



Милюкова О.Ю.

Параллельные варианты
метода неполного
треугольного разложения
второго порядка
сопряженных градиентов на
основе использования
специального
переупорядочения матрицы
коэффициентов

Рекомендуемая форма библиографической ссылки: Милюкова О.Ю. Параллельные варианты метода неполного треугольного разложения второго порядка сопряженных градиентов на основе использования специального переупорядочения матрицы коэффициентов // Препринты ИПМ им. М.В.Келдыша. 2014. № 52. 32 с. URL: <http://library.keldysh.ru/preprint.asp?id=2014-52>

**Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М. В. Келдыша
Российской академии наук**

О. Ю. Милюкова

**Параллельные варианты метода
неполного треугольного разложения
второго порядка сопряженных
градиентов на основе использования
специального переупорядочения
матрицы коэффициентов**

Москва — 2014

Милюкова О.Ю.

Параллельные варианты метода неполного треугольного разложения второго порядка сопряженных градиентов на основе использования специального переупорядочения матрицы коэффициентов

Предлагаются параллельные варианты метода стабилизированного неполного треугольного разложения второго порядка сопряженных градиентов, которые основаны на использовании переупорядочения матрицы коэффициентов, соответствующего упорядочению узлов сетки, согласованному с разбиением области расчета. При этом выбор элементов матрицы предобуславливания осуществляется по значению во внутренних узлах подобластей и по значению и по позициям на разделителях. Проведены исследование скорости сходимости новых параллельных методов, теоретическое исследование безотказности предложенных методов, расчеты модельной задачи на параллельной вычислительной системе МВС 100К с использованием MPI-алгоритма.

Ключевые слова: неполная треугольная факторизация, параллельное предобуславливание, метод сопряженных градиентов

Olga Yurievna Milyukova

Parallel version of the 2nd order incomplete triangular factorization preconditioned conjugate gradient method using special matrix reordering

A parallel version of the stabilized 2nd order incomplete triangular factorization is considered for the use as a preconditioned for the conjugate gradient solution of linear algebraic systems with large sparse symmetric positive definite coefficient matrix. The reordering of the matrix is based on the use of certain domain decomposition type splitting which is further adjusted using a multi-level structure of separators. The incomplete factorization is organized using the truncation of fill-in “by position” at the separators and “by value” otherwise. Some theoretical results are given for the proposed method related, in particular, to its robustness. Numerical results are given obtained on the MVS-100K multiprocessor for the MPI implementation of the iterative linear solver.

Key words: incomplete triangular factorization, parallel preconditioning, conjugate gradient linear solver

Работа выполнена при финансовой поддержке целевой программы П-15 президиума РАН, ОМН РАН №3.

1. Введение

Одной из наиболее часто встречающихся трудоемких вычислительных задач является решение невырожденных систем линейных алгебраических уравнений $Ax=b$ с разреженной симметричной положительно определенной матрицей A . Для многих задач, например, отвечающих дискретным моделям на трехмерных сетках (в том числе неструктурированных), прямые методы треугольного разложения требуют возрастающего нелинейно по размеру задачи объема памяти, а также очень большого числа операций. Кроме того, эффективное распараллеливание соответствующих алгоритмов на большое число процессоров вызывает определенные трудности. Поэтому для задач с разреженными матрицами большого размера привлекательными являются итерационные методы, в частности метод сопряженных градиентов. Использование подходящего предобусловливания позволяет резко сократить число итераций метода. Часто в качестве матрицы предобусловливания используют факторизованную матрицу, представляемую в виде произведения $U^T U$ или $U^T D U$, где U - нижнетреугольная, U^T - верхнетреугольная и D - диагональная матрицы.

При решении многомерных задач на многопроцессорных вычислительных системах обычно используют подход, называемый декомпозицией области расчета. Основная трудность распараллеливания алгоритмов методов с факторизованной матрицей предобусловливания связана с рекурсивным характером вычислений при построении и применении предобусловливания. Для ее преодоления можно использовать переупорядочение узлов сетки. Так, например, в работах [1,2] предложены эффективные параллельные варианты методов ICCG(0) [3], MICCG(0) [4] (неполного разложения Холецкого и модифицированного неполного разложения Холецкого без заполнения сопряженных градиентов), методов GSSOR-CG [5] (обобщенной симметричной верхней релаксации сопряженных градиентов). В этих методах матрица предобусловливания имеет вид: $B=(D^{-1}+L)D(D^{-1}+L^T)$, где L - строго нижнетреугольная часть матрицы A , элементы диагональной матрицы D выбираются из условия совпадения диагональных элементов матриц B и A или критерия равенства строчных сумм $Ae+\sigma D_A e=Be$, где $e=(1,1,\dots,1)^T$, D_A - диагональная часть матрицы A , а выбор малого параметра $0\leq\sigma\ll 1$ зависит от конкретной задачи. При этом используются упорядочения, связанные с разбиением области расчета (DDO - Domain Decomposition ordering) [6]. Однако указанные выше методы и их параллельные варианты имеют ограниченную область применимости, теоретическое обоснование их применимости сделано для случая положительных диагональных, неотрицательных внедиагональных элементов матрицы и наличия диагонального преобладания.

В работе [7] используется упорядочение узлов сетки типа DDO для распараллеливания алгоритма метода с ILU предобусловливанием с отсечением

по позициям для решения пятиточечных уравнений на ортогональной сетке. В работе [8] используется упорядочение узлов сетки типа DDO для распараллеливания алгоритма метода DRIC-CG, проводится численное исследование скорости сходимости с помощью расчетов модельных двумерных задач на ортогональной сетке.

В работе [9] предложен метод, в котором матрица предобусловливания имеет вид: $B = (I + LZ)W^{-1}(I + ZL^T)$, где диагональные матрицы Z, W вычисляются из условия минимизации K -числа обусловленности обратной к предобусловленной матрицы. Этот метод применим для решения системы уравнения с произвольной симметричной положительно определенной матрицей A . Параллельный алгоритм метода, рассмотренный в [9], основан на использовании DDO, в значительной части аналогичен алгоритму из [2]. В методах [1,2,9] можно применять прием Айзенштата [10] для сокращения числа арифметических действий на каждой итерации.

Одним из эффективных и надежных методов решения систем линейных алгебраических уравнений $Ax = b$ с произвольной разреженной симметричной положительно определенной матрицей A является метод сопряженных градиентов с предобусловливанием IC2S(τ) (стабилизированного приближенного треугольного разложения 2-го порядка) [11]. В работах [12,13] предложена и исследована блочная версия неполного обратного треугольного разложения (ВПС2), в которой специальным образом строятся блоки с налеганием. Для построения предобусловливателя внутри блока используется IC2S(τ) разложение. При параллельной реализации расчеты для каждого блока производятся на своем процессоре. Метод показал хорошую эффективность для большого числа процессоров (несколько сотен, иногда даже тысяч процессоров для больших задач), однако требует настройки параметров, например глубины перекрытия.

В работах [14,15] распараллеливание алгоритма метода IC2S(τ)-CG осуществляется на основе построения для всей задачи упорядочения типа ND (вложенных сечений) [6], выделения на разделителях специального дополнительного блочного биения и априорного ограничения заполнения на разделителях для обеспечения параллельной эффективности. Однако для плохо обусловленных задач может иметь место сильный рост числа итераций с ростом числа блоков (процессов). В работе [16] представлен сквозной параллельный MPI-threads алгоритм построения неполного разложения второго порядка точности на основе динамического упорядочения и декомпозиции разделителей. При этом базой параллельных вычислений является одно и то же для разного числа процессов упорядочение вложенных сечений. При использовании алгоритма [16] не наблюдается роста числа итераций с ростом числа процессов, однако время вычисления предобусловливателя может быть велико и заметно уменьшить эффективность алгоритма. В работе [17] предложен параллельный алгоритм построения неполных регулярных разложений второго порядка точности на основе геометрической редукции, при

этом фактически используется упорядочение типа DDO, отличное от рассматриваемого в настоящей работе. Отмечается, что для некоторых задач время счета итерационного процесса велико из-за недостаточного качества предобусловливания.

Рассматриваемые в настоящей работе варианты метода IC2S(τ) используют некоторые идеи относительно построения приближенных треугольных факторизаций, предлагавшиеся в работах [18-23], однако эти технологии применяются в другом контексте, связанном с дополнительным требованием наличия крупноблочного параллелизма в получаемых алгоритмах. В настоящей работе предлагаются параллельные варианты стабилизированного метода IC2S(τ)-CG, который позволяет при меньшем заполнении предобусловливателя получить лучшее качество предобусловливания. Параллельные варианты этого метода основаны на использовании специального переупорядочения матрицы коэффициентов, соответствующего упорядочению узлов сетки (или вершин графа матрицы), согласованному с разбиением области расчета (DDO). Предложенные параллельные методы отличаются от стабилизированного IC2S(τ)-CG при использовании DDO обнулением некоторых элементов при построении предобусловливателя. Это позволяет преодолеть трудность распараллеливания, связанную с рекурсивным характером вычислений при построении и обращении предобусловливателя. Проведены исследование скорости сходимости новых параллельных методов, их безотказности, расчеты модельной задачи на параллельной вычислительной системе МВС 100К.

2. Алгоритм построения предобусловливателя в методе IC2S(τ)-CG

Предобусловливание IC2S(τ), являющееся более экономичной версией метода Тисменецкого [18], было описано в работе [24], а его матричная формулировка, обоснование, теоретический анализ, выбор параметров были даны в работе Капорина И.Е. [11]. Перед построением матрицы предобусловливания в методе IC2S(τ)-CG необходимо выполнить масштабирование матрицы A размера $n \times n$:

$$A_s = (\text{Diag}(A))^{-1/2} A (\text{Diag}(A))^{-1/2},$$

где $\text{Diag}(A)$ – диагональная часть матрицы A . Тогда матрица A_s имеет единичную диагональ. Далее вместо обозначения A_s будем использовать A .

Матричная схема метода предобусловливания стабилизированного IC2S(τ) имеет вид [11]

$$A = U^T U + U^T R + R^T U - S,$$

где U – верхнетреугольная матрица, R – строго верхнетреугольная матрица, $S = 2\tau^2 I + E_{JM}(\tau^2)$, матрица $E_{JM}(\tau^2) = (E_{JM}(\tau^2))^T \geq 0$ формируется, как было предложено в работе [20], $\|R\| = O(\tau)$ и $\|S\| = O(\tau^2)$. Иначе говоря, перед

факторизацией производится сдвиг диагонали с параметром $2\tau^2$ подобно тому, как ранее было предложено в работе [21], кроме того, значения v_{ij} , полученные в процессе факторизации

$$v_{ij} = u_{ii}u_{ij} + u_{ii}r_{ij} = a_{ij} - \sum_{k=1}^{i-1} u_{ki}u_{kj} - \sum_{k=1}^{i-1} u_{ki}r_{kj} - \sum_{k=1}^{i-1} r_{ki}u_{kj}, \quad j=i+1, \dots, n,$$

и меньшие по модулю $O(\tau^2)$, отбрасываются по методу Дженнингса-Малика [20] и их модули добавляются к диагональным элементам в строках с номерами i и j . В матрицу R включаются остальные «не слишком малые» значения v_{ij} : $\tau^2 < |r_{ij}| < \tau$. В качестве предобусловливателя используется матрица $U^T U$, матрица погрешности имеет вид: $A - U^T U = U^T R + R^T U - S$. Алгоритм вычисления стабилизированного треугольного разложения второго порядка имеет следующий вид [11]. Напомним, что $a_{ii} = 1$.

Алгоритм 1.

1. Инициализация вспомогательной диагональной матрицы:

```
for  $i=1, \dots, n$ 
   $d_i := 1 + 2\tau^2$ 
end for
```

2. Цикл по строкам A для вычисления строк U, R :

```
for  $i=1, \dots, n$ 
```

3. Инициализация вектора v при помощи i -той строки A :

```
for  $j=i+1, \dots, n$ 
   $v_j := a_{ij}$ 
end for
```

4. Цикл по уже вычисленным строкам U, R :

```
for  $s=1, \dots, i-1$ 
```

5. Сделать поправку к вектору v :

```
for  $j=i+1, \dots, n$ 
   $v_j := v_j - u_{si}u_{sj} - u_{si}r_{sj} - r_{si}u_{sj}$ 
end for
```

```
end for
```

6. Прореживание вектора v и выполнение поправки к d :

```
for  $j=i+1, \dots, n$ 
  if  $|v_j| \leq \tau^2 \sqrt{d_i}$  then
     $d_i := d_i + |v_j|$ 
     $d_j := d_j + |v_j|$ 
     $v_j := 0$ 
  endif
end for
```

7. Нормализация вектора:

```

 $u_{ii} := \sqrt{d_i}$ 
  for  $j=i+1, \dots, n$ 
     $v_j := v_j / u_{ii}$ 
  end for

```

8. Расщепление вектора v на i -тую строку U и i -тую строку R :

```

  for  $j=i+1, \dots, n$ 
    if  $|v_j| \geq \tau$  then
       $u_{ij} := v_j$ 
    else
       $r_{ij} := v_j$ 
    endif
  end for

```

9. Выполнение поправки к диагональной матрице d :

```

  for  $j=i+1, \dots, n$ 
     $d_j := d_j - u_{ij}^2$ 
  end for
end for

```

3. Параллельные варианты метода IC2S(τ)-CG

Для построения параллельных вариантов метода IC2S(τ)-CG разобьем произвольную трехмерную область расчета на подобласти каким-либо образом, выберем некоторую нумерацию подобластей и будем использовать упорядочение узлов сетки, связанное с разбиением – DDO [6]. Введем множество узлов разделителей – множество узлов сетки в подобластях, у которых имеются соседи из подобластей с б`ольшим номером. Остальные узлы сетки будем называть внутренними. Для облегчения понимания расположения внутренних узлов подобластей и узлов разделителей на рис. 1.а приведен пример разбиения двумерной области расчета на 4 подобласти. Цифрами обозначены номера подобластей. На рис. 1.б в подобластях выделены в рамочки узлы разделителей, остальные узлы – внутренние. Предполагается, что решается задача Дирихле для уравнения Пуассона на ортогональной сетке, используется стандартная пятиточечная аппроксимация, $n=36$.

Множество узлов разделителей разобьем на 3 части. Узел разделителя назовем узлом разделителя первого уровня, если в шаблоне этого узла нет узлов разделителей из других подобластей с номерами большими, чем номер рассматриваемой подобласти (2D размерности). Таким образом, в шаблонах узлов разделителей первого уровня могут быть только внутренние узлы из подобластей с тем же или большими (см. ниже) номерами и узлы разделителей из той же подобласти и подобластей с меньшими номерами. Используя рассуждения от противного, можно показать, что в шаблонах узлов

разделителей первого уровня не могут быть узлы разделителей первого уровня из других подобластей. Узел разделителя назовем узлом разделителя второго уровня, если в шаблоне этого узла нет узлов разделителей более высокого, чем первый уровень, расположенных в подобластях с большим номером. То есть в шаблоне узла разделителя второго уровня могут быть внутренние узлы из подобластей с тем же или большими (см. ниже) номерами, узлы разделителей первого уровня из подобластей с тем же или большими (см. ниже) номерами, а также узлы разделителей более высокого, чем первый уровень, но только из рассматриваемой подобласти и подобластей с меньшими номерами (1D размерности). Используя рассуждения от противного, можно показать, что в шаблонах узлов разделителей второго уровня не могут быть узлы разделителей второго уровня из других подобластей. Остальные узлы разделителей назовем узлами разделителей третьего уровня. Заметим, что отсутствие в шаблонах узлов разделителей узлов сетки, являющихся внутренними из подобластей с меньшими номерами, можно показать с помощью рассуждений от противного и использования определения узла разделителей. Отсутствие в шаблонах узлов разделителей второго уровня узлов разделителей первого уровня из подобластей с меньшими номерами можно показать с помощью рассуждений от противного и использования определения узла разделителей первого уровня. Для рассматриваемой двумерной задачи на рис. 1.в) узлы разделителей первого уровня обведены прямоугольными контурами, узел разделителей второго уровня обведен в кружок, узлы разделителей третьего уровня отсутствуют.

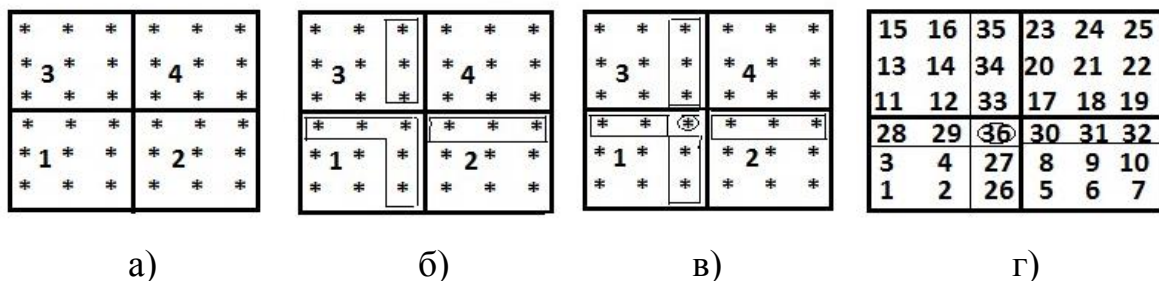


Рис. 1. Разбиение двумерной области расчета на 4 подобласти, расположение внутренних узлов подобластей и узлов разделителей и новый порядок следования узлов сетки для рассматриваемой двумерной задачи.

Вернемся к решению трехмерных задач. Установим следующий порядок следования узлов, отличный от введенного в работе [2]. Сначала идут все внутренние узлы подобластей в порядке следования номеров подобластей, причем сохраняется порядок следования узлов внутри каждой подобласти, введенный ранее. Затем идут узлы разделителей первого уровня в порядке следования номеров подобластей с сохранением порядка следования узлов внутри каждой подобласти, введенного ранее. Далее следуют узлы разделителей второго уровня в порядке следования номеров подобластей с сохранением ранее введенного порядка узлов внутри подобластей. И, наконец,

идут узлы разделителей третьего уровня в порядке следования номеров подобластей и с сохранением ранее введенного порядка следования узлов внутри каждой подобласти. Для рассматриваемой двумерной задачи на рис. 1. г) приведены номера узлов сетки после осуществления переупорядочения.

На рис. 2 приведен пример структуры разреженности матрицы A , полученной после перестановки строк и столбцов в результате переупорядочения в случае разбиения трехмерной области расчета на 4 подобласти. На рис. 2 и далее использованы обозначения: l – число всех внутренних узлов сетки из всех подобластей, $l1$ – число всех внутренних узлов сетки из всех подобластей и всех узлов разделителей первого уровня из всех подобластей, $l2$ – число всех внутренних узлов сетки из всех подобластей и

				l	l1	l2							
l	A_{11}^1	0			*	0	0	*	0	0	*	0	0
		A_{11}^2	0		*	*	0	*	*	0	*	*	0
			A_{11}^3	0		*	*	*	*	*	*	*	*
				A_{11}^4	*	*	*	*	*	*	*	*	*
l1	*	*	*	*	A_{22}^1	0		*	0	0	*	0	0
	0	*	*	*		A_{22}^2	0		*	*	0	*	*
	0	0	*	*	0		A_{22}^3	*	*	*	*	*	*
l2	*	*	*	*	*	*	*	A_{33}^1	0		*	0	0
	0	*	*	*	0	*	*		A_{33}^2	0		*	*
	0	0	*	*	0	0	*	0		A_{33}^3	0		*
	*	*	*	*	*	*	*	*	*	*	*	*	A_{44}

Рис. 2. Пример структуры матрицы A , полученной после перестановки столбцов и строк в результате переупорядочения.

всех узлов разделителей первого и второго уровня из всех подобластей. Строки матрицы, содержащие блочнодиагональные части A_{11}^k , соответствуют внутренним узлам сетки и хранятся в процессоре с номером k . Строки матрицы, содержащие блочнодиагональные части A_{22}^k , соответствуют узлам сетки на разделителях первого уровня из подобласти с номером k и хранятся в процессоре с номером k . Строки матрицы, содержащие блочнодиагональные части A_{33}^k , соответствуют узлам сетки на разделителях второго уровня из подобласти с номером k и хранятся в процессоре с номером k . Строки матрицы, соответствующие блочнодиагональной части A_{44} , соответствуют узлам сетки на разделителях третьего уровня и хранятся в соответствующих процессорах.

Рассмотрим алгоритмы построения матрицы U для параллельных вариантов метода IC2S(τ)-CG, основанных на использовании DDO. Методом математической индукции можно доказать, что в строках матриц U, R , соответствующих внутренним узлам, не могут присутствовать ненулевые значения в столбцах, соответствующих внутренним узлам из других подобластей. Поэтому вычисление элементов матрицы U в строках, соответствующих внутренним узлам подобластей, происходит во всех процессорах одновременно и независимо, каждый процессор производит вычисления в своей подобласти. Используется алгоритм 1 за исключением того, что циклы по i и по s происходят по внутренним узлам соответствующих подобластей. Перед переходом к вычислениям элементов матрицы U в строках, соответствующих узлам разделителей первого уровня, следует вычислить матрицу V , элементы которой определяются по формуле

$$V_{ij} = -\sum_{s=1}^l u_{si} u_{sj} - \sum_{s=1}^l u_{si} r_{sj} - \sum_{s=1}^l r_{si} u_{sj}, \quad i=l+1, \dots, n, j>i. \quad (1)$$

где l – общее число внутренних узлов, i, j – номера узлов разделителей. Сначала в каждом процессоре производится вычисление сумм слагаемых, которые были посчитаны в этом процессоре. Затем происходит суммирование вычисленных в разных процессорах матриц. На этом этапе необходимы обмены. Кроме того, производится суммирование по всем процессорам вычисленных там неокончательных значений d_j для номеров j , соответствующих номерам узлов разделителей, используется операция ALLREDUCE (sum).

Далее можно производить вычисление всех элементов матриц U, R для всех строк i , соответствующих узлам разделителей, во всех процессорах, причем одновременно, используя алгоритм 2.

Алгоритм 2.

1. for $i=l+1, \dots, n$
 $d_i := d_i + 1 + 2\sigma\tau^2$
 end for

Цикл по строкам A для вычисления строк U, R :

- for $i=l+1, \dots, n$
2. Инициализация вектора v :

```

for  $j=i+1, \dots, n$ 
   $v_j := a_{ij} + V_{ij}$ 
end for

```

3. Сделать поправку к вектору v :

Цикл по уже вычисленным строкам U, R :

```

for  $s = l+1, \dots, i-1$ 
  for  $j=i+1, \dots, n$ 
     $v_j := v_j - u_{si}u_{sj} - u_{si}r_{sj} - r_{si}u_{sj}$ 
  end for
end for

```

4. Прореживание вектора v и выполнение поправки к d :

```

for  $j=i+1, \dots, n$ 
  if  $|v_j| \leq \tau^2 \sqrt{d_i}$  then
     $d_i := d_i + |v_j|$ 
     $d_j := d_j + |v_j|$ 
     $v_j := 0$ 
  endif
end for

```

5. Нормализация вектора:

```

 $u_{ii} := \sqrt{d_i}$ 
for  $j=i+1, \dots, n$ 
   $v_j := v_j / u_{ii}$ 
end for

```

6. Расщепление вектора v на i -тую строку U и i -тую строку R :

```

for  $j=i+1, \dots, n$ 
  if  $|v_j| \geq \tau$  then
     $u_{ij} := v_j$ 
  else
     $r_{ij} := v_j$ 
  endif
end for

```

7. Выполнение поправки к диагональной матрице d :

```

for  $j=i+1, \dots, n$ 
   $d_j := d_j - u_{ij}^2$ 
end for

```

Конец цикла по строкам A :

```

end for

```

Затем для обеспечения хорошей параллелизуемости этапа обращения матрицы предобуславливания положить равными нулю значения u_{ij} , если уровень узла разделителя с номером j совпадает с уровнем разделителя узла i , но узлы с

номера i и j находятся в разных подобластях. Этот способ назовем способом 1. В процессе вычисления матрицы V по формуле (1) на этапе вычисления частичных сумм слагаемых, хранящихся в памяти соответствующего процессора (до суммирования по процессорам), можно отбрасывать маленькие по модулю значения V_{ij} и добавлять отброшенные значения $|V_{ij}|$ в d_i и d_j (как в пункте 6 алгоритма 1). Аналогичный подход был использован в работе [16]. Этот способ вычисления предобусловливателя назовем способом 1.а. Однако, следует ожидать большого времени вычисления матрицы предобусловливания при использовании способов 1 и 1.а.

Рассмотрим способы 2, 2.а вычисления матриц U, R , алгоритмы которых обладают значительно большей параллелизуемостью. Они отличаются от способов 1 только вычислением элементов строк матриц U, R , соответствующих узлам разделителей. Введем две целочисленные функции $g_1(i)$ и $g_2(i)$, определенные на множестве узлов разделителей, $g_1(i) = k$, где k – номер подобласти, содержащей узел с номером i , $g_2(i)$ – номер уровня разделителей для узла с номером i .

Перед переходом к вычислениям элементов матриц U, R в строках, соответствующих узлам разделителей первого уровня, следует вычислить матрицу V , не совпадающую с матрицей V в способах 1, 1.а, с использованием алгоритма 3 в способе 2 и алгоритма 4 в способе 2.а.

Алгоритм 3.

```

1. for  $k=1, \dots, p$ 
  for  $i=l+1, \dots, n$ 
    for  $j=i+1, \dots, n$ 
      if  $(g_2(i) = g_2(j)) \wedge (g_1(i) \neq g_1(j))$  then
         $V_{ij}^k := 0$ 
      else

$$V_{ij}^k := - \sum_{s=i_1(k)}^{i_2(k)} u_{si} u_{sj} - \sum_{s=i_1(k)}^{i_2(k)} u_{si} r_{sj} - \sum_{s=i_1(k)}^{i_2(k)} r_{si} u_{sj}$$

      end if
    end for
  end for
end for

```

2. $V_{ij} = \sum_{k=1}^p V_{ij}^k$ для $i=l+1, \dots, n, j=i+1, \dots, n$

Алгоритм 4.

```

1. for  $k=1, \dots, p$ 
  for  $i=l+1, \dots, n$ 
    for  $j=i+1, \dots, n$ 

$$V_{ij}^k := - \sum_{s=i_1(k)}^{i_2(k)} u_{si} u_{sj} - \sum_{s=i_1(k)}^{i_2(k)} u_{si} r_{sj} - \sum_{s=i_1(k)}^{i_2(k)} r_{si} u_{sj}$$

    end for
  end for

```

```

if  $|V_{ij}^k| \leq \tau^2 \sqrt{d_i}$  then
   $d_i := d_i + |V_{ij}^k|$ 
   $d_j := d_j + |V_{ij}^k|$ 
   $V_{ij}^k := 0$ 
end if
if  $(g_2(i) = g_2(j)) \wedge (g_1(i) \neq g_1(j))$  then
   $V_{ij}^k := 0$ 
end if
end for
end for
end for

```

$$2. V_{ij} = \sum_{k=1}^p V_{ij}^k \text{ для } i=l+1, \dots, n, j=i+1, \dots, n$$

В алгоритмах 3, 4 – k -номер подобласти, p - число подобластей, $i_1(k)$, $i_2(k)$ – номера первой и последней строк, соответствующих внутренним узлам в подобласти с номером k . Таким образом, при использовании алгоритмов 3, 4 матрица V имеет структуру, изображенную на рис. 4. Параллельная реализация п.1 алгоритмов 3,4 очевидна, каждый процессор производит вычисления в своей подобласти, пересылок не требуется. Затем (в п. 2) происходит суммирование вычисленных в разных процессорах матриц. На этом этапе необходимы обмены, его можно реализовать по-разному. Кроме того, производится суммирование по всем процессорам вычисленных там неокончательных значений d_j для номеров j , соответствующих номерам узлов разделителей.

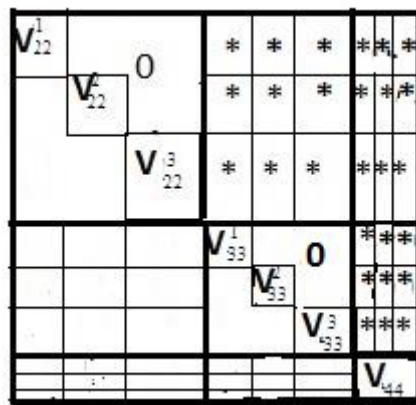


Рис. 4. Структура матрицы V , полученной при вычислении по алгоритмам 3, 4.
 В блоке V_{44} тоже блочно диагональная структура
 с нулевыми элементами вне блоков.

Остановимся на вычислении элементов матриц U, R в строках, соответствующих узлам разделителей первого уровня. Так как $V_{ij} = 0$ и $a_{ij} = 0$ для

i и j , соответствующих узлам разделителей первого уровня и находящихся в разных подобластях, то методом математической индукции можно доказать, что для этих значений ij $u_{ij} = 0$, $r_{ij} = 0$. Следовательно, вычисление элементов в строках матрицы U , соответствующих узлам разделителей первого уровня, может происходить во всех процессорах одновременно, каждый процессор будет производить вычисления в своей подобласти. Используется алгоритм аналогичный алгоритму 2, но с другими параметрами цикла по i : $i = \bar{i}_1(k), \dots, \bar{i}_2(k)$ и по s : $s = \bar{i}_1(k), \dots, i-1$, где $\bar{i}_1(k), \bar{i}_2(k)$ – номера первой и последней строк, соответствующих узлам разделителей первого уровня в подобласти с номером k . Перед переходом к вычислениям элементов матриц U, R в строках, соответствующих узлам разделителей второго уровня, следует вычислить матрицу \bar{V} . Элементы \bar{V} определяются с использованием алгоритма 3 в способе 2 или 4 в способе 2.а, в которых циклы по i начинаются с $l+1$, $i_1(k), i_2(k)$ – номера первой и последней строк, соответствующих узлам разделителей первого уровня в подобласти с номером k . Кроме того, нужно произвести суммирование по всем процессорам вычисленных там значений Δd_j (изменений значений d_j , полученных на этапах 4 и 7 алгоритма 2) для номеров j , соответствующих узлам разделителей второго и третьего уровня, и добавить Δd_j к вычисленным ранее неокончательным значениям d_j .

Вычисление элементов матриц U, R в строках, соответствующих узлам разделителей второго уровня, производится аналогично вычислению элементов в строках, соответствующих узлам разделителей первого уровня. Используется алгоритм аналогичный алгоритму 2, но с другими параметрами цикла по i : $i = \tilde{i}_1(k), \dots, \tilde{i}_2(k)$ и по s : $s = \tilde{i}_1(k), \dots, i-1$, где $\tilde{i}_1(k), \tilde{i}_2(k)$ – номера первой и последней строк, соответствующих узлам разделителей второго уровня в подобласти с номером k . Кроме того, в п. 2 алгоритма 2 $v_j = a_{ij} + V_{ij} + \bar{V}_{ij}$, причем $a_{ij} + V_{ij} + \bar{V}_{ij} = 0$, если $(g_2(i) = g_2(j)) \wedge (g_1(i) \neq g_1(j))$. Можно доказать, что для этих значений ij $u_{ij} = 0$, $r_{ij} = 0$. Следовательно, вычисление элементов матрицы U в строках, соответствующих узлам разделителей второго уровня, может происходить во всех процессорах одновременно. Перед переходом к вычислениям элементов матриц U, R в строках, соответствующих узлам разделителей третьего уровня, следует вычислить матрицу \tilde{V} , элементы которой также следует определять с использованием алгоритмов 3 способе 2 или 4 способе 2.а, где $i_1(k), i_2(k)$ – номера первой и последней строк, соответствующих узлам разделителей второго уровня в подобласти с номером k . Кроме того, нужно произвести суммирование по всем процессорам вычисленных там значений Δd_j для номеров j , соответствующих узлам разделителей третьего уровня, и добавить Δd_j к вычисленным ранее неокончательным значениям d_j .

Если блок A_{44} имеет блочно диагональную структуру с нулевыми элементами вне блоков, то вычисление элементов матриц U, R в строках, соответствующих узлам разделителей третьего уровня, производится аналогично. Только $v_j = a_{ij} + V_{ij} + \bar{V}_{ij} + \tilde{V}_{ij}$ в п. 2 алгоритма 2, причем обеспечивается $V_{ij} = 0, \bar{V}_{ij} = 0, \tilde{V}_{ij} = 0$ и $a_{ij} = 0$, если $(g_2(i) = 3) \wedge (g_1(i) \neq g_1(j))$. Если подматрица A_{44} не является блочно диагональной с нулевыми элементами вне блоков, то этот этап вычислений можно не распараллеливать, а производить вычисление элементов матриц U, R в строках, соответствующих узлам разделителей третьего уровня, во всех процессорах одновременно по одним и тем же формулам. Потом можно произвести отсечение некоторых элементов как в методах 1. Так как узлов разделителей третьего уровня относительно мало, этот этап вычислений должен происходить достаточно быстро.

Заметим, что при использовании способов 2 и 2.а вычисление элементов матриц U, R в строках, соответствующих узлам на разделителях, производится с отсечением не только по значению, но и по позициям. В строках, соответствующих внутренним узлам, вычисление элементов матриц U производится с отсечением только по значению. Заметим, что структура матрицы U в строках, соответствующих узлам разделителей, во всех четырех методах в случае блочно диагональной структуры A_{44} с нулевыми элементами вне блоков аналогична структуре матрицы V , приведенной на рис.4. Заметим, что даже если матрица A_{44} не имеет указанную структуру, то для обеспечения лучшей параллелизуемости можно все равно производить отсечение на разделителях третьего уровня не только по значению, но и по позициям, как бы заменив при этом некоторые значения a_{ij} нулем.

Рассмотрим теперь алгоритм параллельной реализации применения предобусловливателя к вектору r^{k-1} , состоящий из 2 этапов:

$$\hat{w}^k = U^{-T} r^{k-1},$$

$$w^k = U^{-1} \hat{w}^k.$$

Здесь r^{k-1} – невязка после выполнения $k-1$ -ой итерации, w^k – поправка на k -ой итерации. На этапе 1 сначала происходит вычисление значений \hat{w}^k во внутренних узлах подобластей, затем происходит пересылка найденных значений \hat{w}^k , необходимых для расчета значений \hat{w}^k в узлах разделителей первого уровня. Пересылка найденных значений \hat{w}^k осуществляется в процессоры, в которых эти значения нужны для вычисления \hat{w}^k в узлах разделителей первого уровня. Далее происходит вычисление \hat{w}^k в узлах разделителей первого уровня, пересылка в соответствующие процессоры найденных значений \hat{w}^k , необходимых для расчета значений \hat{w}^k в узлах разделителей второго уровня. Затем происходит вычисление \hat{w}^k в узлах разделителей второго уровня и пересылка в соответствующие процессоры найденных значений \hat{w}^k , необходимых для расчета значений \hat{w}^k в узлах

разделителей третьего уровня. И, наконец, происходит вычисление \hat{w}^k в узлах разделителей третьего уровня.

На этапе 2 сначала происходит вычисление w^k в узлах разделителей третьего уровня и пересылка найденных значений w^k , необходимых для расчета значений w^k в узлах разделителей второго уровня. Пересылка найденных значений w^k осуществляется в процессоры, в которых эти значения нужны для вычисления w^k в узлах разделителей второго уровня. Далее происходит вычисление w^k в узлах разделителей второго уровня и пересылка в соответствующие процессоры найденных значений w^k , необходимых для расчета значений w^k в узлах разделителей первого уровня. Затем происходит вычисление w^k в узлах разделителей первого уровня и пересылка в соответствующие процессоры найденных значений w^k , необходимых для расчета значений w^k во внутренних узлах подобластей. И, наконец, осуществляется вычисление w^k во внутренних узлах подобластей. Программная реализация вычисления предобусловливателя и его обращения осуществлена для случая хранения только матрицы U . Основная идея такого подхода описана в [13].

Параллельные варианты IC2S(τ)-CG, в которых элементы матрицы U вычисляются с использованием способа 1 и 1.а назовем PIC2S1-CG и PIC2S1-CG с дополнительным прореживанием. Параллельные варианты IC2S(τ)-CG, в которых элементы матрицы U вычисляются с использованием способа 2 и 2.а назовем PIC2S2-CG и PIC2S2-CG с дополнительным прореживанием.

4. Исследование безотказности предложенных параллельных методов

В настоящей работе под безотказностью метода (алгоритма) будем понимать корректность его выполнения, т. е. обеспечение выполнения неравенства $d_i > 0$ для всех значений $i=1, 2, \dots, n$.

Очевидно, что алгоритм метода PIC2S1-CG является безотказным.

Алгоритмы методов PIC2S1-CG с дополнительным прореживанием, PIC2S2-CG и PIC2S2-CG с дополнительным прореживанием в общем случае нельзя назвать безотказными. Рассмотрим этот вопрос более детально.

Обозначим V_{12} матрицу, состоящую из элементов $(V_{12})_{ij} = U_{ij} + R_{ij}$, где $1 \leq i \leq l$, $j > l$. Так как $V_{ij} = -\sum_{s=1}^l (u_{si} + r_{si})(u_{sj} + r_{sj}) + \sum_{s=il+1}^l r_{si}r_{sj}$, то при вычислении предобусловливателя на разделителях мы фактически осуществляем стабилизированное неполное треугольное разложение второго порядка матрицы $\bar{A}_{22} - V_{12}^T V_{12} + W + \bar{D}(\tau^2)$, где \bar{A}_{22} – часть матрицы A , состоящая из элементов a_{ij} , $i > l$, $j > l$ (i, j – номера узлов разделителей), $W_{ij} = \sum_{s=1}^l r_{si}r_{sj}$, $W = W^T > 0$, \bar{D} – неотрицательная диагональная матрица с положительными элементами

порядка не меньше $O(\tau^2)$. Матрица $\bar{A}_{22} - V_{12}^T V_{12} + W + \bar{D}(\tau^2)$ является симметричной. Будет ли она положительно определенной – это не очевидно, так как $-V_{12}^T V_{12} \leq 0$. При использовании дополнительного прореживания придется осуществлять стабилизированное неполное треугольное разложение второго порядка симметричной матрицы $\bar{A}_{22} - V_{12}^T V_{12} + W + \bar{D}(\tau^2) + \bar{E}_{JM}(\tau^2) \geq \bar{A}_{22} - V_{12}^T V_{12} + W + \bar{D}(\tau^2)$, где $\bar{E}_{JM}(\tau^2)$ симметричная положительно определенная матрица, появившаяся за счет дополнительного прореживания. Можно надеяться, что алгоритм метода PIC2S1-CG с дополнительным прореживанием будет безотказным.

При вычислении предобусловливателя на разделителях первого уровня в методе PIC2S2-CG мы фактически осуществляем стабилизированное неполное треугольное разложение второго порядка матрицы $\bar{A}_{22} - V_{12}^T V_{12} + W + \bar{D}(\tau^2) - \bar{V}$, где \bar{V} состоит из отброшенных элементов V (для некоторых ij по позициям). $\bar{V}_{ij} = \bar{V}_{ij} + \bar{W}_{ij}$, $\bar{V}_{ij} = -\sum_{s=1}^l (u_{si} + r_{si})(u_{sj} + r_{sj})$, $\bar{W}_{ij} = \sum_{s=1}^l r_{si} r_{sj}$, где $(i, j) \in \{l < i \leq n \wedge l < j \leq n\}$. Матрица $-\bar{W} \leq 0$ является симметричной, причем $\bar{W}_{ij} = O(\tau^2)$, а матрица $-\bar{V} \geq 0$ является симметричной с элементами большими, чем $O(\tau^2)$. Можно надеяться, что симметричная матрица $\bar{A}_{22} - V_{12}^T V_{12} + W + \bar{D}(\tau^2) - \bar{V} > \bar{A}_{22} - V_{12}^T V_{12} + W + \bar{D}(\tau^2)$. Аналогичные рассуждения справедливы для вычисления предобусловливателя на разделителях второго и третьего уровня. Можно надеяться, что алгоритм метода PIC2S2-CG будет безотказным. Можно надеяться, что алгоритм метода PIC2S2-CG с дополнительным прореживанием будет безотказным.

Чтобы в общем случае сделать алгоритмы предложенных параллельных методов безотказными, можно при отбрасывании элементов V, \bar{V}, \bar{V} , стоящих в неподходящих позициях, использовать прием Дженнингса-Малика [20], однако при наличии больших по модулю отбрасываемых элементов ухудшается качество предобусловливания и растет число итераций (см. раздел 6). Более подходящим вероятно является увеличение параметра σ при расчете на разделителях в алгоритмах, аналогичных алгоритму 2 (см. раздел 6).

Исследуем вопрос безотказности метода PIC2S1-CG с дополнительным прореживанием для класса матриц $A = A^T > 0$, элементы которых удовлетворяют условиям: $a_{ii} > 0$, $a_{ij} \leq 0, j \neq i$. Докажем ряд утверждений.

Утверждение 1. Пусть элементы матрицы $A = A^T > 0$ удовлетворяют условиям: $a_{ii} > 0$, $a_{ij} \leq 0, j \neq i$. Тогда при построении предобусловливателя IC2S(τ) получим: $u_{ii} > 0, d_i > 0$, $u_{ij} \leq 0, r_{ij} \leq 0, j \neq i$ при любом упорядочении узлов сетки. Кроме того, $u_{ii} > 0, d_i > 0$, $u_{ij} \leq 0, r_{ij} \leq 0, j \neq i$ в методе PIC2S1-CG.

Доказательство проводится методом математической индукции.

Утверждение 2. Пусть элементы матрицы $A = A^T > 0$ удовлетворяют условиям $a_{ii} > 0$, $a_{ij} \leq 0, j \neq i$. Тогда при построении предобусловливателя

способом 1.а при любом $i > l$ $|v_j|$ может не возрастает по сравнению с $|v_j|$ в способе 1, при любом $i > l$ d_i может не убывает по сравнению с d_i в способе 1.

Доказательство проводится с использованием метода математической индукции. Очевидно, что утверждение 2 верно для внутренних узлов сетки. Докажем, что оно верно при $i=l+1$. Действительно, при дополнительном прореживании (при вычислении матрицы V) в способе 1.а d_i может не убывает по сравнению с d_i в способе 1 при $i \geq l+1$, а $|V_{ij}|$ либо тот же, либо 0. Очевидно, в п.2,3 алгоритма 2 $|v_j|$ в способе 1.а могут не возрастает, по сравнению $|v_j|$ в способе 1. В п.4 в способе 1.а может произойти прореживание, однако вместе с дополнительным прореживанием до начала расчета на разделителях увеличение d_i, d_j будет больше, чем в способе 1, так как в способе 1.а это производится для слагаемых одного знака. Очевидно, что в п.5,6 $|v_j|, |u_{ij} + r_{ij}|, |u_{ij}|$ может не возрастает по сравнению с $|v_j|, |u_{ij} + r_{ij}|, |u_{ij}|$ в способе 1 (т.к. $u_{ij}r_{ij}=0$), следовательно, в п.7 d_j ($j > i$) может не убывает (по сравнению со способом 1).

Индуктивное предположение. Пусть для всех $i \leq i_0$, где $i_0 > l+1$, $|v_j|, |u_{ij} + r_{ij}|, |u_{ij}|$ ($j > i$) в способе 1.а может не возрастает по сравнению с $|v_j|, |u_{ij} + r_{ij}|, |u_{ij}|$ ($j > i$) в способе 1, а d_j при $j \geq i$ в процессе вычисления не убывает по сравнению с d_j в способе 1.

Шаг индукции. Пусть $i=i_0+1$. Очевидно, что в п.2 $|v_j|$ в способе 1.а не становится больше, чем в способе 1. В п. 3 имеем:

$$v_j = y - \sum_{s=l+1}^{i_0} (u_{si} + r_{si})(u_{sj} + r_{sj}) + \sum_{s=l+1}^{i_0} r_{si}r_{sj}, \quad j > i, \quad \text{где } y = a_{ij} + V_{ij} \leq 0.$$
 Очевидно, что если $u_{s,i_0+1} = 0$ и одновременно $u_{sj} = 0$, то $(u_{s,i_0+1} + r_{s,i_0+1})(u_{sj} + r_{sj}) - r_{s,i_0+1}r_{sj} = 0$. Если $s \in \Omega = \{s: l+1 \leq s \leq i_0, u_{s,i_0+1} \neq 0 \vee u_{sj} \neq 0\}$, то $r_{s,i_0+1}r_{sj} = 0$. Следовательно, $v_j = y - \bar{x}$, где

$$\bar{x} = \sum_{s \in \Omega} (u_{s,i_0+1} + r_{s,i_0+1})(u_{sj} + r_{sj}) \geq 0.$$
 Вследствие индуктивного предположения значение

$\bar{x} \geq 0$ может не возрастает по сравнению с значением $\bar{x} \geq 0$ в способе 1. Значит, при $i=i_0+1$ $|v_j| = |y - \bar{x}|$ не возрастает по сравнению с $|v_j| = |y - \bar{x}|$ в способе 1.

Следовательно, $|v_j|/d_i$ может не возрастает по сравнению с $|v_j|/d_i$ в способе 1.

Дополнительное прореживание и прореживание в п. 4 алгоритма 2 будут не меньше увеличивать d_i, d_j , чем в способе 1 (без дополнительного прореживания). Следовательно, в п.5, 6 $|v_j|, |u_{ij} + r_{ij}|, |u_{ij}|$ ($j > i$) не возрастает по сравнению с $|v_j|, |u_{ij} + r_{ij}|, |u_{ij}|$ ($j > i$) в способе 1, а в п. 7 а d_j при $j > i$ не убывает по сравнению с d_j в способе 1. Утверждение 2 доказано.

Из утверждения 2 и положительности d_i в методе PIC2S1-CG, следует, что в методе PIC2S1-CG с дополнительным прореживанием для любого i $d_i > 0$.

Значит, если элементы матрицы $A = A^T > 0$ удовлетворяют условиям $a_{ii} > 0$, $a_{ij} \leq 0, j \neq i$, то метод PIC2S1-CG с дополнительным прореживанием является безотказным.

Докажем безотказность метода PIC2S2-CG в случае, если элементы матрицы $A = A^T > 0$ удовлетворяют требованиям

$$a_{ii} > 0, \quad a_{ij} \leq 0, j \neq i, \quad a_{ii} \geq \sum_{j \neq i} |a_{ij}|, \quad 1 \leq i \leq n, 1 \leq j \leq n, \quad \exists i = i_0: a_{i_0 i_0} > \sum_{j \neq i_0} |a_{i_0 j}|. \quad (2)$$

Заметим, что обоснование возможности построения точного LU разложения для М-матриц сделано в [26]. Обоснование возможности треугольного разложения с отбрасыванием по позициям для Н-матриц приведено в [27].

Сначала рассмотрим, как происходит треугольное разложение симметричной положительно определенной матрицы $A = U_0^T U_0$. Предположим,

что используется DDO. На каждом m -ом шаге имеем матрицу $A_m = \begin{bmatrix} \alpha & a^T \\ a & \check{A}_m \end{bmatrix}$.

Строка матрицы U_0 с номером m имеет вид $(0, 0, \dots, \sqrt{\alpha}, a^T / \sqrt{\alpha})$, где $\sqrt{\alpha}$ стоит в позиции m , a – вектор длины $n-m$. Следующая активная подматрица \hat{A}_{m+1} (дополнение Шура), которую следует представить в виде произведения нижнетреугольной и верхнетреугольной матриц, имеет вид

$$\hat{A}_{m+1} = \check{A}_m - \frac{a a^T}{\alpha}.$$

Можно показать, что если элементы матрицы A_m удовлетворяют условиям (2), то элементы матрицы \hat{A}_{m+1} также удовлетворяют этим условиям, и $\hat{A}_{m+1} = \hat{A}_{m+1}^T > 0$. Так как $A_1 = A$, то для любого m элементы \hat{A}_{m+1} удовлетворяют условиям (2).

Для облегчения понимания изложения дальнейшего материала можно считать, что элементы a^T переупорядочены следующим образом: $a^T = (a_1^T, a_2^T, a_3^T)$, где $|(a_1^T)_j| \geq \tau \sqrt{\alpha}$, а $\tau^2 \sqrt{\alpha} < |(a_2^T)_j| < \tau \sqrt{\alpha}$, $0 \leq |(a_3^T)_j| \leq \tau^2 \sqrt{\alpha}$. В случае разложения IC2S(τ) матрицы A при DDO на каждом m -том шаге имеем

$$A_m = \begin{bmatrix} \alpha & a_1^T & a_2^T & a_3^T \\ a_1 & & & \\ a_2 & & \check{A}_m & \\ a_3 & & & \end{bmatrix}, \quad \text{где } \check{A}_m = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix},$$

а следующая активная подматрица имеет вид

$$\hat{A}_{m+1} = \begin{bmatrix} A_{11} - a_1 a_1^T / \alpha & A_{12} - a_1 a_2^T / \alpha & A_{13} \\ A_{21} - a_2 a_1^T / \alpha & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}.$$

Заметим, что все диагональные элементы матрицы \hat{A}_{m+1} (при исходной нумерации) не уменьшились по сравнению с диагональными элементами

матрицы $\hat{A}_{m+1} = \check{A}_m - \frac{aa^T}{\alpha}$ при точном разложении, т. к. $(a_1)_j \leq 0$ $(a_2)_j \leq 0$ в силу утверждения 1. Используя метод математической индукции и сравнение элементов матриц \hat{A}_{m+1} при IC2S(τ), и $\hat{A}_{m+1} = \check{A}_m - \frac{aa^T}{\alpha}$, можно доказать, что для любого m для матрицы \hat{A}_{m+1} при IC2S(τ) выполнены условия (2) и $\hat{A}_{m+1} = \hat{A}_{m+1}^T > 0$. Заметим, что безотказность алгоритма построения IC2S(τ) разложения матрицы $A = A^T > 0$ была доказана ранее [11].

Пусть теперь при вычислении предобусловливателя на разделителях кроме отсечения по значениям происходит отсечение по позициям (метод PIC2S2-CG). При вычислении предобусловливателя в строках, соответствующих внутренним узлам подобластей, алгоритм будет безотказным, т. к. при этом осуществляется ICS2(τ) разложение. Кроме того $u_{ii} > 0, d_i > 0, u_{ij} \leq 0, r_{ij} \leq 0, j \neq i$ для i , соответствующих внутренним узлам подобластей в силу утверждения 1. Следовательно, в результате применения алгоритма 3 $V_{ij} \leq 0$, а в первой строке разделителя первого уровня в каждой подобласти $v_j \leq 0$ (в алгоритме 2).

Теперь докажем, что в первой строке разделителя первого уровня каждой подобласти $u_{ii} > 0, d_i > 0$. При вычислении предобусловливателя в строках, соответствующих узлам разделителей, представим a^T следующим образом: $a^T = (a_1^T, a_2^T, a_3^T, a_4^T)$, где $(a_4^T)_j$ соответствуют тем значениям j , которые будут отбрасываться по позициям, а $|(a_1^T)_j| \geq \tau\sqrt{\alpha}$, $\tau^2\sqrt{\alpha} < |(a_2^T)_j| < \tau\sqrt{\alpha}$, $0 \leq |(a_3^T)_j| \leq \tau^2\sqrt{\alpha}$, причем j не принадлежит множеству индексов, для которых происходит отбрасывание по позициям. Тогда на каждом m -том шаге на разделителях

$$A_m = \begin{bmatrix} \alpha & a_1^T & a_2^T & a_3^T & a_4^T \\ a_1 & & & & \\ a_2 & & \check{A}_m & & \\ a_3 & & & & \\ a_4 & & & & \end{bmatrix}, \text{ где } \check{A}_m = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix},$$

а следующая активная подматрица имеет вид

$$\hat{A}_{m+1} = \begin{bmatrix} A_{11} - a_1 a_1^T / \alpha & A_{12} - a_1 a_2^T / \alpha & A_{13} & A_{14} \\ A_{21} - a_2 a_1^T / \alpha & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}.$$

При вычислении первой строки разделителя первого уровня в каждой подобласти все диагональные элементы матрицы \hat{A}_{m+1} не уменьшились по сравнению с диагональными элементами соответствующей активной подматрицы при точном разложении (и при ICS2(τ) разложении), т. к. $(a_1)_j \leq 0$ $(a_2)_j \leq 0$ в силу утверждения 1. Следовательно, в первой строке разделителя

первого уровня в каждой подобласти $u_{ii} > 0, d_i > 0, u_{ij} \leq 0, r_{ij} \leq 0, j \neq i$. Заметим, что все внедиагональные элементы A_m не положительны, так как до этого применялось ICS2(τ) разложение. Следовательно, все внедиагональные элементы матрицы \hat{A}_{m+1} не положительны, причем по модулю они не превосходят соответствующие элементы при точном разложении. Следовательно, для элементов матрицы \hat{A}_{m+1} выполнены условия (2), в том числе выполнено условие диагонального преобладания, $\hat{A}_{m+1} = \hat{A}_{m+1}^T > 0$, и дальнейшее разложение возможно. Это верно для первой строки разделителя в каждой подобласти.

Используя метод математической индукции, можно доказать, что в процессе вычисления на разделителях первого уровня в каждой подобласти для любого m для матрицы \hat{A}_{m+1} выполнены условия (2), $\hat{A}_{m+1} = \hat{A}_{m+1}^T > 0$, и дальнейшее разложение возможно. Кроме того, для каждого m диагональные элементы матрицы \hat{A}_{m+1} не убывают по сравнению с диагональными элементами активной подматрицы на соответствующем шаге при точном разложении и при ICS2(τ) разложении; а $u_{ij} \leq 0, r_{ij} \leq 0, j \neq i$ для каждой строки i , соответствующей узлу сетки из множества узлов разделителей первого уровня.

Аналогично доказывается, что при расчете на разделителях второго и третьего уровня для любого m $\hat{A}_{m+1} = \hat{A}_{m+1}^T > 0$, неравенства $u_{ii} > 0, d_i > 0, u_{ij} \leq 0, r_{ij} \leq 0, j \neq i$ верны для всех строк разделителей второго и третьего уровня, причем u_{ii} не меньше значения этого элемента при ICS2(τ) разложении. Следовательно, алгоритм метода PIC2S-CG в случае, если элементы матрицы $A = A^T > 0$ удовлетворяют условиям (2), является безотказным.

Заметим, что из приведенных рассуждений следует, что $\det(UA^{-1}U^T) \geq \det(U_1A^{-1}U_1^T)$, где $U_1^T U_1$ – матрица предобусловливателя метода IC2S(τ)-CG при использовании DDO, описанного выше, $U^T U$ – матрица предобусловливателя метода PIC2S2-CG.

Можно доказать, что если элементы матрицы $A = A^T > 0$ удовлетворяют условиям (2), то метод PIC2S2-CG с дополнительным прореживанием является безотказным.

Пусть теперь $U = U_1 - \Delta U$ – матрица, полученная способом 1, или 1.а, или 2, или 2.а. Докажем, что симметричная матрица $U^T U > 0$. Имеем:

$$\begin{aligned} (U^T U \bar{y}, \bar{y}) &= ([U_1 - \Delta U] \bar{y}, [U_1 - \Delta U] \bar{y}) = \|U_1 \bar{y}\|^2 + \|\Delta U \bar{y}\|^2 - 2(U_1 \bar{y}, \Delta U \bar{y}) \\ &\geq \|U_1 \bar{y}\|^2 + \|\Delta U \bar{y}\|^2 - 2|(U_1 \bar{y}, \Delta U \bar{y})| \end{aligned}$$

где $\bar{y} \neq 0$ – произвольный вектор. Учитывая неравенство Коши-Буняковского и неколлинеарность векторов $U_1 \bar{y}$ и $\Delta U \bar{y}$ имеем:

$(U^T U \bar{y}, \bar{y}) > \|U_1 \bar{y}\|^2 + \|\Delta U \bar{y}\|^2 - 2\|U_1 \bar{y}\| \|\Delta U \bar{y}\| \geq 0$. Следовательно, $U^T U > 0$, что гарантирует сходимость предобусловленного метода сопряженных градиентов.

5. Теоретическое исследование скорости сходимости методов PIC2S1-CG и PIC2S2-CG в частном случае

Проведем теоретическое исследование скорости сходимости метода PIC2S1-CG без дополнительного прореживания. В работах [1,2,6] было показано, что использование упорядочений типа DDO может привести к небольшому росту числа итераций методов ICCG(0), GSSORCG со стандартным (лексикографическим) упорядочением (в случае ортогональной сетки) и по сравнению с упорядочениями CM (Катхилл Макки) [6], RCM (обратного Катхилл Макки) [6] в случае неструктурированной сетки. В методе PIC2S1-CG кроме использования упорядочения типа DDO осуществляется обнуление некоторых элементов матрицы U . Посмотрим, как это влияет на скорость сходимости метода PIC2S1-CG.

Будем использовать оценку числа итераций через K -число обусловленности матрицы $UA^{-1}U^T$, где $U^T U$ - матрица предобусловливатель исследуемого метода. В работе [25] получена оценка погрешности предобусловленного метода сопряженных градиентов

$$\frac{\|x_k - x\|_A}{\|x_0 - x\|_A} \leq (K(M)^{2/k} - 1)^{k/2}, \quad n > k > 2 \log K(M),$$

где через $K(M) = \left(\frac{1}{n} \text{trace} M \right)^n / \det M$ обозначено K -число обусловленности матрицы $M = U^{-T} A U^{-1}$. В работе [9] приведена аналогичная оценка погрешности вида

$$\frac{\|x_k - x\|_A}{\|x_0 - x\|_A} \leq (K(M^{-1})^{2/k} - 1)^{k/2}, \quad n > k > 2 \log K(M^{-1}).$$

где $M^{-1} = UA^{-1}U^T$. Рассуждая аналогично [25], для числа итераций предобусловленного метода сопряженных градиентов, достаточного для уменьшения начальной нормы погрешности в ε^{-1} раз, можно вывести верхнюю оценку числа итераций вида

$$i_k \leq \bar{i}_k(\varepsilon) = \left\lceil \frac{2\tilde{\sigma} \log K(M^{-1}) + 2 \log(\varepsilon^{-1})}{\tilde{\sigma} \log \tilde{\sigma} - (\tilde{\sigma} - 1) \log(\tilde{\sigma} - 1)} \right\rceil,$$

где $\tilde{\sigma} \geq 2$. Можно показать, что коэффициент при $\log K(M^{-1})$ монотонно возрастает по $\tilde{\sigma}$, далее будем использовать оценку, получаемую при $\tilde{\sigma} = 2$,

$$i_k(\varepsilon) = \left\lceil 2 \log K(M^{-1}) + \log(\varepsilon^{-1}) \right\rceil. \quad (3)$$

Пусть $U_1^T U_1$ - матрица предобусловливатель метода IC2S(τ)-CG при использовании DDO, описанного выше, $U = U_1 - \Delta U$, где ΔU - строго верхнетреугольная матрица, состоящая из элементов, которые отбрасываются в конце вычисления предобусловливателя метода PIC2S1-CG. Первые l строк ΔU , соответствующие внутренним узлам всех подобластей, нулевые. Имеем

$$K(UA^{-1}U^T) = \left(\frac{1}{n} \text{trac} \mathfrak{e}((U_1 - \Delta U)A^{-1}(U_1^T - \Delta U^T)) \right)^n / \det(UA^{-1}U^T). \quad (4)$$

Поскольку $\Delta U_{ii} = 0$ для $i=1, \dots, n$, то

$$\det(UA^{-1}U^T) = \det(U_1A^{-1}U_1^T). \quad (5)$$

$$\text{trac} \mathfrak{e}((U_1 - \Delta U)A^{-1}(U_1^T - \Delta U^T)) = \text{trac} \mathfrak{e}(U_1A^{-1}U_1^T)(1 - 2\gamma_1 - \alpha^2), \quad (6)$$

где $\alpha = \sqrt{\text{trac} \mathfrak{e}(\Delta UA^{-1}\Delta U^T)} / \sqrt{\text{trac} \mathfrak{e}(U_1A^{-1}U_1^T)}$, $\gamma_1 = \text{trac} \mathfrak{e}(UA^{-1}\Delta U^T) / \text{trac} \mathfrak{e}(U_1A^{-1}U_1^T)$. Из формул (4), (5), (6) следует, что

$$K(UA^{-1}U^T) = \frac{\left(\frac{1}{n} \text{trac} \mathfrak{e}(U_1A^{-1}U_1^T)(1 - 2\gamma_1 - \alpha^2) \right)^n}{\det(U_1A^{-1}U_1^T)} = K(U_1A^{-1}U_1^T)(1 - 2\gamma_1 - \alpha^2)^n.$$

Коэффициент изменения оценки числа итераций имеет вид

$$\xi = \frac{2 \log K(UA^{-1}U^T) + \log(\varepsilon^{-1})}{2 \log K(U_1A^{-1}U_1^T) + \log(\varepsilon^{-1})} = \frac{2 \log K(U_1A^{-1}U_1^T) + \log(\varepsilon^{-1}) + n \log(1 - 2\gamma_1 - \alpha^2)}{2 \log K(U_1A^{-1}U_1^T) + \log(\varepsilon^{-1})}. \quad (7)$$

Если $\alpha^2 \ll 1$ и $|\gamma_1| \ll 1$, то $\log K(UA^{-1}U^T)$ не очень сильно отличается от $\log K(U_1A^{-1}U_1^T)$. Следовательно, ξ должно быть невелико. Если $\gamma_1 \geq 0$ или $\gamma_1 < 0$, но $-2\gamma_1 \leq \alpha^2$, то ξ будет даже меньше единицы.

Оценим коэффициент ξ в случае, если элементы матрицы $A = A^T > 0$ удовлетворяют условиям: $a_{ii} > 0$, $a_{ij} \leq 0, j \neq i$, $a_{ii} \geq \sum_{j \neq i} |a_{ij}|$, причем $A_{ij}^{-1} > 0$ для всех ij . Для этого, например, достаточно, чтобы все элементы матрицы $(A - \text{Diag} A)^n$ были положительны [28]. Пусть

$$\log K(U_1A^{-1}U_1^T) = O(n^\eta), \text{ где } \eta > 0. \quad (8)$$

Сначала оценим

$$\begin{aligned} \tilde{x} &= (-\gamma_1 - 0.5\alpha^2) \text{trac} \mathfrak{e}(U_1A^{-1}U_1^T) = -\text{trac} \mathfrak{e}(UA^{-1}\Delta U^T) - 0.5 \text{trac} \mathfrak{e}(\Delta UA^{-1}\Delta U^T) = \\ &= \sum_{i>l} \sum_{m>i} u_{ii} A_{im}^{-1} |\Delta u_{im}| - \sum_{i>l} \sum_{m>i} \sum_{k>i} (|u_{ik}| + 0.5|\Delta u_{ik}|) A_{km}^{-1} |\Delta u_{im}|. \end{aligned} \quad (9)$$

Очевидно, что

$$\tilde{x} \leq \sum_{i>l} \sum_{m>i} u_{ii} (A_{im}^{-1} - \min_{k>i} A_{km}^{-1} \mu_i) |\Delta u_{im}|, \text{ где } \mu_i = (\sum_{k>i} |u_{ik}| + 0.5|\Delta u_{ik}|) / u_{ii}.$$

Заметим, что $\mu_i > 0$, если существует хотя бы один элемент $\Delta u_{im} \neq 0$. Следовательно, для ненулевых слагаемых $\mu_i > 0$.

Известно, что в случае IC2S(τ) разложения при наложенных условиях на элементы матрицы $A = A^T > 0$ верны неравенства $u_{ii}^1 \geq \sum_{k>i} |u_{ik}^1|$ для всех i . Следовательно, т. к. $u_{ii}^1 = u_{ii}$, верно неравенство $u_{ii} \geq \sum_{k>i} |u_{ik}| + 0.5|\Delta u_{ik}|$ для всех $i > l$, и значит $\mu_i \leq 1$.

Предположим, что

$$\max_{i>l} \max_{m \in \Omega_i} (A_{im}^{-1} - \min_{k>i} A_{km}^{-1} \mu_i) \leq \varphi_1, \text{ где } \varphi_1 = O(n^\theta), \quad (10)$$

$$\Omega_i = \{m : (m > i > l) \wedge (\Delta u_{im} \neq 0) \wedge ((A_{im}^{-1} - \min_{k>i} A_{km}^{-1} \mu_i) > 0)\};$$

$$\sum_{i>l} \sum_{m \in \Omega_i} |\Delta u_{im}| \leq \varphi_2, \text{ где } \varphi_2 = O(n^{\theta_2}); \quad (11)$$

$$\theta_1 + \theta_2 < \eta, \quad (12)$$

где η определено в (8).

Из алгоритмов 1, 2 следует, что $u_{ii} \leq c_1$, где $c_1 = \text{const}$, $c_1 = O(1)$ для всех значений i . Учитывая (10), (11) и (12), получим

$$\tilde{x} \leq c_1 \varphi, \text{ где } \varphi = o(n^\eta). \quad (13)$$

Используя оценки максимального и минимального собственных значений матрицы $U_1^{-T} A U_1^{-1}$ [11], равенство $\lambda(U_1 A^{-1} U_1^T) = \lambda^{-1}(U_1^{-T} A U_1^{-1})$, где через $\lambda(M)$ обозначено собственное значение матрицы $M = M^T > 0$, можно доказать, что

$$\text{trac}(U_1 A^{-1} U_1^T) \geq c_3 n, \text{ где } c_3 = \text{const}, \quad c_3 = O(1). \quad (14)$$

Учитывая (7), (9), (14), (13) и неравенство $\log(1+y) < y$ при $y > 0$, получим

$$\xi \leq \frac{2 \log K(U_1 A^{-1} U_1^T) + \log(\varepsilon^{-1}) + c \varphi}{2 \log K(U_1 A^{-1} U_1^T) + \log(\varepsilon^{-1})}.$$

где $c = 2c_1/c_3 = \text{const}$, $c = O(1)$. Так как $\varphi = o(n^\eta)$, а $\log K(U_1 A^{-1} U_1^T) = O(n^\eta)$, то правая часть в последнем неравенстве больше 1, но для достаточно больших n правая часть в оценке ξ близка к 1. Итак, доказано

Утверждение 3. Предположим, что элементы матрицы $A = A^T > 0$ удовлетворяют условиям: $a_{ii} > 0$, $a_{ij} \leq 0, j \neq i$, $a_{ii} \geq \sum_{j \neq i} |a_{ij}|$, где $1 \leq i \leq n, 1 \leq j \leq n$,

причем $A_{ij}^{-1} > 0$ для всех ij . Пусть верны оценки (10), (11), (12). Тогда для достаточно больших n ξ – коэффициент возрастания оценки (3) числа итераций метода PIC2S1-CG по сравнению с оценкой (3) числа итераций метода IC2S(τ)-CG при использовании DDO, ограничен сверху положительным числом близким к единице.

Сформулированные в утверждении 3 достаточные условия означают, что элементы матрицы A^{-1} не очень сильно отличаются друг от друга хотя бы в некоторых определенных позициях, $|\Delta u_{im}| \neq 0$ достаточно малы, и их не слишком много.

Перейдем к исследованию скорости сходимости метода PIC2S2-CG без дополнительного прореживания для случая, когда элементы матрицы $A = A^T > 0$ удовлетворяют условиям: $a_{ii} > 0$, $a_{ij} \leq 0, j \neq i$, $a_{ii} \geq \sum_{j \neq i} |a_{ij}|$, где $1 \leq i \leq n, 1 \leq j \leq n$.

Пусть, кроме того, $A_{ij}^{-1} > 0$ для всех ij . Будем использовать оценку числа итераций (3), где $K(M^{-1}) = K(UA^{-1}U^T)$, $U^T U$ – матрица предобуславливателя метода PIC2S2-CG. Пусть $\Delta U = U - U_1$, где матрица U_1 , как и ранее, получена в процессе построения IC2S(τ) разложения матрицы A при DDO. Первые l строк ΔU , соответствующие внутренним узлам всех подобластей, нулевые. Теперь матрица ΔU может иметь ненулевые элементы на диагонали. Как доказано выше, $u_{ii}^1 \leq u_{ii}$ для всех i . Следовательно, $\Delta u_{ii} \geq 0$ для всех i . Выше доказано, что

$u_{im} \leq 0$ при $m > i$. Используя метод математической индукции, можно доказать, что $\Delta u_{im} \geq 0$ при $m > i$, $i > l$.

Предположим, что справедлива оценка (8). Ранее было доказано, что $\det(UA^{-1}U^T) \geq \det(U_1A^{-1}U_1^T)$. Поэтому

$$\begin{aligned} K(UA^{-1}U^T) &= \left(\frac{1}{n} \operatorname{trac} \alpha((U_1 + \Delta U)A^{-1}(U_1^T + \Delta U^T)) \right)^n / \det(UA^{-1}U^T) \leq \\ &\leq \left(\frac{1}{n} [\operatorname{trac} \alpha(U_1A^{-1}U_1^T) + 2\operatorname{trac} \alpha(U_1A^{-1}\Delta U^T) + \operatorname{trac} \alpha(\Delta UA^{-1}\Delta U^T)] \right)^n / \det(U_1A^{-1}U_1^T) = \quad (15) \\ &= K(U_1A^{-1}U_1^T) \left(1 + \frac{\operatorname{trac} \alpha(U_1A^{-1}\Delta U^T) + \operatorname{trac} \alpha(\Delta UA^{-1}\Delta U^T)}{\operatorname{trac} \alpha(U_1A^{-1}U_1^T)} \right)^n. \end{aligned}$$

Произведем оценку

$$\bar{x} = \operatorname{trac} \alpha(U_1A^{-1}\Delta U^T) + \operatorname{trac} \alpha(\Delta UA^{-1}\Delta U^T) = \bar{x}_1 + \bar{x}_2, \quad (16)$$

где

$$\begin{aligned} \bar{x}_1 &= \sum_{i>l} \sum_{m>i} (u_{ii} + u_{ii}^1) A_{im}^{-1} \Delta u_{im} - \sum_{i>l} \sum_{m>i} \sum_{k>i} (|u_{ik}^1| + |u_{ik}|) A_{km}^{-1} \Delta u_{im}, \\ \bar{x}_2 &= \sum_{i>l} (u_{ii} + u_{ii}^1) A_{ii}^{-1} \Delta u_{ii} - \sum_{i>l} \sum_{k>i} (|u_{ik}^1| + |u_{ik}|) A_{ki}^{-1} \Delta u_{ii}. \end{aligned}$$

Оценим $\bar{\mu}_i = \sum_{k>i} (|u_{ik}^1| + |u_{ik}|) / (u_{ii} + u_{ii}^1)$. Так как $|u_{ik}| \leq |u_{ik}^1|$, $u_{ii}^1 \leq u_{ii}$, то

$\sum_{k>i} (|u_{ik}| + |u_{ik}^1|) \leq \sum_{k>i} (|u_{ik}^1| + |u_{ik}^1|) \leq 2u_{ii}^1 \leq u_{ii} + u_{ii}^1$ и, следовательно $0 \leq \bar{\mu}_i \leq 1$. Если хотя бы для одного значения $m > i$ $|u_{im}| + |u_{im}^1| \neq 0$, то $\bar{\mu}_i > 0$.

Оценим \bar{x}_1 . Если $\Delta u_{im} \neq 0$ при $m > i$, то $|u_{im}^1| \neq |u_{im}|$ и $|u_{im}^1| + |u_{im}| \neq 0$, следовательно $\bar{\mu}_i > 0$. Тогда имеем

$$\bar{x}_1 \leq \sum_{i>l} \sum_{m \in \bar{\Omega}_i} (u_{ii} + u_{ii}^1) (A_{im}^{-1} - \min_{k>i} A_{km}^{-1} \bar{\mu}_i) \Delta u_{im} - \sum_{i>l} \sum_{m \in \bar{\Omega}_i} (u_{ii} + u_{ii}^1) (\min_{k>i} A_{km}^{-1} \bar{\mu}_i - A_{im}^{-1}) \Delta u_{im},$$

где $\bar{\mu}_i > 0$,

$$\begin{aligned} \bar{\Omega}_i &= \left\{ m : (m > i > l) \wedge (\Delta u_{im} \neq 0) \wedge ((A_{im}^{-1} - \min_{k>i} A_{km}^{-1} \bar{\mu}_i) > 0) \right\}, \\ \bar{\Omega}_i &= \left\{ m : (m > i > l) \wedge (\Delta u_{im} \neq 0) \wedge ((\min_{k>i} A_{km}^{-1} \bar{\mu}_i - A_{im}^{-1}) > 0) \right\}. \end{aligned} \quad (17)$$

Следовательно,

$$\bar{x}_1 \leq \sum_{i>l} \sum_{m \in \bar{\Omega}_i} (u_{ii} + u_{ii}^1) (A_{im}^{-1} - \min_{k>i} A_{km}^{-1} \bar{\mu}_i) \Delta u_{im}.$$

Из алгоритмов 1 и 2 следует, что

$$u_{ii} + u_{ii}^1 \leq \bar{c}_1, \quad \text{где } \bar{c}_1 = \text{const}, \bar{c}_1 > 0, \bar{c}_1 = O(1). \quad (18)$$

Предположим, что

$$\max_{i>l} \max_{m \in \bar{\Omega}_i} (A_{im}^{-1} - \min_{k>i} A_{km}^{-1} \bar{\mu}_i) = \bar{\varphi}_1, \quad \text{где } \bar{\varphi}_1 \leq O(n^{\bar{\eta}_1}), \quad (19)$$

$$\sum_{i>l} \sum_{m \in \bar{\Omega}_i} |\Delta u_{im}| \leq \bar{\varphi}_2, \quad \text{где } \bar{\varphi}_2 = o(n^{\bar{\eta}_2}), \quad (20)$$

где $\bar{\Omega}_i$ определено в (17), и, кроме того,

$$\bar{\eta}_1 + \bar{\eta}_2 < \eta. \quad (21)$$

Тогда, учитывая (18), получим

$$\bar{x}_1 \leq \bar{c}_1 \bar{\phi}_1 \bar{\phi}_2 = \bar{\phi}, \quad \text{где } \bar{\phi} = o(n^\eta). \quad (22)$$

Оценим \bar{x}_2 . Очевидно, что $\bar{x}_2 \leq \sum_{i>l} (u_{ii} + u_{ii}^1)(A_{ii}^{-1} - \min_{k>i} A_{ki}^{-1} \bar{\mu}_i) \Delta u_{ii}$. Заметим,

что если $\Delta u_{ii} \neq 0$, $\bar{\mu}_i$ все же может обращаться в ноль. Это происходит, когда для любого $m > i$ $|u_{im}| + |u_{im}^1| = 0$. Введем два множества индексов

$$\begin{aligned} \Psi &= \{i : (i > l) \wedge (\Delta u_{ii} \neq 0) \wedge (\bar{\mu}_i > 0) \wedge ((A_{ii}^{-1} - \min_{k>i} A_{ki}^{-1} \bar{\mu}_i) > 0)\}, \\ \bar{\Psi} &= \{i : (i > l) \wedge (\Delta u_{ii} \neq 0) \wedge (\bar{\mu}_i = 0)\}. \end{aligned} \quad (23)$$

Тогда

$$\bar{x}_2 \leq \sum_{i \in \Psi} (u_{ii} + u_{ii}^1)(A_{ii}^{-1} - \min_{k>i} A_{ki}^{-1} \bar{\mu}_i) \Delta u_{ii} + \sum_{i \in \bar{\Psi}} (u_{ii} + u_{ii}^1) A_{ii}^{-1} \Delta u_{ii}.$$

Предположим, что

$$\max_{i \in \Psi} (A_{ii}^{-1} - \min_{k>i} A_{ki}^{-1} \bar{\mu}_i) \leq \phi_1, \quad \text{где } \phi_1 = O(n^{\bar{\theta}_1}), \quad (24)$$

$$\sum_{i \in \Psi} \Delta u_{ii} \leq \phi_2, \quad \text{где } \phi_2 = O(n^{\bar{\theta}_2}), \quad (25)$$

и, кроме того,

$$\bar{\theta}_1 + \bar{\theta}_2 < \eta. \quad (26)$$

Предположим, что

$$\sum_{i \in \bar{\Psi}} A_{ii}^{-1} \Delta u_{ii} \leq \phi_3, \quad \text{где } \phi_3 = o(n^\eta). \quad (27)$$

Тогда, учитывая (18), имеем

$$\bar{x}_2 \leq \bar{c}_1 \phi_1 \phi_2 + \bar{c}_1 \phi_3 = \phi, \quad \text{где } \phi = o(n^\eta). \quad (28)$$

Используя (15), (16), (14), (22), (28), получим

$$\log K(UA^{-1}U^T) \leq \log K(U_1A^{-1}U_1) + n \frac{(\bar{\phi} + \phi)}{c_3 n} = \log K(U_1A^{-1}U_1) + 0.5\bar{\phi},$$

где $\bar{\phi} = o(n^\eta)$. Следовательно,

$$\bar{\xi} \leq \frac{2 \log K(U_1A^{-1}U_1^T) + \log(\varepsilon^{-1}) + \bar{\phi}}{2 \log K(U_1A^{-1}U_1^T) + \log(\varepsilon^{-1})}, \quad (29)$$

где $\bar{\xi}$ – коэффициент возрастания оценки числа итераций метода PIC2S2-CG по сравнению с оценкой числа итераций метода IC2S(τ)-CG при использовании DDO. Так как $\bar{\phi} = o(n^\eta)$, а $\log K(U_1A^{-1}U_1^T) = O(n^\eta)$, то для достаточно больших n правая часть в неравенстве (29) близка к единице и положительна. Итак, доказано

Утверждение 4. Предположим, что элементы матрицы $A = A^T > 0$ удовлетворяют условиям: $a_{ii} > 0$, $a_{ij} \leq 0, j \neq i$, $a_{ii} \geq \sum_{j \neq i} |a_{ij}|$, где $1 \leq i \leq n, 1 \leq j \leq n$,

причем $A_{ij}^{-1} > 0$ для всех ij . Пусть верны оценки (19)-(21), где Ω_i определено в (17), и верны оценки (24)-(27), где $\Psi, \bar{\Psi}$ определены в (23). Тогда для достаточно больших n $\bar{\xi}$ – коэффициент возрастания оценки (3) числа итераций метода PIC2S2-CG по сравнению с оценкой (3) числа итераций

метода IC2S(τ)-CG при использовании DDO, ограничен сверху положительным числом близким к единице.

Заметим, что в ряде случаев некоторые условия утверждения 4 можно ослабить. Так, например, условия (19)-(21) можно ослабить, если $\bar{x}_1^1 = \sum_{i>l} \sum_{m>i} (u_{ii} + u_{ii}^1) A_{im}^{-1} \Delta u_{im}$ и $\bar{x}_1^2 = \sum_{i>l} \sum_{m>i} \sum_{k>i} (|u_{ik}^1| + |u_{ik}|) A_{km}^{-1} \Delta u_{im}$ сопоставимы по величине, и $\bar{x}_1^1 - \bar{x}_1^2 = o(n^n)$. Аналогичные рассуждения верны относительно условий (24)-(26). Заметим, что элементов в множестве $\bar{\Psi}$ не должно быть много, при достаточно малых τ их будет конечное число.

Как уже сказано выше, использование упорядочения DDO может привести к небольшому росту числа итераций ряда методов, требующих для сходимости числа итераций, имеющего такой же характер асимптотической зависимости от числа узлов сетки, что и IC2S(τ)-CG. Если число итераций, полученных при применении предложенных параллельных методов, не сильно отличается от числа итераций метода IC2S(τ)-CG при использовании DDO, то следует ожидать, что число итераций при применении предложенных параллельных методов будет не очень сильно отличаться от числа итераций однопроцессорного метода IC2S(τ)-CG. В частности при выполнении достаточных условий утверждений 3,4 число итераций при применении методов PIC2S1-CG и PIC2S2-CG будет не очень сильно отличаться от числа итераций однопроцессорного метода IC2S(τ)-CG.

6. Результаты расчетов

Предложенные в настоящей работе параллельные варианты метода IC2S(τ)-CG решения систем линейных алгебраических уравнений $Ax=b$, где $A=A^T > 0$, были применены для решения трехмерной задачи Дирихле для уравнения Пуассона в единичной кубической области расчета на ортогональной сетке. Использовалась стандартная 7-точечная аппроксимация лапласиана. Матрица A хранилась в CRS формате. Для построения параллельных методов область расчетов разбивалась на кубические подобласти с одинаковым числом узлов сетки. Во всей области расчета и в каждой подобласти использовался лексикографический способ упорядочения узлов сетки. Для нумерации подобластей также применялся лексикографический порядок. Решалась система уравнения $Ax=b$, где $b_i \equiv 1$, начальное приближение $x_0 \equiv 0$, счет продолжался до выполнения условия: $\|Ax-b\| < \varepsilon \|b\|$, где $\varepsilon = 10^{-9}$, $\tau = 0.01$. Сдвиг на диагонали на величину $2\tau^2$ и $2\sigma\tau^2$ не осуществлялся.

Программа была написана на языке FORTRAN с использованием MPI, расчеты производились на многопроцессорной вычислительной системе МВС 100К, установленной на МСЦ РАН.

В таблице 1 приведены значения числа итераций, полученные в расчетах с использованием IC2S(τ)-CG с лексикографическим упорядочением узлов сетки (первый столбец), PIC2S1-CG (первое число), PIC2S2-CG (второе число)

для различного числа узлов сетки $n = N = N_h \times N_h \times N_h$ и различного числа процессоров. В скобках приведены значения числа итераций в методах с дополнительным прореживанием, если они отличаются от числа итераций в методах без дополнительного прореживания.

Как видно из таблицы 1, наблюдается крайне медленный рост числа итераций с ростом числа процессоров (кроме перехода от $p=1$ к $p=8$) при использовании методов PIC2S1-CG, PIC2S2-CG, а также этих методов с дополнительным прореживанием. Для метода PIC2S2-CG с дополнительным прореживанием в таблице 1 приведены числа итераций только для $N_h < 60$, $p < 216$. Числа итераций, полученные в расчетах всеми этими методами, не сильно отличаются при одинаковом числе неизвестных и одинаковом числе процессоров. Максимальное увеличение числа итераций было в 1.28 раза.

Таблица 1

Значения числа итераций, полученные при решении модельной задачи методами IC2S(τ)-CG, PIC2S1-CG (первое число), PIC2S2-CG (второе число), а также этими методами с дополнительным прореживанием

N_h	p=1	p=8	p=27	P=64	p=125	p=216
30	25	28, 28(30) (31*)	28, 29(31)	30(31),32(34) ($N_h = 32$)	28(29),29(32)	29, 29
40	32	36, 36(38) (74*)	36(37),38(40) ($N_h = 42$)	35, 36(40)	35(36),36(40)	38, 39 ($N_h = 42$)
50	39	44, 44(46) (159*)	40, 41(45) ($N_h = 48$)	41, 42(45) ($N_h = 48$)	43, 43(47)	42(43),43 ($N_h = 48$)
60	45	52, 52 (122*)	49, 49	50 50	50, 50 (299*)	51, 52
72		61, 61	57, 57	58 58	57, 57	60, 60

В таблице 1 в скобках и со звездочкой приведены значения числа итераций в методе PIC2S2-CG с дополнительным прореживанием и отбрасыванием элементов V, \bar{V}, \tilde{V} , стоящих в неподходящих позициях, по методу Дженнингса-Малика. Наблюдается сильное увеличение числа итераций, использование такого приема для обеспечения безотказности метода нецелесообразно. В таблице 2 приведены значения числа итераций, полученные при решении модельной задачи методом PIC2S2-CG с дополнительным прореживанием и использованием сдвига диагонали с параметром $2\sigma^2$ только на разделителях при различных значениях σ при $N_h = 60$. Даже при очень большом увеличении параметра σ на разделителях, число итераций почти не меняется.

Таблица 2

Значения числа итераций, полученные при решении модельной задачи методом PIC2S2-CG с дополнительным прореживанием и использованием сдвига диагонали на разделителях

σ	0	2	4	8	16	32	64	128
p=64	54	54	54	54	54	54	54	54
p=125	54	54	54	54	54	55	55	55

В таблице 3 приведены значения числа итераций метода PIC2S2-CG без дополнительного прореживания при решении описанной выше модельной задачи и использовании упорядочения узлов сетки и разбиения области расчета, предложенных в работе [13]. При этом подобласти строятся автоматически, могут иметь сложную форму, не быть односвязными. Как видно из таблицы 3, и при таком разбиении рост числа итераций с ростом числа процессоров достаточно медленный, максимальный коэффициент возрастания на 125 процессорах по сравнению с 8 процессорами был 1.25. Причем с ростом числа узлов сетки для одинакового числа процессоров коэффициент возрастания числа итераций уменьшается.

Таблица 3

Значения числа итераций, полученные при решении модельной задачи методом PIC2S2-CG для различных значений p и N при разбиении области расчета способом из работы [13]

	P=8	P=27	P=64	P=125
N=30*30*30	32	34	39	40
N=40*40*40	40	43	45	46
N=50*50*50	48	48	51	53

Время счета итерационного процесса измерялось при использовании в расчетах 2 ядер из 8 имеющихся в каждом узле МВС 100К и приведено в таблице 3. При этом использовалось разбиение области расчета, описанное в начале раздела 6. Результаты замера времени счета приведены в таблице 4.

Таблица 4

Время (в секундах) счета итерационного процесса в методе PIC2S1-CG при различных значениях N, p

	p=8	p=27	p=64	p=125
$N_h = 72$ (N=373248)	0.41	0.085	0.065	0.083
$N_h = 92$ (N=778688)		0.22	0.11	0.11

Хорошая масштабируемость итерационного процесса наблюдается при числе узлов в подобласти порядка 12000. Ускорение счета с ростом числа

процессоров замедляется главным образом за счет большого числа пересылок в процессе обращения матрицы предобусловливания.

Итак, в настоящей работе предложены параллельные варианты метода $IC2S(\tau)$ -CG, основанные на упорядочении узлов сетки типа DDO и выбора элементов матрицы предобусловливания по значению во внутренних узлах сетки, а на разделителях – по значению и по позициям. Доказана безотказность метода PIC2S1-CG, а также безотказность методов PIC2S1-CG с дополнительным прореживанием и PIC2S2-CG для определенного класса матриц A . Доказано, что для определенного класса матриц A при выполнении некоторых достаточных условий число итераций при использовании методов PIC2S1-CG и PIC2S2-CG должно не очень сильно отличаться от числа итераций однопроцессорного метода $IC2S(\tau)$ -CG. Это подтверждается результатами расчетов модельной задачи. С помощью расчетов модельной задачи показано, что при умеренном числе процессоров наблюдается хорошая масштабируемость при достаточном числе неизвестных.

Автор выражает благодарность Капорину Игорю Евгеньевичу за проявленное внимание к работе, ценные советы и замечания.

Список литературы

1. Milyukova O. Yu. Parallel approximate factorization method for solving discrete elliptic equations // *Parallel Computing*. 2001. № 27. – P. 1365-1379.
2. Милюкова О.Ю. Некоторые параллельные итерационные методы с факторизованными матрицами предобусловливания для решения эллиптических уравнений на треугольных сетках // *Ж. вычисл. матем. и матем. физики*. 2006. Т.46. № 6. – С. 1096-1112.
3. Meijerink J. A., van der Vorst H. A. An Iterative Solution Method for Linear Systems, of which the Coefficient Matrix is a Symmetric M-matrix // *Math. Comp.* 1977. V. 31. № 137. – P. 148-162.
4. Gustafsson I. A Class of First Order Factorization Methods // *BIT*. 1978. V.18. – P. 142-156.
5. Axelsson O. A generalazed SSOR method. // *BIT*. 1972. V.13. – P. 443-467.
6. Duff I.S., Meurant G.A. The effect of ordering on preconditioned conjugate gradients // *BIT*. 1989. V. 29. – P. 625-657.
7. Doi S. On parallelism and convergence of incomplete LU factorizations // *Applied Numerical Mathematics: Transactions of IMACS*. 1991. V. 7. № 5. – P. 417–436.
8. Notay Y. An efficient parallel discrete PDE solver // *Parallel Computing*. 1995. V. 21. – P. 1725-1748.
9. Капорин И.Е., Милюкова О.Ю. Оптимизация факторизованных предобусловливаний метода сопряженных градиентов для решения систем линейных алгебраических уравнений с симметричной положительно определенной матрицей // *Препринты ИПМ им. М.В.Келдыша*. 2013. № 13. 17 с. URL: <http://library.keldysh.ru/preprint.asp?id=2013-13>

10. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем. – М.: Мир. 1991. – 365 с.
11. Kaporin I.E. High quality preconditionings of a general symmetric positive definite matrix based on its $U^T U + U^T R + R^T U$ – decomposition. Numer. Lin. Alg. Appl. 1998. V. 5. – P.483-509.
12. Капорин И. Е., Коньшин И. Н. Параллельное решение симметричных положительно-определенных систем на основе перекрывающегося разбиения на блоки // Ж. вычисл. матем. и матем. физики. 2001. Т. 41. № 4. – С. 515–528.
13. Капорин И.Е., Милюкова О.Ю. Массивно-параллельный алгоритм предобусловленного метода сопряженных градиентов для численного решения систем линейных алгебраических уравнений // Сб. трудов отдела проблем прикладной оптимизации ВЦ РАН (под ред. В.Г. Жадана). М.: Из-во ВЦ РАН. 2011. – С. 32-49.
14. Харченко С.А. Влияние распараллеливания вычислений с поверхностными межпроцессорными границами на масштабируемость параллельного итерационного алгоритма решения систем линейных уравнений на примере уравнений вычислительной гидродинамики. Материалы Международной научной конференции «Параллельные вычислительные технологии» (ПаВТ'2008). Санкт-Петербург, 28 января – 1 февраля 2008г. Челябинск. Изд. ЮУрГУ, 2008. – С. 494-499.
15. Сушко Г.Б., Харченко С.А. Многопоточная параллельная реализация итерационного алгоритма решения систем линейных уравнений с динамическим распределением нагрузки по нитям вычислений. Труды Международной научной конференции «Параллельные вычислительные технологии» (ПаВТ'2008), Санкт-Петербург, 28 января – 1 февраля 2008 г. Челябинск. Изд. ЮУрГУ. 2008. – С. 452-457.
16. Коньшин И.Н., Харченко С.А. Сквозной параллельный алгоритм построения неполного треугольного разложения второго порядка точности с динамическим выбором декомпозиции области и упорядочения. 2008 г. Челябинск. Изд. ЮУрГУ, 2008. – С. 491-494.
17. Якушев В.Л., Симбиркин В.Н., Филимонов А.В., Новиков П.А., Коньшин И.Н., Сушко Г.Б., Харченко С.А. Решение плохообусловленных симметричных СЛАУ для задач строительной механики параллельными итерационными методами. Вестник Нижегородского университета им. Н.И. Лобачевского. 2012. № 4(1) – С. 238-246.
18. Tismenetsky M. A new preconditioning technique for solving large sparse linear systems// Linear Algebra Appls. 1991. V. 154-156. – P.331-353.
19. Tuff A. D., Jennings A. An iterative method for large systems of linear structural equations // International Journal for Numerical Methods in Engineering. 1973. V.7. № 2. – P. 175–183.
20. Jennigs A. Malik G.M. Partial elimination // J.Inst. Math. Appl. 1977. V. 20. – P. 307-316.

21. Manteuffel T.A. An incomplete factorization technique for positive definite linear systems// *Math.Comput.* 1980. V. 34. – P. 473-497.
22. Yamazaki I., Bai Z., Chen W. and Scalettar R. A High-Quality Preconditioning Technique for Multi-Length-Scale Symmetric Positive Definite Linear Systems// *Numer. Math. Theor. Meth. Appl.* 2009. V.2. № 4. – P. 469-484.
23. Scott J., Tuma M. On positive semidefinite modification schemes for incomplete Cholesky factorization. // *SIAM J. Sci. Comput.* 2014. V. 36. №2. – P. A609–A633. <http://epubs.siam.org/doi/abs/10.1137/130917582>
24. Suarjana M., Law K.H. A robust incomplete factorization based on value and space constraints// *Int. J. Numer. Methods Engrg.* 1995. V. 38. – P. 1703-1719.
25. Axelsson O., Kaporin I. On the sublinear and superlinear rate of convergence of conjugate gradient methods // *Numerical Algorithms.* 2000, № 25. – P. 1–22.
26. Meijerink J.A., van der Vorst A.H. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.* 1977. V.31. – P. 148-162.
27. Varga R.S., Saff E.B., Mehrman V. Incomplete factorization of matrices and connection with h- matrices. *SIAM J. Numer. Anal.* 1980. V.17. – P. 787-793.
28. Воеводин В.В., Кузнецов Ю.А. Матрицы и вычисления. – М.: Наука. 1984. – 320 с.

Оглавление

1.	Введение.....	3
2.	Алгоритм построения предобусловливателя в методе IC2S(τ)-CG.....	5
3.	Параллельные варианты метода IC2S(τ)-CG	7
4.	Исследование безотказности предложенных параллельных методов	16
5.	Теоретическое исследование скорости сходимости методов PIC2S1-CG и PIC2S2-CG в частном случае	22
6.	Результаты расчетов	27
	Список литературы.....	30