На правах рукописи

Германенко Максим Игоревич

КОМПЛЕКС ПРОГРАММ ДЛЯ БЕЗОШИБОЧНЫХ ДРОБНО-РАЦИОНАЛЬНЫХ ВЫЧИСЛЕНИЙ И ЕГО ПРИМЕНЕНИЕ ДЛЯ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

05.13.18 – математическое моделирование,
численные методы и комплексы программ

Автореферат
диссертации на соискание ученой степени
кандидата физико-математических наук

Челябинск – 2009

Работа выполнена на кафедре экономико-математических методов и статистики Южно-Уральского государственного университета.

Научный руководитель – доктор физико-математических наук, профессор Панюков Анатолий Васильевич.

Официальные оппоненты: доктор физико-математических наук, профессор Кипнис Михаил Маркович;

доктор физико-математических наук, профессор Сурнев Виктор Борисович.

Ведущая организация – Институт математики и механики УрО РАН.

Защита диссертации состоится 18 ноября 2009 г., в 12 часов, на заседании диссертационного совета Д 212.298.14 при Южно-Уральском государственном университете по адресу: 454080, г. Челябинск, пр. им. В.И. Ленина, 76.

С диссертацией можно ознакомиться в библиотеке Южно-Уральского государственного университета.

Автореферат разослан 25 сентября 2009 г.

Ученый секретарь
диссертационного совета Л.Б. Соколинский

Общая характеристика работы

Актуальность темы. Решение систем линейных алгебраических уравнений является одной из фундаментальных задач математики. В частности, она возникает при решении краевых задач для дифференциальных и интегральных уравнений, к которым сводятся реальные проблемы техники, физики, экономики, математики и др.

Некоторые методы решения данной задачи, такие как метод *Гаусса*, метод *Жордана–Гаусса*, метод прогонки, прямое использование формул *Крамера* и др., определены в терминах точных вычислений. Но использование стандартных типов данных известных языков программирования, существенно сужает множество рациональных чисел, представимых без погрешности. Таким образом, арифметические операции приходится выполнять приближенно, что часто дает неудовлетворительные результаты решения задач.

Интересно заметить, что в последнее время теория и практика решения плохо обусловленных линейных систем развивается в направлении разработки алгоритмов, устойчивых к погрешностям округления промежуточных результатов. Примерами таких методов являются: метод вращений, метод отражений и др. Они содержат операции извлечения квадратного корня, вычисление синуса, косинуса и прочих иррациональных функций, т.е. ориентированы на вычисления с приближенными числами. Методы, не ориентированные на безошибочные вычисления, как правило, не распознают случаи, когда система имеет бесконечное множество решений или не имеет их вообще, выдавая ошибочные ответы. При вычислениях с округлениями, возможно, что 1) не будет найдено ни одного подходящего решения, даже если оно имеется; 2) найдены корни при их отсутствии; 3) найдено только одно решение, при их бесконечном множестве. При безошибочных вычислениях все три случая легко идентифицировать.

Общеизвестные алгоритмы решения систем линейных алгебраических уравнений: метод *Гаусса*, метод *Жордана–Гаусса*, метод прогонки, – для преобразования данных используют только основные арифметические операции. Но на сегодняшний день нам не известно языков программирования, представляющих программисту целочисленные типы данных с более чем 64 двоичными разрядами. Однако, при использовании в указанных алгоритмах безошибочных вычислений будут исключены все возможные методические погрешности решения (так как все промежуточные операции будут выполняться точно, без округлений), останутся только погрешности, обусловленные неточностью исходных данных.

По-видимому, первый комплекс программ для безошибочных дробно-рациональных вычислений реализован в 1986 г. в Пермском государственном университете под руководством А.Н. Румянцева. Однако факты массового использования данного комплекса программ для реальных вычислений не известны.

Развитие объектно-ориентированного программирования позволило пользователям дополнить стандартные числовые типы данных. Используя символьное программирование, японские программисты К.Ш.Тан, В.Х.Стиб, Й.Харди разработали класс, позволяющий выполнять операции со сверхдлинными числами. Данный класс основан на символьном программировании с использованием десятичной системы счисления. Для 32-битных операционных систем более рациональным является использование систем счисления с основанием 2^{16}.

Указанного недостатка не имеет библиотека GMP. Данная библиотека разработана под *операционные системы* Unix, Linux и подобные малопопулярные в нашей стране операционные системы, ее использование требует компиляции и дополнительных знаний, что затрудняет использование данной библиотеки для рядового программиста. Стоит также отметить, что для объектов GMP не предоставляется возможность их использования в параллельных вычислениях.

Целью диссертации является разработка программного обеспечения для безошибочных дробно-рациональных вычислений с использованием суперкомпьютерных технологий и его применение для решения систем линейных алгебраических уравнений. В связи с поставленной целью решаются следующие задачи:

1. Разработка классов, обеспечивающих безошибочные дробно-рациональные и матричные вычисления.
2. Разработка параллельного программного обеспечения для безошибочного решения систем линейных алгебраических уравнений и нахождения обобщенной обратной матрицы.
3. Анализ сложности безошибочного решения систем линейных алгебраических уравнений и нахождения обобщенной обратной матрицы и эффективности распараллеливания.

Научная новизна. Разработано программное обеспечение для безошибочного решения систем линейных алгебраических уравнений, найдены битовые сложности безошибочного решения систем различными методами.

Наиболее существенные результаты, полученные автором в результате исследования и выносимые на защиту, состоят в следующем:

1. Доказано, что при безошибочном решении систем линейных алгебраических уравнений методом *Жордана–Гаусса* пространственная битовая сложность не превышает $O\left(l^2\right)$, а вычислительная битовая сложность при использовании алгоритма умножения столбиком и быстрого алгоритма не превышает $O\left(l^{3,5}\right)$ и $O\left(l^{2,5}\right)$ соответственно, где l – число бит требуемых для представления исходных данных.
2. Доказано, что при безошибочном решении систем линейных алгебраических уравнений с трехдиагональной матрицей методом прогонки пространственная битовая сложность не превышает $O\left(l^{1,5}\right)$, а вычислительная битовая сложность при использовании алгоритма умножения столбиком и быстрого алгоритма не превышает $O\left(l^3\right)$ и $O\left(l^{1,5}\right)$ соответственно.
3. Доказано, что при безошибочном нахождении обобщенной обратной матрицы *Мура–Пенроуза* методом Эрмита пространственная битовая сложность не превышает $O\left(l^4\right)$, а вычислительная битовая сложность при использовании алгоритма умножения столбиком и быстрого алгоритма не превышает $O\left(l^7\right)$ и $O\left(l^5\right)$. Результаты теоретического исследования показывают возможность практического использования программы, как при анализе, так и при безошибочном решении систем линейных алгебраических уравнений.
4. Разработана библиотека классов ***ExactComp***, в состав которой включены классы ***overlong, rational*** и ***matrix***, расширяющие возможности безошибочных вычислений и позволяющие использовать безошибочные матричные вычисления при решении плохо обусловленных и некорректных систем линейных алгебраических уравнений в программах пользователя. Разработанные классы адап-

тированы для параллельных вычислений. Проведен анализ ускорения использования параллельных вычислений для решения данных задач.

Теоретическая значимость. Получены оценки безошибочного решения систем линейных алгебраических уравнений и нахождения обобщенной обратной матрицы различными методами. Используя полученные оценки можно выбрать оптимальный метод, при использовании безошибочных вычислений. Также были получены оценки ускорения безошибочных вычислений при использовании суперкомпьютеров, что поможет рассчитать требуемые ресурсы для решения задач.

Практическая значимость. К вычислению обобщенной обратной матрицы сводятся реальные задачи экономики, математики, физики. Представленные в работе библиотеки (свид. РОСПАТЕНТ об официальной регистрации программы для ЭВМ № 2009612777, № 990607) позволяют реализовать безошибочное решение данной задачи.

Проведенные практические эксперименты являются подтверждением теоретических исследований, а также показывают не только возможность, но и необходимость безошибочного решения систем линейных алгебраических уравнений. Необходимость вызвана тем, что при приближенном решении конечный результат будет существенно отличаться от правильного, но это можно избежать, если выполнять все операции точно.

Методы исследования. Проведенные в работе исследования базируются на методах решения систем линейных алгебраических уравнений, методы нахождения обобщенной обратной матрицы, методы безошибочных вычислений, нахождение битовой вычислительной сложности, практические эксперименты.

Апробация работы. Основные положения диссертации докладывались и обсуждались на следующих конференциях:

1. Международная научная конференция «Параллельные вычислительные технологии ПАВТ’2009», г. Нижний Новгород, март – апрель 2009 г.
2. Международная научная конференция «Параллельные вычислительные технологии ПАВТ’2008», г. Санкт-Петербург, январь – февраль 2008 г.
3. Международная научная конференция «Параллельные вычислительные технологии ПАВТ’2007», г. Челябинск, январь – февраль 2007 г.
4. «Третья российско-германская школа по параллельным вычислениям на высокопроизводительных системах», г. Новосибирск, сентябрь 2006 г.
5. «Всероссийский конкурс студенческих работ в области естественных наук», г. Москва, декабрь 2004 г.
6. Российская молодежная научная и инженерная выставка «Шаг в будущее», г. Москва, МГТУ им. Н. Э. Баумана, апрель 2002 г.
7. Российская молодежная научная и инженерная выставка «Шаг в будущее», г. Москва, МГТУ им. Н. Э. Баумана, февраль 2001 г.
8. Седьмая научная конференция молодых исследователей «Шаг в будущее», г. Москва, МГТУ им. Н. Э. Баумана, февраль 2000 г.
9. Российская молодежная научная и инженерная выставка «Шаг в будущее», г. Москва, МГТУ им. Н.Э. Баумана, апрель 2000 г.

Кроме того, результаты работы были представлены на ежегодных научно-практических конференциях Южно-Уральского государственного университета.

Работа награждена дипломами лауреата Российских научных мероприятий:

- Диплом 1 степени по итогам 2 тура Всероссийского конкурса студенческих работ в области естественных наук, Саратов 2004 год.

- Диплом министерства промышленности, науки и технологий РФ за первое место в номинации «Абсолютное первенство – лучшая работа в области естественных наук», Москва, МГТУ им. Н.Э. Баумана, 2001 год.
- Диплом комитета общественных и межрегиональных связей Правительства Москвы за первое место в номинации «Лучшая работа в области Естественных наук», МГТУ им. Н.Э.Баумана, 2001 год.

Публикации. Основные результаты диссертации опубликованы в 15 научных работах, в том числе 2 публикации их списка ВАК, также получены 2 свидетельства РОСПАТЕНТ об официальной регистрации программы.

В работах [2–8] научному руководителю принадлежит постановка задачи и обсуждение результатов.

В работе [3] описание класса ***rational***, написание ряда алгоритмов класса принадлежит автору диссертации.

В работах [4, 5] доработка для параллельных вычислений классов ***overlong*** и ***rational***, написание класса ***matrix*** и программы алгоритмов решения линейных систем осуществлено автором диссертации.

Объем работы. Диссертационная работа состоит из введения, трех глав, списка используемой литературы и 4-х приложений; изложена на 131 страницах машинописного текста, содержит 35 рисунков и 7 таблиц, библиография содержит 77 наименований. В приложения сведения о внедрении работы и фрагменты исходных текстов программ.

Содержание работы

Во **введении** обоснована актуальность темы диссертационной работы, проведен обзор аналогов и описаны их недостатки, сформулирована цель работы, кратко охарактеризована научная новизна, возможности научного и практического применения, отмечена связь проблемы с планами научных исследований. Кроме того, приведено краткое изложение материала работы по главам и параграфам.

В **первой главе** произведен обзор алгоритмов основных арифметических операций для целочисленных вычислений. Описан класс ***overlong***, который существенно расширяет *логические* возможности целочисленных вычислений: диапазон представимых чисел расширяется до $(-(2^{16})^{65536}, +(2^{16})^{65536})$. Таким образом, имеется возможность представлять целые числа, имеющие более 600 000 десятичных разрядов. Ранее доказано, что битовая пространственная сложность результата арифметической операции $\circ \in \{+,-,/,\times\}$ над рациональными числами p, q не превосходит величины $L(p \circ q) \le L(p) + L(q)$. Проведенный анализ битовой вычислительной сложности, при выполнении алгоритма столбиком, показал что имеет место следующее неравенство $C(p \circ q) \le L(p) \cdot L(q)$. При использовании быстрого алгоритма умножения вычислительная сложность операций с дробно-рациональными числами не будет превосходить значения

$$O((L(p)+L(q))\log_2(L(p)+L(q))).$$

Описан класс ***rational***, который дает потенциальную возможность использовать в программах пользователя безошибочное выполнение основных арифметических операций над полем рациональных чисел. По определению, тип данных ***rational*** представляет собой пару <*nmr, dmr*>. Здесь *nmr* – целочисленная переменная типа ***overlong***, обозначающая числитель, а *dmr* – целочисленная переменная типа ***over-***

long, обозначающая знаменатель. Минимальный шаг дискретизации представляемых чисел существенно лучше, чем у стандартных типов данных и равен $2^{-1048575}$.

Проведенное практическое исследование, показало, что наиболее оптимальный метод сокращения – попарно сократить операнды до выполнения операции.

Для использования классов ***overlong*** и ***rational*** достаточно подключить заголовочный файл *ExactComp.h* в свою программу, после этого можно пользоваться объектами данных классов как объектами стандартных типов данных. Над объектами классов определены все основные арифметические операции, бинарные отношения, операторы ввода/вывода. Объем памяти, требуемый для представления объекта, зависит от значения представляемого числа.

В работе показано, что битовая пространственная и вычислительная сложности сложения $n\times m$ матрицы A и B не превосходят величин

$$L(A+B)\le L(A)+L(B),\quad C(A+B)\le nm(M_A+M_B)$$

соответственно, а битовая пространственная и вычислительная сложности умножения $n\times m$ матрица A и $m\times l$ матрица B не превосходят величин

$$L(A\cdot B)\le nml(M_A+M_B),\quad C(A\cdot B)\le O\left(nm^2lC_*(A,B)\right)$$

соответственно, где C_* – битовая вычислительная сложность выбранного метода умножения. Здесь M_A – число бит требуемых для представления матрицы A.

Разработанный класс ***matrix***, предназначен для облегчения программирования, а также улучшений визуального восприятия программ, использующих матричные вычисления. В данный класс встроены методы решения систем линейных уравнений с заданной матрицей, нахождения обратной матрицы и нахождения обобщенной обратной матрицы. Добавлена возможность использования параллельных вычислений.

Во **второй главе** рассмотрен метод решения систем линейных алгебраических уравнений с гарантированной оценкой погрешности при приближенно заданных исходных данных. Известно, что система $Ax=b$ с приближенно заданной матрицей имеет единственное решение, если

$$\|\Delta\|\le\frac{1}{\|A^{-1}\|},$$

где $\|\Delta\|$ – норма матрицы погрешностей, а $\|A^{-1}\|$ – норма обратной матрицы системы. Если система имеет единственное решение, тогда имеет место оценка

$$\|\Delta_x\|\le\frac{\|\Delta_A\|}{1-\|\Delta_A\|\cdot\|A^{-1}\|}\cdot\|A^{-1}\|^2\cdot\|b\|.$$

Для безошибочного вычисления, как обратной матрицы A^{-1}, так и решения x можно использовать метод *Жордана–Гаусса*, а при использовании разработанного класса ***rational*** возможно исключить все методические погрешности. Известно, что алгоритм *Жордана–Гаусса* имеет алгебраическую пространственную сложность $O(n^2)$ и алгебраическую вычислительную сложность $O(n^3)$. Данные оценки позволяют определить количество переменных, требуемых для решения задачи и количество арифметических операций с этими переменными. При использовании безошибочных вычислений длина переменной зависит от представляемого ей значения, следовательно, количество переменных не позволяет оценить ресурсы, необходимые для нахождения результата. Для практического использования алгоритма требуется определить количество бит, требуемых для нахождения результата, а также количество операций с битами. Ответ на данный вопрос дают теоремы изложенные ниже.

Лемма 1. *Пусть A = (a_{ij}) – невырожденная целочисленная матрица n×n,* $m = \max_{i,j=1,2,..,n} L(a_{ij})$. *Тогда*

$$L(\det A) \le n(\log_2 n + m).$$

Лемма 2. *Пусть A = (a_{ij}) – невырожденная n×n рациональна матрица,* $m = \max_{i,j=1,2,..,n} L(a_{ij})$. *Тогда*

$$L(\det A) \le n(\log_2 n + (2n+1)m).$$

Теорема 1. *Пусть – система линейных алгебраических уравнений, A – невырожденная матрица n×n с рациональными элементами, b – n-мерный вектор с рациональными элементами,*

$$m = \max\left\{ \max_{i,j=1,2,..,n} L(a_{ij}), \max_{i=1,2,..,n} L(b_i) \right\},$$

тогда

$$L(x) = \sum_{i=1}^{n} L(x_i) \le 2n^2(\log_2 n + (2n+1)m).$$

Теорема 2. *Пусть даны невырожденная n × n система уравнений* $Ax = b$ *с рациональными коэффициентами, являющаяся приближением некоторой n×n системы уравнений* $\tilde{A}\tilde{x} = \tilde{b}$ *; матрицы абсолютных погрешностей:*

$$\Delta_A = (\delta_{ij}) : (\forall i, j = 1,2,...,n)\, \delta_{ij} \le \left| a_{ij} - \tilde{a}_{ij} \right|,$$

$$\Delta_b = (\delta_j) : (\forall j = 1,2,...,n)\, \delta_j \le \left| b_j - \tilde{b}_j \right|;$$

а также верхняя оценка числа бит, требуемых для одного элемента исходных данных

$$m = \max\left\{ \max_{i,j=1,2,..,n} \{L(a_{ij}), L(\delta_{ij})\}, \max_{i=1,2,..,n} \{L(b_i), L(\delta_i)\} \right\}.$$

Тогда, для нахождения как решения x, так и гарантированной нормы погрешности $\|\Delta_x\| = \|x - \tilde{x}\|$ *потребуется не более $O(n^4 m)$ бит памяти и не более $O(n^7 m^2)$ битовых операций.*

Пусть l – число бит, требуемых для представления исходных данных. При ограниченном m будем иметь $l = \Theta(n^2 m)$. Из теоремы [2] следует, что в этом случае зависимость битовой пространственной сложности от величины l не превосходит величины $O(l^2)$, а зависимость битовой вычислительной сложности (при использовании быстрых алгоритмов умножения) от величины l не превосходит $O(l^{2,5})$.

Численные методы решения краевых задач для линейных дифференциальных и интегральных уравнений сводятся к решению систем линейных алгебраических уравнений с (2*к*+1)-диагональной матрицей. Для решения таких систем используется метод прогонки. С одной стороны, данный метод имеет низкую алгебраическую вычислительную сложность (т. е. количество операций с числами) равна $O(n)$, где n – порядок системы. С другой стороны он не позволяет решать плохо обусловленные системы. Для устранения последнего можно использовать безошибочные вычисления. Однако, при безошибочных вычислениях, как и в случае алгоритма *Жордана–Гаусса*, адекватной оценкой сложности будет не алгебраическая, а битовая вычислительная сложность (т. е. количество операций с битами).

Теорема 3. *Пусть даны невырожденная система уравнений* $Ax = b$, *A – трехдиагональная* $n \times n$ *матрица с рациональными коэффициентами,* b *–* $n \times k$ *матрица свободных членов, пусть также дана верхняя оценка числа бит, требуемых для одного элемента исходных данных*

$$m = \max\left[\max_{i,j=1,..n}\{L(a_{ij}), L(\delta_{ij})\}, \max_{i=1,..n}\{L(b_i), L(\delta_i)\}\right].$$

Тогда, для нахождения решения потребуется не более $O(kmn^3)$ *бит памяти и не более* $O(kmn^3)$ *операций с ними.*

При решении практических задач бывают случаи, когда система уравнений бывает недоопределенной или переопределенной. При решении плохо обусловленных систем в этом случае может помочь нормальное псевдорешение. Для его нахождения будем использовать обобщенную обратную матрицу, называемую также g-обратной.

Существует несколько видов g-обратных матриц. Наиболее приемлемой матрицей для нашей является матрица *Мура–Пенроуза,* так как с ее помощью можно получить нормальное псевдорешение задачи.

Известно множество методов нахождения g-обратной матрицы *Мура–Пенроуза*. Теоретический расчет битовой сложности алгоритмов и практические эксперименты показали, что наиболее оптимальным методом для вычисления обобщенной обратной матрицы является метод Эрмита. Для сохранения общности изложения приведем описание алгоритма.

Теорема 4. *Пусть* $L(A_{m\times n})$ *длина исходной матрицы. Тогда, для нахождения обобщенной обратной матрицы Мура–Пенруоза алгоритмом Эрмита потребуется не более*$O(l^{3,5})$ *бит памяти и не более*$O(l^5)$ *операций с ними.*

Также во второй главе были проведены практические эксперименты. Эксперимент состоял в решении системы $Hx = He$, где

$$H = \left\{\frac{1}{i + j - 1}\right\}_{i,j=1,2,\ldots,n}, \quad e = \{1\}_{i=1,2,\ldots,n}.$$

Очевидно, что в этом случае x должен быть единичным вектором. Результаты, полученные при n=15, с использованием популярных коммерческих программ *MS Excel* и *MathCAD*, приведены на рис. 1..

Легко заметить, что полученные решения не верны, а программы даже не выдают сообщения, что решение может быть неправильным

Для апробации разработанного программного обеспечения и анализа практической достижимости полученных оценок вычислительной сложности с использованием безошибочных вычислений был проведен второй вычислительный эксперимент, он состоял в решении систем линейных алгебраических уравнений

$$Hx = e, \quad H = \left\{\frac{1}{i + j - 1}\right\}_{i,j=1}^{n}, \quad e = \{e_i = 1\}_{i=1}^{n}.$$

	MS Excel (*n*=15)		MathCAD (*n*=15)
	1.95		0.99
	72.01		1.04
	-747.44		0.96
	2648.23		0.59
	-4625.26		1.47
	4207.05		4.28
	-1710.44		-3.74
x=	524.03	*x*=	-14.85
	-996.25		56.99
	974.50		-78.42
	-439.50		69.18
	106.75		-40.54
	-12.63		20.19
	0.84		-5.05
	1.16		1.91

Рис. 1. Результаты эксперимента использования программ *MS Excel* и *MathCAD*

Матрица *H*, известная как матрица Гильберта, является плохообусловленной. Подобные системы, с использованием приближенных вычислений, не удается решить уже при $n \geq 5$. Были решены все системы линейных уравнений, для $n = 3, \ldots, 250$. Расчеты осуществлялись по методу *Жордана–Гаусса*, методу *Гаусса* и методу прогонки. При решении системы методом прогонки также использовалась матрица *Гильберта* с ненулевыми элементами только на трех главных диагоналях. Результаты вычислительных экспериментов приведены рис. 2 и рис. 3.

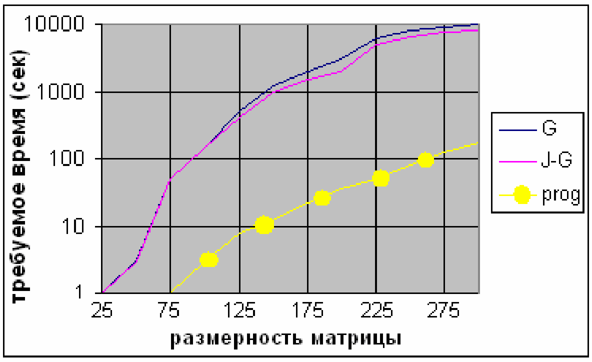

Рис. 2. Время (секунды) решения системы методами *Гаусса*, *Жордана–Гаусса* и *методом прогонки*

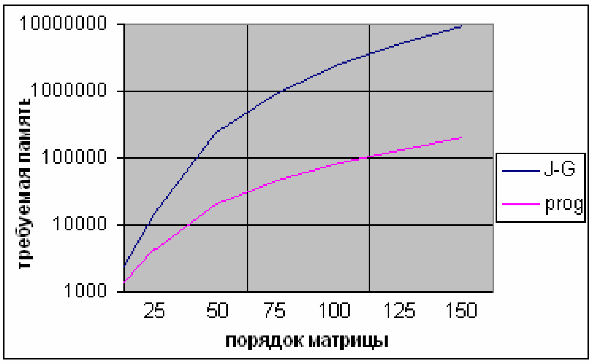

Рис. 3. Память (число слов памяти), требуемая для решения системы

Проведенные практические эксперименты являются подтверждением теоретических исследований, а также показывают не только возможность, но и необходимость безошибочного решения систем линейных алгебраических уравнений. Необходимость вызвана тем, что при приближенном решении конечный результат будет существенно отличаться от правильного, но это можно избежать, если выполнять все операции точно. Многие реальные задачи имеют большую размерность *n*, следовательно, вычислительная и пространственная сложность этих задач будет достаточно большой, а следовательно и время, требуемое для решения будет большим. Увеличить скорость решения во многих случаях позволяет использование параллельных вычислений на нескольких компьютерах одновременно, при этом сократятся как количество операций, выполняемых на одном компьютере, так и память, требуемая для хранения промежуточных результатов.

В **третьей главе** предлагается адаптация разработанных классов ***overlong***, ***rational*** и ***matrix*** к параллельным вычислениям. При организации параллельных вычислений было принято использовать *MPICH*, которая поддерживает стандарт ***MPI*** и имеет *GNU* лицензию.

Передача класса ***rational*** между узлами внутренними средствами MPI невозможна, так как данный класс содержит два объекта класса ***overlong***, которые в свою очередь содержат длину массива и указатель на массив содержащий число. Поэтому объявить структуру ***rational*** по стандарту ***MPI*** не возможно по двум причинам:

- необходимо хранить указатель;
- каждый объект ***rational*** может иметь произвольную длину, которая может измениться при следующей математической операции – невозможно создать универсальную структуру.

Из вышесказанного следует, что стандартными средствами передачу типа ***rational*** осуществить не удастся. Очевидными решениями будет упаковка ***rational*** в буфер с целью дальнейшей передачи. Данный вариант позволяет обойтись одной транзакцией, и поэтому взят за основу модификации. Для передачи типов ***overlong***, ***rational*** были переопределены стандартные методы передачи данных в среде ***MPI***.

С целью целостности изложения, рассмотрим предварительно используемый для распараллеливания последовательный алгоритм. Фрагменты листинга его программной реализации приведены на рис. 4. Система хранится в матрице *m*, включая столбец свободных членов.

```
#define N 500
#include <stdio.h>
#include <iostream>
#include "rational.h"
using namespace std;

rational m[N][N+1];
int rowN[N];

void SubLine(int i, int j) {
    rational R,z;
    R=m[rowN[i]][j];
    for(int k=j; k < N+1; k++) {
        z=R*m[rowN[j]][k];
        m[rowN[i]][k]-=z;
    }
}
void DivLine(int i) {
    rational R;
    R=m[rowN[i]][i];
    for(int l=i ;l<N+1; l++)
    m[rowN[i]][l]/=R;
}

void Swap(int i, int j){
    k=rowN[i]; rowN[i]= rowN[j];
    rowN[j]=k;
}
```

```
int Solve() {
    rational zero;
    int i,j,k;
    // Инициализация rowN
    for (i=0; i<N; i++) rowN[i]=i;
    for (i=0; i<N; i++){
        j=i;
        // Поиск   ведущей строки
        while((m[rowN[j]][i] ==zero)
              &&(j<N)) j++;
        if  (j==N)  break;
        // Определитель 0
        if (i!=j) Swap(i,j);

        // Ведущее преобразование
        DivLine(i);
        for (k=0; k<i; k++) SubLine(k, i);
        for (k=i+1; k<N; k++) SubLine(k, i);
        }
    return (j!=N);
}

int main(){
    ReadFile();
    if (Solve()) WriteResult();
    else cout << "Det=0\n";
    return 0;
}
```

Рис. 4. Алгоритм решения систем уравнений методом *Жордана–Гаусса*

Для строк матрицы вводятся состояния: *открытая*, *закрытая* и *ведущая*. Открытыми строками будем называть строки с номером *j*, у которых на *i*-м шаге выполнения алгоритма модифицированный номер *rowN*[*j*]≥*i*. Первый цикл в теле функции *Solve*() выполняет присваивания *rowN*[*j*]=*j*, что соответствует присваиванию всем строкам статуса открытая. Число выполнений тела второго цикла равно числу переменных. При *i*-м выполнении тела цикла из открытых строк по выбирается строка *j*, имеющая ненулевой элемент в столбце *i*. Найденной строке присваивается статус ведущая, при необходимости производится модификация значений *rowN*[*j*] и *rowN*[*i*]. Далее с помощью процедуры *DivLine*() производится нормировка ведущей строки, а с помощью процедуры *SubLine*() осуществляется ведущее преобразование других строк матрицы, состоящее в обнулении элементов *i*-го столбца, не принадлежащих ведущей строке. После этого ведущая строка переводится в состояние *закрытой*.

Из описания последовательного алгоритма следует, что следует распараллелить процесс инициализации, поиск ведущей строки и ведущее преобразование. Из этого следует, что наиболее предпочтительной является декомпозиция исходных данных по строкам, т.е. разрезание матрицы системы на горизонтальные полосы, содержащие примерно равное число строк. При этом, каждая полоса загружается в соответствующий компьютер: нулевая полоса – в нулевой компьютер, первая полоса – в первый компьютер, и т. д., последняя полоса - в последний компьютер.

В начале параллельной реализации производим инициализацию переменных и выделение памяти под обрабатываемый локальным процессом фрагмент матрицы системы, ведущую строку, массив модифицированных номеров, массив флагов открытых строк и буфер передачи данных.

Внешний цикл функции *Solve*() выполняется для каждой переменной системы уравнений. При *i*-м выполнении тела цикла в каждом процессе из открытых строк выбирается строка *j*, имеющая максимальный по абсолютной величине элемент в столбце *i* локального фрагмента матрицы системы. Затем определяется ведущий процесс, в котором находится максимальный по модулю элемент столбца *i*. Если модуль найденного элемента равен 0, то выполнение программы прекращается с выводом сообщения "*Det*=0". В противном случае строке *j* ведущего процесса присваивается статус «*ведущая*» и устанавливается модифицированное значение *rowN*[*j*] = *i*.

Далее в ведущем процессе производится нормировка ведущей строки и ее рассылку остальным процессам. Функция *SubLine*() осуществляется ведущее преобразование строк матрицы, состоящее в обнулении элементов *i*-го столбца, не принадлежащих ведущей строке. После этого ведущая строка переводится в состояние *закрытой*.

Если исходными данными является $n\times n$ матрица, m – количество бит, требуемых для представления одного элемента исходной матрицы, и задача решается на N компьютерах, то ускорение параллельной реализации алгоритма *Жордана–Гаусса* при использовании параллельный вычислений составит:

$$\frac{N}{1+\dfrac{CN}{m^2n^2}\left(\dfrac{\upsilon_{процессора}}{\upsilon_{передачи}}\right)}$$

Вычислительный эксперимент проводился на кластере кафедры ЭММиС ЮУрГУ. Данный кластер состоит из основного и восьми вспомогательных узлов, объединенных в локальную сеть посредством коммутатора Allied Telesyn AT-FS716E 100Base-TX.

Примерная расчетная пиковая производительность кластера в соответствии с данными, взятыми с сайта производителя, составляет величину

ПРППК=8*2,4*2+2*2,8*2 = 49,6 Gflops.

На узлах кластера присутствуют только средства необходимые для функционирования среды ***MPI***. На основном узле установлены оконные менеджеры *XFCE4* и *GNOME*, а также другие пакеты необходимые для написания, отладки и запуска программ.

Для апробации разработанного программного обеспечения и анализа практической достижимости полученных оценок вычислительной сложности был проведен вычислительный эксперимент. Эксперимент состоял в решении систем линейных алгебраических уравнений $Ax = b$, где $A = [1/(i + j + 1)]_{i,j=0}^{n}$ – матрица Гильберта, $b = [1]_{i=0}^{n}$. Результаты вычислительного эксперимента приведены на рис. 5.

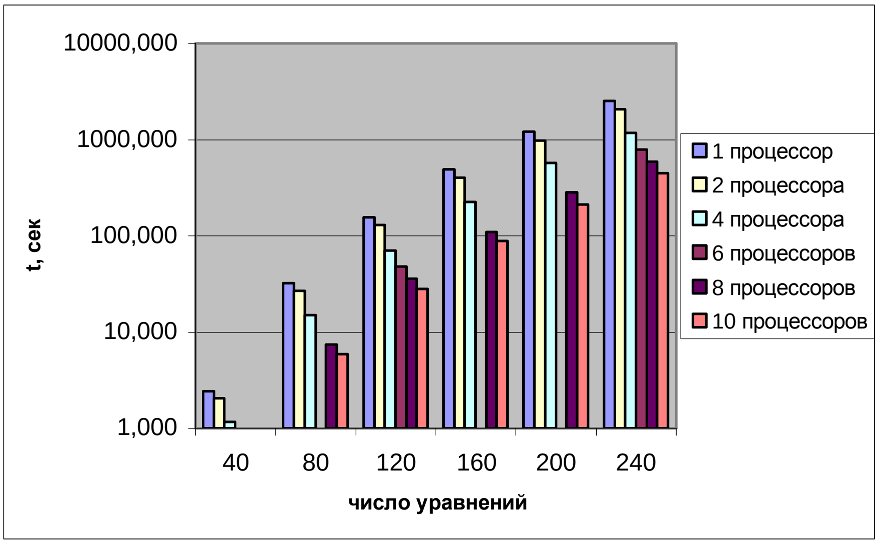

Рис. 5. Время решения систем уравнений на разных количествах процессоров

Известны реализации параллельных алгоритмов умножения матриц и метода *Жордана–Гаусса*. Используя разработанные классы ***overlong*** и ***rational*** в этих реализациях можно выполнять все операции точно, не задумываясь о длине передаваемых операндов. Метод Эрмита использует умножение матриц и диагональную редукцию, которая выполняется методом *Жордана–Гаусса*. Следовательно, не составит труда написать его программную реализацию. Для параллельной реализации метода *Жордана–Гаусса* в класс ***matrix*** были добавлены возможности параллельных вычислений. Таким образом при использовании классов ***overlong***, ***rational*** и ***matrix*** последовательные программы практически не будут отличатся параллельных.

Если исходными данными является $n_1 \times n_2$ матрица, m – количество бит, требуемых для представления одного элемента исходной матрицы, тогда коэффициент ускорения параллельной реализации метода Эрмита, выполняемого на N компьютерах не будет превышать ускорения на участке с наибольшей вычислительной сложностью:

$$\frac{N}{1+\frac{CN}{n_2^5 m}\left(\frac{\upsilon_{процессора}}{\upsilon_{передачи}}\right)}.$$

Проведенные вычислительные эксперименты подтвердил теоретически рассчитанные коэффициенты ускорения (при достаточно больших исходных данных скорость вычисления увеличится в N раз, где N – количество компьютеров, на которых решается задача). Результаты аналитического исследования и проведенного вычислительного эксперимента показывают, что использование параллельного программирования существенно уменьшает время, требуемое для безошибочного решения систем линейных уравнений.

В **заключении** подведены итоги проведенного исследования.

1. Создана библиотека ***ExactComp***, позволяющая использовать безошибочное выполнение основных арифметических операций над полем рациональных чисел и матричные вычисления использовать в программах пользователя. Данная библиотека состоит из классов ***overlong***, ***rational*** и ***matrix***. Диапазон чисел, представляемых объектами класса ***overlong*** расширен до $(-2^{1048575}, 2^{1048575})$, а минимальный шаг дискретизации чисел, представляемых объектами класса ***rational*** может достигать $(2^{-1048575})$. Помимо матричных операций в класс ***matrix*** добавлены возможности: решение систем уравнений с заданной матрицей, обращение матрицы, нахождение обобщенной обратной матрицы и использование параллельных вычислений.
2. Проведен анализ, который показал, что при безошибочном решении систем линейных алгебраических уравнений методом *Жордана–Гаусса* пространственная битовая сложность не превышает $O(l^2)$, а вычислительная битовая сложность при использовании алгоритма умножения столбиком и быстрого алгоритма не превышает $O(l^{3,5})$ и $O(l^{2,5})$ соответственно, где l – число бит требуемых для представления исходных данных. При безошибочном решении систем линейных алгебраических уравнений с трехдиагональной матрицей методом прогонки пространственная битовая сложность не превышает $O(l^{1,5})$, а вычислительная битовая сложность при использовании алгоритма умножения столбиком и быстрого алгоритма не превышает не превышает $O(l^3)$ и $O(l^{1,5})$ соответственно. При безошибочном нахождении обобщенной обратной матрицы *Мура–Пенроуза* методом Эрмита пространственная битовая сложность не превышает $O(l^4)$, а вычислительная битовая сложность при использовании алгоритма умножения столбиком и быстрого алгоритма не превышает $O(l^7)$. Результаты теоретического исследования показывают возможность практического использования программы, как при анализе, так и при безошибочном решении систем линейных алгебраических уравнений. Проведены практические эксперименты, подтверждающие теоретические исследования, а также показывающие не только возможность, но и необходимость безошибочного решения систем линейных алгебраических уравнений.
3. Разработано параллельное программное обеспечение для безошибочного решения систем линейных алгебраических уравнений и нахождения псевдообратной матрицы. Проведен анализ ускорения использования параллельных вычислений для решения данных задач.

Публикации автора по теме диссертации

В журналах, рекомендованных ВАК РФ

1. Программное обеспечение безошибочных дробно-рациональных вычислений и его применение для решения линейных систем / М.И. Германенко // Вестник Нижегородского университета им. Н. И. Лобачевского. Серия: «Математика и информационные технологии». – 2009. – № 4. – С. 172–180.
2. Безошибочное решение систем линейных алгебраических уравнений / А.В. Панюков, М.И. Германенко // Вестник Южно-Уральского государственного университета. Серия: «Математика, физика, химия» – 2009. – Вып. 12. – №10 (143). – С. 28–35.

В других изданиях

3. Свидетельство РОСПАТЕНТ № 990607 «Класс rational» / А.В. Панюков, М.М. Силаев, М.И. Германенко. – 1999.
4. Распараллеливание алгоритмов решения систем линейных алгебраических уравнений с применением вычислений без округлений / А.В. Панюков, М.И. Германенко, В.В. Горбик // Параллельные вычислительные технологии (ПаВТ’2007): Труды международной научной конференции (г. Челябинск, 29 января – 2 февраля 2007 г.). – Том 2. – С. 238–249.
5. Свидетельство РОСПАТЕНТ № 2009612777. Библиотека классов «Exact Computational» / А.В. Панюков, М.И. Германенко, В.В. Горбик. – 2009.
6. Параллельные алгоритмы безошибочного вычисления матрицы Мура–Пенроуза / А.В. Панюков, М.И. Германенко // Параллельные вычислительные технологии (ПаВТ'2008): Труды международной научной конференции (Санкт-Петербург, 28 января – 1 февраля 2008 г.). – С. 215–223.
7. Оценка сложности безошибочного решения систем линейных алгебраических уравнений с использованием класса RATIONAL / А.В. Панюков, М.И. Германенко // Вычислительные технологии – 2000 – Новосибирск: Институт вычислительных технологий СО РАН. URL: http://www.nsc.ru/ws/ct-2000/125.
8. Сложность нахождения гарантированной оценки решения приближенно заданной системы линейных алгебраических уравнений / А.В. Панюков, М.И. Германенко // Известия Челябинского научного центра. – 2000. – 4(9). – С. 11–15. URL: http://www.sci.urc.ac.ru/news/2000_3.
9. Оценка сложности решения систем линейных алгебраических уравнений с использованием класса Rational / М.И. Германенко // Научные труды молодых исследователей программы «Шаг в будущее» - М.: НТА «АПФН». – 2000 г. – Том 3. – С. 90–98.
10. Нахождение гарантированных оценок решения приближенно заданной системы линейных алгебраических уравнений / М.И. Германенко // Научные труды молодых исследователей программы «Шаг в будущее» – М.: НТА «АПФН». – 2001 г. – Том 4. – С. 170–178.
11. Аналитическое и экспериментальное исследование сложности безошибочного решения систем линейных алгебраических уравнений / М.И. Германенко // Тезисы второго Международного конгресса студентов, молодых ученых и специалистов «Молодежь и наука – третье тысячелетие» / YSTM`02 (Москва, 15 – 19 апреля 2002 г.) – М.: НТА «АПФН». – 2002 г. – Часть 2. – С. 11–12.
12. Программное обеспечение безошибочных дробно-рациональных вычислений и его применение для решения линейных систем / М.И. Германенко // «Всерос-

сийский конкурс на лучшие научные работы студентов по естественным, техническим (проекты в области высоких технологий) и инновационным научно-образовательным проектам». Материалы итоговой конференции – М.: МИЭМ. – 2004 г. – С. 328–331.

13. Программное обеспечение безошибочных дробно-рациональных вычислений и его применение для решения линейных систем / М.И. Германенко // Всероссийский конкурс на лучшие научные работы по техническим наукам (проекты в области высоких технологий): Тезисы проектов – М.: МИЭМ. – 2004 г. – Том 2. – С. 451–456.
14. Программное обеспечение безошибочных дробно-рациональных вычислений и его применение для решения линейных систем / М.И. Германенко // Всероссийский конкурс среди учащейся молодежи высших учебных заведений Российской Федерации на лучшие научные работы студентов по естественным наукам: Тезисы научных работ. Саратов: Сарат. гос. техн. ун-т. – 2004 г. – С. 28–30.
15. Использование параллельных и распределенных вычислений для безошибочных дробно-рациональных вычислений / М.И. Германенко // Новосибирск: Институт вычислительных технологий СО РАН. – 2006. URL: http://www.nsc.ru/ws/show_abstract.dhtml?ru+152+10461.
16. Приложение для безошибочного нахождения обобщенной обратной матрицы методом Мура–Пенроуза и Безошибочное решение систем линейных алгебраических уравнений / М.И. Германенко // Информационные технологии моделирования и управления. – 2009. – №1 (53). – С. 78–87.
17. Программное обеспечение безошибочных дробно-рациональных вычислений и его применение для решения линейных систем / М.И. Германенко // Параллельные вычислительные технологии (ПаВТ'2009): Труды международной научной конференции (Нижний Новгород, 30 марта – 3 апреля 2009 г.). – С. 147–156.

Работа выполнена при поддержке
грант губернатора Челябинской области 2004 г.,
гранта РФФИ, №07-01-96035-р_урал_а.

Германенко Максим Игоревич

**КОМПЛЕКС ПРОГРАММ
ДЛЯ БЕЗОШИБОЧНЫХ ДРОБНО-РАЦИОНАЛЬНЫХ ВЫЧИСЛЕНИЙ
И ЕГО ПРИМЕНЕНИЕ ДЛЯ РЕШЕНИЯ СИСТЕМ
ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ**

05.13.18 – математическое моделирование,
численные методы и комплексы программ

Автореферат
диссертации на соискание ученой степени
кандидата физико-математических наук

Издательство Южно-Уральского государственного университета

Подписано в печать . Формат 60*84 1/16 Печать трафаретная. Усл. печ.л. 0,93. Уч.-изд. л. 1. Тираж 110 экз. Заказ 512/92.

Группа МЭНП Издательства 454080, г.Челябинск, пр. им. В.И.Ленина, 76.