

УДК 519.6

## ПАРАЛЛЕЛЬНЫЙ ПРЕДОБУСЛОВЛИВАТЕЛЬ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ ДЛЯ РЕШЕНИЯ НЕЛИНЕЙНОГО УРАВНЕНИЯ ЛУЧИСТОЙ ТЕПЛОПРОВОДНОСТИ В МЕТОДИКЕ "КОРОНА"

С. В. Чеботарь

(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Работа посвящена решению систем линейных уравнений, которые возникают при расчетах процессов теплопроводности на ЭВМ с распределенной памятью. Представлено описание параллельного предобусловливателя, используемого при решении систем линейных уравнений в блоке теплопроводности методики КОРОНА итерационными схемами на подпространствах Крылова. В его основе лежит параллельное многоуровневое неполное  $LU$ -разложение разреженных матриц в сочетании с неполным блочным разложением обратной матрицы. Суть метода — в рекурсивном построении иерархии матриц дополнения Шура с последовательным уменьшением числа процессов. Реализована возможность остановки рекурсии на заданном уровне и построения для очередной матрицы дополнения Шура неполной блочной аппроксимации обратной матрицы.

Приведены результаты вычислительных экспериментов, полученные при решении задач по методике КОРОНА в параллельном режиме. Для получения сравнительных показателей реализовано еще несколько общепринятых предобусловливателей, которые используют в качестве основы  $ILLU(t)$ -разложение: блочный метод Якоби, ограниченный аддитивный метод Шварца, а также метод из семейства Spike, использующий редуцированную матрицу.

*Ключевые слова:* параллельные вычисления, системы линейных алгебраических уравнений, предобусловливатель, методика КОРОНА, теплопроводность.

### Введение

Решение систем линейных алгебраических уравнений (СЛАУ) является одним из наиболее затратных этапов численного моделирования задач механики сплошной среды с помощью неявных методов. Методы решения СЛАУ постоянно развиваются, в частности, в связи с необходимостью их адаптации к усложняющейся с течением времени архитектуре вычислительной техники. Одним из ключевых моментов в развитии вычислительной техники было создание машин с распределенной памятью. Это, соответственно, потребовало создания параллельных алгоритмов и программ решения СЛАУ для таких компьютеров.

Предобусловливатели на основе неполного треугольного разложения ( $ILLU$ ) в комбинации с итерационными методами на базе подпространств Крылова [1] хорошо зарекомендовали себя с точки зрения надежности и эффективности при решении СЛАУ большого размера с разреженными матрицами. Как правило, адаптация этих подходов к параллельным вычислительным системам выполняется двумя способами — перенумерацией неизвестных (reordering) либо применением идей методов декомпозиции области (domain decomposition). К примеру, в работах [2–4] параллельная неполная факторизация основана на разбиении графов и многоцветной раскраске. Методы декомпозиции области, в свою очередь, позволяют использовать стратегии построения глобальных предобусловливателей [5]. При этом начальная область разбивается на подобласти, в том числе с перекрытием получающихся подобластей. В этом случае получается распределенная система уравнений.

В статье представлено описание параллельного предобусловливателя, используемого в блоке теплопроводности методики КОРОНА [6] при проведении двумерных и трехмерных расчетов на ЭВМ с распределенной памятью. В его основе лежит параллельное многоуровневое неполное  $LU$ -разложение разреженных матриц в сочетании с неполным блочным разложением обратной матрицы [7]. Приведены результаты, полученные при решении задач по методике КОРОНА в параллельном режиме. Для получения сравнительных показателей реализовано несколько общепринятых предобусловливателей (например, блочный метод Якоби), которые используют в качестве основы  $ILU(t)$ -разложение [1].

Описание и отдельные результаты применения представленного параллельного предобусловливателя при решении уравнения теплопроводности изложены также в работе [8].

### Обзор методики КОРОНА

Методика КОРОНА разработана для параллельных расчетов процессов газовой динамики с учетом лучистой теплопроводности. Одно пространственное направление трехмерной сетки рассчитывается на основе лагранжева подхода на подвижной расчетной сетке, два других направления — на неподвижной расчетной сетке (эйлеров подход). Используется неявная разностная схема с отслеживанием границ веществ с помощью составных ячеек. Если в ячейке больше одного вещества, то она разбивается на компоненты, каждый из которых отвечает одному веществу со своими параметрами: температурой, плотностью, уравнением состояния (УРСом) и т. д. Решение уравнений газовой динамики и теплопроводности для составной ячейки такое же, как для чистой. Разностный шаблон переменный, размер его для каждой ячейки зависит от степени неортогональности сетки и числа компонентов в соседних составных ячейках. Вся область задачи делится на фрагменты, в которых свои правила расстановки сетки, а также свои параметры декомпозиции. Внутри каждого фрагмента декомпозиция на процессы выполняется только по двум эйлеровым направлениям, как показано на рис. 1.

Решение уравнения теплопроводности находится по полностью неявной схеме одновременно во всех направлениях по всей многофрагментной области задачи. На каждом временном шаге выполняется линеаризация методом Ньютона системы нелинейных уравнений и рассчитывается несколько итераций Ньютона по нелинейности коэффициента теплопроводности, на каждой из которых порождается СЛАУ, которую надо решить с определенной точностью. При этом портрет положительно определенной матрицы СЛАУ на каждой итерации постоянный, несимметричный и перестраивается только между шагами.

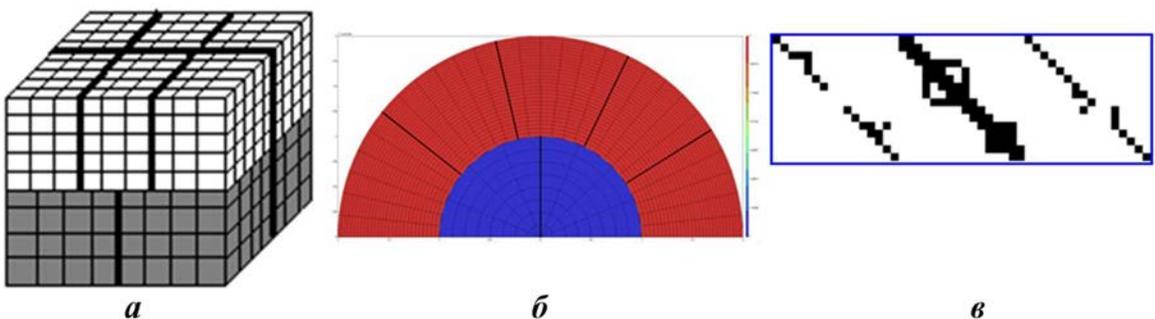


Рис. 1. Примеры декомпозиции и портрета матрицы в методике КОРОНА: *a* — два фрагмента при расчете трехмерной задачи на 10 процессах; *б* — два фрагмента при расчете двумерной задачи на 7 процессах; *в* — фрагмент портрета матрицы из блока теплопроводности расчета двумерной задачи

### Описание предобусловливателя, разработанного для методики КОРОНА

Пусть на ЭВМ с распределенной памятью необходимо решить систему уравнений  $Ax = f$  с положительно определенной матрицей  $A$ , имеющей размеры  $n \times n$  и хранящейся в памяти  $N$  процессов

группами строк, или, по-другому, блоками:  $A = \begin{pmatrix} A_{11} & & A_{1N} \\ & \ddots & \\ A_{N1} & & A_{NN} \end{pmatrix}$ .

Большинство блоков  $A_{ij}$ ,  $i \neq j$ , — нулевые, а оставшиеся имеют число ненулевых строк и столбцов много меньше, чем в диагональных квадратных блоках  $A_{ii}$ . Все блоки  $A_{ij}$ ,  $j = 1, \dots, N$ , хранятся на процессе с номером  $i$ , т. е. любая строка матрицы хранится целиком на одном из процессов.

Построение предобусловливателя состоит из четырех этапов (алгоритм 1):

1. Разделение строк матрицы на каждом процессе на граничные и внутренние, представление

$$\text{диагональных блоков в виде } A_{ii} = R_i^T \begin{pmatrix} B_i & C_i \\ D_i & E_i \end{pmatrix} R_i.$$

2.  $ILLU(t)$ -разложение блоков  $B_i$ .

3. Вычисление приближенной матрицы дополнения Шура  $S$ .

4. Построение аппроксимации для матрицы  $S$  или  $S^{-1}$  (в зависимости от уровня разложения).

При этом первые три этапа выполняются на всех процессах полностью независимо.

**Разделение строк матрицы на граничные и внутренние.** На каждом процессе  $I = 1, \dots, N$  строку  $i$  матрицы назовем *граничной*, если существует  $a_{ji} \neq 0$  или  $a_{ij} \neq 0$  и строка  $j$  хранится на процессе  $J$ , отличном от  $I$  (рис. 2). Строки, которые не являются граничными, назовем *внутренними*.

Выполним симметричные перестановки так, чтобы на каждом процессе располагались сначала внутренние строки, а потом граничные. Таким образом, диагональный блок представим в виде блочной матрицы с размерами  $2 \times 2$   $R_i A_{ii} R_i^T =$

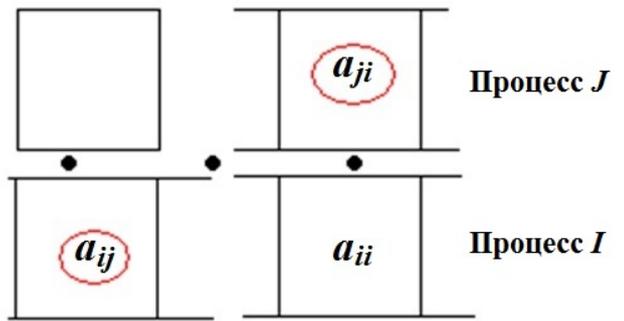


Рис. 2. Понятие граничной строки

$$= \begin{pmatrix} B_i & C_i \\ D_i & E_i \end{pmatrix}, \text{ где } R = \begin{pmatrix} R_1 & & \\ & \ddots & \\ & & R_N \end{pmatrix} \text{ — блочно-диагональная матрица перестановок; бло-$$

ки  $B_i, C_i$  содержат внутренние строки, а  $D_i, E_i$  — граничные.

**$ILLU(t)$ -разложение блоков  $B_i$ .** После симметричных перестановок исходная матрица преобразуется. На рис. 3 такое преобразование показано для матрицы, соответствующей двумерной сетке из  $6 \times 6$  ячеек, которая распределена на 4 процессора в декомпозиции  $2 \times 2$ .

Видно, что для диагональных блоков черного цвета на рис. 3 справа можно параллельно выполнить неполное  $ILLU(t)$ -разложение. Таким образом, получим  $B_i \approx L_i U_i$ ,  $i = 1, \dots, N$ .

**Вычисление приближенной матрицы дополнения Шура и предобусловливателя.** Вычислим диагональные блоки распределенной по процессам матрицы дополнения Шура  $S$  [3]:  $S_{ii} = E_i - D_i (L_i U_i)^{-1} C_i$ .

Предобусловливатель для исходной задачи можно записать в виде  $RAR^T \approx M = L\tilde{S}U$ , где  $L, U$  — блочно-диагональные матрицы с блоками вида  $L_{ii} = \begin{pmatrix} L_i & \\ D_i L_i^{-1} & I \end{pmatrix}$ ,  $U_{ii} = \begin{pmatrix} U_i & U_i^{-1} C_i \\ & I \end{pmatrix}$  и

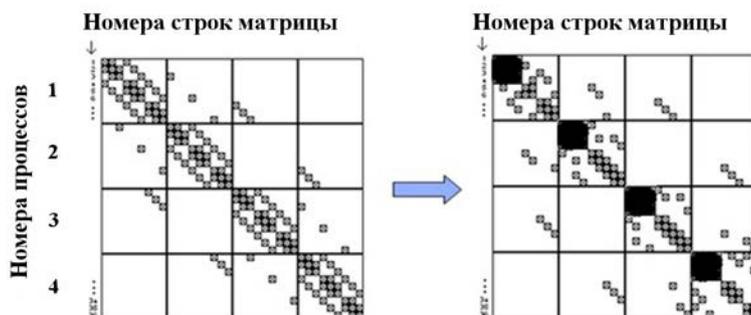


Рис. 3. Трансформация матрицы при симметричных перестановках

$L_i U_i \approx B_i$ ,  $i = 1, \dots, N$ ;  $I$  — квадратный единичный блок. В свою очередь, диагональные блоки  $\tilde{S}$  запишем как  $\tilde{S}_{ii} = \begin{pmatrix} I \\ S_{ii} \end{pmatrix}$ . Недиагональные блоки  $\tilde{S}_{ij} = R_i A_{ij} R_j^T$ ,  $i \neq j$ , являются перенумерованными копиями блоков исходной матрицы. При таком построении блочный портрет  $\tilde{S}$  полностью совпадает с таковым для исходной матрицы. Умножение обратной матрицы предобусловливателя сопровождается решением двух треугольных систем с матрицами  $L$ ,  $U$  и решением системы с матрицей  $\tilde{S}$ .

Обозначим через  $P$  оператор сужения на подпространство граничных переменных и пусть далее  $cols(P)$  и  $rows(P)$  — число столбцов и число строк матрицы, соответствующей оператору  $P$ . Тогда  $rows(P) = n$ ;  $cols(P) = \sum_{i=1}^N rows(E_i)$ . Аналогично блочному разбиению исходной матрицы получим

блочную структуру  $P = \begin{pmatrix} P_1 & & 0 \\ & \ddots & \\ 0 & & P_N \end{pmatrix}$ , где  $P_i = \begin{pmatrix} 0 \\ I_i \end{pmatrix}$ ;  $cols(P_i) = rows(E_i)$ ;  $rows(P_i) =$

$= cols(A_{ii})$ , а квадратный единичный блок  $I_i$  такой, что  $cols(I_i) = rows(I_i) = cols(E_i) = rows(E_i)$ . Тогда решение системы с матрицей  $\tilde{S}$  эквивалентно решению системы с матрицей  $S = P^T \tilde{S} P$ , в которой  $S_{ii} = E_i - D_i (L_i U_i)^{-1} C_i$ ;  $S_{ij} = P_i^T R_i A_{ij} R_j^T P_j$ ,  $i \neq j$ .

**Построение аппроксимации для матрицы  $S$  или  $S^{-1}$ .** По построению все строки  $S$  будут граничными. Выполним перераспределение строк этой матрицы между процессами так, чтобы можно было рекурсивно применить алгоритм 1. Очевидно, что в таком случае часть процессов можно выключить из работы, передав все имеющиеся на них элементы матрицы  $S$  соседям.

В качестве альтернативы в предлагаемом подходе вводится номер уровня, на котором вместо перераспределения элементов и продолжения рекурсии для приближения текущей матрицы  $S$  (или  $S^{-1}$ ) используется какой-либо метод декомпозиции области. Например, в работе [9] используется аддитивный метод Шварца. В предобусловливателе для задач из блока теплопроводности методики КОРОНА применен метод неполной блочной аппроксимации обратной матрицы [7].

Суть данного метода в следующем (алгоритм 2):

1.  $S$  разбивается на подматрицы с перекрытием *наверх* (высота перекрытия  $q$  — параметр метода):
  - 1) строится неориентированный граф матрицы  $S$ ;
  - 2) в расширенную подобласть процесса  $p$  входят строки, удовлетворяющие условиям:
    - строка  $i$  находится в матрице  $S$  выше первой строки с процесса  $p$ ;
    - из какой-либо вершины, соответствующей строке с процесса  $p$ , можно дойти до вершины  $i$  не больше чем за  $q$  шагов.
2. В расширенных областях строится  $ILLU(t)$ -разложение  $S_p \approx L_p U_p$ ,  $p = 1, \dots, N$ .

3. Приближение для обратной матрицы записывается как

$$S^{-1} \approx \sum_{p=1}^N U_p^{-1} \begin{pmatrix} 0_p & 0 \\ 0 & I_p \end{pmatrix} L_p^{-1}, \quad \text{rows}(I_p) = \text{cols}(I_p) = \text{rows}(E_p).$$

Представленный метод далее будем называть Schur+OI (Schur complement + Overlap Inverse). В процессе проведения расчетов по методике КОРОНА опытным путем получено, что наиболее эффективно использовать один уровень построения матрицы дополнения Шура и для нее сразу применять неполную блочную аппроксимацию. В случае возникновения трудностей при решении какой-либо СЛАУ (например, не достигается точность за данное число итераций) происходит увеличение количества уровней, часто в ущерб времени счета. Как правило, увеличение числа уровней редукции случается редко. Это требуется при решении СЛАУ, которых много меньше 1% от их общего порождаемого числа при решении задачи по методике КОРОНА, что практически не влияет на время счета. В вычислительных экспериментах, приведенных далее, предобусловливатель Schur+OI применялся с одним уровнем.

### Описание предобусловливателя на базе методов семейства Spike

Одним из методов, относительно которого далее будут проводиться сравнительные исследования, является предобусловливатель на основе методов семейства Spike [10]. В данном подходе также вычисляется редуцированная матрица и решается соответствующая СЛАУ. В отличие от построения матрицы дополнения Шура в методах типа Spike нет необходимости выполнять перестановки и выделять граничные строки матрицы. Для вычислительных экспериментов реализован одноуровневый подход.

Предположим, что исходная матрица распределена на три процесса и имеет вид

$$A = \begin{pmatrix} A_{11} & A_{12} & 0 \\ A_{21} & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{pmatrix}, \quad \text{где } A_{ij} = \begin{pmatrix} 0 \\ 0 \\ B_{ij} \end{pmatrix} \text{ при } i < j; \quad A_{ij} = \begin{pmatrix} B_{ij} \\ 0 \\ 0 \end{pmatrix} \text{ при } i > j.$$

Построение предобусловливателя выполняется в следующей последовательности:

1. На всех процессах строим неполную факторизацию  $A_{ii} = L_{ii}U_{ii}$ .

2. Вычисляем  $C_{ij} = \begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix}_{ij} = (L_{ii}U_{ii})^{-1} A_{ij}$ .

3. Получаем

$$A \approx M = \begin{pmatrix} L_{11}U_{11} & & \\ & L_{22}U_{22} & \\ & & L_{33}U_{33} \end{pmatrix} \begin{pmatrix} I & & \begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix}_{12} & 0 & 0 & 0 \\ & I & & & & \\ 0 & 0 & \begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix}_{21} & I & & \begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix}_{23} \\ & & & & I & \\ 0 & & 0 & 0 & \begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix}_{32} & I \\ & & & & & I \end{pmatrix}.$$

4. Формируем редуцированную матрицу  $S = \begin{pmatrix} I & (C_3)_{12} & & \\ (C_1)_{21} & I & & (C_1)_{23} \\ (C_3)_{21} & & I & (C_3)_{23} \\ & & (C_1)_{32} & I \end{pmatrix}$ .

5. Вычисляем блочное приближение обратной матрицы  $S^{-1} \approx \sum_{p=1}^N U_p^{-1} \begin{pmatrix} 0_p & 0 \\ 0 & I_p \end{pmatrix} L_p^{-1}$ .

Далее этот предобусловливатель обозначим Spike+OI (Spike + Overlap Inverse).

### Вычислительные эксперименты на двумерной задаче

Все расчеты в двумерном и трехмерном случаях выполнены только с распараллеливанием по процессам на стандарте MPI без использования двухуровневого подхода MPI+OpenMP.

Для исследований в двумерном случае взята сферическая осесимметричная задача с размером сетки  $4800 \times 5100$  счетных ячеек. Исследовалась работа решателей на шести временных отрезках  $T_1, T_2, \dots, T_6$ , каждый из которых характеризует определенную стадию процесса. На каждом временном отрезке рассчитывалось 40-50 временных шагов, время работы решателя суммировалось.

Проведено исследование масштабирования решения СЛАУ с помощью предобусловливателя, полученного по алгоритму 1, а также его сравнение:

- с блочным предобусловливателем Якоби (BJ);
- с треугольной блочной с перекрытием аппроксимацией обратной матрицы (метод приближения дополнения Шура применительно к исходной задаче, обозначенный далее как Overlap Inverse);
- с предобусловливателем Spike+OI (см. алгоритм 2).

Отметим, что все применяемые в экспериментах предобусловливатели разработаны на одной и той же собственной базе низкоуровневых подпрограмм линейной алгебры, что позволяет исключить влияние качества реализации на процесс сравнения.

Параметры  $ILU(t)$ -разложений на процессах брались одинаковыми во всех методах, в частности, пороговое значение для фильтрации бралось равным 0,001. Итерации проводились мето-

дом BiCGStab [11] с критерием точности  $\mu = \max_i \frac{|f_i - \sum_j a_{ij}x_j|}{|f_i| + \sum_j |a_{ij}x_j|} \leq 10^{-12}$ . Число уровней раз-

ложения в Spike+OI и Schur+OI равнялось 1.

В табл. 1 представлено время решения СЛАУ с предобусловливателем Schur+OI при изменении количества процессов от 56 до 560. Эффективность распараллеливания относительно 56 процессов составляет от 50 до 80%. В некоторых случаях наблюдается удвоение числа итераций.

В табл. 2 приведено сравнение времени решения СЛАУ на всех временных отрезках с использованием четырех предобусловливателей. Хуже всего показатели у блочного метода Якоби, наилучшими с практически идентичными результатами оказались реализации методов Spike + OI и Schur + OI.

### Вычислительные эксперименты на трехмерных задачах

Для исследований взято три задачи. В задаче 1 задана совокупность сферических и цилиндрических тел с разными наборами веществ. В ней бралось восемь временных интервалов:  $T_{11}, T_{12}, \dots, T_{18}$ . Число ячеек сетки менялось от 40 до 20 миллионов. В задачах 2 и 3 — сферы с различными наборами веществ. Размеры сеток — 1 млн ячеек и 9,15 млн соответственно. В задаче 2 бралось два

Таблица 1

**Время решения двумерной задачи, число итераций и эффективность распараллеливания при использовании метода Schur+OI**

Кол-во процессов	Показатель	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$
56	Время решения	695,85	617,11	1 226,78	969,20	844,92	464,88
	Итерации	3 716	3 506	6 751	5 054	3 957	1 916
	Эффективность	100 %	100 %	100 %	100 %	100 %	100 %
140	Время решения	260,54	400,42	713,43	379,53	331,31	186,16
	Итерации	3 722	5 996	10 291	5 172	3 987	2 041
	Эффективность	107 %	62 %	69 %	102 %	102 %	100 %
280	Время решения	130,11	204,60	366,68	202,39	191,99	107,77
	Итерации	3 753	6 064	10 977	5 531	4 719	2 317
	Эффективность	107 %	60 %	67 %	96 %	88 %	86 %
560	Время решения	85,14	126,76	246,59	143,73	150,77	81,18
	Итерации	4 609	7 227	13 494	7 604	7 569	3 597
	Эффективность	82 %	49 %	50 %	67 %	56 %	57 %

Таблица 2

**Время решения двумерной задачи и число итераций при использовании четырех методов на 280 процессах**

Метод	Показатель	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$
BJ	Время решения	344,7	647,4	1 756,0	2 108,0	2 098,0	1 058,2
	Итерации	13 178	23 693	75 500	79 640	78 204	34 543
Overlap Inverse	Время решения	178,76	379,34	496,65	291,50	520,71	322,88
	Итерации	5 377	11 645	15 166	8 633	15 408	8 754
Schur + OI	Время решения	130,11	204,60	366,68	202,39	191,99	107,77
	Итерации	3 753	6 064	10 977	5 531	4 719	2 317
Spike + OI	Время решения	123,8	200,4	374,5	201,7	183,6	98,5
	Итерации	3 722	6 065	10 931	5 456	4 565	2 212

временных интервала:  $T_{21}$  и  $T_{22}$ , в задаче 3 — четыре интервала:  $T_{31}, \dots, T_{34}$ . На всех временных интервалах выполнялось 50 шагов по времени, время работы решателя суммировалось по всем шагам.

Проведено исследование масштабирования решения СЛАУ с помощью предобусловливателя Schur+OI, а также сравнение:

- с блочным предобусловливателем Якоби;
- с неполной блочной аппроксимацией обратной матрицы [7] (применительно к исходной матрице);
- с предобусловливателем Spike + OI.

Параметры  $LU(t)$ -разложений на процессах брались одинаковыми во всех методах. Итерации

проводились методом BiCGStab [11] с критерием точности  $\mu = \max_i \frac{\left| f_i - \sum_j a_{ij}x_j \right|}{|f_i| + \sum_j |a_{ij}x_j|} \leq 10^{-12}$ .

На рис. 4–6 приведены времена решения задач 1–3 в зависимости от числа процессов при использовании предобусловливателя Schur+OI. При увеличении числа процессов в 8 раз ускорение счета составляет от 3,5 до 6,3 раза.

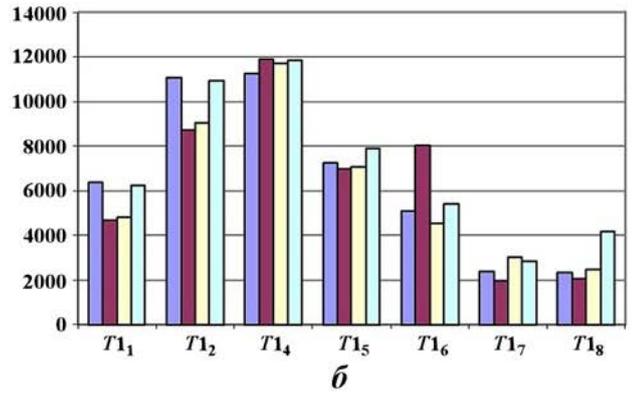
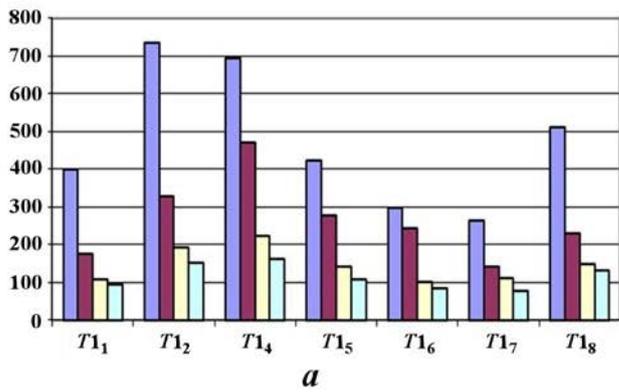


Рис. 4. Время решения ( $a$ ) и число итераций ( $b$ ) в задаче 1 при использовании метода Schur + OI на различном числе MPI-процессов: ■ 224; ■ 420; ■ 840; ■ 1680

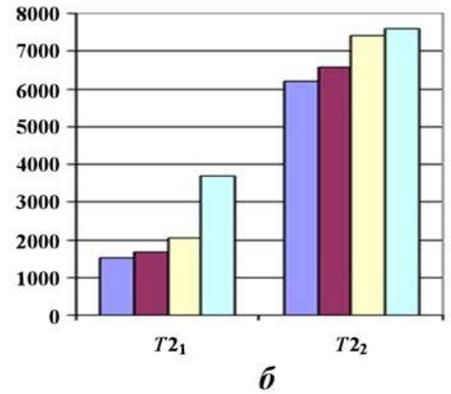
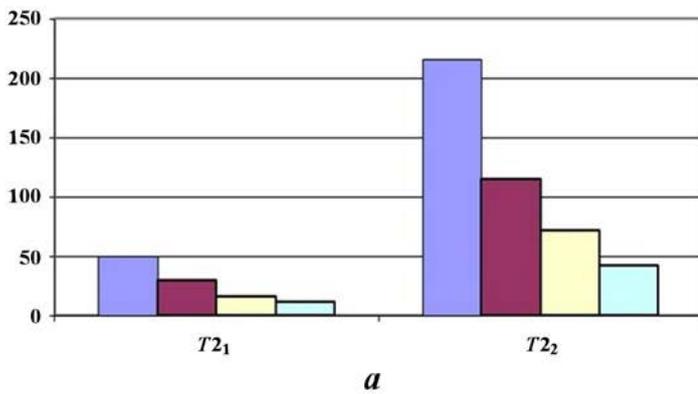


Рис. 5. Время решения ( $a$ ) и число итераций ( $b$ ) в задаче 2 при использовании метода Schur + OI на различном числе MPI-процессов: ■ 28; ■ 56; ■ 112; ■ 224

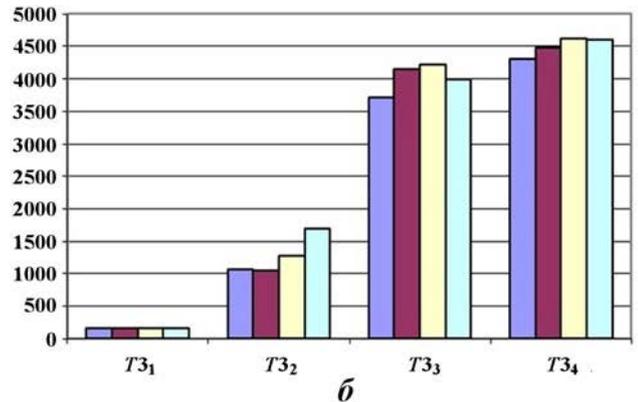
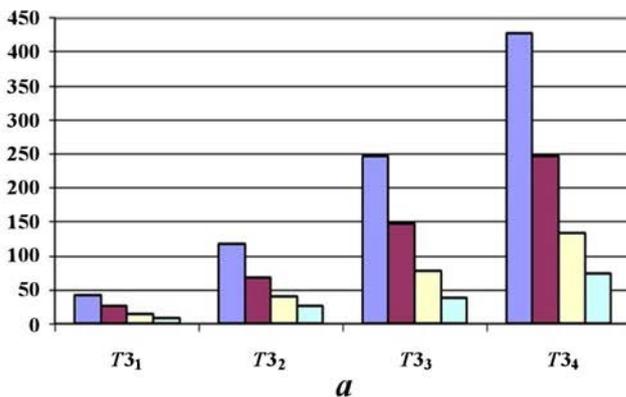


Рис. 6. Время решения ( $a$ ) и число итераций ( $b$ ) в задаче 3 при использовании метода Schur + OI на различном числе MPI-процессов: ■ 56; ■ 112; ■ 224; ■ 448

На рис. 7, 8 приведены сравнительные времена и число итераций при решении трехмерных задач с помощью всех четырех предобуславливателей на 420 процессорах. В подавляющем большинстве случаев лучшим оказался метод Schur+OI. Spike+OI в половине случаев считает так же, как и Schur+OI, в остальных — заметно медленнее. Блочный метод Якоби показал самые большие времена решения. Итерация с методом Spike+OI более затратна по времени, чем с методом Schur+OI

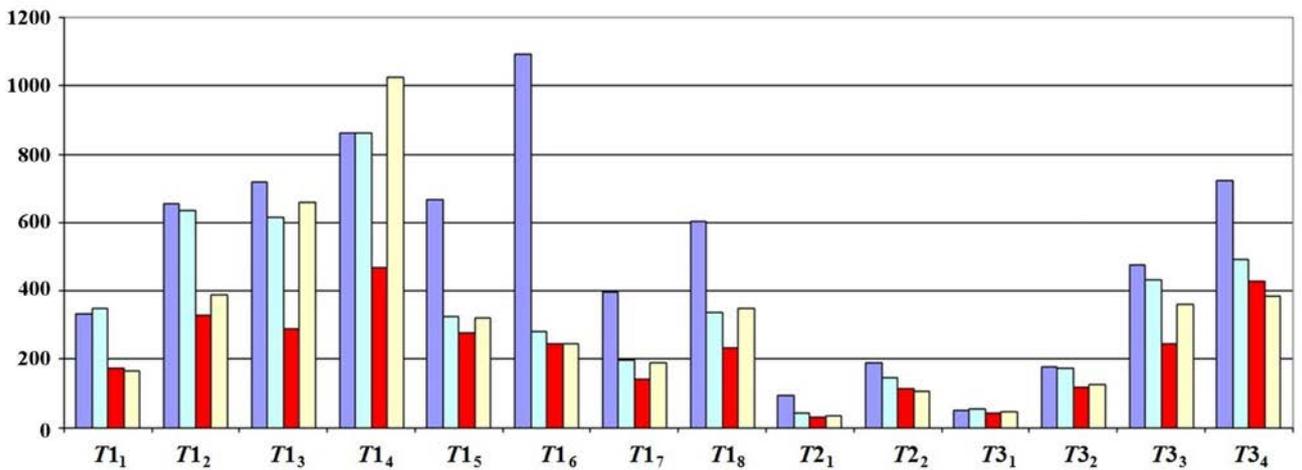


Рис. 7. Времена решения СЛАУ четырьмя методами в трехмерных задачах: ■ BJ; ■ Overlap Inverse; ■ Schur + OI; ■ Spike + OI

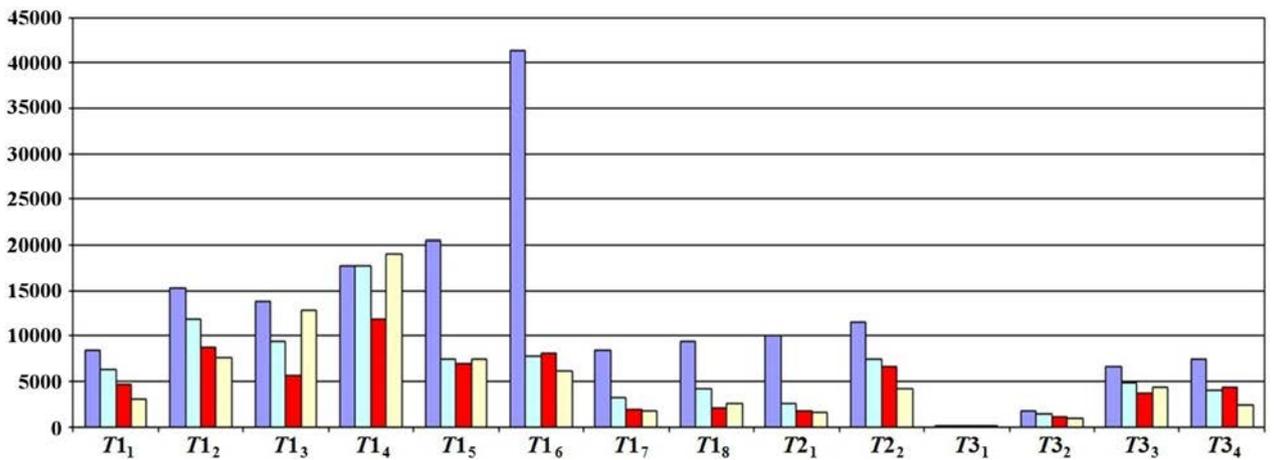


Рис. 8. Количество итераций при решении трехмерных задач: ■ BJ; ■ Overlap Inverse; ■ Schur + OI; ■ Spike + OI

(например, показательны расчеты задачи 1 на интервалах  $T1_2$  и  $T1_6$ ). Также стоит отметить, что отношение времени работы лучшего метода к худшему заметно меньше, чем в двумерных расчетах.

### Заключение

В статье приведено описание параллельного предобусловливателя, основанного на многоуровневом  $ILU(t)$ -разложении матриц дополнения Шура в комбинации с неполной блочной аппроксимацией обратной матрицы, а также предобусловливателя из семейства методов Spike. Приведены сравнительные результаты параллельного решения СЛАУ с помощью различных предобусловливателей на примере использования методов в блоке теплопроводности методики КОРОНА.

Во всех сериях расчетов блочный метод Якоби оказался явным аутсайдером. Предобусловливатели, основанные на редуцированных матрицах, в двумерных расчетах показали наилучшие и почти идентичные результаты. С переходом к трехмерной постановке при использовании метода на основе алгоритма Spike в части задач возникли проблемы со сходимостью итерационного процесса.

Эффективность распараллеливания метода Schur+OI в двумерном случае составила от 50 до 80% при росте числа процессов в 10 раз. В трехмерном случае с ростом числа процессов в 7-8 раз ускорение составило от 3,4 до 6,2 раза.

### Список литературы

1. *Saad Y.* Iterative Methods for Sparse Linear Systems. Second Edition. SIAM, Philadelphia, 2003.
2. *Milyukova O. Yu.* Parallel approximate factorization method for solving discrete elliptic equation // Parallel Computing. 2001. Vol. 27(10). P. 1365–1379.
3. *Karyps G., Kumar V.* Parallel Threshold-based ILU Factorization. Technical Report MN 55455. Minneapolis: University of Minnesota, Department of Computer Science; Army HPS Research Center, 1998.
4. *Hysom D., Pothen A.* A scalable parallel algorithm for incomplete factor preconditioning // SIAM J. Sci. Comput. 2001. Vol. 22(6). P. 2194–2215.
5. *Cai X. C., Saad Y.* Overlapping domain decomposition algorithms for general sparse matrices // Numerical Linear Algebra with Applications. 1996. Vol. 3(3). P. 221–237.
6. *Скрыпник С. И., Рудько Н. М., Королёв Р. А., Ляпин В. В., Попов А. В.* Численное решение двумерных уравнений газовой динамики с теплопроводностью с использованием переменного по размеру разностного шаблона // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2007. Вып. 1. С. 14–26.  
*Skrypnik S. I., Rudko N. M., Korolyev R. A., Lyapin V. V., Popov A. V.* Chislennoe reshenie dvumernykh uravneniy gazovoy dinamiki s teploprovodnostyu s ispolzovaniem peremennogo po razmeru raznostnogo shablona // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2007. Vyp. 1. S. 14–26.
7. *Kaporin I. E., Konshin I. N.* A parallel block overlap preconditioning with inexact submatrix inversion for linear elasticity problems // Numerical Linear Algebra with Applications. 2002. Vol. 9. P. 141–162.
8. *Чеботарь С. В., Сухих А. С.* Параллельный решатель систем линейных алгебраических уравнений "модуль NEWT". Его применение в методике КОРОНА // XVI Всерос. конф.-школа молодых исследователей "Современные проблемы математического моделирования". Абрау-Дюрсо, 14–19 сентября 2015 г. <https://durso.uginfo.sfedu.ru/sites/default/files/Presentation/Chebotar.pdf>.  
*Chebotar S. V., Sukhikh A. S.* Parallelnyy reshatel sistem lineynykh algebraicheskikh uravneniy "modul NEWT". Ego primeneniye v metodike KORONA // XVI Vseros. konf.-shkola molodykh issledovateley "Sovremennye problemy matematicheskogo modelirovaniya". Abrau-Dyurso, 14-19 sentyabrya 2015 g. <https://durso.uginfo.sfedu.ru/sites/default/files/Presentation/Chebotar.pdf>.
9. *Haidar A.* On the Parallel Scalability of Hybrid Linear Solvers for Large 3D Problems. Toulouse: CERFACS, 2008.
10. *Polizzi E., Sameh A.* A Parallel hybrid banded system solver: the SPIKE algorithm // Parallel Comput. 2006. Vol. 32(2). P. 177–194.
11. *Van Der Vorst H. A.* Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems // SIAM J. Sci. Statist. Comput. 1992. Vol. 13. P. 631–644.

Статья поступила в редакцию 20.03.23.