

УДК 519.612

© *Н. С. Недождогин, С. П. Копысов, А. К. Новиков***ВАРИАНТ ПАРАЛЛЕЛЬНОГО РАЗЛОЖЕНИЯ
В ПРЕДОБУСЛАВЛИВАТЕЛЕ AISM¹**

Предлагается вариант параллельного разложения при формировании предобуславливателя, основанного на приближенном обращении Шермана–Моррисона. Проведены численные эксперименты по решению тестовых систем уравнений на графических ускорителях.

Ключевые слова: решение систем линейных алгебраических уравнений, предобуславливание, параллельные алгоритмы, графические ускорители вычислений.

Введение

Разработка и реализация новых подходов к решению больших систем линейных алгебраических уравнений предобусловленными итерационными методами являются достаточно трудоемкой задачей. Матрица предобуславливателя не только должна быть близка к обратной матрице коэффициентов системы уравнений, но и должна допускать эффективно распараллеливаемый алгоритм формирования. Наиболее используемыми на сегодняшний день методами, ориентированными на разреженные матрицы, можно считать метод неполного LU-разложения и метод неполного разложения Холецкого [1]. Имея высокую эффективность, данные методы сталкиваются с известными проблемами при их параллельной реализации.

Особенности архитектуры GPU (ставшей широко используемой в вычислениях) накладывают определенные ограничения на подходы к реализации параллельных алгоритмов формирования предобуславливателей. Исходя из этого, высоким потенциалом обладают предобуславливатели на основе аппроксимации обратной матрицы: полиномиальные (TNS [1] и др.), разреженные аппроксимации обратной матрицы (например, AINV), аппроксимации обратной матрицы в факторизованной форме (такие как FSAI, SPAI и др.) [2–4], а также методы, основанные на приближенном обращении матриц по формулам Шермана–Моррисона (AISM — Approximate Inverse Sherman–Morrison formula) [5].

Эффективное распараллеливание алгоритмов построения предобуславливателей связано с возможностью использования блочных вариантов формирования предобуславливателя, что дает сокращение перемещения данных по уровням иерархии памяти GPU, с операциями линейной алгебры, таких как матрично-векторные и матричные произведения, обладающих большим потенциалом распараллеливания.

В работе будет рассмотрен один из подходов к параллельному разложению при формировании обратного предобуславливателя AISM [6].

§ 1. Последовательный алгоритм

Рассмотрим оригинальный вариант алгоритма построения предобуславливателя (предложенного в [7]), так как именно он наиболее близок к предлагаемой в статье параллельной форме.

За основу построения предобуславливателя вида $P = A^{-1}$ возьмем матрицу B той же размерности, что и A , но с известной обратной матрицей.

Т е о р е м а 1.1 ([5]). *Пусть B — невырожденная матрица, векторы u и v такие, что*

$$r = 1 + v^T B^{-1} u, \quad r \neq 0.$$

Тогда матрица $A = B + uv^T$ является обратимой и ее обращение находится как

$$A^{-1} = B^{-1} - r^{-1} B^{-1} u v^T B^{-1}. \quad (1.1)$$

¹Работа поддержана РФФИ (гранты №№ 14-01-00055_а, 14-01-31066-мол_а).

Обозначим A_0 — невырожденную матрицу, обращение которой легко вычисляется, например диагональную или единичную. Тогда $A_k = A_0 + \sum_{i=1}^k u_i v_i^T$, где $k = 1, \dots, n$ и $A = A_n$. Если A_k , u_k , v_k удовлетворяют выражению (1.1), тогда обращение матрицы A может быть вычислено при n -кратном использовании (1.1):

$$A^{-1} = A_0^{-1} - \sum_{k=1}^n r_k^{-1} A_{k-1}^{-1} u_k v_k^T A_{k-1}^{-1}. \quad (1.2)$$

Представим (1.2) в матричной форме:

$$A_0^{-1} - A^{-1} = \Phi \Omega^{-1} \Psi^T, \quad (1.3)$$

где

$$\begin{aligned} \Phi &= [A_0^{-1} u_1, A_1^{-1} u_2, \dots, A_{n-1}^{-1} u_n], \\ \Psi &= [v_1^T A_0^{-1}, v_2^T A_1^{-1}, \dots, v_n^T A_{n-1}^{-1}], \\ \Omega^{-1} &= \text{diag} [r_1^{-1}, r_2^{-1}, \dots, r_n^{-1}]. \end{aligned}$$

Покажем, что факторизация (1.3) записывается без явного вычисления A_k^{-1} через векторы u_k , v_k как

$$s_k = u_k - \sum_{i=0}^{k-1} \frac{t_i^T A_0^{-1} u_k}{r_i} s_i, \quad t_k = v_k - \sum_{i=0}^{k-1} \frac{v_k^T A_0^{-1} s_i}{r_i} t_i, \quad k = 1, \dots, n. \quad (1.4)$$

Тогда выполняются соотношения

$$A_{k-1}^{-1} u_k = A_0^{-1} s_k, \quad u_k^T A_{k-1}^{-1} = t_k^T A_0^{-1}, \quad (1.5)$$

$$r_k = 1 + v_k^T A_0^{-1} s_k = 1 + t_k^T A_0^{-1} u_k. \quad (1.6)$$

С учетом (1.5) соотношение (1.3) запишем в виде

$$A_0^{-1} - A^{-1} = A_0^{-1} S \Omega^{-1} T^T A_0^{-1}, \quad (1.7)$$

где матрицы $S = [s_1, s_2, \dots, s_n]$, $T = [t_1, t_2, \dots, t_n]$, столбцы которых вычисляются по u_k, v_k .

Выбор A_0 , u_k , v_k определим следующим образом [7]:

$$A_0 = gI_n, \quad u_k = e_k, \quad v_k = (a^k - a_0^k)^T, \quad k = 1, \dots, n,$$

где I_n , e_k — единичная матрица и ее k -столбец; вектора a^k , a_0^k — k -ая строка матриц A , A_0 ; g — скаляр. Тогда аппроксимация обратной матрицы и соответствующий предобуславливатель примут вид

$$P_1 = A^{-1} = gI_n - g^{-2} U \Omega^{-1} V^T.$$

Рассмотрим еще один вариант разложения обрабатываемой матрицы $A = W - Z$, где W — обратимая матрица, $Z = UV^T = \sum_{k=1}^n u_k v_k^T$, а v_k, u_k такие, что $d_k = 1 - v_k^T W_{k-1}^{-1} u_k \neq 0$, где

$$W_k = W_0 - \sum_{i=1}^k u_i v_i^T.$$

Задавая выбор матриц $W = \beta \cdot \text{diag}(A)$, $\beta > 0$, $U = I$, $V = Z^T$ и следуя соотношениям (1.4), (1.5) и (1.6), получим выражения для вычисления столбцов матриц S и T в виде

$$s_k = u_k - \sum_{i=1}^{k-1} \frac{t_i^T W^{-1} u_k}{d_i} s_i, \quad t_k = v_k - \sum_{i=1}^{k-1} \frac{v_k^T W^{-1} s_i}{d_i} t_i$$

и обратной матрицы в виде

$$A^{-1} = W^{-1} - W^{-1} S D^{-1} T^T W^{-1}. \quad (1.8)$$

Алгоритм 1: Формирование предобуславливателя AISM

$$A = W - Z; W = \beta \cdot \text{diag}(A); Z = W - A; U = I; V = Z^T;$$

```
1 for  $k = 1$  to  $n$  do
2    $s_k = u_k$  ;
3    $t_k = v_k$  ;
4   for  $i = 1$  to  $k - 1$  do
5      $\delta = (t_i^T W^{-1}, u_k)$  ;
6     if  $|\frac{\delta}{d_i}| > \tau_S$  then
7        $s_k = u_k - \frac{\delta}{d_i} \cdot s_i$ 
8      $\delta = (v_k^T W^{-1}, s_i)$  ;
9     if  $|\frac{\delta}{d_i}| > \tau_T$  then
10       $t_k = v_k - \frac{\delta}{d_i} \cdot t_i$ 
11     for  $j = 1$  to  $n$  do
12       if  $|(s_k)_j| < \tau_S$  then
13          $(s_k)_j = 0$ 
14       if  $|(t_k)_j| < \tau_T$  then
15          $(t_k)_j = 0$ 
16    $d_k = 1 - (t_k^T W^{-1}, u_k)$  ;
17  $\tilde{S} = \{s_1, s_2, \dots, s_n\}, \tilde{T} = \{t_1, t_2, \dots, t_n\}$  ;
18  $D = \{d_1, d_2, \dots, d_n\}$  ;
19  $P_2 = W^{-1} - W^{-1} \tilde{S} D^{-1} \tilde{T}^T W^{-1}$  ;
```

Используя стратегию фильтрации по значениям элементов (оставляем элементы, значения которых больше некоторой величины τ) при вычислении матриц S , T и основываясь на (1.8), выпишем предобуславливатель, аппроксимирующий обратную матрицу в виде

$$P_2 = W^{-1} - W^{-1} \tilde{S} D^{-1} \tilde{T}^T W^{-1}. \quad (1.9)$$

Последовательный процесс формирования рассматриваемого предобуславливателя представлен в виде **алгоритма 1**.

Основные затраты последовательного алгоритма составляет вложенный цикл (строка 4 алгоритма 1), в котором определяются матрицы разложения \tilde{S} , \tilde{T} . Отметим, что матрицы формируются последовательно так, что существует зависимость по данным.

§ 2. Параллельный алгоритм

Первоначальным вариантом распараллеливания данного алгоритма является вычисление скалярных произведений на GPU. Такой подход показал, что достижение эффективного ускорения существенно ограничено доступом к памяти, и не дал существенного увеличения производительности.

Рассмотрим параллельный **алгоритм 2** формирования предобуславливателя P_2 (1.9) на GPU, в котором скалярные произведения в строках 5 и 8 **алгоритма 1** заменены матрично-векторными произведениями.

Для этого введены матрицы $S_k = \{s_1, \dots, s_{k-1}\}$ и $T_k = \{t_1, \dots, t_{k-1}\}$, состоящие из столбцов, вычисленных на шаге $k - 1$. В этом случае $k - 1$ скалярное произведение выполняется в виде матрично-векторных произведений (строка 4 **алгоритма 2**), результаты которых обозначены x_{d_1} и x_{d_2} , а через $(x_{d_1})_i$, $(x_{d_2})_i$ — соответствующие компоненты векторов. Также стоит отметить, что в памяти GPU₁ и GPU₂ хранятся локальные копии векторов x_{d_1} и x_{d_2} , в то время как x_{h_1} и x_{h_2} — глобальные векторы в оперативной памяти CPU.

Алгоритм 2: Формирование предобуславливателя AISM на двух GPU

$$A = W - Z; W = \beta \cdot \text{diag}(A); U = I; V = Z^T$$

```
1 for k = 1 to n do
  // GPU1
2   tk = vk;
3   y1 = W-1uk;
4   xd1 = Tky1, Tk = (t1, ..., tk-1);
5   xd1 → xh1 // Пересылка GPU1 → CPU
6   xh2 → xd2 // Пересылка CPU → GPU1
7   for i = 1 to k - 1 do
8     if |(xd2)i / di| > τS then
9       tk = tk - (xd2)i / di · ti;
10  for j = 1 to n do
11    if |(tk)j| < τS then
12      (tk)j = 0;
13  dk = 1 - (tkTW-1, uk);
14   $\tilde{S} = \{s_1, s_2, \dots, s_n\}, \tilde{T} = \{t_1, t_2, \dots, t_n\}; D = \{d_1, d_2, \dots, d_n\};$ 
15  P2 = W-1 - W-1 $\tilde{S}D^{-1}\tilde{T}^T$ W-1
  // GPU2
   sk = uk;
   y2 = W-1vk;
   xd2 = Sky2, Sk = (s1, ..., sk-1);
   xd2 → xh2 // Пересылка GPU2 → CPU
   xh1 → xd1 // Пересылка CPU → GPU2
   for i = 1 to k - 1 do
     if |(xd1)i / di| > τT then
       sk = sk - (xd1)i / di · si;
   for j = 1 to n do
     if |(sk)j| < τT then
       (sk)j = 0;
```

В рамках одной итерации цикла **алгоритма 2** при вычислении столбцов матриц s_k и t_k не возникает ситуации блокировки памяти, что позволило выполнить операции матрично-векторного и скалярных произведений (строки 4, 15) независимо в параллельных нитях. Такой подход применяется при вычислениях на нескольких GPU. В этом случае каждая последующая итерация цикла зависит от данных, полученных на предыдущем шаге, и требуется выполнение обмена векторами s_k и t_k между памятью различных GPU (строки 5 и 6).

Шаги алгоритма, содержащие матричные операции (строка 15), были распараллелены в рамках технологии CUDA с разработкой ядра, в котором произведение матриц вычисляется в виде последовательности матрично-векторных произведений. Эффективность распараллеливания данного этапа очень высокая, и затраты существенно меньше, чем на предыдущих шагах алгоритма.

При построении предобуславливателя P_{AISM} разреженность матриц неизвестна. Промежуточные вычисления на этапе формирования выполнялись над векторами s_k, t_k , являющимися столбцами матриц \tilde{S}, \tilde{T} , хранящихся по строкам. Расходы по памяти увеличивались, но сокращалось время обращения к элементам векторов. Для преобразования из сжатого формата хранения матриц CSR в формат хранения полных строк (этап построения предобуславливателя) и обратное преобразование (матрично-векторное произведение при решении СЛАУ) были разработаны эффективные параллельные алгоритмы, позволяющие пренебречь затратами на преобразование матриц [6].

§ 3. Численные эксперименты

В численных экспериментах были использованы матрицы из коллекции The University of Florida Sparse Matrix Collection. Решались системы уравнений $Ay = f$ с известным точным решением $y = [1, 1, \dots, 1]$, матрицы которых хранились в сжатом строчном формате (CSR). В качестве начального приближения выбиралось $y_0 = [0, 0, \dots, 0]$, а критерий сходимости — $\|r_i\| \leq 10^{-6}\|r_0\|$, где $r_i = f - Ay_i$.

В расчетах рассматривался один из вариантов представления матрицы $W = \beta \cdot \text{diag}(A)$, хотя возможны и другие, например $W = \beta I$. Выбор стратегии фильтрации по значениям элементов и предельных значений τ основывался на рекомендациях, приведенных в работе [7]. Отметим, что затраты на формирование возрастают несущественно при уменьшении порогового значе-

ния. Выбор оптимального значения параметра τ для рассматриваемых матриц заключался в минимизации числа итераций и времени решения. Точность фильтрации для матриц \tilde{S} и \tilde{T} выбиралась следующая: $\tau_u = 0.0001$, $\beta = 100$.

Сравним сначала результаты, полученные при решении несимметричных систем. В таблицах приведены затраты на построение предобуславливателей неполного разложения с контролем заполнения ILU(p) (рассматривались варианты $p = 0$ и $p = 1$), явном вычислении приближенной обратной матрицы AINV, TNS в реализации пакета PARALUTION (<http://www.paralution.com/>) на центральных процессорах и графических ускорителях.

Предобуславливатели ILU(p) формировались на центральном процессоре, а итерационный процесс решения BiCGStab выполнялся на ускорителе вычислений. В таблице 1 приведены времена формирования предобуславливателя (t_p), итерационного процесса (t_{its}) и число итераций (its), необходимых для решения систем линейных уравнений с использованием предобуславливателей LU и AISM. Временные затраты на пересылку предобуславливателя, сформированного на CPU, на графический ускоритель в приведенных табличных данных не учитывались.

Таблица 1. Вычислительные затраты предобуславливателей при решении систем с несимметричными матрицами, $t_p/t_{its}(its)$

Матрица A	$P_{ILU(0)}$	$P_{ILU(1)}$	$P_{AISM}(\tau = 0.0001)$	
	CPU/GPU	CPU/GPU	GPU/GPU	2 × GPU/GPU
cdde5	0.0002/0.13 (140)	0.002/0.12 (106)	0.57/0.13(167)	0.76/0.13 (165)
ex37	0.003/0.03 (3)	1.4/0.03 (2)	15.44/0.004 (4)	8.76/0.004 (4)
rajat03	—	—	169/0.11 (135)	79.08/0.28 (333)
flowmeter5	0.002/0.36 (63)	0.04/0.34 (32)	357.9/0.15 (112)	165.41/0.14 (105)
ex19	0.04/—	699/0.5 (279)	703/0.39 (128)	325.96/0.44 (127)
sme3Da	0.15/13.23(1700)	495/16.61(158)	843/13.5(1338)	389.113/11.5 (1005)
poisson3Da	0.04/0.13 (24)	56.3/0.58 (11)	1066/0.24 (30)	495.475/0.24 (30)

При рассмотрении результатов экспериментов основное внимание будет уделяться вычислительным затратам. Прежде всего остановимся на сравнении вариантов распараллеливания при формировании предобуславливателя P_{AISM} . Для **алгоритма 2** при использовании двух GPU для параллельного вычисления матриц S , T наблюдается двукратное ускорение. В этом случае коммуникационные затраты при обмене векторами x_{d_i}, x_{h_i} между GPU через общую память в рамках OpenMP существенны только для небольших матриц до $N = 3665$ (cdde5, ex37).

По времени работы алгоритмов решения систем методом BiCGStab видны преимущества предобуславливателя P_{AISM} . Затраты на одну итерацию в этом случае существенно ниже, чем при ILU(p). Достигнутое ускорение при формировании явного предобуславливателя P_{AISM} все-таки сохраняет достаточно большие вычислительные затраты и требует дальнейших исследований.

При рассмотрении плохо обусловленных матриц большего размера (ex19, sme3Da) затраты на формирование предобуславливателя P_{AISM} сравнимы с вариантом $P_{ILU(1)}$. Как видно из таблицы 1, в ряде случаев предобуславливатели на основе неполного разложения не обеспечили сходимость к решению. Так, использование ILU(0) в случае матрицы ex19 не привело к решению системы, а при решении системы с матрицей rajat03 из приведенных предобуславливателей только AISM обеспечил сходимость к точному решению.

Исключительно для целей тестирования в таблице 2 приведены результаты для явных предобуславливателей AINV, TNS при решении симметричных систем уравнений методом сопряженных градиентов с формированием матрицы предобуславливателя на центральных процес-

сорах и графических ускорителях. В этом случае симметричность матриц при формировании предобуславливателя P_{AISM} не учитывалась. Потенциально сокращение затрат при формировании рассматриваемого предобуславливателя в случае симметричных матриц представляется возможным и перспективным.

По скорости сходимости результаты многих тестов для различных предобуславливателей сопоставимы, а для матриц *bcsstk15*, *vibrobox* применение P_{AISM} позволило получить меньшее число итераций. Затраты на решение систем уравнений с предобуславливателями P_{AISM} и P_{AINV} примерно одинаковы.

Дальнейшее развитие предобуславливателей, основанных на приближенном обращении Шермана–Моррисона, должно быть связано блочным представлением разложения, выполняемом на нескольких GPU.

Таблица 2. Вычислительные затраты явных предобуславливателей при решении систем с симметричными матрицами, $t_p/t_{its}(its)$

Матрица <i>A</i>	P_{AINV}	P_{TNS}	P_{AISM}	
	CPU/GPU	GPU/GPU	GPU/GPU	2× GPU/GPU
nasa2910	2.5/0.65 (314)	0.002/0.32 (962)	8.78/0.62 (387)	5.62/0.57 (339)
bcsstk15	0.15/0.6 (293)	0.002/0.08 (259)	20.37/0.17 (81)	11.23/0.16 (70)
Kuu	4.03/0.16 (75)	0.004/0.1 (241)	142.03/0.18 (103)	66.7/0.18 (102)
msc10848	1.48/2.68 (1190)	0.006/14.7 (21871)	505.5/5.12 (846)	230.715/7.03 (757)
vibrobox	1.29/1.42 (683)	0.003/1.98 (4682)	814/0.81 (52)	391.8/0.8 (52)

Список литературы

1. Saad Y. Iterative methods for sparse linear systems. SIAM, 2003. xviii + 528 p. ISBN-10: 0-89871-534-2
2. Benzi M. Preconditioning techniques for large linear systems: a survey // Journal of Computational Physics. 2002. Vol. 182. № 2. P. 418–477.
3. Kolotilina L.Yu., Yeremin A.Yu. Factorized sparse approximate inverse preconditionings. I: Theory // SIAM Journal on Matrix Analysis and Applications. 1993. Vol. 14. № 1. P. 45–58.
4. Grote M., Huckle T. Parallel preconditioning with sparse approximate inverses // SIAM Journal on Scientific Computing. 1997. Vol. 18. № 3. P. 838–853. DOI: 10.1137/S1064827594276552
5. Sherman J., Morrison W.J. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix // The Annals of Mathematical Statistics. 1950. Vol. 21. № 1. P. 124–127.
6. Недожогин Н.С., Копысов С.П., Новиков А.К. Параллельное формирование предобуславливателя, основанного на аппроксимации обращения Шермана–Моррисона // Вычислительные методы и программирование. 2015. Т. 16. С. 86–93.
7. Bru M., Cerdán J., Marín J., Mas J. Preconditioning sparse nonsymmetric linear systems with Sherman–Morrison formula // SIAM Journal on Scientific Computing. 2003. Vol. 25. № 2. P. 701–715. DOI: 10.1137/S1064827502407524

Поступила в редакцию 30.09.2015

Недоожогин Никита Сергеевич, младший научный сотрудник, Институт механики УрО РАН, 426067, Россия, г. Ижевск, ул. Т. Барамзиной, 34.

E-mail: nedozhogin@inbox.ru

Копысов Сергей Петрович, д. ф.-м. н., профессор, кафедра вычислительной механики, Удмуртский государственный университет, 426034, Россия, г. Ижевск, ул. Университетская, 1; заведующий лабораторией, Институт механики УрО РАН, 426067, Россия, г. Ижевск, ул. Т. Барамзиной, 34.

E-mail: s.kopysov@gmail.com

Новиков Александр Константинович, к. ф.-м. н., старший научный сотрудник, Институт механики УрО РАН, 426067, Россия, г. Ижевск, ул. Т. Барамзиной, 34.

E-mail: sc_work@mail.ru

N. S. Nedozhogin, S. P. Kopysov, A. K. Novikov

Parallel decomposition version in the AISM preconditioner

Keywords: solving systems of linear algebraic equations, preconditioners, parallel algorithms, graphics processing.

We propose a variant of the parallel decomposition in the formation of preconditioner based on the approximate inversion of the Sherman-Morrison. We conduct numerical experiments for solving test systems of equations using GPUs.

REFERENCES

1. Saad Y. *Iterative methods for sparse linear systems*, SIAM, 2003. xviii + 528 p. ISBN-10: 0-89871-534-2
2. Benzi M. Preconditioning techniques for large linear systems: a survey, *Journal of Computational Physics*, 2002, vol. 182, no. 2, pp. 418–477.
3. Kolotilina L.Yu., Yeremin A.Yu. Factorized sparse approximate inverse preconditionings. I: Theory, *SIAM Journal on Matrix Analysis and Applications*, 1993, vol. 14, no. 1, pp. 45–58.
4. Grote M., Huckle T. Parallel preconditioning with sparse approximate inverses, *SIAM Journal on Scientific Computing*, 1997, vol. 18, no. 3, pp. 838–853. DOI: 10.1137/S1064827594276552
5. Sherman J., Morrison W.J. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix, *The Annals of Mathematical Statistics*, 1950, vol. 21, no 1, pp. 124–127.
6. Nedozhogin N.S., Kopysov S.P., Novikov A.K. Parallel forming preconditioner based on the approximate Sherman–Morrison inversion, *Vychislitel'nye metody i programirovanie*, 2015, vol. 16, pp. 86–93 (in Russian).
7. Bru M., Cerdán J., Marín J., Mas J. Preconditioning sparse nonsymmetric linear systems with Sherman–Morrison formula, *SIAM Journal on Scientific Computing*, 2003, vol. 25, no. 2, pp. 701–715. DOI: 10.1137/S1064827502407524

Received 30.09.2015

Nedozhogin Nikita Sergeevich, Junior Researcher, Institute of Mechanics, Ural Branch of RAS, ul. T. Baramzinoi, 34, Izhevsk, 426067, Russia.

E-mail: nedozhogin@inbox.ru

Kopysov Sergei Petrovich, Doctor of Physics and Mathematics, Professor, Department of Computational Mechanics, Udmurt State University, ul. Universitetskaya, 1, Izhevsk, 426034, Russia; Head of the Laboratory, Institute of Mechanics, Ural Branch of RAS, ul. T. Baramzinoi, 34, Izhevsk, 426067, Russia.

E-mail: s.kopysov@gmail.com

Novikov Aleksandr Konstantinovich, Candidate of Physics and Mathematics, Associate Professor, Department of Computational Mechanics, Udmurt State University, ul. Universitetskaya, 1, Izhevsk, 426034, Russia; Senior Researcher, Institute of Mechanics, Ural Branch of RAS, ul. T. Baramzinoi, 34, Izhevsk, 426067, Russia.

E-mail: sc_work@mail.ru