистического моделирования на независимых вычислительных ядрах предлагается следующий *распределительный способ использования базовых псевдослучайных чисел* (термин предложен член-корр. Г.А. Михайловым). Базовая последовательность предварительно разбивается на подпоследовательности, начинающиеся с чисел

$$\{\alpha_{m\mu}\}, m = 0, 1, 2, \dots,$$

где  $\mu$  — длина «прыжка», после чего полученные таким образом подпоследовательности распределяются по разным ядрам. Значение «прыжка» генератора должно выбираться так, чтобы такого количества псевдослучайных чисел хватало для моделирования на каждом ядре. Легко показать, что для метода вычетов начальные значения указанных подпоследовательностей получаются по формуле

$$u_{(m+1)\mu} = u_{m\mu} A_{\mu} \pmod{2^{128}}, \alpha_{m\mu} = u_{m\mu} 2^{-128}, m = 0, 1, \dots \quad (3)$$

$$A_{\mu} \equiv A^{\mu} \pmod{2^{128}}.$$

Отметим, что длина периода генератора позволяет независимым образом распределять псевдослучайные числа по реализациям на практически неограниченное число вычислительных ядер. Параллельный генератор, приведенный выше, успешно используется в ряде институтов СО РАН на протяжении последних десяти лет.

### 2.3. Библиотека PARMONC

С целью унификации использования распределенного статистического моделирования при решении широкого круга задач методом Монте-Карло автором разработана и внедрена программная библиотека **PARMONC** (**PAR**allel **MON**te **C**arlo) [12]. Библиотека PARMONC установлена на кластерах Сибирского суперкомпьютерного центра (ЦКП ССКЦ СО РАН) и может использоваться на вычислительных системах с аналогичной архитектурой. Библиотека предназначена для распараллеливания программ, написанных на языках Fortran или C, причем библиотечные подпрограммы применяются без явного использования команд MPI. Инструкции по использованию библиотеки приведены в [13].

Возможность применения библиотеки PARMONC определяется «естественной» крупноблочной фрагментированностью программ статистического моделирования. Общая структура такого рода программ, в упрощенном виде, представлена на рис. 1 (в нотации языка C++).

```

void main( void ) {
    long int i, L;
    TypeRL RL, SUBT;
    SUBT = 0.0;
    // цикл по реализациям
    for( i = 0; i < L; i++ ) {
        // далее операторы, вычисляющие реализацию RL
        ...
        SUBT= SUBT + RL;
    }
    SUBT = SUBT / L;
}
    
```

Рис. 1. Общая структура программ статистического моделирования

Здесь  $L$  – общее число независимых реализаций случайного объекта, задаваемых композитным типом данных  $TypeRL$  (допускается поэлементное суммирование переменных такого типа); реализации  $RL$  моделируются внутри цикла по переменной  $i$ . Полученные таким образом реализации  $RL$  (статистически независимые в совокупности) добавляются к «счетчику»  $SUBT$  и по выходу из цикла осредняются, что дает статистическую оценку искомого математического ожидания случайного объекта.

При распараллеливании последовательных программ с помощью PARMONC определяется процедура *realization* (моделирующая подпрограмма), возвращающая одну реализацию (возвращение осуществляется через аргумент процедуры). При этом считается, что моделирующая подпрограмма использует потоки псевдослучайных чисел, генерируемых внешней по отношению к ней подпрограммой. Цикл по независимым реализациям и финальное осреднение заменяются вызовом библиотечной процедуры следующего вида:

```
parmoncc( realization, L, SUBT, ... );
```

Здесь имя моделирующей подпрограммы и общее число независимых реализаций передаются в библиотечную процедуру *parmoncc* в качестве входных аргументов; выборочное среднее будет возвращаться в переменную  $SUBT$ ; для простоты остальные аргументы процедуры *parmoncc* опущены и заменены многоточием. Процедура *parmoncc* автоматически распределяет моделирование независимых реализаций по вычислительным ядрам. Все остальные операторы пользовательской программы остаются без изменений. На рис. 2 представлен модифицированный код, пригодный для компиляции и сборки с помощью библиотеки PARMONC.

```
void main( void ) {
    TypeRL SUBT;
    parmoncc (realization, L, SUBT , ... );
}

void realization( TypeRL RL ){
    // далее операторы моделирующей подпрограммы
    ...
    // вычисленная реализация возвращается в переменную RL
}
```

**Рис. 2.** Модифицированный код программы, пригодный для компиляции и сборки с помощью библиотеки PARMONC

В процессе распределенных вычислений на каждом ядре используются потоки независимых псевдослучайных чисел, получаемые в результате работы подпрограммы, реализующей параллельный генератор. В процедуре *realization* библиотечный параллельный генератор вызывается следующим образом:

$$a = rnd128(),$$

Здесь  $a$  — очередное псевдослучайное число, равномерно распределенное в интервале от нуля до единицы. Инициализация параллельного генератора выполняется автоматически при запуске программы, скомпилированной и собранной с помощью библиотеки PARMONC.

Следует упомянуть о практически важной возможности коррелирования результатов различных расчетов для одной и той же задачи, когда варьируется лишь ряд ее параметров. При использовании библиотеки PARMONC это делается на основе

использования одних и тех же псевдослучайных чисел в каждом из расчетов. Предлагаемый выше параллельный генератор позволяет распределять псевдослучайные числа также и по отдельным траекториям в каждом из расчетов. С использованием такого подхода проводились многократные численные расчеты по подбору параметров нашей вероятностной модели, в частности, величины временного шага в модели.

#### **2.4. Комбинирование крупнозернистого и мелкозернистого распараллеливания**

Для рассматриваемой задачи *методика крупнозернистого распараллеливания* заключается в распределении моделирования отдельных реализаций электронной лавины по отдельным ядрам. С помощью библиотеки PARMONS такого рода распределение делается автоматически. С одной стороны, среднее время моделирования отдельной реализации может быть достаточно большим (в некоторых расчетах оно достигало нескольких суток). С другой стороны, при крупнозернистом распараллеливании возникают проблемы, связанные с обработкой быстро растущего числа частиц в лавине.

Расчеты по методике крупнозернистого распараллеливания производились на кластере НКС-30Т в ЦКП ССКЦ СО РАН. Объем оперативной памяти, доступный каждому CPU ядру на узле, составляет от 4 до 8 ГБ. Таких объемов оперативной памяти вполне достаточно для реализации методики распараллеливания. Количество используемых ядер варьировалось в пределах от 128 до 512.

*Методика мелкозернистого распараллеливания* для рассматриваемой задачи заключается в моделировании отдельной реализации электронной лавины на одном многоядерном процессоре, например, на графическом сопроцессоре или акселераторе Intel Xeon Phi. С этой целью при реализации лексикографической схемы моделирование развития условно-независимых «ветвей» дерева (т.е. части лавины частиц) от вторичных частиц, появляющихся при актах ионизации, распределяется по разным ядрам сопроцессора. Естественно, необходимо сбалансированно распределять моделирование «ветвей» по ядрам сопроцессора с целью недопущения простоя ядер, таким образом, осуществляя балансировку вычислительной нагрузки. Такая балансировка (т.е. пересылка массивов вторичных частиц), очевидно, требует затрат машинного времени.

При лавинообразном росте числа частиц в реализации лавины следует также передавать «лишние» частицы в память CPU ядра для временного размещения в «магазине» памяти и при окончании моделирования всех «ветвей» на сопроцессоре, догружать частицы в память сопроцессора из «магазина» для дальнейшей обработки.

Ясно, что при таком распараллеливании необходимо распределять псевдослучайные числа также и по отдельным ядрам сопроцессора. Применение методики мелкозернистого распараллеливания помогает обрабатывать быстро растущее число частиц в лавине.

Поскольку узлы гибридного кластера имеют в своем составе разные вычислители (CPU ядра и сопроцессоры), то целесообразно комбинировать крупно- и мелкозернистое распараллеливание следующим образом. На каждом вычислительном узле часть CPU ядер будет моделировать реализации согласно методике крупнозернистого распараллеливания, в то время как сопроцессоры узла (и «прикрепленные» к ним CPU ядра) будут моделировать реализации лавины по методике мелкозернистого распараллеливания. Таким образом, при использовании такой *методики*

комбинированного распараллеливания вычислительные ресурсы гибридного узла будут задействованы максимально полно.

В следующем разделе приведены результаты численных расчетов с целью сравнения предложенных методик распараллеливания. Полученные результаты позволяют сделать вывод о целесообразности применения комбинированной методики распараллеливания на основе использования сопроцессоров Intel Xeon Phi для моделирования развития электронных лавин.

### 3. Результаты статистического моделирования

#### 3.1. Результаты моделирования электронных лавин

Расчеты показали, что данная вычислительная модель хорошо согласуется с теоретическими и экспериментальными данными при  $E_z/p < 200$  В/(см·Торр). Во всех расчетах относительная погрешность результатов ELSHOW не превосходит 2,6 %.

На рис. 3а представлены графики для скорости дрейфа, за которую принималась среднестатистическая скорость по ансамблю частиц  $V_{dr} = \langle V_z \rangle$ . На рисунке видно, что при  $E_z/p > 200$  В/(см·Торр) наблюдается хорошее совпадение полученных с помощью ELSHOW данных с результатами расчетов с помощью программы BOLSIG+ [3]. Это, по всей видимости, связано с тем, что в ELSHOW используются преимущественно те же сечения взаимодействий, что и в BOLSIG+. Функция  $3,3 \cdot 10^6 (E_z/p)^{1/2}$  (рис. 3а, штриховая линия), построенная по данным экспериментов [4], при больших  $E_z/p$  лежит ниже результатов ELSHOW (разница доходит до 21 %). Значения  $V_{dr}$  (рис. 3а, «точки»), полученные в [15], выше данных ELSHOW на 9–16 %.

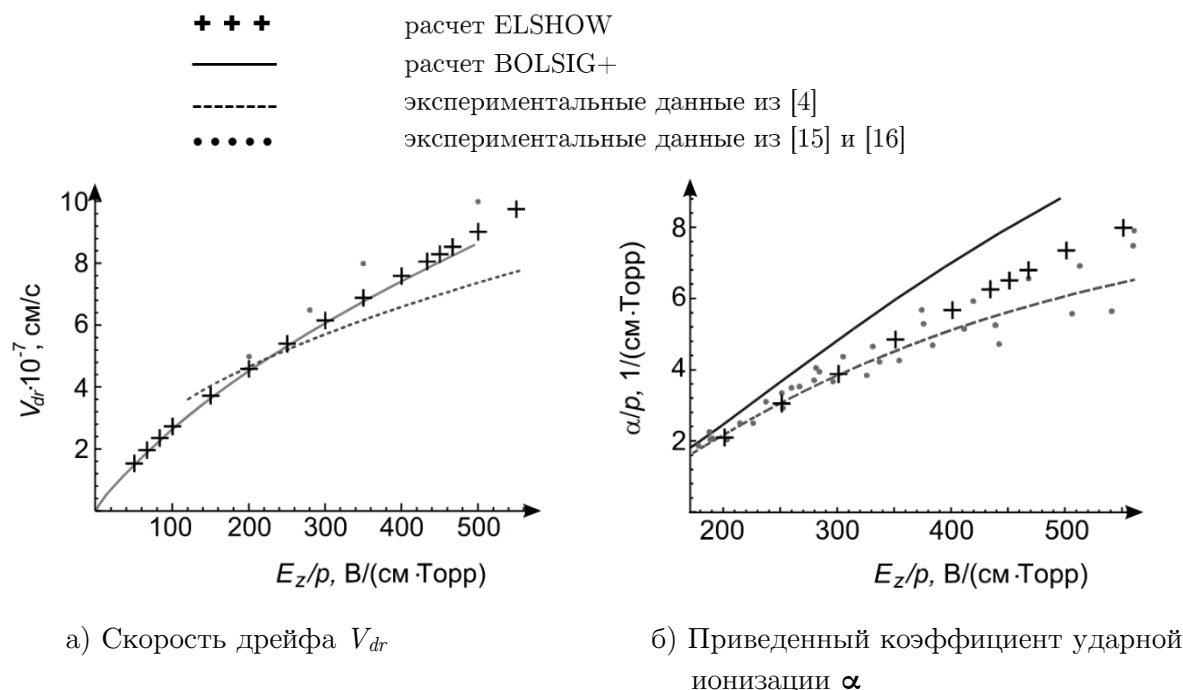


Рис. 3. Сравнение расчетных и экспериментальных данных

На рис. 3б представлены графики для приведенного коэффициента ударной ионизации  $\alpha$ . Из рисунка видно, что при  $E_z/p > 200$  В/(см·Торр) результаты ELSHOW лежат



ближе к экспериментальным данным ([4]—штриховая линия и [16]—«точки»), чем у результатов BOLSIG+. Функция  $12 \cdot \exp(-342 p/E_z)$ , построенная по результатам экспериментов (штриховая линия), лежит ниже значений ELSHOW до 19 %.

Важно отметить, что решение уравнения Больцмана с помощью метода Монте-Карло позволяет учесть маловероятные события при развитии лавины. Так, на рис. 4 приведены «портреты» электронных лавин, полученные в расчетах с помощью ELSHOW, в азоте при  $p = 300$  Торр,  $E_z/p = 50$  В/(см·Торр),  $t_{\max} = 90$  нс (рис. 4а) и  $500$  В/(см·Торр),  $t_{\max} = 0.055$  нс (рис. 4б). Из рисунков видно, что при внешнем поле порядка пробивного ( $1.2 E_{np} = 15$  кВ/см) лавина имеет практически правильную сферическую форму (рис. 4а). При увеличении напряженности поля до значений около  $12 E_{np}$  сильное влияние на форму лавины оказывает высокоэнергетичная часть функции распределения электронов по энергии. Помимо четко выраженной сферической формы имеются «ветвления», которые искажают форму лавины (рис. 4б). Этими соображениями, по-видимому, объясняется расхождение в значениях приведенного коэффициента ударной ионизации между результатами ELSHOW и BOLSIG+ (рис 3б).

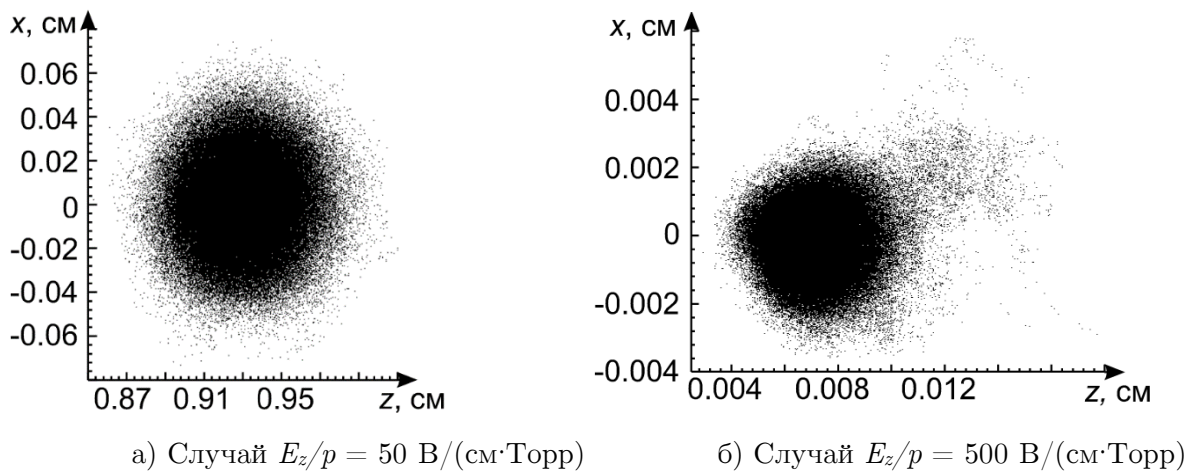


Рис. 4. «Портреты» электронных лавин, полученные в расчетах с помощью ELSHOW

### 3.2. Результаты распараллеливания на многоядерных сопроцессорах

Расчеты по методике комбинированного распараллеливания производились на прототипе суперкомпьютера МВС-10П в МСЦ РАН. На этом кластере на каждом вычислительном узле доступны два 8 ядерных процессора Intel Xeon E5-2690 и два 60 ядерных сопроцессора Intel Xeon Phi SE10X. Объем оперативной памяти, доступный каждому ядру на узле, составляет 4 ГБ. На каждом сопроцессоре для моделирования доступно примерно 8 ГБ оперативной памяти (за вычетом объема оперативной памяти, требуемой операционной системе сопроцессора). Таких объемов оперативной памяти вполне достаточно для реализации описанных выше методик распараллеливания. При программной реализации использовалась технология т.н. offload-режима исполнения задач на сопроцессоре [14].

Для одной из физических постановок задачи сравнивался эффект от применения методики крупнозернистого распараллеливания с эффектом от применения методики комбинированного распараллеливания (на одном вычислительном узле). При вычислениях варьировалась только величина  $t_{\max}$ , что давало в конце моделирования различное число частиц в лавине (большему значению времени соответствовало большее число частиц).

Для каждого сочетания методики распараллеливания и величины  $t_{\max}$  оценивались значения функции  $L = L(t)$ , где  $L(t)$  - среднее количество реализаций лавины частиц, полученных к моменту машинного времени  $t$ .

Как показали расчеты, для  $t_{\max} = 0.01$  нс в лавине образуется в среднем примерно  $10^6$  частиц, а при  $t_{\max} = 1$  нс – в среднем около  $8 \cdot 10^7$  частиц. В программе, реализующей методику комбинированного распараллеливания, при числе частиц в лавине более  $10^6$  намеренно производилось периодическое перераспределение частиц между ядрами сопроцессора и «прикрепленного» CPU ядра.

На рис. 5 представлено сравнение эффективности двух методик распараллеливания. По горизонтальной оси отложено машинное время  $t$  в секундах, по вертикальной оси – соответствующее количество реализаций  $L$ . Линия с ромбами — крупнозернистое распараллеливание,  $t_{\max} = 0,01$  нс; линия с кругами — крупнозернистое распараллеливание,  $t_{\max} = 1$  нс; линия с квадратами — комбинированное распараллеливание,  $t_{\max} = 0,01$  нс; линия с треугольниками — комбинированное распараллеливание,  $t_{\max} = 1$  нс.

Как видно из рис. 5, для одной и той же величины  $t_{\max}$  методика комбинированного распараллеливания позволяет получить значительно большее число реализаций в течение заданного машинного времени. Таким образом, при большой величине  $t_{\max}$  методика комбинированного распараллеливания предпочтительнее методики крупнозернистого распараллеливания (несмотря на затраты машинного времени на балансировку вычислительной нагрузки).

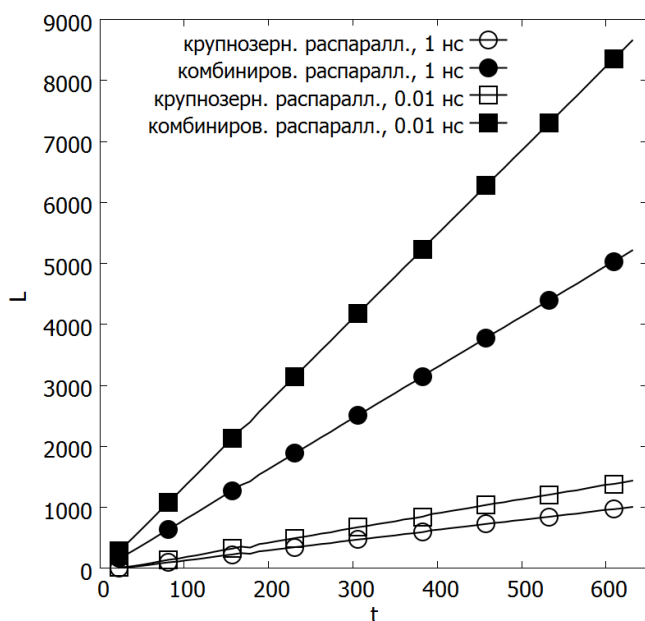


Рис. 5. Сравнение эффективности двух методик распараллеливания

## Заключение

В работе представлены трехмерный параллельный алгоритм метода Монте-Карло и программа ELSHOW, предназначенные для моделирования развития электронных лавин в газе. В численных экспериментах показывается удовлетворительное соответствие результатов расчетов с помощью ELSHOW известным из литературы экспериментальными данными и результатам расчетов с помощью других пакетов программ. Изучается эффективность применения различных методик распараллеливания (крупнозернистого и

мелкозернистого параллелизма и их комбинации), делается вывод о целесообразности применения предложенной комбинированной методики распараллеливания на основе использования сопроцессоров Intel Xeon Phi для моделирования развития электронных лавин.

В дальнейшем предполагается усовершенствовать разработанный параллельный алгоритм с целью моделирования облака электронов начиная с момента, когда собственное электрическое поле электронов и ионов лавины становится сравнимым с внешним электрическим полем [4]. С целью повышения эффективности моделирования следует, возможно, отказаться от лексикографической схемы «ветвления» траекторий и использовать «метод поколений» [1]. Следует также оценивать влияние высокоэнергичных электронов на развитие пробоя в газе [2, 4]. Эта задача связана с оценкой т.н. «редких» событий и требует применения специальных методик статистического моделирования. Для таких постановок статистическое моделирование на гибридных суперкомпьютерах с использованием методики комбинированного распараллеливания представляется весьма перспективным.

*Настоящая работа проводилась при финансовой поддержке грантов РФФИ №№ 13-07-00589, 13-01-00746, 12-01-00034, 12-01-00727; МИП №№ 39, 47, 126, 130 СО РАН.*

*Пользуясь случаем, автор выражает признательность гл. специалисту ЦКП ССКЦ СО РАН Н.В. Кучину за помощь и плодотворные обсуждения.*

## Литература

1. Ермаков, С.М. Курс статистического моделирования / С.М. Ермаков, Г.А. Михайлов — М.: Наука, 1976. — 320 с.
2. Аккерман, А.Ф. Моделирование траекторий заряженных частиц в веществе / А.Ф. Аккерман — М.: Энергоатомиздат, 1991. — 200 с.
3. Hagelaar, G.J.M. Solving the Boltzmann equation to obtain electron transport coefficients and rate coefficients for fluid models / G.J.M. Hagelaar, L.C. Pitchford // Plasma Sources Sci. Technol. — 2005. — Vol. 14. — P. 722–733.
4. Королёв, Ю.Д. Физика импульсного пробоя газов / Ю.Д. Королев, Г.А. Месяц. — М.: Наука. Гл. ред. физ.-мат. лит., 1991. — 224 с.
5. Параллельная реализация метода Монте-Карло для моделирования развития электронных лавин в газе / Г.З. Лотова, М.А. Марченко, Г.А. Михайлов и др. // Известия высших учебных заведений. Физика. — 2013.
6. Itikawa, Y. Cross Sections for Collisions of Electrons and Photons with Nitrogen Molecules. / Y. Itikawa, M. Hayashi, A. Ichimura, K. Onda, K. Sakimoto, K. Takayanagi, M. Nakamura, H. Nishimura, T. Takayanagi // J. Phys. Chem. Ref. Data — 1986. — Vol. 15, No. 3. P. 985–1010.
7. Okhrimovskyy, A. Electron anisotropic scattering in gases: A formula for Monte Carlo simulations / A. Okhrimovskyy, A. Bogaerts, R. Gijbels // Phys. Rev. E. — 2002. — Vol. 65, No. 037402. — P. 1–4.
8. Sun, W. Detailed theoretical and experimental analysis of low-energy electron-N<sub>2</sub> scattering / W. Sun, M.A. Morrison, W.A. Isaacs, W.K. Trail, D.T. Alle, R.J. Gulley, M.J. Brennan, S.J. Buckman // Phys. Rev. A. — 1995. — Vol. 52, No. 2. — P. 1229–1256.

9. Tagashira, H. The development of electron avalanches in argon at high E/N values. II. Boltzmann equation analysis / H. Tagashira, Y. Sakai, S. Sakamoto // J. Phys. D: Appl. Phys. — 1977 — Vol. 10. — P. 1051.
10. Жуковский, М.Е. Математическое моделирование радиационной эмиссии электронов на гибридных суперкомпьютерах / М.Е. Жуковский, Р.В. Усков // Вычислительные методы и программирование. — 2012. — Т. 13, № 1. — С. 189–197.
11. Марченко, М.А. Распределенные вычисления по методу Монте-Карло / М.А. Марченко, Г.А. Михайлов // Автоматика и телемеханика. — 2007. — Вып. 5. — С. 157–170.
12. Марченко, М.А. Библиотека PARMONC для решения «больших задач по методу Монте-Карло» / М.А. Марченко // Вестник ННГУ. — 2012. — № 5. — С. 392–397.
13. Марченко, М.А. Библиотека PARMONC на сайте ЦКП ССКЦ СО РАН / М.А. Марченко. URL: <http://www2.sccc.ru/SORAN-INTEL/paper/2011/parmonc.htm> (дата обращения: 19.08.2013).
14. Jeffers, J. Intel Xeon Phi Coprocessor High - Performance Programming. / J. Jeffers, J. Reinders — Elsevier, 2013. — 432 p.
15. Lisovskiy, V. Electron drift velocity in argon, nitrogen, hydrogen, oxygen and ammonia in strong electric fields determined from rf breakdown curves / V. Lisovskiy, J.P. Booth, K. Landry, D. Douai, V. Cassagne, V. Yegorenko // J. Phys. D: Appl. Phys. — 2006. — Vol. 39. — P. 660–665.
16. Dutton, J. A survey on electron swarm data / J. Dutton // J. Phys. Chem. Ref. Data. — 1975. — Vol. 4, No. 3. — P. 577–851.

Марченко Михаил Александрович, к.ф.-м.н., ученый секретарь, Институт вычислительной математики и математической геофизики СО РАН, доцент кафедры вычислительной математики, механико-математический факультет, Новосибирский государственный университет (Новосибирск, Российская Федерация), [mam@osmf.sccc.ru](mailto:mam@osmf.sccc.ru).

---

## EFFECTIVE USE OF MULTICORE COPROCESSORS IN SUPERCOMPUTER STOCHASTIC SIMULATION OF ELECTRON AVALANCHES

*M.A. Marchenko, Institute of Computational Mathematics and Mathematical Geophysics of SB RAS, Novosibirsk State University (Novosibirsk, Russian Federation)*

Three-dimensional parallel Monte Carlo algorithm for modelling the electron avalanches in gases is developed. Parallel Implementation is made on supercomputers with MPP architecture and on hybrid supercomputers with Intel Xeon Phi coprocessors. The well-working library PARMONC is used to implement parallel computations. The use of the library enables fast calculation of functionals such as the number of particles in avalanche, first Townsend coefficient, drift velocity, etc.

*Keywords: electron avalanche, Monte Carlo method, parallelization, supercomputer.*

## References

1. Ermakov S.M., Mikhailov G.A. Kurs statisticheskogo modelirovaniya [Course of stochastic simulation]. Moscow, Nauka, 1976. 320 p.
2. Akkerman A.F. Modelirovanie traektorij zaryazhennykh chastic v veshestve [Simulation of trajectories of charged particles in medium]. Moscow, Energoatomizdat, 1991. 200 p.
3. Hagelaar G.J.M., Pitchford L.C. Solving the Boltzmann equation to obtain electron transport coefficients and rate coefficients for fluid models. Plasma Sources Sci. Technol. 2005. Vol. 14. P. 722–733.
4. Korolev Ju.D., Mesyats G.A. Physics of impulse breakdown in gases. Moscow, Nauka, 1991. 224 p.
5. G.Z. Lotova, M.A. Marchenko, G.A. Mikhailov, et al. Parallel realization of Monte Carlo method for modelling of electron avalanches in gases // Izvestiya vyshykh uchebnykh zavedeniy. 2013.
6. Itikawa Y., Hayashi M., Ichimura A., et al. Cross Sections for Collisions of Electrons and Photons with Nitrogen Molecules. // J. Phys. Chem. Ref. Data. 1986. Vol. 15, No. 3. P. 985–1010.
7. Okhrimovskyy A., Bogaerts A., Gijbels R. Electron anisotropic scattering in gases: A formula for Monte Carlo simulations // Phys. Rev. E. 2002. Vol. 65, No. 037402. P. 1–4.
8. Sun W., Morrison M.A., Isaacs W.A., et al. Detailed theoretical and experimental analysis of low-energy electron-N<sub>2</sub> scattering // Phys. Rev. A. 1995. Vol. 52, No. 2. P. 1229–1256.
9. Tagashira H., Sakai Y., Sakamoto S. The development of electron avalanches in argon at high E/N values. II. Boltzmann equation analysis // J. Phys. D: Appl. Phys. 1977. Vol. 10. P. 1051.
10. Zhukovskiy M.E., Uskov R.V. Mathematical modeling of radiative electron emission using hybrid supercomputers // Numerical methods and programming. 2012. Vol. 13, P. 271–279.
11. Marchenko M.A., Mikhailov G.A. Distributed computing by the Monte Carlo method. // Automation and Remote Control. 2007. Vol. 68, Iss. 5, P. 888–900.
12. Marchenko M. PARMONC - A Software Library for Massively Parallel Stochastic Simulation. // LNCS. 2011. Vol. 6873. P. 302–315.
13. Marchenko M.A. Page of PARMONC on the web site of Siberian Supercomputer Center. URL: <http://www2.sccc.ru/SORAN-INTEL/paper/2011/parmonc.htm> (accessed: 19.08.2013).
14. Jeffers J., Reinders J. Intel Xeon Phi Coprocessor High-Performance Programming. Elsevier, 2013. 432 p.
15. Lisovskiy V., Booth J.P., Landry K., et al. Electron drift velocity in argon, nitrogen, hydrogen, oxygen and ammonia in strong electric fields determined from rf breakdown curves // J. Phys. D: Appl. Phys. 2006. Vol. 39. P. 660–665.
16. Dutton J. A survey on electron swarm data // J. Phys. Chem. Ref. Data. 1975. Vol. 4, No. 3. P. 577–851.

*Поступила в редакцию 20 августа 2013 г.*

# Краткие сообщения

УДК 519.6, 615.281.8, 578.828

## ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ ПРОЦЕССОМ ПРИМЕНЕНИЯ АНТИВИРУСНЫХ ПРЕПАРАТОВ ПРИ ЛЕЧЕНИИ ВИЧ-ИНФЕКЦИИ

*И.П. Болодурин, Ю.П. Иванова*

Рассмотрена методологическая проблема применения современных математических и информационных методов для управления восстановлением иммунной системы человека в целях увеличения продолжительности его жизни. Разработана и численно решена задача оптимального управления динамикой ВИЧ-инфекции на основе применения принципа максимума Л.С. Понтрягина для нелинейных систем с постоянным запаздыванием и негладкой правой частью. На основе программной реализации численного алгоритма приведены результаты моделирования динамики иммунной системы при использовании различных стратегий применения лекарственных средств. Путем преобразования базовой системы дифференциальных уравнений, проведено исследование влияния вторичных инфекционных заболеваний на динамику ВИЧ-инфекции.

*Ключевые слова: негладкие нелинейные динамические системы с запаздыванием, оптимальное управление, математические модели ВИЧ-инфекции.*

### Введение

ВИЧ-инфекция представляет серьезную опасность, так как является одним из факторов, способствующих снижению численности человеческих популяций. Стремительный рост числа случаев ВИЧ-инфекции на всей территории России обуславливает необходимость изучения механизмов становления и развития данного вида инфекционного заболевания. Одним из подходов, направленных на изучение общей картины протекания заболевания, является математическое моделирование. Построение и исследование математических моделей заболеваний и процессов иммунной защиты, часто формулируемых в виде систем обыкновенных дифференциальных уравнений с постоянными коэффициентами, вычисляемыми путем обработки статистических данных, является объектом научных исследований, развитых в работах Ж.И. Белла, Г.И. Марчука, Л.Н. Белых, Г.А. Бочарова, И.Б. Погожева, С.М. Зуева, А.А. Романюхи, В.В. Величенко и других ученых.

Разработанные математические модели иммунных механизмов при ВИЧ-инфекции [2–5] описывают столь сложную динамику иммунной системы человека и внедряющихся в нее вирусов, что позволяют выявлять количественные характеристики, определяющие течение и исход заболевания. В случае ВИЧ-инфекции одним из таких количественных показателей служит концентрация Т-клеток в крови, определяющая категории тяжести заболевания: первая – более 600 ед/мм<sup>3</sup>, вторая – 200-600 ед/мм<sup>3</sup>, третья, приводящая к летальному исходу на фоне развития синдрома приобретенного иммунодефицита (СПИД), – менее 200 ед/мм<sup>3</sup> [6].

В последнее время выработка гибкой программы лечения ВИЧ-инфекции, основанной на управлении функционированием иммунной системы, становится одной из важ-

нейших задач медицины. В связи с этим представляют интерес задачи оптимального управления иммунным ответом, где управления можно рассматривать как функции от времени, отражающие возможные фармакологические или физиологические воздействия на иммунный процесс с целью увеличения продолжительности жизни больному.

В данной статье рассматривается применение математических и информационных методов теории управления для выявления иммунологических закономерностей развития ВИЧ-инфекции на фоне вторичных инфекционных заболеваний и оценки эффективности антивирусной терапии ВИЧ-инфицированных больных, направленной на prolongation их жизни.

Статья состоит из трех самостоятельных разделов. В первом разделе представлено исследование закономерностей динамики иммунного ответа при ВИЧ-инфекции на фоне воздействия вторичных вирусных заболеваний. Во втором разделе осуществляется управление улучшением функционированием иммунной системы для увеличения продолжительности жизни больного путем минимизации количества вируса в организме и увеличения концентрации неинфицированных Т-клеток. В качестве средства поиска оптимальных управляющих воздействий и формирования оптимальной программы лечения предложен вычислительный метод, основанный на условиях оптимальности для негладких систем с постоянным запаздыванием в фазовых переменных, который позволяет на основе компьютерной реализации оперативно дать предварительное заключение об эффективности выбранной стратегии лечения. В третьем разделе представлены оптимальные программы лечения, описано их воздействие на организм, приведены рекомендации по выбору наиболее эффективных из них. В заключении сформулированы основные выводы и результаты работы.

## 1. Математическое моделирование ВИЧ-инфекции на фоне вторичного инфекционного заболевания

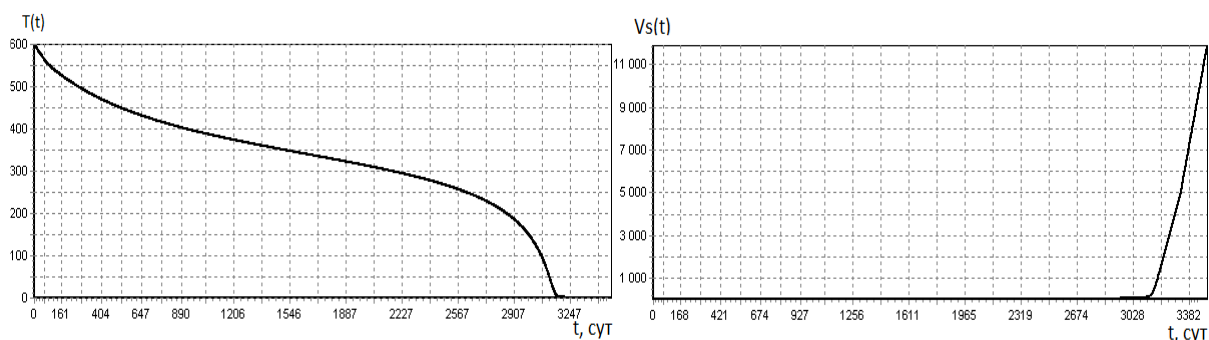
Для выявления количественных характеристик, существенных для динамики протекания ВИЧ-инфекции, в основу исследования положена модель, предложенная американскими математиками Д.Е. Киршнером и Г.Ф. Веббом, которая отражает основные механизмы взаимодействия иммунной системы и ВИЧ-инфекции [2–4]. В модели описываются процессы взаимодействия ВИЧ-инфекции ( $V_S(t)$ ) с Т-лимфоцитами, переход Т-лимфоцитов под действием вируса из разряда неинфицированных Т-клеток ( $T(t)$ ) в разряд Т-клеток, инфицированных вирусом иммунодефицита ( $T_S(t)$ ). Модель, основанная на принципах функционирования как гуморальной так и клеточной иммунной системы организма человека, представлена нелинейной системой дифференциальных уравнений с полиномиальной правой частью, записанной в нормальной форме Коши

$$\begin{aligned} \dot{V}_S(t) &= \frac{\lambda_3}{C_S + V_S(t)} T_S(t) V_S(t) - k_{i_s} T(t) V_S(t) + \frac{G_S}{B + V_S(t)} V_S(t), \\ \dot{T}(t) &= S_T - \frac{S_2}{B_S + V_S(t)} V_S(t) - \mu_T T(t) + \frac{\lambda_1}{C_S + V_S(t)} T(t) V_S(t) - k_S T(t) V_S(t), \\ \dot{T}_S(t) &= k_S T(t) V_S(t) - \mu_{T_S} T_S(t) - \frac{\lambda_2}{C_S + V_S(t)} T_S(t) V_S(t). \end{aligned} \quad (1)$$

Подробное описание модели (1) изложено в работах [2–4].

На рисунке представлены результаты численного интегрирования модели (1), полученные методом Рунге-Кутты 4-го порядка, реализованного на языке Delphi. Интегриро-

вание проводилось на отрезке времени равном 3500 дней ( $\approx 10$  лет) с начальными условиями  $T(0)=600$  ед/мм<sup>3</sup>,  $T_s(0)=0$  ед/мм<sup>3</sup>,  $V_s(0)=10$  ед/мм<sup>3</sup> при значениях параметров, представленных в работе [2].



Динамика здоровых Т-клеток и антигенов при ВИЧ-инфекции

Динамика изменения числа здоровых Т-клеток и антигенов при ВИЧ-инфекции, представленная на рисунке, хорошо согласуется с результатами клинических исследований ВИЧ [6]. Время жизни больного составило 3243 дня ( $\approx 9$  лет). При этом время наступления последней стадии развития ВИЧ-инфекции – СПИДа составило 2860 дней.

Согласно клиническим исследованиям, продолжительность жизни после инфицирования ВИЧ обычно составляет 8–10 лет, однако в некоторых случаях она может не превышать даже и одного года [4]. Одной из причин снижения продолжительности жизни ВИЧ-инфицированных людей являются наличие вторичного вирусного заболевания, вероятность заражения которым при снижении иммунитета велика. По классификации стадий ВИЧ-инфекции, предложенной В.В. Покровским [6], как правило, время особой опасности заболевания вторичным вирусом совпадает с третьей стадией развития ВИЧ-инфекции, когда концентрация Т-клеток в крови менее 200 ед/мм<sup>3</sup>.

Для исследования динамики развития ВИЧ-инфекции на фоне вторичного заболевания рассмотрен синтез моделей стороннего вирусного заболевания, основная идея построения которой предложена в работе [5], и ВИЧ-инфекции, описанной системой (1). Уравнения динамики клеточных популяций, участвующих в иммунном ответе против вируса ВИЧ-инфекции на фоне вторичного заболевания, примут вид

$$\begin{aligned} \dot{T}(t) &= S_T - \frac{S_2}{B_S + V_S(t)} V_S(t) - \frac{S_3}{B_T + V_f(t)} V_f(t) - \mu_T T(t) + \frac{\lambda_1}{C_s + V_S(t)} T(t) V_S(t) + \\ &+ \frac{\lambda_4}{C_f + V_f(t)} V_f(t) T(t) - k_S T(t) V_S(t), \\ \dot{T}_S(t) &= k_S T(t) V_S(t) - \mu_{T_S} T_S(t) - \frac{\lambda_2}{C_s + V_S(t)} T_S(t) V_S(t), \\ \dot{V}_S(t) &= \frac{\lambda_3}{C_s + V_S(t)} T_S(t) V_S(t) - k_{V_S} T(t) V_S(t) + \frac{G_S}{B + V_S(t)} V_S(t), \\ \dot{M}(t) &= S_M - \frac{S_4}{B_M + V_f(t)} V_f(t) - \alpha_M M(t) - \beta M(t) V_f(t), \\ \dot{I}(t) &= \beta M(t) V_f(t) - \alpha_I I(t) - k_I I(t) T(t), \quad \dot{V}_f(t) = p I(t) - k_{V_f} T(t) V_f(t). \end{aligned} \tag{2}$$



В модели (2)  $V_f(t)$  – клетки вируса, отличного от ВИЧ;  $M(t)$  – клетки, подвергшиеся атаке со стороны вируса вторичного заболевания;  $I(t)$  – клетки, пораженные вирусом вторичного заболевания. Параметры модели (2) – постоянные, положительные величины, часть значений которых определена в работах [2–4], а другая часть, представленная в табл. 1, получена в ходе вычислительных экспериментов по настройке модели вторичного вирусного заболевания на данные субклинической формы его протекания.

В табл. 2 представлены количественные оценки численной реализации динамики развития ВИЧ-инфекции на фоне вторичного заболевания, присоединенного на поздних стадиях развития ВИЧ. Результаты численного интегрирования получены методом Рунге-Кутты 4-го порядка, реализованным на языке Delphi. Интегрирование проводилось на отрезке времени равном 3500 дней с начальными условиями  $T(0)=T^*$  ед./мм<sup>3</sup>,  $T_s(0)=T_s^*$  ед./мм<sup>3</sup>,  $V_s(0)=V_s^*$  ед./мм<sup>3</sup>,  $M(0)=0$  ед./мм<sup>3</sup>,  $I(0)=0$  ед./мм<sup>3</sup>,  $V_f(0)=10$  ед./мм<sup>3</sup> при ранее рассматриваемых значениях параметров.

Таблица 1

Значения параметров модели инфекционного заболевания, отличного от ВИЧ

Обозначение	Параметр	Значение
$P$	Скорость воспроизводства вирусов, пораженной клеткой	0,08/день
$k_{Vf}$	Скорость уничтожения вируса здоровыми Т-клетками, вызванная иммунным ответом	0,00014 мм <sup>3</sup> /день
$S_T, S_M$	Естественная скорость размножения неинфицированных Т-клеток в отсутствие вируса	4 мм <sup>3</sup> день
$\mu_T, \alpha_M$	Естественная скорость смертности неинфицированных Т-клеток	0,005 /день
$\beta$	Скорость поражения клеток, подвергшихся атаке со стороны вируса	0,00015 мм <sup>3</sup> /день
$\alpha_I$	Скорость естественной гибели клеток, пораженных вирусом	0,23 /день
$k_I$	Скорость уничтожения клеток, пораженных вирусом, в результате иммунного ответа	0,0004 мм <sup>3</sup> /день
$S_3, S_4$	Скорость уменьшения притока неинфицированных Т-клеток	2,8 мм <sup>3</sup> день
$B_T, B_M$	Коэффициент насыщения источника неинфицированных Т-клеток	13,8 / мм <sup>3</sup>
$\lambda_4$	Скорость воспроизводства неинфицированных Т-клеток	0,025 / день
$C_f$	Коэффициент насыщения неинфицированных Т-клеток	47 /мм <sup>3</sup>

Результаты численного интегрирования модели (2) показали, что динамика вторичного заболевания на начальных стадиях развития ВИЧ-инфекции не вызывает нарушений функционирования иммунной системы организма человека и не приводит к сокращению продолжительности жизни больного. При малых концентрациях ВИЧ-инфекции, несмотря на истощение популяции Т-клеток, происходит вторичный ответ на не-ВИЧ-антигены, приводящий к избавлению организма от вторичного заболевания.

Таблица 2

Количественные оценки динамики развития ВИЧ-инфекции  
на фоне вторичного заболевания

$T^*$ , ед/мм <sup>3</sup>	$T_s^*$ , ед/мм <sup>3</sup>	$V_s^*$ , ед/мм <sup>3</sup>	Время жизни больного с ВИЧ-инфекцией, сут.		Время полного выздоровления от стороннего заболевания, сут.
			$V_f(0)=0$ ед/мм <sup>3</sup>	$V_f(0)=10$ ед/мм <sup>3</sup>	
500	9,8	11	2980	3033	48
400	9,8	15	2327	2403	66
300	9,9	21	1079	682	Выздоровление не наступает
200	9,9	33	379	179	Выздоровление не наступает

По результатам численного интегрирования модели (2), представленным в табл. 2, можно утверждать, что иммунная система на более поздних стадиях развития ВИЧ-инфекции не в состоянии эффективно справиться с вторичным заболеванием. Концентрация неинфицированных Т-клеток снижается (в то время как при отсутствии вторичного заболевания она была бы постоянной), количество вирусных клеток начинает с большей скоростью расти. Организм не в состоянии избавиться от вторичной инфекции, что приводит к тому, что больной ВИЧ-инфекцией погибает от вторичной инфекции, а не от вируса иммунодефицита в более ранние сроки.

Таким образом, как показали результаты моделирования, если организм ослаблен ВИЧ-инфекцией, то любая доза заражения другими вирусами на поздних стадиях приводит к повышению вирусной нагрузки ВИЧ и снижению иммунных показателей. Иммунная система в данном случае не в состоянии эффективно справиться с заболеванием, что требует своевременного лечебного вмешательства.

## 2. Формирование задачи оптимального управления динамикой ВИЧ-инфекции и реализация алгоритма оптимизации

Результаты, описывающие закономерности иммунного ответа при ВИЧ-инфекции, позволяют осуществлять управление улучшением ее функционирования. Управляемая модель построена на базе модели, предложенной Д.Е. Киршнером и Г.Ф. Веббом [2-4]. Модель описывает взаимодействие процессов размножения и нейтрализации вирусов ( $V(t) = V_s(t) + V_r(t)$ ), чувствительных к воздействию лекарственных препаратов ( $V_s(t)$ ) и резистентных к химиотерапии ( $V_r(t)$ ); переход под действием вируса неинфицированных Т-клеток ( $T(t)$ ) в разряд Т-клеток, инфицированных вирусом иммунодефицита, чувствительного к воздействию лекарственных препаратов ( $T_s(t)$ ), и Т-клеток, инфицированных вирусом иммунодефицита, резистентного к химиотерапии ( $T_r(t)$ ). В качестве управления, регулирующего интенсивность процессов иммунного ответа при ВИЧ-инфекции, выступают используемые при лечении препараты, приводящие к подавлению темпа поражения Т-клеток вирусом ВИЧ-инфекции ( $u_1(t)$ ) и препараты, приводящие к снижению притока вируса из внешнего лимфоидного источника ( $u_2(t)$ ).

Таким образом, управляемая модель динамики иммунной защиты при ВИЧ-инфекции описывается нелинейной системой дифференциальных уравнений с разрывной правой частью и запаздывающим аргументом

$$\begin{aligned} \frac{dT(t)}{dt} &= S_1 - \frac{S_2 V(t)}{B_s + V(t)} - \mu_r T(t) + \frac{\lambda_1 T(t-\tau)V(t-\tau)}{C+V(t)} - ((k_s - u_1(t))V_s(t) + k_r V_r(t))T(t), \\ \frac{dT_s(t)}{dt} &= (k_s - u_1(t))V_s(t)T(t) - \mu_{r_s} T_s(t) - \frac{\lambda_2 T_s(t)V(t)}{C+V(t)}, \quad \frac{dT_r(t)}{dt} = \\ &= k_r V_r(t)T(t) - \mu_{r_k} T_r(t) - \frac{\lambda_2 T_r(t)V(t)}{C+V(t)}, \\ \frac{dV_s(t)}{dt} &= \frac{(1-q)\lambda_3 T_s(t)V(t)}{C+V(t)} - k_v V_s(t)T(t) + \frac{(G_s - u_2(t))V_s(t)}{B+V(t)}, \quad \frac{dV_r(t)}{dt} = \\ &= \frac{\lambda_3 T_r(t)V(t)}{C+V(t)} + \frac{q\lambda_3 T_s(t)V(t)}{C+V(t)} - k_v V_r(t)T(t) + \frac{G_r(V(t))V_r(t)}{B+V(t)}. \end{aligned} \quad (3)$$

с начальными условиями

$$T(0) = T_0; \quad T_s(0) = T_{s0}; \quad T_r(0) = T_{r0}; \quad V_s(0) = V_{s0}; \quad V_r(0) = V_{r0}, \quad (4)$$

и фазовыми ограничениями

$$T(t) \geq 0; \quad T_s(t) \geq 0; \quad T_r(t) \geq 0; \quad V_s(t) \geq 0; \quad V_r(t) \geq 0, \quad (5)$$

где функции управления  $u_1(t)$ ,  $u_2(t)$  характеризуют скорость введения лекарственных препаратов и удовлетворяют ограничениям

$$0 \leq u_1(t) \leq b_1, b_1 < k_s; \quad 0 \leq u_2(t) \leq b_2, b_2 < G_s, \quad (6)$$

где  $b_i > 0$ ,  $i=1,2$  – максимальная доза лекарственных препаратов, вводимых в организм больного человека.

Наличие разрыва в правой части системы дифференциальных уравнений (3) связано с функцией

$$G_r(V(t)) = \begin{cases} 0, & V(t) < V^* \\ G_s, & V(t) > V^* \end{cases}, \quad (7)$$

учитывающей замещение чувствительного вируса резистентным при большой концентрации лекарственных препаратов, где  $V^*$  – пороговая концентрация вируса. Постоянное запаздывание  $\tau > 0$  объясняется тем, что вирус, попадая в организм, проявляется не сразу, а спустя некоторый промежуток времени.

Для исследования закономерностей противоинфекционной защиты предположим, что среди допустимых вариантов иммунной защиты реализуются те, которые минимизируют количество вируса в организме человека и максимизируют концентрацию неинфицированных Т-клеток за фиксированное время  $\hat{T}$ .

Тогда, в рамках построенной управляемой модели (3)-(7), выбор оптимального управления рассматривается как задача минимизации функционала  $J(u)$  на временном отрезке  $[0, \hat{T}]$

$$J(u) = \int_0^{\hat{T}} (l_1 V(t) - l_2 T(t)) dt \rightarrow \min, \quad (8)$$

где  $l_i > 0, i = 1, 2$  – весовые коэффициенты, придающие значимость одному из критериев аддитивной свертки.

Для решения краевой задачи принципа максимума Понтрягина для негладких систем с постоянным запаздыванием, технология построения которой представлена в работе [1], использован метод множителей Лагранжа, основанный на сведении исходной непрерывной задачи оптимального управления к дискретной задаче. Для реализации численного алгоритма получены условия стационарности функции Лагранжа и условия дополняющей нежесткости, которым с необходимостью удовлетворяет оптимальный процесс. Для поиска оптимальных динамических траекторий и оптимального управления программно реализован алгоритм, основанный на итерационном методе, позволяющий получить численные результаты решения поставленной задачи.

### 3. Результаты численного моделирования

Количественные оценки программ лечения ВИЧ-инфекции получены с использованием среды Delphi. Численное моделирование проводилось на отрезке времени равном 3500 дней (10 лет) с начальными условиями  $T(0)=600$  ед./мм<sup>3</sup>,  $T_s(0)=0$  ед./мм<sup>3</sup>,  $V_s(0)=10$  ед./мм<sup>3</sup> при значениях параметров, представленных в работе [2] и величиной запаздывания 0,5 дней.

Результаты расчетов задачи оптимального управления (3)–(8) показали, что после заражения организма вирусом ВИЧ-инфекции при отсутствии лечебного вмешательства уровень неинфицированных Т-клеток держится почти на постоянном уровне (около 600 ед./мм<sup>3</sup>) в течение 2,5 месяцев. Через 2,5 года развития ВИЧ-инфекции уровень здоровых Т-клеток снижается до 400, что свидетельствует о развитии второй стадии болезни. Организм устает постоянно бороться с инфекцией и уже не в состоянии вырабатывать необходимое для организма количество здоровых Т-клеток. К восьмому году развития болезни уровень здоровых Т-клеток снижается ниже допустимой нормы (меньше 200 ед./мм<sup>3</sup>), что свидетельствует о последней стадии развития болезни – СПИД. Стадия эта длится недолго – около 1 года, – после чего человек умирает.

При применении лечебного препарата, приводящего к подавлению темпа поражения Т-клеток вирусом ВИЧ-инфекции, удастся продлить первую стадию развития болезни до трех месяцев и отодвинуть стадию наступления СПИДа до 11 лет, что подтверждает положительный эффект влияния лекарственного препарата, применять который необходимо с самого начала заболевания.

Применение лечебного препарата, приводящего к снижению притока вируса из внешнего лимфоидного источника, является не столь эффективным по отношению к предыдущему методу терапии, так как при применении препарата с начала заболевания отодвигает стадию наступления СПИДа до 10 лет.

При применении сразу двух препаратов с первого дня лечения позволит продлить первую стадию развития ВИЧ-инфекции, то есть удержать стабильный уровень здоровых Т-клеток на протяжении четырех месяцев и отодвинуть стадию наступления СПИДа до 14 лет.

Результаты расчетов показали, что значение функционала, определяющее количество вируса в организме человека, заметно уменьшается с применением лечебных препаратов, в то время как суммарный уровень неинфицированных Т-клеток на всем пери-

оде исследования, несмотря на применении различных методов терапии, остается стабильным.

Таким образом, анализируя все программы лечения, можно сделать вывод, что самая лучшая программа лечения ВИЧ-инфекции – это лечение с самого начала заболевания одновременно двумя препаратами. Рассмотренные программы лечения ВИЧ-инфекции, полученные при численной реализации алгоритмов оптимизации, приводят к результатам, которые допускают содержательную интерпретацию и не противоречат имеющимся фактическим данным.

## Заключение

Работа посвящена методологической проблеме использования современных математических и информационных методов для управления иммунной системой человека при ВИЧ-инфекции. Полученные в ходе исследования модели программы лечения доказывают возможность эффективного использования математических методов теории управления при лечении ВИЧ-заболеваний. Результаты демонстрируют теоретическую возможность увеличения жизни больного за счет оптимизации программ лечения.

Рассмотренный в работе метод может быть положен в основу новых схем терапии ВИЧ-инфекции с учетом влияния вторичных заболеваний, так как компьютерная реализация модели способна оперативно дать предварительное заключение об их эффективности, подробностях воздействия на организм и рекомендации по их применению.

*Работа выполнена при финансовой поддержке гранта РФФИ, проект 12-01-31325.*

## Литература

1. Андреева, Е.А. Управление системами с последействием / Е.А. Андреева, В.Б. Колмановский, Л.Е. Шайхет. – М.: Наука, 1992. – 336 с.
2. Величенко, В.В. Возможности искусственного интеллекта и компьютерных технологий в построении программ лечения сложных иммунных заболеваний / В.В. Величенко, Д.А. Притыкин // *Фундаментальная и прикладная математика*. – М.: Открытые системы, 2009. Т. 15, № 5. – С. 21–42.
3. Величенко, В.В. Управление лечением СПИДа / В.В. Величенко, Д.А. Притыкин // *Автоматика и телемеханика*. – 2006. – № 3. – С. 166–185.
4. Величенко, В.В. Численные методы оптимального управления динамикой ВИЧ-инфекции / В.В. Величенко, Д.А. Притыкин // *Изв. РАН. Теория и системы управления*. – 2006. – № 6. – С. 53–64.
5. Куимов, В.И. Моделирование динамики ВИЧ-инфекции / В.И. Куимов, А.Н. Рапопорт // *Технологии Microsoft в теории и практике программирования*. – Нижний Новгород: Нижегородский госуниверситет, 2010. – С. 237–240.
6. Покровский, В.В. Эпидемиология и профилактика ВИЧ-инфекции / В.В. Покровский – М.: Медицина, 1996. – 275 с.

Болодурина Ирина Павловна, д.т.н., профессор, зав. кафедрой прикладной математики, Оренбургский государственный университет (Оренбург, Российская Федерация), prmat@mail.osu.ru.

Иванова Юлия Петровна, к. ф.-м.н., ст. преподаватель кафедры прикладной математики, Оренбургский государственный университет (Оренбург, Российская Федерация), ulia\_lugovskova@inbox.ru.

## OPTIMUM CONTROL OF PROCESS OF APPLICATION OF ANTIVIRUS PREPARATIONS AT HIV INFECTION TREATMENT

*I.P. Bolodurina*, Orenburg State University (Orenburg, Russian Federation),

*Yu.P. Ivanova*, Orenburg State University (Orenburg Russian Federation)

The methodological problem of application of modern mathematical and information methods for management of restoration of immune system of the person for increase in duration of his/her life is considered. Developed and numerically solved the problem of optimum control by dynamics of HIV infection on the basis of application of the principle of a maximum of Pontryagin for nonlinear systems with continuous delay and rough right part. On the basis of program realization of numerical algorithm results of modeling of dynamics of immune system are given when using various strategy of application of medicines. By transformation of basic system of the differential equations, research of influence of secondary infectious diseases on dynamics of HIV infection is conducted.

*Keywords: rough nonlinear dynamic systems with delay, optimum control, mathematical models of HIV infection.*

### References

1. Andreeva E.A., Kolmanovskii V.B., Shaikhet L.E. Upravleniye sistemami s posled-eystviem [Management of systems with aftereffect]. Moscow, Nauka, 1992. 336 p.
2. Velichenko V.V., Pritykin D.A. Vozmozhnosti iskusstvennogo intellekta i komputernix texnologii v postroenii programm lechenia slognix immunix zabolevanii [Possibilities of artificial intelligence and computer technologies in development of programs for treatment of difficult immune diseases]. Fundamental and applied mathematics. Moscow, izdatelskii dom Otkritie sistemi [open systems publishing house], 2009. Vol. 15, No. 5. P. 21-42.
3. Velichenko V.V., Pritykin D.A. Control of the medical treatment of AIDS // Automation and Remote Control. 2006. Vol. 67, No. 3. P. 493-511.
4. Velichenko V.V., Pritykin D.A. Chislennii metodi optimalnogo upravlenia dinamikoii VIT-infekciei [Numerical methods optimum control of dynamics HIV infection]. Izv. RAN. Teoria i sistemi upravlenia. [News RAS. Theory and systems control.]. 2006. No. 6. P. 53-64.
5. Kuimov V.I., Rapoport A.N. Modelirovanie dinamiki VIT-infekcii [Modeling of dynamics of HIV infection]. Texnologii Microsoft v teorii i praktike programmirovania [Microsoft Technologies in the theory and practice of programming]. Nizhny Novgorod, Nizhny Novgorod state university, 2010. P. 237-240.
6. Pokrovsk V.V. Ipedemiologia i profilaktika VIT-infekcii [Epidemiology and prevention of HIV infection]. Moscow, Medicine, 1996. 275 p.

*Поступила в редакцию 1 октября 2013 г.*

# ПРЕДСТАВЛЕНИЕ ТОРГОВЫХ СИГНАЛОВ НА ОСНОВЕ АДАПТИВНОЙ СКОЛЬЗЯЩЕЙ СРЕДНЕЙ КАУФМАНА В ВИДЕ СИСТЕМЫ ЛИНЕЙНЫХ НЕРАВЕНСТВ

*М.М. Дышаев, И.М. Соколинская*

Рассматривается применение задачи сильной отделимости для получения решений о покупке или продаже финансовых активов, таких как акции, иностранная валюта, фьючерсы и т.д. на биржевом рынке. Для этого выполнено построение двух систем линейных неравенств, задающих области в  $n$ -мерном пространстве, которые описывают экспертные торговые сигналы на основе адаптивной скользящей средней Кауфмана.

*Ключевые слова:* задача сильной отделимости, фейеровское отображение, адаптивная скользящая средняя Кауфмана, торговые сигналы для робота.

## Введение

В настоящее время около 40 % объема торгов на мировых биржах осуществляется роботами — программами, работающими по различным алгоритмам, призванным определить наилучший момент времени для покупки и/или продажи активов [1]. Рост автоматизации торговых операций на биржах, растущая сложность применяемых алгоритмов неуклонно ведут к повышению скорости изменения цен и усилению нестационарности в рыночных процессах. В качестве перспективной модели автоматического принятия решений на финансовом рынке является модель, основанная на задаче сильной отделимости.

Задача сильной отделимости заключается в нахождении слоя наибольшей толщины между двумя выпуклыми непересекающимися многогранниками [2]. Каждый многогранник описывается системой линейных неравенств в  $n$ -мерном пространстве. Система линейных неравенств для многогранника формируется на основании экспертных оценок. Такие оценки описывают практическое использование экспертного набора рыночных показателей следующего вида:

$$I_i = I_i(p, v, \tau),$$

где  $p$  — массив временных рядов цен по различным активам, по которым совершались сделки,  $v$  — соответствующие объемы сделок,  $\tau$  — набор параметров для расчета рыночного показателя.

Для каждого рыночного показателя, как правило, существует несколько эмпирически установленных условий на значения, при которых генерируется сигнал к покупке или продаже актива. Например: «покупаем, если  $I_i > 0$ , продаем, если  $I_i < 0$ » или «покупаем, при  $I_i > I_{i-1}$ , продаем при  $I_i < I_{i-1}$ ». Таким образом, из соответствующих наборов условий на значения рыночных показателей формируются две системы неравенств. Эти системы описывают две непересекающиеся области в  $n$ -мерном пространстве, где  $n$  — общее количество переменных всех используемых рыночных показателей  $I$ . Геометрически это можно представить как два выпуклых непересекающихся многогранника в  $n$ -мерном пространстве. Текущее рыночное состояние описывается точкой в этом пространстве. Если для текущего рыночного состояния одновременно выполняются все неравенства, описывающие один из многогранников, это означает, что точка рыночного состояния оказалась «внутри», и появился экспертный сигнал на покупку или продажу актива.

Как только рыночное состояние изменяется (например, на бирже заключена сделка, поменялись котировки, изменилась величина спроса или предложения и т.д.), сразу же одновременно изменяются и координаты точки, и координаты вершин многогранников в  $n$ -мерном пространстве. Таким образом, решаемая задача имеет ярко выраженный нестационарный характер.

Решение задачи сильной отделимости позволяет найти слой наибольшей толщины между указанными многогранниками и на основе этого принимать решение о покупке или продаже актива даже в том случае, когда нет «точного попадания» точки рыночного состояния в один из многогранников. В силу нестационарности исходных данных, фактически единственным эффективным методом нахождения разделяющего слоя наибольшей толщины является метод решения задачи сильной отделимости с использованием фейеровских отображений [3]. Особенностью данного метода является его высокая адаптивность к динамическому изменению исходных данных непосредственно в ходе вычислений. Недостатком указанного метода является медленная сходимость фейеровских процессов. Для преодоления этого недостатка в работах [4–6] был предложен и исследован параллельный алгоритм решения задачи сильной отделимости с использованием фейеровских отображений.

## 1. Пример построения системы линейных неравенств для адаптивного скользящего среднего Кауфмана

Рассмотрим пример построения системы линейных неравенств. В качестве примера будем использовать адаптивное скользящее среднее (частный случай экспоненциального скользящего среднего), разработанное Кауфманом [7]. Для удобства далее будем использовать аббревиатуру АМА (Adaptive Moving Average).

Согласно [8], экспоненциальное скользящее среднее (Exponential Moving Average, ЕМА) находится по рекуррентному соотношению:

$$EMA_i = a \cdot x_i + (1 - a) \cdot EMA_{i-1}, \quad (1)$$

при этом:

$EMA_0 = x_0$  — т.е. первое значение ЕМА принимается равным первому значению анализируемого числового ряда,

$a \in (0; 1)$  — «сглаживающий фактор».

Для учета нестационарности рыночных процессов Кауфман предложил использовать переменный «сглаживающий фактор», т.е. коэффициенты, которые зависят от текущей волатильности («изменчивости») рыночных цен. Авторами предложен следующий вид указанных коэффициентов:

$$a_i = a_i(x, n, f, s) = \left( \frac{|x_i - x_{i-n+1}|}{\sum_{k=0}^{n-1} |x_{i-k} - x_{i-k-1}|} \left( \frac{2}{f+1} - \frac{2}{s+1} \right) + \frac{2}{s+1} \right)^2, \quad (2)$$

где  $f$  и  $s$  — «сглаживающие» константы,  $n$  — количество периодов (сделок) для расчета среднего. Авторы рекомендуют на практике использовать  $n = 10$ ,  $f = 2$  и  $s = 30$ .



Используя соотношение (1) и учитывая, что «сглаживающий фактор» Кауфмана зависит от волатильности и меняется со временем, для АМА получаем:

$$\begin{aligned} АМА_i &= a_i \cdot x_i + (1 - a_i) \cdot АМА_{i-1}, \\ АМА_{i-1} &= a_{i-1} \cdot x_{i-1} + (1 - a_{i-1}) \cdot АМА_{i-2}, \\ &\dots \\ АМА_{i-n+1} &= a_{i-n+1} \cdot x_{i-n+1} + (1 - a_{i-n+1}) \cdot АМА_{i-n}. \end{aligned} \quad (3)$$

Учитывая, что  $АМА_{i-n} = x_{i-n}$  и подставляя соответствующие рекуррентные соотношения, получаем следующую формулу расчета  $АМА_i$  через значения числового ряда:

$$АМА_i = a_i x_i + \sum_{j=1}^{n-1} \left[ a_{i-j} x_{i-j} \prod_{k=0}^{j-1} (1 - a_{i-k}) \right] + x_{i-n} \cdot \prod_{j=0}^{n-1} (1 - a_{i-j}). \quad (4)$$

Экспертные торговые сигналы (согласно предложениям авторов) генерируются с помощью двойной фильтрации — на основе среднеквадратичного отклонения и по направлению изменения АМА:

$$\begin{aligned} \text{Покупаем, если: } &\begin{cases} АМА_i - \min_{i-n \dots i} (АМА) > \delta_i, \\ АМА_i > АМА_{i-1}. \end{cases} \\ \text{Продаем, если: } &\begin{cases} \max_{i-n \dots i} (АМА) - АМА_i > \delta_i, \\ АМА_i < АМА_{i-1}. \end{cases} \end{aligned}$$

В указанных соотношениях:

$$\delta_i = K \cdot \sigma_i - \text{значение фильтра,}$$

$\sigma_i = \sqrt{\frac{1}{n} \sum_{j=0}^{n-1} (\Delta_{i-j} - \bar{\Delta}_i)^2}$  — среднеквадратичное отклонение изменений АМА в соседних периодах.

$K$  — доля стандартного отклонения для фильтрации, (авторы рекомендуют использовать  $K = 0,1$  (т.е. 10 %) для рынка фьючерсов и форекс, и  $K = 1$  для рынка акций).

$$\Delta_i = АМА_i - АМА_{i-1} - \text{разница между «соседними» значениями АМА.}$$

$$\bar{\Delta}_i = \frac{1}{n} \sum_{j=0}^{n-1} \Delta_{i-j} - \text{матожидание } \Delta_i \text{ за } n \text{ периодов (сделок).}$$

$$\min_{i-n \dots i} (АМА) - \text{минимальное значение АМА за } n \text{ последних периодов (сделок).}$$

$$\max_{i-n \dots i} (АМА) - \text{максимальное значение АМА за } n \text{ последних периодов (сделок).}$$

## 2. Получение системы линейных неравенств в явном виде

Рассмотрим случай построения многогранника с неравенствами, отвечающими условию «покупать».

Первое неравенство (фильтрация по среднеквадратичному отклонению):

$$АМА_i - \min_{i-n \dots i} (АМА) > \delta_i.$$

Значения  $АМА_i$  для расчета  $\min_{i-n \dots i} (АМА)$  и  $\delta_i$  находятся с помощью рекуррентной формулы (3).

Неравенство в явном виде:

$$a_i x_i + \sum_{j=1}^{n-1} \left[ a_{i-j} x_{i-j} \prod_{k=0}^{j-1} (1 - a_{i-k}) \right] + x_{i-n} \cdot \prod_{j=0}^{n-1} (1 - a_{i-j}) - \min_{i-n \dots i} (AMA) - \delta_i > 0,$$

или, для наглядности:

$$\begin{aligned} & a_i x_i + a_{i-1} (1 - a_i) \cdot x_{i-1} + a_{i-2} (1 - a_i) (1 - a_{i-1}) \cdot x_{i-2} + \dots \\ & \dots + a_{i-n+1} (1 - a_i) \dots (1 - a_{i-n+2}) \cdot x_{i-n+1} + x_{i-n} (1 - a_i) \dots (1 - a_{i-n+1}) - \\ & - \min_{i-n \dots i} (AMA) - \delta_i > 0. \end{aligned}$$

Для получения второго неравенства (фильтрация по направлению изменения АМА) воспользуемся рекуррентным соотношением (3):

$$AMA_i - AMA_{i-1} = a_i \cdot (x_i - AMA_{i-1}).$$

По определению  $a_i > 0$ , и, следовательно, условие  $AMA_i > AMA_{i-1}$  тождественно неравенству  $x_i - AMA_{i-1} > 0$ .

Таким образом, с учетом вида (4) получаем второе неравенство в явном виде:

$$x_i - a_{i-1} x_{i-1} - \sum_{j=2}^n \left[ a_{i-j} x_{i-j} \prod_{k=1}^{j-1} (1 - a_{i-k}) \right] - x_{i-n-1} \cdot \prod_{j=1}^n (1 - a_{i-j}) > 0,$$

или, для наглядности:

$$\begin{aligned} & x_i - a_{i-1} x_{i-1} - a_{i-2} (1 - a_{i-1}) \cdot x_{i-2} - a_{i-3} (1 - a_{i-1}) (1 - a_{i-2}) \cdot x_{i-3} - \dots \\ & \dots - a_{i-n} (1 - a_{i-1}) \dots (1 - a_{i-n+1}) \cdot x_{i-n} - x_{i-n-1} (1 - a_{i-1}) \dots (1 - a_{i-n}) > 0. \end{aligned}$$

Окончательно, система неравенств для построения многогранника (условие «покупать») принимает вид:

$$\begin{cases} a_i x_i + \sum_{j=1}^{n-1} \left[ a_{i-j} x_{i-j} \prod_{k=0}^{j-1} (1 - a_{i-k}) \right] + x_{i-n} \cdot \prod_{j=0}^{n-1} (1 - a_{i-j}) - \min_{i-n \dots i} (AMA) - \delta_i > 0, \\ x_i - a_{i-1} x_{i-1} - \sum_{j=2}^n \left[ a_{i-j} x_{i-j} \prod_{k=1}^{j-1} (1 - a_{i-k}) \right] - x_{i-n-1} \cdot \prod_{j=1}^n (1 - a_{i-j}) > 0. \end{cases}$$

Система неравенств для построения многогранника (условие «продавать»):

$$\begin{cases} a_i x_i + \sum_{j=1}^{n-1} \left[ a_{i-j} x_{i-j} \prod_{k=0}^{j-1} (1 - a_{i-k}) \right] + x_{i-n} \cdot \prod_{j=0}^{n-1} (1 - a_{i-j}) - \max_{i-n \dots i} (AMA) + \delta_i < 0, \\ x_i - a_{i-1} x_{i-1} - \sum_{j=2}^n \left[ a_{i-j} x_{i-j} \prod_{k=1}^{j-1} (1 - a_{i-k}) \right] - x_{i-n-1} \cdot \prod_{j=1}^n (1 - a_{i-j}) < 0. \end{cases}$$

## Заключение

В данной работе проведено построение двух систем неравенств на основе экспертных сигналов, базирующихся на адаптивной скользящей средней Кауфмана. Попадание в один

многогранник дает сигнал к покупке, в другой — к продаже. Наибольший интерес представляет случай отсутствия четкого экспертного сигнала, когда точка рыночного состояния оказывается не точно «внутри» одного из многогранников, описываемых системами линейных неравенств, а находится в некоторой окрестности. Принятие обоснованного решения до момента появления четкого экспертного сигнала позволит максимизировать прибыль от торговых операций. Реализация механизма генерации сигнала на покупку или продажу актива в этом случае основывается на решении задачи сильной отделимости и определении расположения точки рыночного состояния относительно слоя наибольшей толщины, разделяющего эти многогранники.

*Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 12-01-00452 и Фонда содействия развитию малых форм предприятий в научно-технической сфере в рамках проекта № 11533р/20979.*

## Литература

1. Володин, С.Н. Проблемы распространения алгоритмической торговли на крупнейших мировых биржах / С.Н. Володин // Информационно-аналитический журнал «Политическое образование». — 2012. — URL: <http://www.lawinrussia.ru/node/252999>.
2. Ту, Дж. Принципы распознавания образов / Дж. Ту, Р. Гонсалес. Пер. с англ. Под ред. Ю.И. Журавлёва. — М.: Мир, 1978. — 411 с.
3. Еремин, И.И. Фейеровские методы сильной отделимости выпуклых полиэдральных множеств / И.И. Еремин // Известия вузов. Сер. Математика. — 2006. — № 12. — С. 33–43.
4. Ершова, А.В. Параллельный алгоритм решения задачи сильной отделимости на основе фейеровских отображений / А.В. Ершова, И.М. Соколинская // Вычислительные методы и программирование: новые вычислительные технологии. — 2011. — Т. 12. № 1. — С. 423–434.
5. Ершова, А.В. О сходимости масштабируемого алгоритма построения псевдопроекции на выпуклое замкнутое множество / А.В. Ершова, И.М. Соколинская // Вестник Южно-Уральского государственного университета. Серия: Математическое моделирование и программирование. — 2011. — № 37 (254). — С. 12–21.
6. Ершова, А.В. Исследование устойчивости параллельного алгоритма решения задачи сильной отделимости на базе фейеровских отображений / А.В. Ершова, И.М. Соколинская // Вестник Южно-Уральского государственного университета. Серия: Математическое моделирование и программирование. — 2012. — № 18 (277). — С. 5–12.
7. Kaufman, P.J. Smarter Trading: Improving Performance in Changing Markets / P.J. Kaufman. — McGraw-Hill, 1995. — 257 p.
8. Hyndman, R.J. Forecasting with Exponential Smoothing. The State Space Approach / R.J. Hyndman, A.B. Koehler, J.K. Ord, R.D. Snyder. — Springer, 2008. — 360 p.

Михаил Михайлович Дышаев, начальник отдела валютного дилинга, ОАО «Челябинскбанк» (Челябинск, Российская Федерация), [Mikhail.Dyshaev@gmail.com](mailto:Mikhail.Dyshaev@gmail.com).

Ирина Михайловна Соколинская, к.ф.-м.н., доцент, кафедра вычислительной математики, Южно-Уральский государственный университет (Челябинск, Российская Федерация), [irinasokolinsky@gmail.com](mailto:irinasokolinsky@gmail.com).

## REPRESENTATION OF TRADING SIGNALS BASED KAUFMAN ADAPTIVE MOVING AVERAGE AS A SYSTEM OF LINEAR INEQUALITIES

*M.M. Dyshaev*, JSCB CHELINDBANK (Chelyabinsk, Russian Federation),  
*I.M. Sokolinskaya*, South Ural State University (Chelyabinsk, Russian Federation)

This paper considers the adaptation of problem of strong separability for decisions about buying or selling financial assets, such as equities, currencies, futures, etc. on the stock exchange. There were constructed two systems of linear inequalities that define the regions in  $n$ -dimensional space. These systems describe the expert trading signals that based on adaptive moving average of Kaufman.

*Keywords: the problem of strong separability, Fejer mapping, adaptive moving average of Kaufman, trading signals for robot.*

### References

1. Volodin S.N. Problemy rasprostraneniya algoritmicheskoy trgovli na krupneyshikh mirovykh birzhakh. Informatcionno-analiticheskii zhurnal "Politicheskoe obrazovanie". 2012. URL: <http://www.lawinrussia.ru/node/252999>.
2. Tu Dzh., Gonsales R. Printcipy raspoznavaniia obrazov: Per. s angl. Pod red. Iu.I. Zhuravlyova. M.: Mir, 1978. 411 p.
3. Eremin I.I. Feierovskie metody silnoi otdelimosti vypuclykh poliedralnykh mnozhestv. Izvestiia vuzov. Ser. Matematika. 2006. No 12. P. 33–43.
4. Yershova, A.V., Sokolinskaya, I.M. Parallel'nyy algoritm resheniya zadachi silnoy otdelimosti na osnove feyyerovskikh otobrazheniy. Vychislitel'nyye metody i programmirovaniye: novyye vychislitel'nyye tekhnologii. 2011. T. 12. No 1. P. 423–434.
5. Yershova, A.V., Sokolinskaya, I.M. O skhodimosti masshtabiruyemogo algoritma postroyeniya psevdoproektsii na vypukloye zamknutoye mnozhestvo. Vestnik Yuzhno-Ural'skogo gosudarstvennogo universiteta. Seriya: Matematicheskoye modelirovaniye i programmirovaniye. 2011. No 37 (254). P. 12–21.
6. Yershova, A.V., Sokolinskaya, I.M. Issledovaniye ustoychivosti parallel'nogo algoritma resheniya zadachi sil'noy otdelimosti na baze feyyerovskikh otobrazheniy. Vestnik Yuzhno-Ural'skogo gosudarstvennogo universiteta. Seriya: Matematicheskoye modelirovaniye i programmirovaniye. 2012. No 18 (277). P. 5–12.
7. Kaufman, P.J. Smarter Trading: Improving Performance in Changing Markets. McGraw-Hill, Inc. 1995. 257 p.
8. Hyndman R.J., Koehler A.B., Ord J.K., Snyder R.D. Forecasting with Exponential Smoothing. The State Space Approach. Springer. 2008. 360 p.

*Поступила в редакцию 24 сентября 2013 г.*

# РАЗРАБОТКА ВИРТУАЛЬНОГО ИСПЫТАТЕЛЬНОГО СТЕНДА ДЛЯ ЧИСЛЕННОГО МОДЕЛИРОВАНИЯ ГИДРОДИНАМИЧЕСКИХ ТЕЧЕНИЙ В БЕСФЛАНЦЕВЫХ ВИХРЕВЫХ РАСХОДОМЕРАХ С ИСПОЛЬЗОВАНИЕМ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ

*Е.В. Сафонов, К.А. Бромер, В.А. Дорохов*

В работе изложены результаты по созданию виртуального стенда и отработке CFD моделей проточных частей бесфланцевых расходомеров на суперкомпьютере «Горнадо ЮУрГУ». Представлена структура виртуального стенда для проведения параметрических расчетов. Представлены результаты численного моделирования течения в проточной части бесфланцевого вихревого расходомера для сжимаемой (воздух) и несжимаемой среды (вода).

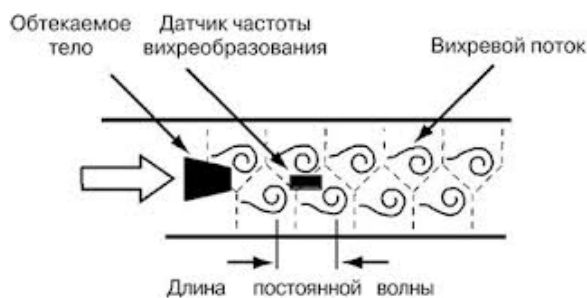
*Ключевые слова:* вихревой расходомер, тело обтекания, дорожка кармана, моделирование внутренних течений, виртуальный стенд, кластер, CFD

## Введение

Объектом исследования является проточная часть бесфланцевого вихревого расходомера, внутри которой установлено тело обтекания и сенсор, включающий датчик частоты вихреобразования (рис. 1).



а) схема проточной части



б) схема образования вихрей

**Рис. 1.** Проточная часть вихревого расходомера

Вихревые расходомеры предназначены для измерения объемного расхода жидких или газообразных сред. Принцип действия вихревого расходомера основан на зависимости частоты срыва вихрей с поверхности плохообтекаемого тела, помещенного в трубопроводе диаметрально оси трубы, от объемного расхода. Сигналы с чувствительного элемента (сенсора), расположенного в проточной части, усиливаются, преобразуются и передаются для дальнейшей обработки.

Особенность исследования заключается в том, что разрабатываются и анализируются модели бесфланцевых расходомеров, имеющих ступенчатый и конусный переход с

большого диаметра подводящего трубопровода на диаметр проточной части расходомера (рис. 2, 3).

Для отработки и поверки расходомеров в настоящее время применяются проливной метод. Поверка проливным методом осуществляется на проливной установке, где через поверяемый прибор пропускается строго определенное количество жидкости. Достоинством проливных установок является возможность исследования на них расходомеров любых конструкций. К сожалению, для такого метода требуется не только проливная установка, но и разработанный расходомер, что является крайне дорогим и трудозатратным, поэтому стоимость проливной поверки достаточно высока и сопоставима с ценой самого расходомера.

Задача по отработке конструкции вихревого расходомера может быть решена с помощью виртуального испытательного стенда (ВИС). Идея виртуального стенда позволяет решить вопрос об использовании суперкомпьютеров для решения сложных задач инженерного моделирования путем аренды вычислительных и программных ресурсов в режиме удаленного доступа у центров коллективного пользования. В случае использования ВИС конечному пользователю посредством Интернет предоставляется специализированный графический интерфейс, предназначенный для задания параметров вычислительных моделей, запуска вычислений на кластере и отображения результатов расчета.

В данной статье рассматривается технология виртуального испытательного стенда для численного моделирования гидродинамических процессов в вихревых расходомерах с помощью Ansys CFX. В разделе 1 рассматривается расчетная модель расходомера, описывается интерфейс и схема организации передачи файлов. В разделе 2 приводится сравнение экспериментальных данных численного и физического эксперимента. В заключении делается вывод о значимости использования виртуального испытательного стенда и о точности расчетов.

## 1. Технология виртуального испытательного стенда

Для создания ВИС использовались технологии, обеспечивающие автоматизированную генерацию проблемно-ориентированных сервисов на основе оболочки CAEVan и позволяющих использовать программные системы для инженерного проектирования и анализа в распределенных вычислительных средах [1, 2]. Данная технология представляет собой комплекс моделей, методов и алгоритмов, направленных на автоматизированное создание иерархий распределенных проблемно-ориентированных оболочек для работы с CFD пакетами на основе сервисно-ориентированного подхода и концепции облачных вычислений.

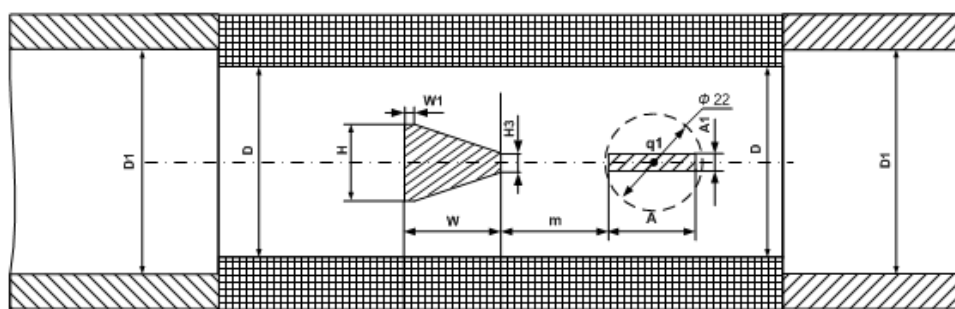


Рис. 2. Эскиз проточной части бесфлацевого расходомера со ступенчатым входом

Проблемно-ориентированная оболочка САЕВеап представляет собой клиентский графический интерфейс, в котором пользователь может задавать параметры моделируемого процесса, посредством доступа через каналы сети интернет, защищенных системой аутентификации пользователей и шифрованием передаваемых данных.

При этом для работы с проблемно-ориентированной оболочкой не требуется установки какого-либо программного обеспечения на компьютер пользователя, поскольку для этого достаточно стандартного интернет-обозревателя. На рис. 4 представлены элементы графического интерфейса ВИС по расчету бесфланцевых расходомеров.

В ходе выполнения работ было разработано три вычислительных сервиса различного назначения:

- вычислительный сервис для экспериментов на воде;
- вычислительный сервис для экспериментов на воздухе;
- вычислительный сервис для прямого доступа к решателю ANSYS CFX.

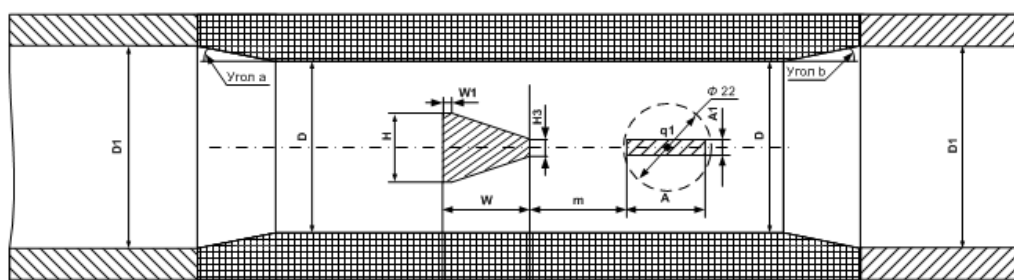


Рис. 3. Эскиз зауженной проточной части бесфланцевого расходомера

Графический интерфейс позволяет задавать геометрические размеры всех внутренних частей проточной камеры, а также их расположение друг относительно друга также следующие параметры CFD расчета: давление среды (Па); скорость среды (м/с); количество шагов расчета; продолжительность каждого шага (с); модель турбулентности (SST,  $k-\epsilon$ ; максимальное количество итераций на один шаг; минимальное количество итераций на один шаг; критерий сходимости; продолжительность реального времени процесса моделирования).

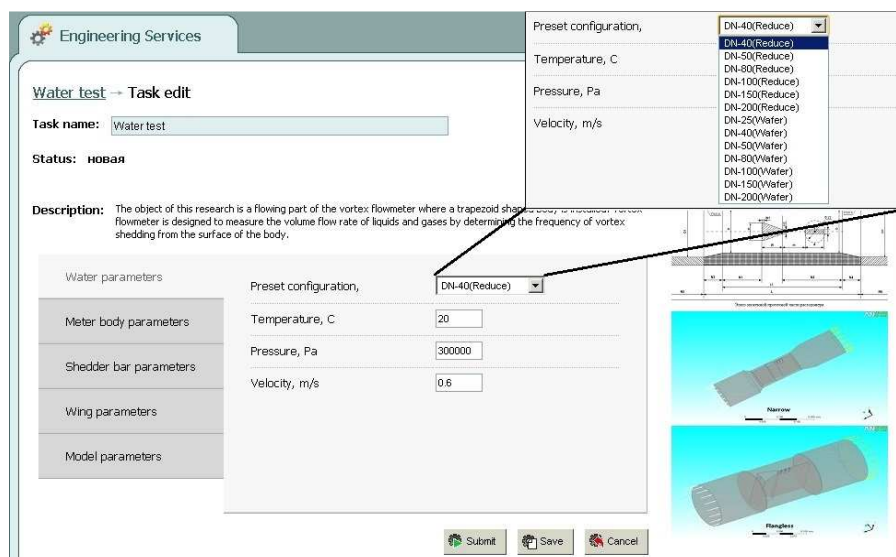


Рис. 4. Интерфейс вычислительного сервиса по воде. Параметры среды

Процесс компьютерного моделирования турбулентных течений в проточной части вихревого расходомера осуществляется в CFD-пакете Ansys CFX, поэтому для создания виртуального испытательного стенда с проблемно-ориентированной оболочкой реализовано автоматическое построение геометрии и расчетной сетки в соответствии с заданными параметрами и запуск процесса моделирования.

Для автоматизации обозначенных процессов были разработаны параметризованные шаблоны геометрии (SolidWorks), расчетной сетки (ICEM CFD) и сценарии (Shell script), выполняющие действия по обновлению геометрии и построению расчетной сетки. Схема, представленная на рис. 5, работает следующим образом.

На основе шаблона геометрии \*.sldprt, файла сценария и параметров геометрии в соответствии с эскизом (см. рис. 2, 3) запускается процесс генерации выбранной геометрии с последующим сохранением ее в формате ParaSolid в файлы InOut.x\_b и flowmeter\_geometry.x\_b. Данное действие выполняет SolidWorks.

На основе файла геометрии (flowmeter\_geometry.x\_b – расходомер, InOut.x\_b – входной, выходной участок) и сценария построения расчетной сетки (mesh.rpl, inout.rpl) в модуле ICEM производится построение новой расчетной сетки с сохранением в файл формата \*.cfx5 (flowmeter.cfx5, inout.cfx5).

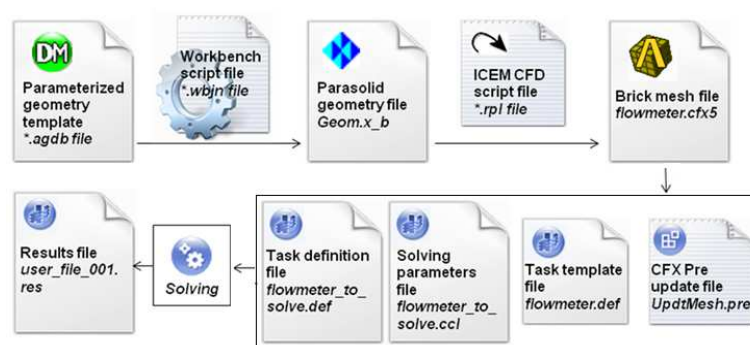


Рис. 5. Схема процесса автоматизации в ВИС

В программный модуль Ansys CFX Pre подгружается шаблон CFD модели flowmeter.def, в который с помощью скрипта update\_mesh.pre импортируется новая расчетная сетка и задаются параметры модели.

В файле flowmeter.def заранее указан тип среды: вода или воздух. В результате работы данного блока образуются файл постановки задачи flowmeter\_to\_solve.def и файл настроечных параметров flowmeter\_to\_solve.ccl, в котором внешней консольной программой прописываются значения параметров расчета. Далее запускается сценарий расчета на основе сгенерированного def файла на узлах суперкомпьютера.

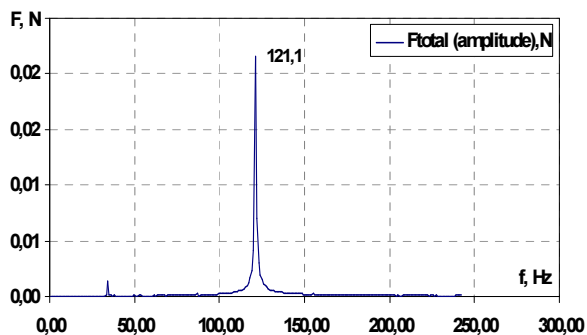
Все дальнейшие обращения к суперкомпьютеру производятся через сервер распределенных расчетов Unicore 6.3.

## 2. Результаты тестирования ВИС

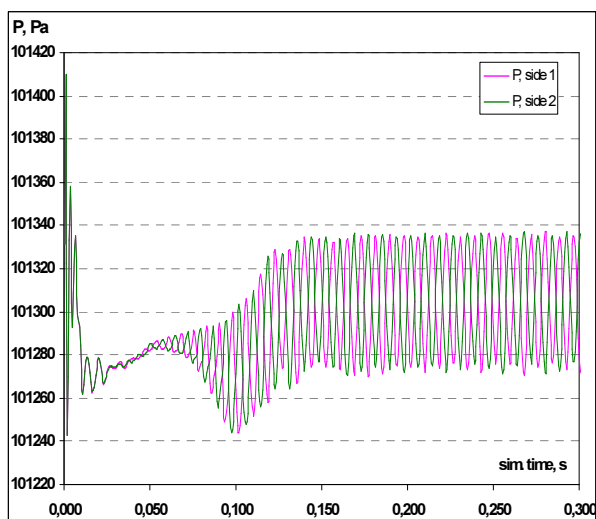
В данной статье отображены результаты расчетов с использованием ВИС для расходомера Ду50 при продувке воздухом. Сравнение результатов численного моделирования гидродинамических процессов в проточной части расходомера с результатами натуральных испытаний приведены на рис. 8.



В ходе исследования определялись частота и амплитуды пульсаций давления в потоке за телом обтекания (рис. 6, 7), зависимости для безразмерных чисел Струхалия и Рейнольдса (рис. 8).



**Рис. 6.** Спектр результирующей силы, действующей на сенсор ДУ 50

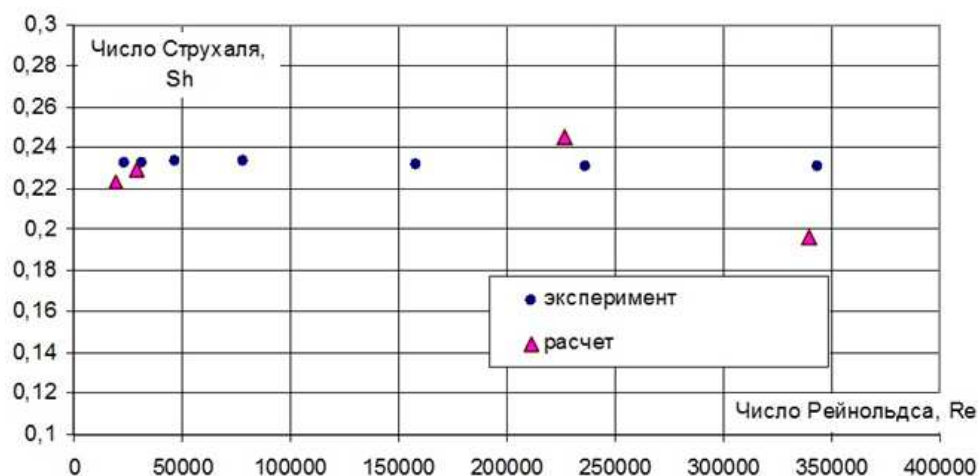


**Рис. 7.** Реализация во времени статического давления с двух сторон сенсора ДУ 50

Результаты моделирования процессов в проточной части расходомеров представлялись интегральными параметрами в виде частоты колебаний давления, графиков изменения давления во времени, графиков спектров частот пульсаций давления, а также зависимостями числа Струхалия ( $Sh$ ) от числа Рейнольдса ( $Re$ ), рассчитанными по параметрам потока перед телом обтекания как

$$Sh = \frac{f \cdot H}{V_{cp}} \quad \text{и} \quad Re = \frac{V_{cp} \cdot D}{\nu},$$

где  $f$  – частота генерации вихрей,  $H$  – поперечный размер тела обтекания;  $\nu$  – кинематическая вязкость при температуре среды;  $D$  – диаметр проточной части;  $V_{cp}$  – осредненная скорость по сечению.



**Рис. 8.** Зависимость безразмерной частоты пульсаций ( $Sh$ ) от числа Рейнольдса ( $Re$ ) для проточной части ДУ50 для численного и физического экспериментов

В результате численного моделирования установлено, что устойчивый вихревой след с приемлемой амплитудой давления генерировался телом обтекания для всего диапазона рабочих величин  $Re$ .

## Заключение

Статья посвящена проблеме сокращения затрат на разработку новых или модернизацию существующих вихревых расходомеров путем использования программного обеспечения, реализующего моделирование гидрогазодинамических процессов в многомерной постановке (CFD-технология) с привлечением вычислительных возможностей современных многопроцессорных вычислительных устройств.

Предложено решение этой проблемы за счет создания ВИС, привлечения суперкомпьютера и CFD пакетов. С помощью ВИС проведено численное моделирование гидродинамических процессов в проточной части бесфланцевого вихревого расходомера, а также выполнены физические проливки расходомеров на стенде.

Сравнение данных численного и физического экспериментов показало удовлетворительное количественное совпадение. Относительная погрешность численного моделирования составляет не более  $\pm 8,0$  % для крайних точек интервала скоростей потока и не более  $\pm 5$  % для середины диапазона скоростей потока.

Таким образом, применение ВИС с программным обеспечением CFD, реализующего моделирование гидрогазодинамических процессов в проточной части расходомера, позволяет сократить затраты на создание и модернизацию вихревых расходомеров при разработке и серийном производстве.

В качестве направлений будущих исследований целесообразно сосредоточится на оптимизации расчетной сетки цифровых 3D-моделей при моделировании гидрогазодинамических процессов, сокращении машинного времени при моделировании гидрогазодинамических процессов в вихревых расходомерах, модернизации интерфейса виртуального испытательного стенда, оптимизации процедур и процессов взаимодействия виртуального стенда с CFD программой ANSYS CFX при запуске расчетов через удаленный доступ.

*Работа выполнена по государственному заданию в рамках тематического плана исследований ФГБОУ ВПО ЮУрГУ (НИУ) № 7.5141.2011/01201255644.*

## Литература

1. Kesselman, C. The Anatomy of the Grid: Enabling Scalable Virtual Organizations / C. Kesselman, I. Foster, S. Tuecke // International Journal of Supercomputer Applications and High Performance Computing. — 2001. — Vol. 15. — №. 3. — P. 200–222.
2. Радченко Г.И. Методы организации грид-оболочек системного слоя в технологии CAEBeans / Г.И. Радченко // Вестник ЮУрГУ. Серия «Математическое моделирование и программирование». — 2008. — № 15 (115), Вып. 1. — С. 69–80.

Сафонов Евгений Владимирович, кандидат технических наук, доцент, и.о. заведующего кафедрой двигателей летательных аппаратов, Южно-Уральский государственный университет (Челябинск, Российская Федерация), e-safonov@yandex.ru.

Бромер Константин Александрович, инженер кафедры двигателей летательных аппаратов, Южно-Уральский государственный университет (Челябинск, Российская Федерация), bromer75@mail.ru.

Дорохов Валентин Александрович, программист Суперкомпьютерного центра, Южно-Уральский государственный университет (Челябинск, Российская Федерация), dorohovv85@mail.ru

## DEVELOPMENT OF VIRTUAL TEST BENCH FOR CFD IN FLANGELESS VORTEX FLOWMETER WITH APPLICATION OF HIGH-PERFORMANCE COMPUTING

*E. V. Safonov*, South Ural State University (Chelyabinsk, Russian Federation)

*K.A. Bromer*, South Ural State University (Chelyabinsk, Russian Federation)

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

This paper describes results of development of virtual test bench and using of CFD models for inner part of flangeless vortex flowmeter for virtual test bench running on Tornado SUSU supercomputer. The structure of virtual test bench is presented. The results of numerical simulation of flow (air and water) in inner part of flangeless vortex flowmeter are introduced.

*Keywords:* vortex flowmeter, blunt body, Karman vortex street, simulation of fluid, virtual bench, cluster, CFD

### References

1. Kesselman C., Foster I., Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations // International Journal of Supercomputer Applications and High Performance Computing. 2001. Vol. 15. № 3. P. 200–222.
2. Radchenko G.I. Methodi organizacii grid-tehnologii systemnogo sloya v tehnologii CAEBeans [Methods of organization grid- cloud of system in CAEBeans technology]. Vestnik Yuzho-Uralskogo gosudarstvennogo universiteta. Seriya "Matematicheskoe modelirovanie i programmirovaniye" [Bulletin of South Ural State University. Series: Mathematical Modeling, Programming & Computer Software]. 2008. No. 15(115). Issue 1. – P.69 – 80.

*Поступила в редакцию 11 октября 2013 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](mailto: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).