

## Параллельный блочно-диагональный переобуславливатель с проекторами для задачи диффузии

Крамаренко В. К.<sup>1</sup>✉, Кузнецов Ю. А.<sup>2</sup>, and Коньшин И. Н.<sup>1,3</sup>

<sup>1</sup> Институт вычислительной математики им. Г. И. Марчука Российской Академии Наук

<sup>2</sup> University of Houston

<sup>3</sup> Вычислительный центр им. А. А. Дородницына, ФИЦ ИУ РАН

**Аннотация** В статье представлена параллельная реализация блочно-диагонального переобуславливателя с проекторами. В качестве модельной задачи была взята задача диффузии с граничными условиями Дирихле и локальными включениями с гетерогенными коэффициентами диффузии. Было проведено сравнение эффективности разработанного переобуславливателя с переобуславливателями  $ILU(k)$ ,  $ILU2(\tau)$  и алгебраического многосеточного метода при использовании BiCGstab итераций. Приведены результаты численных экспериментов.

**Keywords:** уравнение диффузии, гетерогенные коэффициенты диффузии, параллельный переобуславливатель

### Введение

Задачи с локальными включениями часто возникают в различных приложениях, таких, как, например, задачи гомогенизации. В настоящее время имеется большое количество теоретических (см., например, [1]) и практических исследований на эту тему (см. [2,3]). В данной работе представлена параллельная реализация алгоритма, описанного в работе [4] и исследованного в работе [5]. Данная реализация была проведена на основе программной платформы INMOST [6] с использованием ее функционала как для построения расчетной сетки и модели дискретизации, так и для реализации описываемого переобуславливателя.

Работа состоит из 6 разделов. В первом разделе описывается математическая постановка задачи. Во втором и третьем разделах приведены постановка задачи и способ построения переобуславливателя [4,7]. В четвертом разделе описывается реализация параллельного переобуславливателя, описанного в работе [7]. В пятом разделе представлены результаты численных экспериментов, а шестой раздел содержит краткие выводы работы.

## 1 Постановка задачи

Рассмотрим классическое уравнение диффузии с граничными условиями Дирихле

$$\begin{aligned} -\nabla(D\nabla p) &= f && \text{in } \Omega, \\ p &= p_0 && \text{on } \partial\Omega, \end{aligned} \quad (1)$$

где  $\Omega$  – многоугольник/многогранник в области  $\mathbb{R}^2/\mathbb{R}^3$  с границей  $\partial\Omega$ ,  $\mathbf{n}$  – внешняя нормаль единичной длины к границе  $\partial\Omega$ ,  $D = D(\mathbf{x})$  – тензор диффузии, и  $f = f(\mathbf{x})$  – функция источника, а  $p_0$  – функция граничного условия Дирихле.

Задача (1) является типичной для описания многих физических процессов, таких как фильтрация воды или нефти сквозь горные породы.

В общем случае тензор  $D = D(\mathbf{x})$  может изменяться от ячейки к ячейке, но в данной работе будет рассмотрен случай кусочно-постоянного коэффициента диффузии.

В работе [4] подробно описана постановка в смешанной формулировке, в этой же статье будет рассмотрена упрощенная модель.

## 2 Описание тестовой задачи

Будем считать область  $\Omega$  единичным квадратом, разбитым на квадратные ячейки, каждая из которых в свою очередь разбита на два треугольника. В самой области задан скалярный коэффициент диффузии  $D = d_0 I$ . Внутри области выделены  $m$  подобластей  $\omega_s$ , соответствующие включениям, в которых коэффициент диффузии  $d_s$  существенно больше коэффициента в остальной области,  $d_s \gg d_0$ ,  $s = 1, \dots, m$ , как показано на Рис. 1. Включения не имеют общей границы друг с другом и не соприкасаются с границей области, т.е.  $\partial\omega_s \cap \partial\omega_t = \emptyset$ ,  $\partial\omega_s \cap \partial\Omega = \emptyset$ ,  $s, t = 1, \dots, m$ .

Для дискретизации был применен метод конечных элементов с элементами P1, что является эквивалентным применением стандартной пятиугольной схемы конечных разностей [8].

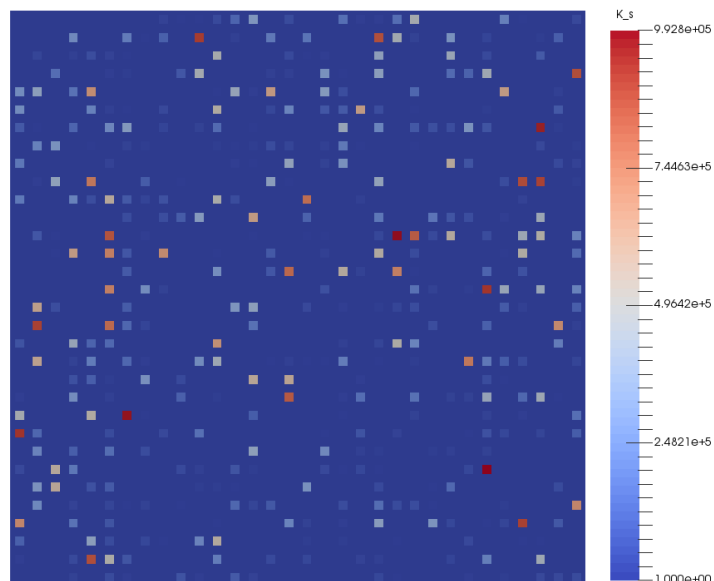
В итоге, после дискретизации получается систем линейных алгебраических уравнений

$$A\bar{p} = \bar{g} \quad (2)$$

с симметричной положительно определенной матрицей  $A$ .

Включения неоднородности коэффициента диффузии не граничат друг с другом, а также с границей области  $\Omega$ , тогда матрица  $A$  может быть представлена в виде суммы матрицы  $\hat{A}$ , которая отвечает за вклад ячеек вне включений и суммы матриц  $A_s$ , каждая из которых соответствует  $s$ -тому вкладу включений.

Учитывая предположение, что  $d_s \gg d_0$ , матрицу  $A_s$  можно представить в виде  $A_s = d_s A_{\lambda,s} = A_{\lambda,s} + (d_s - d_0) A_{\lambda,s}$ . В таком случае мы можем разделить матрицу  $A$  на две части: матрицу  $A_0$ , которая соответствует случаю



**Рис. 1:** Пример распределения включений с коэффициентом диффузии, выбранным случайным образом из интервала  $[1; 10^6]$ .

отсутствия включений ( $d_s \equiv d_0, s = 1, \dots, m$ ) и матрицу  $A_1$ , характеризующую вклад включений

$$A_1 = \sum_{s=1}^m (d_s - 1) N_s A_{\lambda,s} N_s^T, \quad (3)$$

где матрицы  $N_s$  – это матрицы ассемблирования.

Легко заметить, что матрицы  $A_{\lambda,s}$  являются матрицами для решения задачи диффузии в области  $\omega_s$  с условиями Неймана.

### 3 Построение переобуславливателя

Рассмотрим обобщенную проблему собственных значений

$$A_{\lambda,s} \bar{w} = \mu M_s \bar{w}, \quad s = 1, \dots, N, \quad (4)$$

где  $M_s$  диагональная  $m_s \times m_s$  матрица с положительными элементами ( $m_s$  – размерность матрицы  $A_{\lambda,s}$ ). Обозначим через  $\mu_{s,i}, i = 1, \dots, m_s$ , собственные значения матрицы  $A_{\lambda,s}$ , а через  $A_k$  диагональную матрицу с этими значениями, упорядоченными в порядке неубывания (заметим, что  $0 = \mu_{s,1} < \mu_{s,2}$ );  $\bar{w}_{s,i}, i = 1, \dots, m_s$ ,  $M_s$ -ортонормированные собственные векторы матрицы  $A_{\lambda,s}$ , а  $W_s$  – матрица, составленная из этих векторов. Во введенных

обозначениях спектральное разложение матрицы  $A_{\lambda,s}$  будет иметь вид:

$$A_{\lambda,s} = M_s W_s \Lambda_s W_s^T M_s. \quad (5)$$

Отметим, что

$$\bar{w}_{k,1} = \frac{1}{\|\bar{e}_s\|_{M_s}} \bar{e}_s, \quad (6)$$

где  $\bar{e}_s$  вектор с компонентами равными единице и  $\|\bar{e}_s\|_{M_s}$  обозначает  $M_s$ -норму  $\bar{e}_s$ ,  $s = 1, \dots, m$ .

В работах [4,7] предлагается использовать переобуславливатель

$$B_s = \alpha_k (M_s - M_s \bar{w}_{s,1} \bar{w}_{s,1}^T M_s), \quad (7)$$

где  $\alpha_s$  являются некоторыми положительными числами.

Рассмотрим также матрицу

$$B = \sum_{k=1}^m N_k B_k N_k^T. \quad (8)$$

Тогда блочно-диагональный переобуславливатель  $B$  для матрицы  $A$  задается формулой

$$B = Z + \sum_{k=1}^m (d_k - 1) N_k B_k N_k^T, \quad (9)$$

где  $Z$  – диагональная матрица с положительными элементами, например  $Z = \|M^{-1} A_0\|_{\infty} M$ , где  $M = \sum_{k=1}^m N_k M_{\lambda,k} N_k^T$ .

Очевидно, что матрица  $B$  является блочно-диагональной матрицей, все блоки которой  $B_{kk}$ ,  $k = 1, \dots, m$ , являются одноранговыми возмущениями диагональных матриц, а остальные блоки – диагональными матрицами.

#### 4 Параллельная реализация VDP переобуславливателя

Вместо прямого обращения матрицы  $B$  на каждой итерации будем решать линейную систему  $B\bar{u} = \bar{z}$ , как это делается в работе [7].

При решении системы линейных уравнений на каждой итерации необходимо решать систему  $\bar{u} = B^{-1}\bar{g}$  с переобуславливателем  $B$ , что эквивалентно решению системы  $B\bar{u} = \bar{g}$ .

В нашем случае справедливо равенство

$$B = Z - \sum_t \alpha_t N_t M_t \bar{w}_{t,1} \bar{w}_{t,1}^T M_t N_t^T. \quad (10)$$

Таким образом систему  $Bu = g$  можно записать в виде

$$Z\bar{u} - \sum_t \alpha_t N_t M_t \bar{w}_{t,1} \bar{w}_{t,1}^T M_t N_t^T \bar{u} = \bar{g} \quad (11)$$

или

$$\bar{u} - Z^{-1} \sum_t \alpha_t N_t M_t \bar{w}_{t,1} \bar{w}_{t,1}^T M_t N_t^T \bar{u} = Z^{-1} \bar{g}. \quad (12)$$

Вводя новую переменную  $\xi_t = \bar{w}_{t,1}^T M_t N_t^T u$  и домножая уравнение (12) на  $\bar{w}_{t,1}^T M_t N_t^T$  для всех включений мы получаем систему  $(I - Q) \bar{\xi} = \bar{\psi}$ , где  $Q = \{q_{ts}\}$ ,  $\bar{\psi} = \{\psi_t\}$ ,

$$q_{ts} = \bar{w}_{t,1}^T M_t N_t^T \alpha_t M^{-1} N_s M_s \bar{w}_{s,1}, \quad (13)$$

$$\psi_t = \bar{w}_{t,1}^T M_t N_t^T \bar{g}. \quad (14)$$

В нашем случае все внедиагональные элементы матрицы  $Q$  являются нулевыми,  $q_{ts} = 0$ ,  $t \neq s$ , т.к. включения не имеют общей границы. Учитывая это, а также тот факт, что диагональная часть переобуславливателя не зависит от правой части при решении системы  $B\bar{u} = \bar{g}$  мы можем разбить область так, что каждое отдельное включение во время расчета будет расположено только на одном процессоре. Вследствие этого процедура переобуславливания будет производиться полностью параллельно, а обмены между процессорами будут нужны только для дополнительного обмена частями вектора  $\bar{u}$  при параллельном умножении исходной матрицы  $A$  на вектор.

Блочно-диагональный переобуславливатель был реализован на базе платформы INMOST [6].

## 5 Результаты численных экспериментов

Численные эксперименты по сравнению эффективности параллельной реализации предложенного блочно-диагонального переобуславливателя (BDP) с другими переобуславливателями проводились с использованием платформы INMOST [6], которая позволяет вызывать различные вычислительные пакеты линейной алгебры, используя единый интерфейс.

Для сравнения свойств переобуславливателей были использованы три различных вычислительных пакета PETSc [9], Trilinos [10] и INMOST. Для всех 4 вариантов решения системы линейных уравнений (2), включая вариант с применением предложенного блочно-диагонального переобуславливателя, в качестве итерационной схемы для единообразия был выбран метод стабилизированных бисопряженных градиентов (BiCGstab). Этот метод был использован для корректного сравнения различных переобуславливателей реализованных в различных пакетах.

В самой платформе INMOST был выбран переобуславливатель  $ILU2(\tau)$  с параметром отсечения малых элементов  $\tau$ , равным  $10^{-3}$ . В пакете PETSc был использован традиционный переобуславливатель на основе неполного треугольного разложения  $ILU(k)$  с достаточно большим значением параметром расширения структуры переобуславливателя,  $k = 7$ , что было обусловлено необходимостью повышения его надежности. В качестве схемы распараллеливания использовался аддитивный метод Шварца с перекрытием,

размер которого в данных экспериментах был равен 1. В пакете Trilinos был использован алгебраический многосеточный переобуславливатель. Количество уровней переобуславливателя было выбрано равным 5, а пред- и пост-сглаживание было произведено при помощи пакета Ifpack.

Отметим что в случае блочно-диагонального и ILU2 переобуславливателя использовался итерационный метод BiCGstab из платформы INMOST, тогда как Trilinos и PETSc использовали свои встроенные реализации итерационной схемы BiCGstab.

Были рассмотрены различные значения  $m$  количества включений и различные размеры включений  $n$ , и, таким образом, различные размерности решаемой задачи  $N_A$ . Все численные эксперименты проводились на кластере ИВМ РАН [11] на 32 вычислительных ядрах двух вычислительных узлов очереди x8core, имеющих следующие характеристики:

- Compute Node Arbyte Alkazar+ R2Q50.
- 16 ядер (два 8-ядерных процессора Intel Xeon E5-2665@2.40ГГц).
- Оперативная память: 64 Гб.
- Операционная система: SUSE Linux Enterprise Server 11 SP1 (x86\_64).

Коэффициент диффузии для основной области был выбран равный единице, т.е.  $d_0 = 1$ . Были проведены две серии экспериментов, в которых скачки коэффициентов имели разную структуру. В первой серии все включения имели одинаковый коэффициент диффузии  $d_s = d_{\max}$ , сильно отличающийся от коэффициента диффузии основной области, т.е.  $d_{\max} \gg d_0$ . Во второй серии экспериментов коэффициенты диффузии выбирались для каждого включения случайным образом в диапазоне от коэффициента диффузии основной области до некоего значения  $d_{\max}$ , т.е.  $d_s \in [1; d_{\max}]$ .

Правая часть во всех экспериментах выбиралась нулевой, тогда как начальное приближение выбиралось случайным образом в диапазоне  $(0; 1)$ . Критерием остановки было падение  $L_2$ -нормы невязки в  $10^{12}$  раз.

В таблицах ниже величина  $d_s$  обозначает коэффициент диффузии во включениях. Он может быть одинаковым во всех включениях или может быть выбран случайным образом из диапазона  $[1; d_{\max}]$ , “#iter” – количество итераций метода BiCGstab, “Time” – время, затраченное на построение переобуславливателя и решения линейной системы.

В таблицах 1, 2 и 3 проявляется характерная черта [4] независимости количества итераций при использовании блочно-диагонального переобуславливателя от скачка коэффициентов диффузии. Блочно-диагональный переобуславливатель также является самым быстрым из всех рассматриваемых в пересчете на одну итерацию.

Результаты, представленные в таблицах 4, 5 и 6 также показывают невосрастание количества итераций для блочно-диагонального переобуславливателя при увеличении скачков коэффициента диффузии. Однако следует отметить, что при одинаковом скачке коэффициента блочно-диагональный переобуславливатель BDP сходится быстрее чем при случайном разбросе коэффициентов диффузии во включениях.

**Таблица 1:**  $m = 32, n = 2, N_A = 16129$ .

Переобуславливатель		#iter	Time		#iter	Time
PETSc ILU(7)	$d_s = 10^4$	48	0.02	$d_s = 10^6$	54	0.02
Trilinos AMG		107	0.20		285	0.32
INMOST ILU2( $10^{-3}$ )		62	0.06		82	0.12
BDP		163	0.13		130	0.05
PETSc ILU(7)	$d_s \in [1; 10^4]$	51	0.02	$d_s \in [1; 10^6]$	50	0.02
Trilinos AMG		133	0.15		775	0.81
INMOST ILU2( $10^{-3}$ )		50	0.03		61	0.03
BDP		176	0.04		160	0.03

**Таблица 2:**  $m = 64, n = 2, N_A = 65025$ .

Переобуславливатель		#iter	Time		#iter	Time
PETSc ILU(7)	$d_s = 10^4$	86	0.14	$d_s = 10^6$	90	0.02
Trilinos AMG		123	0.34		440	1.15
INMOST ILU2( $10^{-3}$ )		119	0.14		175	0.16
BDP		292	0.18		271	0.16
PETSc ILU(7)	$d_s \in [1; 10^4]$	91	0.15	$d_s \in [1; 10^6]$	103	0.17
Trilinos AMG		128	0.36		1055	2.70
INMOST ILU2( $10^{-3}$ )		66	0.21		97	0.21
BDP		334	0.21		285	0.17

**Таблица 3:**  $m = 128, n = 2, N_A = 261121$ .

Переобуславливатель		#iter	Time		#iter	Time
PETSc ILU(7)	$d_s = 10^4$	115	0.91	$d_s = 10^6$	134	1.04
Trilinos AMG		153	2.01		369	4.81
INMOST ILU2( $10^{-3}$ )		243	1.87		338	1.98
BDP		613	2.61		451	1.88
PETSc ILU(7)	$d_s \in [1; 10^4]$	134	1.05	$d_s \in [1; 10^6]$	146	1.17
Trilinos AMG		125	1.67		890	11.59
INMOST ILU2( $10^{-3}$ )		110	1.66		227	2.31
BDP		681	2.91		447	1.87

**Таблица 4:**  $m = 32, n = 4, N_A = 65025$ .

Переобуславливатель		#iter	Time		#iter	Time
PETSc ILU(7)	$d_s = 10^4$	83	0.27	$d_s = 10^6$	80	0.14
Trilinos AMG		120	1.22		414	1.10
INMOST ILU2( $10^{-3}$ )		62	0.06		163	0.23
BDP		284	0.50		290	0.15
PETSc ILU(7)	$d_s \in [1; 10^4]$	85	0.14	$d_s \in [1; 10^6]$	90	0.15
Trilinos AMG		64	0.19		538	1.43
INMOST ILU2( $10^{-3}$ )		54	0.27		84	0.30
BDP		375	0.20		309	0.17

**Таблица 5:**  $m = 64, n = 4, N_A = 261121$ .

Переобуславливатель		#iter	Time		#iter	Time
PETSc ILU(7)	$d_s = 10^4$	125	0.99	$d_s = 10^6$	120	0.94
Trilinos AMG		122	1.66		750	9.71
INMOST ILU2( $10^{-3}$ )		131	1.93		279	2.42
BDP		544	2.16		528	1.98
PETSc ILU(7)	$d_s \in [1; 10^4]$	126	0.99	$d_s \in [1; 10^6]$	134	1.04
Trilinos AMG		85	1.15		551	7.14
INMOST ILU2( $10^{-3}$ )		86	1.80		133	1.99
BDP		644	2.43		472	1.77

**Таблица 6:**  $m = 128, n = 4, N_A = 1046529$ .

Переобуславливатель		#iter	Time		#iter	Time
PETSc ILU(7)	$d_s = 10^4$	206	6.34	$d_s = 10^6$	205	6.37
Trilinos AMG		146	11.26		663	50.04
INMOST ILU2( $10^{-3}$ )		242	11.08		338	1.98
BDP		1122	17.53		866	13.13
PETSc ILU(7)	$d_s \in [1; 10^4]$	218	6.71	$d_s \in [1; 10^6]$	183	5.71
Trilinos AMG		111	8.57		891	67.93
INMOST ILU2( $10^{-3}$ )		149	10.78		355	19.45
BDP		1445	21.92		1336	20.48

**Таблица 7:**  $m = 32, n = 8, N_A = 261121$ .

Переобуславливатель		#iter	Time		#iter	Time
PETSc ILU(7)	$d_s = 10^4$	1341	9.89	$d_s = 10^6$	403	3.02
Trilinos AMG		91	1.23		463	6.01
INMOST ILU2( $10^{-3}$ )		94	1.56		421	5.63
BDP		714	2.66		400	1.51
PETSc ILU(7)	$d_s \in [1; 10^4]$	574	4.25	$d_s \in [1; 10^6]$	1012	7.51
Trilinos AMG		63	0.87		380	4.96
INMOST ILU2( $10^{-3}$ )		69	1.62		171	3.39
BDP		740	2.78		576	2.12

**Таблица 8:**  $m = 64, n = 8, N_A = 1046529$ .

Переобуславливатель		#iter	Time		#iter	Time
PETSc ILU(7)	$d_s = 10^4$	565	17.10	$d_s = 10^6$	1982	59.12
Trilinos AMG		88	6.87		483	36.63
INMOST ILU2( $10^{-3}$ )		191	11.56		1094	53.49
BDP		1016	15.00		966	14.10
PETSc ILU(7)	$d_s \in [1; 10^4]$	1177	35.2	$d_s \in [1; 10^6]$	1223	36.63
Trilinos AMG		71	5.58		513	39.08
INMOST ILU2( $10^{-3}$ )		126	10.49		299	21.29
BDP		1262	18.97		1152	17.15



**Таблица 9:**  $m = 128, n = 8, N_A = 4190209$ .

Переобуславливатель		#iter	Time		#iter	Time
PETSc ILU(7)	$d_s = 10^4$	3641	449.99	$d_s = 10^6$	3996	495.52
Trilinos AMG		92	34.41		518	189.17
INMOST ILU2( $10^{-3}$ )		322	74.85		1441	278.21
BDP		2609	165.9		1839	117.47
PETSc ILU(7)	$d_s \in [1; 10^4]$	1517	189.21	$d_s \in [1; 10^6]$	3871	368.87
Trilinos AMG		76	28.57		544	105.23
INMOST ILU2( $10^{-3}$ )		206	63.98		492	80.80
BDP		2996	192.85		2539	76.18

В таблицах 7, 8 и 9 можно увидеть резкое увеличение времени решения с переобуславливателем  $ILU(k)$  при  $k = 7$ . Также следует заметить, что при использовании меньшего количества уровней расширения структуры в этом переобуславливателе, например, использование  $k = 6$  не дает гарантии решения линейной системы.

Стоит также отметить, что во всех сериях экспериментов многосеточный метод из пакета Trilinos дает значительный рост числа итераций и времени решения при увеличении скачка коэффициентов.

Исследование зависимости скорости сходимости от количества процессоров для различных переобуславливателей не проводилось. Причиной этого было то, что ввиду блочно-диагональной структуры переобуславливателя BDP скорость сходимости итерационного метода с ростом количества процессоров должна оставаться постоянной. В этом заключается основное преимущество рассматриваемого переобуславливателя BDP перед остальными способами построения переобуславливателей.

## 6 Выводы

В статье представлена процедура построения параллельного блочно-диагонального переобуславливателя для решения уравнения диффузии с кусочно-постоянными коэффициентами. Этот переобуславливатель является идеально распараллеливаемым и во многих случаях не уступает по скорости сходимости другим, широко используемым переобуславливателям. Также стоит отметить, что идея построения такого переобуславливателя легко обобщается для уравнений реакции-диффузии или конвекции-реакции-диффузии, а также для решения задач упругости.

**Благодарности.** Теоретическая часть данной работы была выполнена при финансовой поддержке гранта РФФ 14-31-00024. Экспериментальная часть была частично поддержана грантом РФФИ 17-01-00886.

## Литература

1. *Kuznetsov Yu. A.*, New homogenization method for diffusion equations // Russ. J. Numer. Anal. Math. Modelling, **33** (2018), No. 2, 85–93.
2. *Aarnes J., Hou Th. Y.* Multiscale domain decomposition methods for elliptic problems with high aspect ratios // Acta Mathematicae Applicatae Sinica, **18** (2002), No. 1, 63–76.
3. *Aksoylu B., Graham I. G., Klie H., Scheichl R.* Towards a rigorously justified algebraic preconditioner for high-contrast diffusion problems // Computing and Visualization in Science **11** (2008), No. 4, 319–331.
4. *Kuznetsov Y., Kramarenko V.* Preconditioners with projectors for mixed hybrid finite element methods // Russ. J. Numer. Anal. Math. Modelling, **32** (2017), No. 1, 39–45.
5. *Крамаренко В. К.* Предобуславливатель с проекторами для смешанного метода конечных элементов // Современные проблемы математического моделирования. Сборник трудов XVII Всероссийской конференции-школы молодых исследователей Абрау-Дюрсо, 11–16 сентября 2017 г., 91–98.
6. INMOST – a toolkit for distributed mathematical modeling. URL: <http://www.inmost.org> (дата обращения: 15.04.2018)
7. *Kuznetsov Yu. A.* Two-level preconditioners with projectors for unstructured grids // Russ. J. Numer. Anal. Math. Modelling, **15** (2000), No. 3–4, 247–255.
8. *Dobrowolski M.* On finite difference schemes for elliptic equations with discontinuous coefficients. Computational Methods in Applied Mathematics, **13** (2013), No. 3, 281–289.
9. PETSc is a suite of data structures and routines for the scalable (parallel) solution of scientific applications modeled by partial differential equations. URL: <http://www.mcs.anl.gov/petsc> (дата обращения: 15.04.2018)
10. Trilinos – platform for the solution of large-scale, complex multi-physics engineering and scientific problems. URL: <http://trilinos.org/> (дата обращения: 15.04.2018)
11. Кластер ИВМ РАН. URL: <http://cluster2.inm.ras.ru> (дата обращения: 15.04.2018)