

УДК 519.61

Р.Г. Куликов, А.А. ЗвягинПермский национальный исследовательский
политехнический университет, Пермь, Россия**ЭФФЕКТИВНОСТЬ ИСПОЛЬЗОВАНИЯ ИДЕОЛОГИИ
МАСТЕР-ПРОЦЕССА ПРИ ПАРАЛЛЕЛЬНОЙ
РЕАЛИЗАЦИИ ИТЕРАЦИОННЫХ ПРОЦЕДУР
РЕШЕНИЯ ЛИНЕЙНЫХ СИСТЕМ**

В работе рассматривается эффективность использования идеологии мастер-процесса при построении параллельной реализации процедуры решения систем линейных алгебраических уравнений. Исследования проведены применительно к итерационным алгоритмам методов сопряженных градиентов и Якоби. Использовался разреженный формат RR(C)U хранения матрицы коэффициентов алгебраической системы. Исследования проводились применительно к системе линейных алгебраических уравнений, сформированной в ходе решения двумерной краевой задачи линейной теории упругости методом конечных элементов. В качестве критерия количественного анализа использовалось «ускорение», равное отношению времени выполнения последовательного алгоритма ко времени выполнения параллельного алгоритма. Анализ проведен для алгебраических систем от 100 до 50000 уравнений с использованием ЭВМ на базе шестиядерного процессора AMD® Phenom II X6 1075T. Программная реализация итерационных алгоритмов решения системы линейных алгебраических уравнений была произведена на языке программирования C#. Для обеспечения взаимодействия параллельных процессов использовался стандарт MPI 2.0. Полученные данные позволяют сделать вывод, что использование мастер-процесса при параллельной реализации метода сопряженных градиентов приводит к незначительному, от 10 до 15 процентов, снижению величины коэффициента ускорения по отношению к варианту реализации, не использующему мастер-процесс. В случае использования метода Якоби эффект выражен намного слабее. Принимая во внимание структурное удобство использования идеологии мастер-процесса, можно говорить о допустимости применения данного архитектурного решения при реализации рассмотренных методов.

Ключевые слова: параллельные вычисления, итерационные методы, системы линейных алгебраических уравнений.

R.G. Kulikov, A.A. Zviagin

Perm National Research Polytechnic University, Perm, Russian Federation

**EFFICIENCY OF MASTER-PROCESS IDEOLOGY
FOR PARALLEL REALIZATION OF ITERATIVE SOLVING
PROCEDURES FOR LINEAR ALGEBRAIC SYSTEM**

The effectiveness of master-process ideology application for realization of parallel solving procedures for system of linear algebraic equations is considered. Researches were conducted for iterative algorithms of conjugate gradient and Jacobi methods. The sparse format RR(C)U for coefficient matrix was used. Researches were conducted using system of linear algebraic equations computed with method of finite elements for plane boundary elastic problem. "Acceleration" calculated as ratio of successive algorithm execution time to parallel algorithm execution time was used as a criterion of quantity analysis. Analysis was conducted for algebraic systems from 100 to 50000 equations using six core AMD® Phenom II X6 1075T processor based computer. Program realization of iteration algorithms was created using C#. Standard MPI 2.0 was used to provide parallel processes communication. On base of obtained results it is possible to make a conclusion that usage of master-process parallel architecture for conjugate gradient iterative algorithm results in insignificant, from ten to fifteen percents, reduce of acceleration value as regards realization without master-process. This effect is much more lower for Jacobi method. Taking into account structural convenience of master-process parallel architecture the conclusion of possibility of this program architecture solution for considered methods was made.

Keywords: parallel computing, iterative methods, system of linear algebraic equations.

Введение

Развитие аппаратных мощностей вычислительных систем и численных методик решения сложных задач обусловили широкое применение ЭВМ в различных областях науки и техники. Зачастую для получения результатов решения с высокой точностью программные реализации алгоритмов решения задачи предъявляют повышенные требования к вычислительным ресурсам, соответственно, время получения решения оказывается неприемлемо велико. Применение многопроцессорных систем и параллельных вычислительных технологий существенно расширяет возможности исследователей, занимающихся численными экспериментами.

Достаточно часто при численном анализе краевых задач возникает необходимость решения системы линейных алгебраических уравнений (СЛАУ) большой размерности, которая может быть представлена в виде

$$[A]\{x\} = \{b\}, \quad (1)$$

где $[A]$ – матрица коэффициентов системы; $\{b\}$ – вектор правой части; $\{x\}$ – вектор неизвестных.

При получении решения системы линейных алгебраических уравнений наиболее эффективно работают алгоритмы, построенные по принципу параллельной обработки данных в рамках различных процессов, когда длительные математические операции над матрицами и векторами производятся одновременно несколькими независимыми процессами. Данный способ позволяет эффективно использовать в программе как многопроцессорные (кластерные) системы, так и современные многоядерные процессоры.

Эффективность применения параллельных вычислений подтверждена большим числом работ на данную тему. В работе [1] рассматривается параллельная реализация метода квадратного корня при нахождении решения системы алгебраических уравнений, полученной в ходе реализации метода граничных элементов. Работы [2] и [3] посвящены аспектам построения параллельной реализации итерационных методов решения систем линейных алгебраических уравнений. Как отмечается в [4], построение эффективного параллельного алгоритма представляет собой сложную задачу и автор прибегает к готовым библиотекам параллельных алгоритмов. Основным минусом данного подхода является отсутствие учета особенностей исследуемой задачи. Как правило, все реализации предполагают сборку матрицы коэффициентов системы в рамках одного процесса с последующей рассылкой ее блоков по рабочим процессам системы. При этом разделение производится построено.

На сегодняшний день при решении СЛАУ большой размерности большее распространение получили итерационные методы. К их преимуществам следует отнести существенно меньшее накопление погрешности при получении решения, высокую эффективность выполнения матричных операций при использовании разреженных форматов хранения матрицы коэффициентов, возможность более быстрого получения начальных приближений искомого решения, наличие эффективных способов организации параллельных вычислений. Дополнительно следует упомянуть неизменность структуры матрицы коэффициентов, что существенно при использовании разреженных форматов хранения.

В настоящее время используются два подхода в организации архитектуры параллельных вычислительных программных средств: с использованием и без использования мастер-процесса, функции которого заключаются в координировании работы остальных вычислительных процессов системы. Целью данного исследования является анализ эффективности применения идеологии мастер-процесса при построении параллельных алгоритмов методов сопряженных градиентов и Якоби.

В данной работе рассматриваются две архитектуры параллельных вычислительных систем. Первый подход подразумевает наличие мастер-процесса, совершающего все векторные операции, при этом параллельный алгоритм строится только для нахождения произведения матрицы коэффициентов на вектор. Основное преимущество данного подхода заключается в минимизации количества операций передачи данных между параллельными потоками вычислений. Операции передачи данных производятся значительно медленнее операций вычисления, таким образом, данный подход позволит существенно сократить издержки синхронизации процессов.

Второй подход подразумевает отсутствие мастер-процесса, все математические операции на каждой итерации методов решения СЛАУ производятся децентрализованно. Повышение количества передаваемых данных компенсируется снижением количества последовательных вычислений в рамках каждого процесса.

1. Применение параллельных вычислений при выполнении основных операций

Для построения параллельного алгоритма умножения матрицы коэффициентов на вектор предлагается разделить матрицу коэффициентов по строкам на p блоков, где p соответствует числу параллельных вычислительных процессов. Вектор, на который производится умножение, передается всем процессам. Для обеспечения равномерности нагрузки на параллельно выполняющиеся ветви алгоритма рекомендуется определить равное число строк для каждого блока. В результате произведения каждый процесс определяет свой блок элементов результирующего вектора (рис. 1).

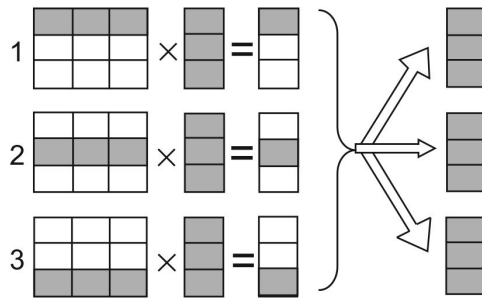


Рис. 1. Схема параллельного умножения матрицы на вектор

Для реализации математических операций над векторами предлагается разделение векторов на блоки с закреплением каждого блока за конкретным процессом. Таким образом, операции сложения и вычитания векторов будут производиться одновременно над несколькими блоками; результат данных операций будет сразу представлен в виде, предназначенном для параллельного алгоритма (рис. 2, *а*). Аналогично предлагается строить операцию умножения вектора на скаляр (рис. 2, *б*). Для получения результатов операции перемножения векторов потребуется также провести коллективную операцию передачи данных с суммированием (рис. 3). После проведения данной операции каждый процесс получает свою копию вектора результата.

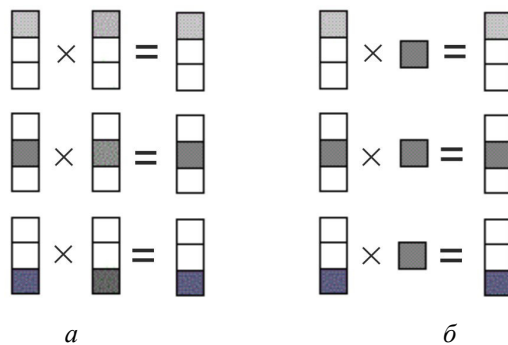


Рис. 2. Скалярные операции над векторами: *а* – сложение векторов; *б* – умножение вектора на скаляр

В случае применения первого подхода требуется выполнение процедуры сборки данных и получение полного результирующего вектора в рамках мастер-процесса. В случае применения второй методики объединение данных не требуется.

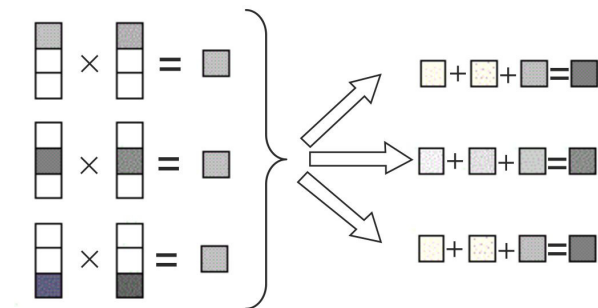


Рис. 3. Схема параллельного алгоритма произведения векторов

Численные эксперименты проводились применительно к системе линейных алгебраических уравнений, полученной при реализации метода конечных элементов для двумерной задачи. В силу разреженности матрицы коэффициентов системы использовалась строчная схема хранения RR(C)U [5], применение которой позволяет существенно повысить эффективность операции произведения матрицы на вектор.

Реализация параллельного алгоритма была построена на основе интерфейса передачи сообщений между процессами MPI (Message Passing Interface) [6]. Данный инструмент позволяет использовать программный продукт на различных аппаратных архитектурах, не требуя дополнительного изменения алгоритма и его реализации.

Численный эксперимент был поставлен на ЭВМ класса «персональный компьютер», работающей под управлением шестиядерного процессора AMD® Phenom II X6 1075T. Программная реализация итерационных алгоритмов решения системы линейных алгебраических уравнений была произведена на языке программирования C#. Для обеспечения взаимодействия параллельных процессов использовался стандарт MPI 2.0.

В качестве критерия количественного анализа в соответствии с [7] использовалось «ускорение», равное отношению времени выполнения последовательного алгоритма ко времени выполнения параллельного алгоритма.

2. Применение метода сопряженных градиентов

Метод сопряженных градиентов – итерационный метод для безусловной оптимизации в многомерном пространстве [8]. Основное достоинство метода заключается в достижении точного решения при

обработке систем с положительно определенной матрицей коэффициентов за конечное число итераций.

Схема данного метода (рис. 4) подразумевает выполнение на каждой итерации одной операции произведения матрицы на вектор и девяти скалярных операций над векторами.

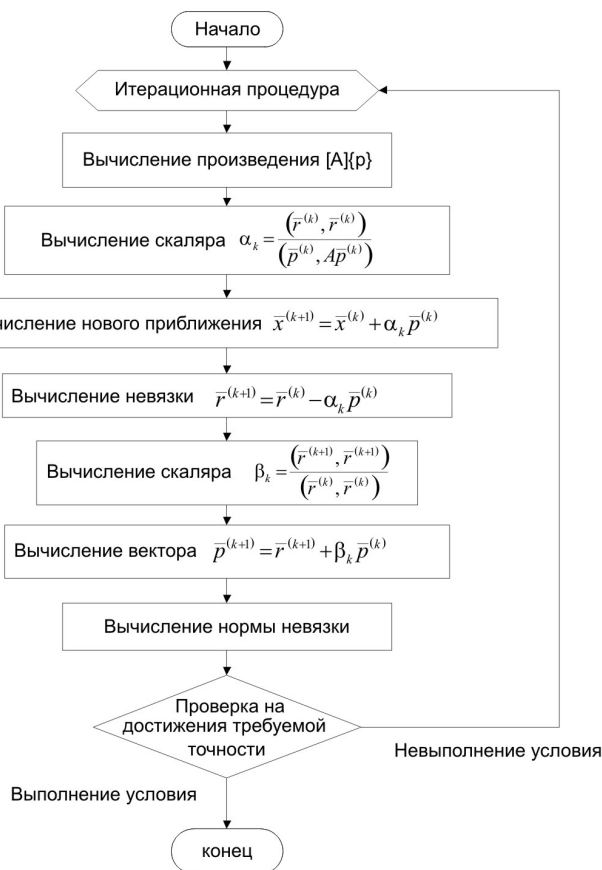


Рис. 4. Схема метода сопряженных градиентов

На рис. 5, 6 представлены зависимости ускорения от числа параллельных процессов при решении систем различной размерности. Рассматриваются варианты построения параллельного алгоритма без главного процесса и с главным процессом соответственно. Увеличение длительности работы параллельных алгоритмов для СЛАУ малой размерности объясняется тем, что трудоемкость обмена данными между процессорами больше, чем вычислительная трудоемкость частей задачи, выполняемых в рамках разных процессов. Построение параллель-

ного алгоритма, производящего математические операции над векторами, позволяет получить более высокие коэффициенты ускорения.

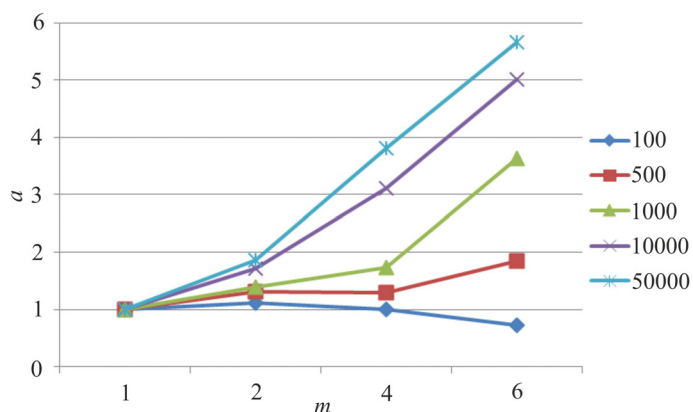


Рис. 5. Зависимость ускорения a параллельного алгоритма от числа процессоров m для сеток разной размерности в параллельной модели без мастер-процесса

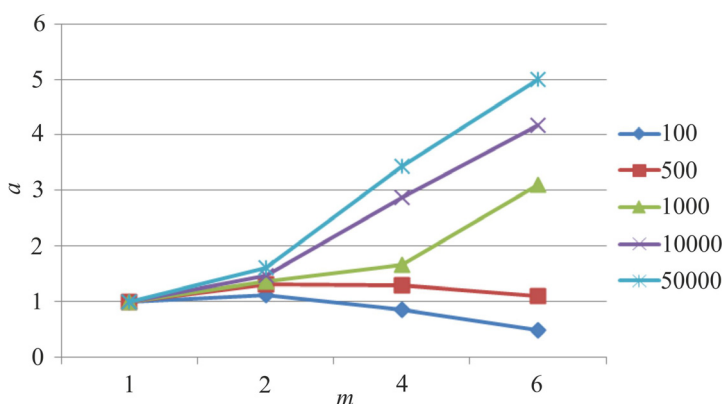


Рис. 6. Зависимость ускорения a параллельного алгоритма от числа процессоров m для сеток разной размерности в параллельной модели, содержащей мастер-процесс

3. Применение метода Якоби

Схема данного метода [9] (рис. 7) подразумевает выполнение на каждой итерации одной операции произведения матрицы на вектор и двух скалярных операций над векторами. Метод Якоби требует предварительного представления матрицы коэффициентов уравнения (1) в следующем виде:

$$[A] = [D] + [L] + [U],$$

где $[D]$ – диагональная матрица; $[U]$ и $[L]$ – верхняя и нижняя треугольные матрицы с нулевыми главными диагоналями. Итерационная схема метода имеет вид

$$\{x\}^{k+1} = [B]\{x\}^k + \{g\},$$

где $[B] = [D]^{-1}([L] + [U])$, $\{g\} = [D]^{-1}\{b\}$.

При анализе результатов время, необходимое на построение $[B]$ и $\{g\}$, не учитывалось.



Рис. 7. Схема метода Якоби

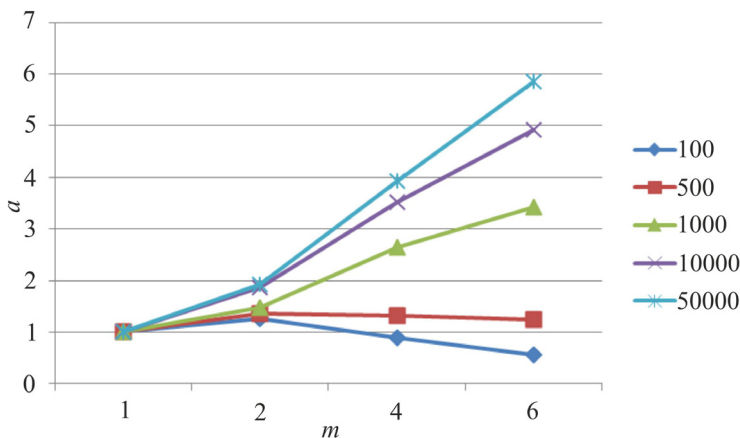


Рис. 8. Зависимость ускорения a параллельного алгоритма от числа процессоров m для сеток разной размерности в параллельной модели без мастер-процесса

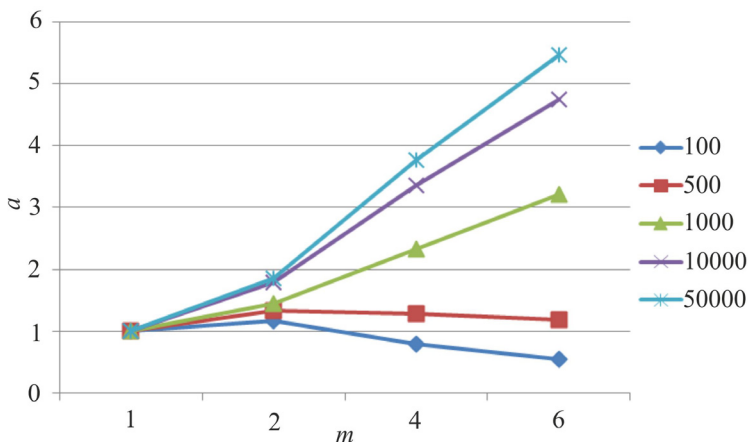


Рис. 9. Зависимость ускорения a параллельного алгоритма от числа процессоров m для сеток разной размерности в параллельной модели, содержащей мастер-процесс

На рис. 8 и 9 представлены зависимости ускорения от числа параллельных процессов для систем различной размерности с использованием и без использования мастер-процесса. Аналогично методу сопряженных градиентов наблюдается снижение коэффициента ускорения для задач малой размерности. Близкие результаты обеих схем распараллеливания объясняются малым числом векторных операций в методе Якоби.

Заключение

Полученные данные позволяют сделать вывод, что использование мастер-процесса при параллельной реализации метода сопряженных градиентов приводит к незначительному, от 10 до 15 процентов, снижению величины коэффициента ускорения по отношению к варианту реализации, не использующему мастер-процесс. В случае использования метода Якоби эффект выражен намного слабее. Принимая во внимание структурное удобство использования идеологии мастер-процесса, применение данного архитектурного решения при реализации рассмотренных методов предпочтительнее.

Библиографический список

1. Кузьмин А.О., Олейников А.И. Параллельная реализация метода граничного элемента по расчету тел с тонкими покрытиями на кластере рабочих станций // Информатика и системы управления. – Благовещенск, 2002. – № 1 (03). – С. 24–37.

2. Головашкин Д.Л., Горбунов О.Е. Параллельное решение СЛАУ методом Зейделя // Вестн. Самар. гос. техн. ун-та. Сер. Физ.-мат. науки. – Самара, 2004. – № 27. – С. 10–13.

3. Сухинов А.И., Чистяков А.Е. Параллельная реализация трехмерной модели гидродинамики мелководных водоемов на супервычислительной системе // Вычислительные методы и программирование: новые вычислительные технологии. – М., 2012. – Т. 13, № 1. – С. 290–297.

4. Губайдуллин Д.А., Садовников Р.В. Применение параллельных алгоритмов для решения задачи фильтрации жидкости в трещиноватопористом пласте к скважинам со сложной траекторией // Вычислительные методы и программирование: новые вычислительные технологии. – М., 2007. – Т. 8, № 1. – С. 244–251.

5. Pissanetzky S. Sparse Matrix Technology. – NY.: Academic Press, 1984. – 312 с.

6. Антонов А.С. Параллельное программирование с использованием технологии MPI. – М.: Московский университет, 2004. – 72 с.

7. Гергель В.П. Теория и практика параллельных вычислений. – М.: Бином, 2007. – 424 с.

8. Воеводин В.В. Численные методы алгебры (теория и алгоритмы). – М.: Наука, 1966. – 243 с.

9. Saad Y. Iterative methods for sparse linear system. Second edition. Society for Industrial and Applied Mathematics, 2003. – 528 с.

References

1. Kuz'min A.O., Olejnikov A.I. Parallel'naya realizaciya metoda granichnogo e'lementa po raschetu tel s tonkimi pokryt'yami na klasterne rabochix stancij [Parallel realization of boundary element method for bodies with thin covering calculations using cluster systems]. *Informatika i sistemy upravleniya*. Blagoveschensk, 2002, no. 1 (03), pp. 24-37.

2. Golovashkin D.L., Gorbunov O.E. Parallel'noe reshenie slau metodom Zejdelya [Parallel solution of SLAE using Zeidel method]. *Vestnik Samarskogo gosudarstvennogo texnicheskogo universiteta. Fiziko-matematicheskie nauki*, 2004. no. 27. pp. 10-13.

3. Suxinov A.I., Chistyakov A.E. Parallel'naya realizaciya trexmernoj modeli gidrodinamiki melkovodnyx vodoemov na supervychislitel'noj sisteme [Parallel realization of shallow reservoir hydrodynamics 3D-model using high performance computer system]. *Vychislitel'nye metody i programmirovaniye: novye vychislitel'nye texnologii*, 2012, vol. 13, no 1, pp. 290-297.

4. Gubajdullin D.A., Sadovnikov R.V. Primenenie parallel'nyx algoritmov dlya resheniya zadachi fil'tracii zhidkosti v treshhinovato-poristom plaste k skvazhinam so slozhnoj traektoriej [Parallel algorithms appliance to problems on liquid filtration through pore layer with chinks with complicated geometry]. *Vychislitel'nye metody i programmirovaniye: novye vychislitel'nye texnologii*. Moscow, 2007, vol. 7, no. 1, pp. 244-251.

5. Pissanetzky S. Sparse Martix Technology. NY.: Academic Press, 1984. 312 p.

6. Antonov A.S. Parallel'noe programmirovaniye s ispol'zovaniem texnologii MPI. Uchebnoe izdanie [Parallel programming using MPI technology]. Moscow: Moskovskij Universitet, 2004. 72 p.

7. Gergel' V.P. Teoriya i praktika parallel'nyx vychislenij [Theory and practices of parallel calculations]. Moscow: Binom, 2007. 424 p.

8. Voevodin V.V. Chislennyye metody algebrы (teoriya i algoritmy) [Numerical algebra methods (theory and algorithms)]. Moscow: Nauka, 1966. 243 p.

9. Saad Y. Iterative methods for sparse linear system. Second edition. Society for Industrial and Applied Mathematics, 2003. 528 p.

Об авторах

Куликов Роман Георгиевич (Пермь, Россия) – кандидат физико-математических наук, доцент кафедры вычислительной математики и механики Пермского национального исследовательского политехнического университета (614990, г. Пермь, Комсомольский пр., 29, e-mail: kulrtg@mail.ru).

Звягин Александр Андреевич (Пермь, Россия) – студент кафедры вычислительной математики и механики Пермского национального исследовательского политехнического университета (614990, г. Пермь, Комсомольский пр., 29, e-mail: azvyagin@hotmail.com).

About authors

Kulikov Roman Georgievich (Perm, Russian Federation) – Ph. D. in Physical and Mathematical Sciences, Ass. Professor, Department of Computational Mathematics and Mechanics, Perm National Research Polytechnic University (29, Komsomolsky av., 614990, Perm, Russian Federation, e-mail: kulrtg@mail.ru).

Zviagin Alexandr Andreevich (Perm, Russian Federation) – Student of Department of Computational Mathematics and Mechanics, Perm National Research Polytechnic University (29, Komsomolsky av., 614990, Perm, Russian Federation, e-mail: azvyagin@hotmail.com).

Получено 15.02.2013