



Милюкова О.Ю.

Сочетание числовых и структурных подходов к построению неполного треугольного разложения второго порядка в параллельных алгоритмах предобусловленного метода сопряженных градиентов

**Рекомендуемая форма библиографической ссылки:** Милюкова О.Ю. Сочетание числовых и структурных подходов к построению неполного треугольного разложения второго порядка в параллельных алгоритмах предобусловленного метода сопряженных градиентов // Препринты ИПМ им. М.В.Келдыша. 2015. № 10. 32 с. URL: <http://library.keldysh.ru/preprint.asp?id=2015-10>

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

**О. Ю. Милюкова**

**Сочетание числовых и структурных  
подходов к построению неполного  
треугольного разложения второго  
порядка в параллельных алгоритмах  
предобусловленного метода  
сопряженных градиентов**

**Москва — 2015**

*Милюкова О.Ю.*

**Сочетание числовых и структурных подходов к построению неполного треугольного разложения второго порядка в параллельных алгоритмах предобусловленного метода сопряженных градиентов**

Рассматриваются предложенные ранее и предлагаются новые параллельные варианты метода стабилизированного неполного треугольного разложения второго порядка, предназначенные для предобусловливания метода сопряженных градиентов. При этом используется переупорядочение матрицы коэффициентов, соответствующее упорядочению типа расщепления на подобласти с разделителями. Построение неполного треугольного разложения осуществляется отсечением по значению во внутренних узлах подобластей, а на разделителях используется отсечение как по значению, так и по позициям. Проводится исследование безотказности и скорости сходимости рассматриваемых параллельных методов. Построенные алгоритмы реализованы с использованием MPI, приводятся результаты расчетов тестовых задач с матрицами из коллекции университета Флориды.

**Ключевые слова:** итерационное решение систем линейных алгебраических уравнений, разреженные матрицы, неполная треугольная факторизация, параллельное предобусловливание, метод сопряженных градиентов

*Olga Yurievna Milyukova*

**Combining incomplete factorization strategies “by value” and “by position” for the 2<sup>nd</sup> order incomplete triangular factorization in parallel algorithms for the preconditioned conjugate gradient method**

Some earlier and newly developed parallel versions of the stabilized 2<sup>nd</sup> order incomplete triangular factorization are considered as preconditioners 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 with separators. The incomplete factorization is organized using the truncation of fill-in “by value” within the subdomains and “by position” and “by value” at the separators. Some theoretical results are given for the proposed preconditionings related to its quality and robustness. For an MPI implementation of the iterative linear solver, numerical results are given obtained for matrices from the University of Florida collection.

**Key words:** iterative linear solvers, sparse matrices, incomplete triangular factorization, parallel preconditioning, conjugate gradient method

Работа выполнена при финансовой поддержке целевой программы 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 решения задач с разреженными симметричными положительно определенными матрицами большого размера целесообразно использовать итерационные методы, в частности предобусловленный метод сопряженных градиентов. Часто в качестве матрицы предобусловливания используют факторизованную матрицу, представляемую в виде произведения  $U^T U$  или  $U^T D U$ , где  $U$  - нижнетреугольная,  $U^T$  - транспонированная к  $U$  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 предобусловливателя. Для ее преодоления можно использовать переупорядочение узлов сетки и соответствующую ему перестановку строк и столбцов матрицы. 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 [4,1]. 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 работах [7] и [8] используются упорядочения узлов сетки типа DDO для распараллеливания алгоритмов метода с предобусловливанием ILU и метода DRIC-CG для решения уравнений, полученных в результате аппроксимации задачи 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 матрицы, обратной к предобусловленной. 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].

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( $\tau$ ) (стабилизированного приближенного треугольного разложения 2-го порядка) [10]. В работах [11,12] предложена и исследована блочная версия неполного обратного треугольного разложения (BIC2), в которой специальным образом строятся блоки с налеганием. Для построения предобусловливателя внутри блока используется IC2S( $\tau$ ) 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 [13,14] распараллеливание алгоритма метода IC2S( $\tau$ )-CG осуществляется на основе построения для всей задачи упорядочения типа ND (вложенных сечений, nested dissection) [6]. Однако при использовании подходов из работ [13,14] для плохо обусловленных задач может иметь место сильный рост числа итераций с ростом числа блоков (процессов). В работе [15] представлен сквозной параллельный MPI-threads алгоритм построения неполного разложения второго порядка точности на основе динамического упорядочения и декомпозиции разделителей, при использовании которого не наблюдается роста числа итераций с ростом числа процессов. Однако время вычисления предобусловливателя при этом может быть велико и заметно уменьшить эффективность алгоритма [15]. В работе [16] 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.

В работе [17] предлагаются параллельные варианты метода IC2S( $\tau$ )-CG, основанные на использовании специального переупорядочения матрицы коэффициентов, соответствующего упорядочению узлов сетки (или вершин графа матрицы), согласованного с разбиением области расчета (DDO). Предложенные в [17] параллельные методы, в частности методы PIC2S1-CG (параллельного стабилизированного приближенного треугольного разложения второго порядка 1 сопряженных градиентов) и PIC2S2-CG (параллельного стабилизированного приближенного треугольного разложения второго порядка 2 сопряженных градиентов), отличаются от метода IC2S( $\tau$ )-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.

В настоящей работе проводится дальнейшее исследование методов PIC2S1-CG и PIC2S2-CG, предложены еще три новых параллельных варианта метода IC2S( $\tau$ )-CG. Все пять параллельных методов основаны на использовании специального переупорядочения матрицы коэффициентов, соответствующего упорядочению узлов сетки, согласованному с разбиением области расчета (DDO), предложенного в [17]. Сформулированы теоретические результаты [17] по исследованию безотказности методов PIC2S1-CG, PIC2S2-CG, теоретически доказана безотказность одного из предложенных в настоящей работе методов в случае произвольной матрицы  $A=A^T > 0$ . С помощью решения ряда задач из Флоридской коллекции [18] проведено численное исследование безотказности и скорости сходимости рассматриваемых параллельных вариантов IC2S( $\tau$ )-CG в случае, когда матрица системы уравнений не удовлетворяет достаточным условиям безотказности метода PIC2S2-CG [17] и достаточным условиям утверждений из работы [17] о небольшом росте числа итераций с ростом числа процессоров.

## 2. Алгоритм построения предобусловливателя в методе IC2S( $\tau$ )-CG

Предобусловливание с использованием двух порогов отсечения, являющееся более экономичной версией метода Тисменецкого [19], было описано в работе [20], а соответствующая матричная формулировка, обоснование, теоретический анализ и выбор параметров были даны в работе Капорина [10]. В работе [10] также была предложена версия метода IC2S( $\tau$ ), отличающаяся предварительной равномерной модификацией всех элементов диагонали аналогично подходу [21].

Перед построением матрицы предобусловливания в методе IC2S( $\tau$ )-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( $\tau$ ) имеет вид [10]

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

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 [22],  $\|R\| = O(\tau)$  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 [22], и их модули добавляются к диагональным элементам в строках с номерами  $i$  и  $j$ . В матрицу  $R$  включаются остальные «не слишком малые» значения  $v_{ij}$ :  $\tau^2 < |r_{ij}| < \tau$ . Заметим, что  $E_{JM}(\tau^2) = \sum_{i < j, e_{ij} \neq 0} Z_{ij} Q_{ij} Z_{ij}^T$ , где  $e_{ij} = v_{ij}$ , если

$$|v_{ij}|/u_{ii} \leq \tau^2, \quad e_{ij} = 0 \text{ в противном случае; } Q_{ij} = \begin{pmatrix} |v_{ij}| & -v_{ij} \\ -v_{ij} & |v_{ij}| \end{pmatrix}, \quad Z_{ij} = [e_i \ e_j], \quad e_k - k\text{-ый}$$

столбец единичной матрицы. В качестве предобусловливателя используется матрица  $U^T U$ .

Алгоритм вычисления стабилизированного треугольного разложения второго порядка имеет следующий вид [10]. 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. Некоторые параллельные варианты метода IC2S( $\tau$ )-CG, основанные на использовании специального переупорядочения матрицы коэффициентов**

Для построения параллельных вариантов метода IC2S( $\tau$ )-CG разобьем произвольную трехмерную область расчета на  $p$  подобластей каким-либо образом, выберем некоторую нумерацию подобластей и будем использовать упорядочение узлов сетки [17], связанное с разбиением (DDO). Введем множество узлов разделителей – множество узлов сетки в подобластях, у которых имеются соседи из подобластей с бóльшим номером. Остальные узлы сетки будем называть внутренними.

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. Таким образом, в шаблонах узлов разделителей первого уровня могут быть только внутренние узлы из подобластей с тем же или большими номерами и узлы разделителей из той же подобласти и подобластей с меньшими номерами. 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 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. Остальные узлы разделителей назовем узлами разделителей третьего уровня. В работе [17] доказано, что в шаблонах узлов разделителей первого уровня не может быть узлов разделителей первого уровня из других подобластей; в шаблонах узлов разделителей второго уровня не может быть узлов разделителей второго уровня из других подобластей, а также узлов разделителей первого уровня из подобластей с меньшими номерами.

Установим следующий порядок следования узлов сетки [17]. 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.

На рис. 1 приведен пример структуры разреженности матрицы  $A$ , полученной после перестановки строк и столбцов в результате переупорядочения в случае разбиения области расчета на 4 подобласти. На рис. 1 и далее использованы обозначения:  $l$  – число всех внутренних узлов сетки из всех подобластей,  $l_1$  – число всех внутренних узлов сетки из всех подобластей и всех узлов разделителей первого уровня из всех подобластей,  $l_2$  – число всех внутренних узлов сетки из всех подобластей и всех узлов разделителей первого и второго уровня из всех подобластей. Строки матрицы, содержащие блочно-диагональные части  $A_{11}^k$ , соответствуют внутренним узлам сетки и хранятся в процессоре с номером  $k$ . Строки матрицы, содержащие блочно-диагональные части  $A_{22}^k$  и  $A_{33}^k$ , соответствуют узлам сетки соответственно на разделителях первого и второго уровня из подобласти с номером  $k$  и хранятся в процессоре с номером  $k$ . Строки матрицы, соответствующие блочно-диагональной части  $A_{44}$ , соответствуют узлам сетки на разделителях третьего уровня и хранятся в соответствующих процессорах.

Рассмотрим алгоритмы [17] построения матрицы  $U$  для параллельных методов PIC2S1-CG и PIC2S2-CG. В работе [17] доказано, 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, в котором циклы по  $i$  и по  $s$  происходят по внутренним узлам соответствующих подобластей. Перед началом вычислений следует  $d_i := 0$ . Перед переходом к вычислениям элементов матрицы  $U$  в строках, соответствующих узлам разделителей первого уровня,

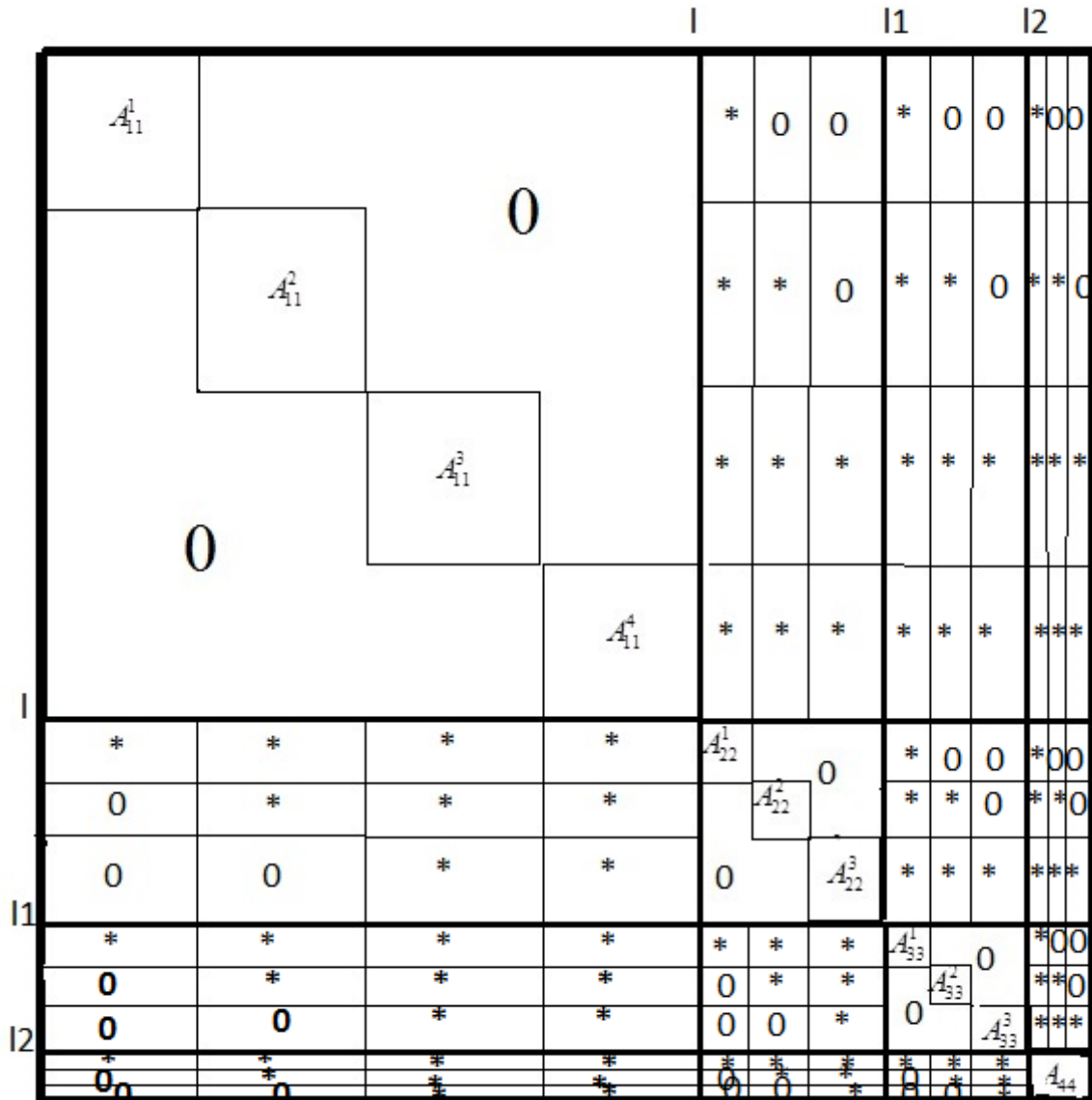


Рис. 1. Пример структуры матрицы  $A$ , полученной после перестановки столбцов и строк в результате переупорядочения.

следует вычислить матрицу  $V$ , элементы которой определяются по формуле

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

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).

В методе PIC2S1-CG используется способ 1 вычисления элементов матриц  $U, R$  на разделителях. При применении способа 1 производится вычисление всех элементов матриц  $U, R$  для всех строк  $i$ , соответствующих узлам разделителей, во всех процессорах, причем одновременно, используется алгоритм 2 [17].

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  $j=i+1, \dots, n$

```

 $v_j := v_j / u_{ij}$ 
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 в способе 1 для обеспечения хорошей параллелизуемости этапа обращения матрицы предобуславливания следует положить равными нулю значения  $u_{ij}$ , если уровень узла разделителя с номером  $j$  совпадает с уровнем разделителя узла  $i$ , но узлы с номерами  $i$  и  $j$  находятся в разных подобластях. Заметим, что следует ожидать большого времени вычисления матрицы предобуславливания при использовании способа 1, т.к. вычисление значений  $u_{ij}$  на разделителях не распараллелено.

В методе PIC2S2-CG используется способ 2 вычисления элементов матриц  $U, R$  на разделителях, алгоритм которого обладает значительно большей параллелизуемостью. 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 способе 1, с использованием алгоритма 3 [17].

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

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

В алгоритме 3  $i_1(k)$ ,  $i_2(k)$  – номера первой и последней строк, соответствующих внутренним узлам в подобласти с номером  $k$ . Параллельная реализация п.1 алгоритма 3 очевидна, каждый процессор производит вычисления в своей подобласти, пересылок не требуется. Одним из эффективных параллельных алгоритмов реализации вычислений п.2 является следующий алгоритм. В каждом процессоре производится сборка строк матриц  $V^s$ , соответствующих узлам разделителей, расположенных в подобласти с номером  $k$ , обслуживаемой этим процессором, и суммирование  $V_{ij} = \sum_{s=1}^p V_{ij}^s$  для строк  $i \in Y_k$ , где  $Y_k$  – множество узлов разделителей в подобласти с номером  $k$ . В результате в каждом процессоре находятся как раз те строки матрицы  $V$ , которые необходимы для дальнейших расчетов в этом процессоре.

Кроме того, производится суммирование по всем процессорам вычисленных там неокончательных значений  $d_j$  для номеров  $j$ , соответствующих номерам узлов разделителей.

Далее происходит вычисление 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, как доказано в [17], для этих значений  $ij$   $u_{ij} = 0$ ,  $r_{ij} = 0$ . Вычисление элементов в строках матрицы  $U$ , соответствующих узлам разделителей первого уровня, может происходить во всех процессорах одновременно. 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 котором 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  $k$ . Параллельная реализация аналогична параллельной реализации при вычислении матрицы  $V$ . Кроме того, нужно произвести суммирование по всем процессорам вычисленных там значений  $\Delta d_j$  (изменений значений  $d_j$ ,

полученных на этапах 4 и 7 алгоритма 2) для номеров  $j$ , соответствующих узлам разделителей второго и третьего уровня, и добавить  $\Delta d_j$  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 и подробно описано в работе [17]. Заметим, что при этом меняются индексы начала и конца некоторых циклов в алгоритме 2, в п.2 алгоритма 2 добавляется еще одно слагаемое:  $\bar{V}_{ij}$ .

Перед вычислением элементов матриц  $U, R$  в строках, соответствующих узлам разделителей третьего уровня, следует вычислить матрицу  $\tilde{V}$ , что делается аналогично вычислению матрицы  $\bar{V}$ . Кроме того, нужно произвести суммирование по всем процессорам вычисленных там значений  $\Delta d_j$  (изменений значений  $d_j$ , полученных на этапах 4 и 7 алгоритма 2) для номеров  $j$ , соответствующих узлам разделителей третьего уровня, и добавить  $\Delta d_j$  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, кратко описано в [17]. 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 способе 1. Так как узлов разделителей третьего уровня относительно мало, этот этап вычислений должен происходить достаточно быстро. 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.

Итак, при использовании способов 1,2 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. Используя метод математической индукции, можно доказать, что для любых  $j, i_0$ , если  $a_{ij} = 0$  для  $i = 1, 2, \dots, i_0$ , то  $u_{ij} = 0$  и  $r_{ij} = 0$  для  $i = 1, 2, \dots, i_0$ . На рис. 2 приведен вид структуры матрицы  $U$ , полученной для матрицы  $A$ , структура которой изображена на

рис.1, при использовании способов 1 и 2.  $U_0^k$  – блочно-диагональная часть матрицы  $U$  в строках, соответствующих внутренним узлам сетки, хранящихся в процессоре с номером  $k$ .

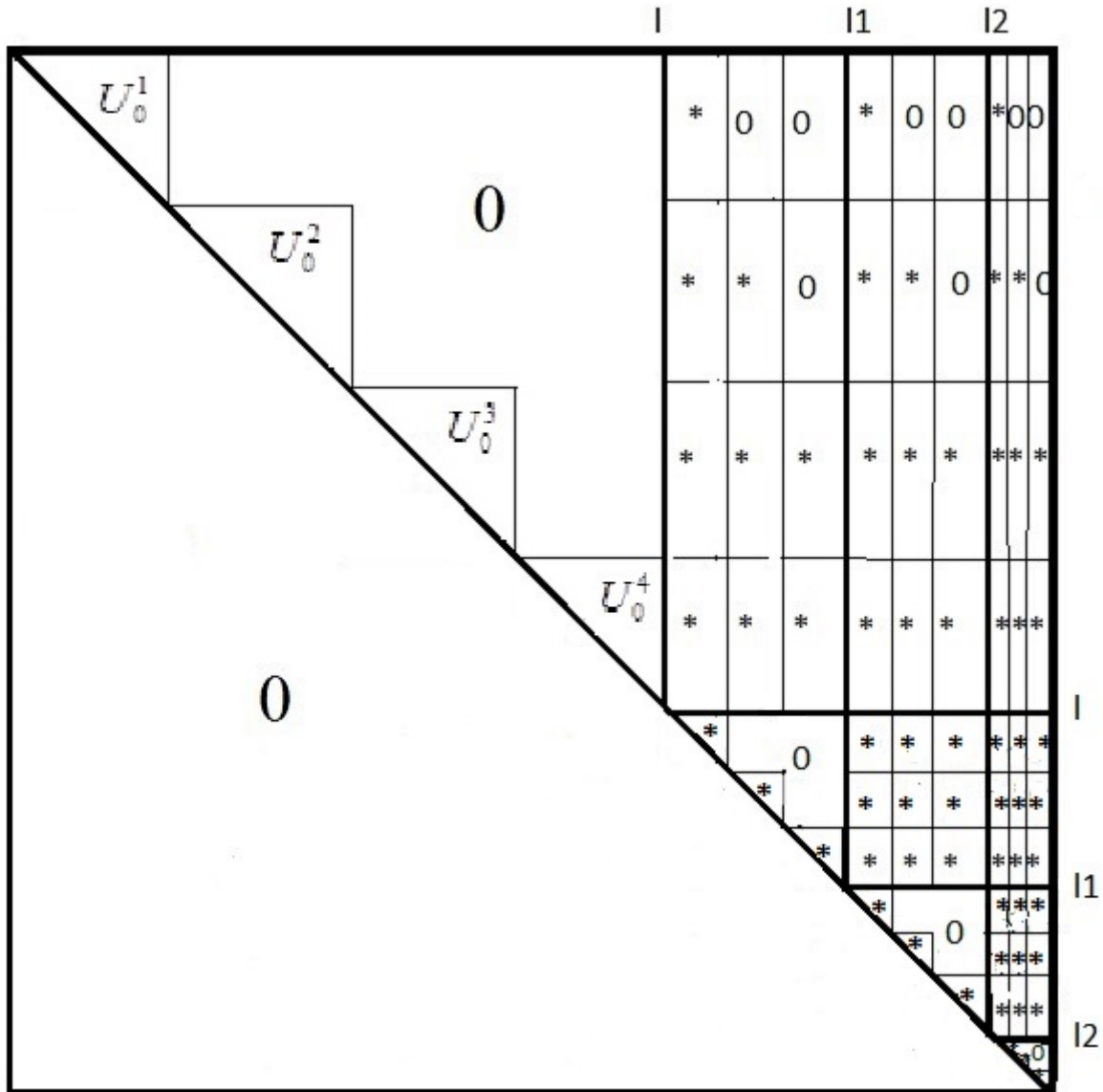


Рис. 2. Структура заполненности матрицы  $U$  в методах PIC2S1-CG, PIC2S2-CG для матрицы  $A$ , структура которой изображена на рис.1.

Матричная схема метода PIC2S2-CG при  $\sigma=1$  имеет вид

$$A + 2\tau^2 I + E_{JM}(\tau^2) + E = U_2^T U_2 + U_2^T R_2 + R_2^T U_2,$$

где  $U_2 = U$  – верхнетреугольная матрица,  $R_2 = R$  (строго верхнетреугольная матрица) создаются, как описано выше, матрица  $E_{JM}(\tau^2) = (E_{JM}(\tau^2))^T \geq 0$  формируется, как описано выше в разделе 2 для метода IC2S( $\tau$ )-CG,

$E = \sum_{i < j, (i,j) \in \Theta} Z_{ij} \tilde{Q}_{ij} Z_{ij}^T$ , где  $\tilde{Q}_{ij} = \begin{pmatrix} 0 & -v_{ij} \\ -v_{ij} & 0 \end{pmatrix}$ ,  $\Theta$  множество индексов  $ij$ , для которых  $V_{ij} \neq 0$ , или  $\bar{V}_{ij} \neq 0$ , или  $\tilde{V}_{ij} \neq 0$  отбрасываются по позициям,  $v_{ij}$  – отбрасываемые значения. В матрице  $E$  первые  $l$  строк нулевые.

В методе PIC2S3-CG используется способ 3 вычисления элементов матриц  $U, R$  на разделителях. В способе 3, в отличие от способа 2, вместо алгоритма 3 вычисления матрицы  $V (\bar{V}, \tilde{V})$  используется алгоритм 4, а вместо алгоритма 2 используется алгоритм 5.

Алгоритм 4.

```

1. for  $k=1, \dots, p$ 
  for  $i=l+1, \dots, n$ 
    for  $j=i+1, \dots, n$ 
       $V_{ij}^k := - \sum_{s=i_1(k)}^{i_s(k)} u_{si} u_{sj} - \sum_{s=i_1(k)}^{i_s(k)} u_{si} r_{sj} - \sum_{s=i_1(k)}^{i_s(k)} r_{si} u_{sj}$ 
    end for
  end for
end for

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

```

Заметим, что в способе 3 при вычислении элементов матриц  $U, R$  на разделителях первого уровня матрица  $V$  совпадает с матрицей  $V$ , полученной в способе 1.

Алгоритм 5.

```

1. for  $i = i'_1(k), \dots, i'_2(k)$ 
   $d_i := d_i + 1 + 2\sigma\tau^2$ 
end for

```

Цикл по строкам  $A$  для вычисления строк  $U, R$ :

```

for  $i = i'_1(k), \dots, i'_2(k)$ 
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  $s = i'_1(k), \dots, j-1$ 
  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
     $d_i := d_i + |v_j|$ 
     $d_j := d_j + |v_j|$ 
     $v_j := 0$ 
  else
    if  $(i, j) \in \Theta$  then
       $v_j := 0$ 
    endif
  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
```

В алгоритме 5  $i'_1(k), i'_2(k)$ - номера первой и последней строк на разделителях первого уровня в подобласти с номером  $k$ . При использовании алгоритма 5 для вычислений на разделителях второго или третьего уровня  $i'_1(k), i'_2(k)$ - номера первой и последней строк на разделителях второго или третьего уровня соответственно, кроме того в пункте 2 алгоритма 5 добавляются слагаемые  $\bar{V}_{ij}$  или  $\bar{V}_{ij}$  и  $\tilde{V}_{ij}$ .

Таким образом, при вычислении предобусловливателя на разделителях всех уровней в методе PIC2S3-CG отсечение по позициям происходит после отсечения по значениям, что отличает его от метода PIC2S2-CG. Как правило,

вычисление предобусловливателя в методе PIC2S3-CG требует больше машинного времени и машинной памяти, чем в методе PIC2S2-CG.

Матричная схема метода PIC2S3-CG при  $\sigma=1$  имеет вид

$$A + 2\tau^2 I + E_{JM}(\tau^2) + \hat{E} = U_3^T U_3 + U_3^T R_3 + R_3^T U_3,$$

где матрица  $\hat{E}$  соответствует отсечению по позициям в пункте 4 алгоритма 5, может быть определена по формулам, аналогичным формулам, определяющим матрицу  $E$ , в матрице  $\hat{E}$  первые  $l$  строк нулевые.

Наряду с методами PIC2S2-CG, PIC2S3-CG рассмотрим методы PIC2S4-CG, PIC2S5-CG – модификации этих методов, в которых используются соответственно способы 4, 5 вычисления элементов матриц предобусловливания на разделителях. Матричная схема метода PIC2S4-CG при  $\sigma=1$  имеет вид

$$A + 2\tau^2 I + E_{JM}(\tau^2) + \hat{E}_{JM}(\alpha\tau^2) + E = U_4^T U_4 + U_4^T R_4 + R_4^T U_4,$$

где матрица  $\hat{E}_{JM}(\alpha\tau^2)$  соответствует отсечению по значению в пункте 4 алгоритма 2, однако это отсечение происходит при  $\tau^2 \sqrt{d_i} < |v_j| \leq \alpha\tau^2 \sqrt{d_i}$ , где  $\alpha > 1$  - параметр, подбираемый экспериментально. Первые  $l$  строк матрицы  $\hat{E}_{JM}(\alpha\tau^2)$  нулевые. Иначе говоря, по сравнению с алгоритмом вычисления предобусловливателя в PIC2S2-CG, прореживание при вычислении во внутренних узлах происходит при выполнении условия  $|v_j| \leq \tau^2 \sqrt{d_i}$ , а прореживание при вычислении на разделителях происходит при выполнении условия  $|v_j| \leq \alpha\tau^2 \sqrt{d_i}$ .

Пусть  $\hat{E} = \hat{E}_1 + \hat{E}_2$ , где  $\hat{E}_1$  соответствует отсечению по позициям в пункте 4 алгоритма 5 не очень больших по модулю значений  $v_j$ :  $\tau^2 \sqrt{d_i} < |v_j| \leq \alpha\tau^2 \sqrt{d_i}$ . Для таких значений на разделителях также будем производить отсечение по значению. То есть в способе 5 и, соответственно, в методе PIC2S5-CG, в отличие от способа 3, будем производить отсечение по значению в пункте 4 алгоритма 5 при  $|v_j| \leq \alpha\tau^2 \sqrt{d_i}$ , где  $\alpha > 1$  – параметр, подбираемый экспериментально. Матричная схема метода PIC2S5-CG при  $\sigma=1$  имеет вид

$$A + 2\tau^2 I + E_{JM}(\tau^2) + \tilde{E}_{JM}(\alpha\tau^2) + \hat{E}_2 = U_5^T U_5 + U_5^T R_5 + R_5^T U_5. \quad (1)$$

Еще раз подчеркнем, что во всех пяти параллельных методах вычисление элементов матриц  $U, R$  в строках, соответствующих внутренним узлам, происходит одинаково. Структура матрицы  $U$  для методов PIC2S3-CG PIC2S4-CG, PIC2S5-CG в случае матрицы  $A$ , структура которой изображена на рис. 1, такая же, как изображено на рис.2.

Рассмотрим теперь алгоритм [17] 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, 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. Алгоритм параллельной реализации вычислений на втором этапе  $w^k = U^{-1}\hat{w}^k$  аналогичен алгоритму параллельной реализации вычислений на первом этапе, но вычисления происходят в обратном порядке.

Программная реализация вычисления предобусловливателя и его обращения осуществлена для случая хранения только матрицы  $U$ . Основная идея такого подхода описана в [12].

#### 4. Безотказность методов PIC2S1-CG, PIC2S2-CG, PIC2S3-CG, PIC2S4-CG, PIC2S5-CG

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 (т. к. метод IC2S( $\tau$ )-CG является безотказным при любом упорядочении узлов сетки). Алгоритм метода PIC2S2-CG в общем случае нельзя назвать безотказным. В работе [17] приведены некоторые рассуждения, согласно которым можно надеяться, что метод PIC2S2-CG в ряде случаев будет безотказным. В работе [17] теоретически доказана безотказность алгоритма метода PIC2S2-CG для класса матриц  $A = A^T > 0$ , элементы которых удовлетворяют условиям:  $a_{ii} > 0$ ,  $a_{ij} \leq 0, j \neq i$ . Алгоритм метода PIC2S3-CG в общем случае тоже нельзя назвать безотказным.

Как показали расчеты задач offshore и cfd2 из Флоридской коллекции, для которых методы PIC2S2-CG, PIC2S3-CG не обладали свойством безотказности (см. раздел 6), для обеспечения условия  $d_i > 0$  для всех  $i$  пришлось очень сильно увеличить параметр  $\sigma$  и сдвиг по диагонали, что привело к очень большому числу итераций в этих методах. Заметим, что, как показали результаты расчетов [17], если вместо отбрасывания элементов по позициям всегда использовать прием Дженнингса-Малика, то при наличии больших по модулю элементов, которые предполагалось отбросить по позициям, сильно ухудшается качество предобусловливания и сильно растет число итераций.

Если метод PIC2S2-CG не является безотказным, то можно использовать параметры  $\tau_2 = \theta\tau_1^2$ , где  $1 \leq \theta < \tau_1^{-1}$ . При этом прием Дженнингса-Малика будет

использоваться чаще, что позволит увеличить ряд значений  $d_i$  и может привести к безотказности метода.

Поскольку на этапе вычисления предобусловливателя во внутренних узлах подобластей алгоритм во всех пяти рассматриваемых методах совпадает с алгоритмом вычисления в методе IC2S( $\tau$ )-CG при DDO, то на этом этапе  $d_i > 0$ . Возникновение  $d_i \leq 0$  возможно только на разделителях. В методах PIC2S4-CG, PIC2S5-CG благодаря выбору параметра  $\alpha > 1$  можно увеличить значения  $d_i$  на разделителях. Можно надеяться, что при подходящем значении параметра  $\alpha > 1$  метод PIC2S4-CG будет безотказным.

**Утверждение.** Начиная с некоторого значения  $1 < \alpha = \alpha_1$  алгоритм метода PIC2S5-CG становится безотказным.

**Д о к а з а т е л ь с т в о.** Пусть всегда вместо отбрасывания по позициям в п.4 алгоритма 5 применяется прием Дженнингса-Малика. Тогда фактически производится приближенное неполное треугольное разложение с отложенным отсечением [10] симметричной положительно определенной матрицы

$$\bar{A} = A + 2\tau^2 I + E_{JM}(\tau^2) + \bar{E}_{JM} = \bar{U}^T \bar{U} + \bar{U}^T \bar{R} + \bar{R}^T \bar{U}, \quad (2)$$

где матрица  $\bar{E}_{JM} = (\bar{E}_{JM})^T \geq 0$  получена в результате применения приема Дженнингса-Малика вместо отбрасывания по позициям в п.4 алгоритма 5.

Матричная схема метода PIC2S5-CG при  $\sigma = 1$  имеет вид (1) (см. раздел 3). С ростом  $\alpha$  количество ненулевых элементов в матрице  $\tilde{E}_{JM}(\alpha\tau^2)$  увеличивается, а в матрице  $\hat{E}_2$ , являющейся незнакоопределенной, уменьшается. Поэтому начиная с некоторого  $\alpha_1 \leq \alpha_2$ , где  $\alpha_2$  соответствует матричной схеме (2), симметричная матрица  $A + 2\tau^2 I + E_{JM}(\tau^2) + \tilde{E}_{JM}(\alpha\tau^2) + \hat{E}_2$  становится положительно определенной. Для любой симметричной положительно определенной матрицы алгоритм построения приближенного неполного треугольного разложения с отложенным отсечением является безотказным [10]. Следовательно, при  $\alpha \geq \alpha_1$  алгоритм метода PIC2S5-CG является безотказным.

Можно организовать следующий алгоритм вычисления предобусловливателя:

1. Производим вычисление элементов матриц  $U$  и  $R$  во всех строках, соответствующих внутренним узлам сетки.
2.  $\alpha = 1$ .
3. Производим вычисление элементов матриц  $U$  и  $R$  на разделителях с использованием способа 2 или 3 (при  $\alpha = 1$ ), способа 4 или 5 (при  $\alpha > 1$ ). Если не возникло значение  $d_i \leq 0$ , то переходим к п. 6.
4.  $\alpha = \alpha + 1$ .
5. Переходим к пункту 3.
6. Конец.

Этот алгоритм будет безотказным в случае использования способов 3,5. Можно надеяться, что этот алгоритм будет безотказным и в случае

использования способов 2,4, при этом, как правило, следует ожидать меньших затрат машинной памяти и, возможно, меньшего времени счета.

Матрица предобусловливания  $U^T U$  симметричная, при условии безотказности алгоритма вычисления матрицы  $U$  матрица  $U^T U$  положительно определена (так как при построении матрицы  $U$   $d_i > 0$  для всех значений  $i$ ). Это гарантирует сходимость предобусловленного метода сопряженных градиентов.

Для ряда методов, имеющих такой же характер асимптотической зависимости числа итераций от числа узлов сетки, как в методе IC2S( $\tau$ )-CG, показано, что использование упорядочений типа DDO может привести лишь к небольшому росту числа итераций по сравнению с их однопроцессорными вариантами [1,2,6]. Можно надеяться, что использование DDO не приведет к большому росту числа итераций метода IC2S( $\tau$ )-CG. В рассматриваемых параллельных методах производится также отсечение по позициям. В работе [17] для матриц  $A = A^T > 0$ , удовлетворяющих условиям:  $a_{ii} > 0$ ,  $a_{ij} \leq 0, j \neq i$ ,

$$a_{ii} \geq \sum_{j \neq i} |a_{ij}|, \text{ где } 1 \leq i \leq n, 1 \leq j \leq n, \text{ причем } A_{ij}^{-1} > 0 \text{ для всех } ij, \text{ получены}$$

достаточные условия, когда изменение оценки числа итераций методов PIC2S1-CG, PIC2S2-CG по сравнению с оценкой числа итераций метода IC2( $\tau$ )-CG при DDO незначительное. Можно надеяться, что, по крайней мере, при выполнении этих достаточных условий для указанного класса матриц число итераций при применении методов PIC2S1-CG и PIC2S2-CG будет не очень сильно отличаться от числа итераций однопроцессорного метода IC2S( $\tau$ )-CG.

## 5. Результаты расчетов

Все программы, реализующие использование методов IC2S( $\tau$ )-CG, PIC2S1-CG, PIC2S2-CG, PIC2S3-CG, PIC2S4-CG, PIC2S5-CG для решения системы уравнений  $Ax = b$ , где  $A = A^T > 0$ , были написаны на языке FORTRAN с использованием MPI, расчеты производились на многопроцессорной вычислительной системе МВС 100К, установленной на МСЦ РАН. Матрица  $A$  хранилась в CRS формате.

В работе [17] методы IC2S( $\tau$ )-CG, PIC2S1-CG, PIC2S2-CG были применены для решения трехмерной задачи Дирихле для уравнения Пуассона в единичной кубической области расчета на ортогональной сетке. Использовалась стандартная 7-точечная аппроксимация лапласиана. Для построения параллельных методов область расчета 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. Предполагалось, что  $b_i \equiv 1$ , начальное приближение  $x_0 \equiv 0$ , счет продолжался до выполнения условия:  $\|Ax - b\| < \varepsilon \|b\|$ , где  $\varepsilon = 10^{-9}$ ,  $\tau = 0.01$ ,  $\tau_1 = \tau, \tau_2 = \tau_1^2$ . Сдвиг на диагонали на величину  $2\tau^2$  и  $2\sigma\tau^2$  не осуществлялся. Заметим, что для этой модельной задачи согласно

теоретическому исследованию [17] методы PIC2S1-CG, PIC2S2-CG безотказны. Приведем результаты из работы [17].

В таблице 1 приведены значения числа итераций, полученные в расчетах с использованием IC2S( $\tau$ )-CG (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  $p$ . 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. Максимальное увеличение числа итераций было в 1.28 раза.

Таблица 1

Значения числа итераций, полученные при решении модельной задачи методами IC2S( $\tau$ )-CG, PIC2S1-CG, PIC2S2-CG

$N_h$	$p=1$	$p=8$	$p=27$	$P=64$	$p=125$	$p=216$
30	25	28, 28	28, 29	30,32( $N_h = 32$ )	28, 29	29, 29
40	32	36, 36	36, 38 ( $N_h = 42$ )	35, 36	35, 36	38, 39 ( $N_h = 42$ )
50	39	44, 44	40, 41 ( $N_h = 48$ )	41, 42 ( $N_h = 48$ )	43, 43	42, 43 ( $N_h = 48$ )
60	45	52, 52	49, 49	50, 50	50, 50	51, 52
72	53	61, 61	57, 57	58, 58	57, 57	60, 60

В таблице 2 приведены значения числа итераций метода PIC2S2-CG при решении этой модельной задачи и использовании упорядочения узлов сетки и разбиения области расчета, предложенных в работе [12]. При этом подобласти строятся автоматически, могут иметь сложную форму, не быть односвязными. Из таблицы 2 видно, что и при таком разбиении рост числа итераций с ростом числа процессоров достаточно медленный, однако он больше выражен, чем в таблице 1.

Таблица 2

Значения числа итераций, полученные при решении модельной задачи методом PIC2S2-CG для различных значений  $p$  и  $N$  при разбиении области расчета способом из работы [12]

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

Время счета итерационного процесса и вычисления предобусловливателя измерялось при использовании в расчетах 2 ядер из 8 имеющихся в каждом

узле МВС 100К и приведено в таблицах 3,4. При этом использовалось разбиение области расчета, описанное в начале раздела 6.

Таблица 3

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

	p=1	p=8	p=27	p=64	p=125
$N_h = 72 (N=373248)$	3.07	0.41	0.085	0.065	0.083
$N_h = 92 (N=778688)$			0.22	0.11	0.11

Ускорение счета с ростом числа процессоров замедляется главным образом за счет большого числа пересылок в процессе обращения матрицы предобусловливания. Заметим, что, как показывают расчеты, использование более современных вычислительных систем, приводит к значительно лучшей масштабируемости при счете итерационного процесса.

Таблица 4

Время (в секундах) вычисления предобусловливателя в методе PIC2S2-CG при различных  $p, N=373248$

	p=1	p=27	p=64	p=125
$N_h = 72 (N=373248)$	2.93	0.279	0.173	0.130

Как видно из таблицы 4, замедление ускорения вычисления предобусловливателя происходит при бóльших значениях  $p$ , с ростом  $p$  происходит сближение времени построения предобусловливателя и времени счета итерационного процесса.

Для тестирования рассматриваемых параллельных методов использовались также некоторые матрицы из Флоридской коллекции [18]. Перечислим имена используемых тестовых матриц и укажем источник их происхождения.

**apache2** – трехмерная конечно-разностная схема;

**parabolic\_fem** – уравнение диффузии-конвекции с постоянным переносом;

**tmt\_sym** – моделирование задач электромагнетизма;

**ecology2** – приложение теории электрических цепей к задаче передачи генов;

**thermal2** – стационарная термальная задача;

**boneS01** – модель трубчатой кости;

**cf2** – вычислительная гидродинамика (уравнение давления);

**offshore** – диффузия электрического поля (3-мерная тетраэдральная сетка).

В таблице 5 перечислены некоторые свойства этих матриц, причем значения  $Cond(A_s)$ , где  $A_s = (Diag(A))^{-1/2} A (Diag(A))^{-1/2}$  – матрица системы уравнений после масштабирования, взяты из работы [23],  $I_d$  – количество строк без диагонального преобладания,  $I_p$  – количество положительных

внедиагональных элементов,  $NZA$  – число ненулевых элементов матрицы  $A$ ,  $nz_{\min}$ ,  $nz_{\max}$  – минимальное и максимальное число ненулевых элементов в строках матрицы  $A$ . Для большинства из перечисленных матриц не выполнены достаточные условия [17] безотказности метода PIC2S2-CG.

Таблица 5

Свойства некоторых матриц из Флоридской коллекции

Матрица	N	NZA	Id	Ip	$nz_{\min}$	$nz_{\max}$	$Cond(A_S)$
apache2	715176	4817870	2	0	4	8	0.12+7
parabolic_fem	525825	3674625	0	1048576	3	7	0.20+6
tmt_sym	726713	5080961	676871	724378	3	9	0.25+9
ecology2	999999	4995991	670	0	3	5	0.63+8
thermal2	1228045	8580313	327235	840	1	11	0.45+7
boneS01	127224	5516602	127222	2064830	12	67	0.13+8
cf2	123440	3085406	123440	936464	8	30	0.15+7
offshore	259789	4242673	194617	1813920	5	31	0.34+5

Решалось уравнение  $Ax=b$ , где правая часть  $b_i \equiv 1$ , начальное приближение  $x_0 \equiv 0$ , счет продолжался до выполнения условия:  $\|Ax-b\| < \varepsilon \|b\|$ , где  $\varepsilon = 10^{-8}$ . Для разбиения области расчета использовался способ, предложенный в [12]. При построении предобусловливателей всех методов использовались значения параметров  $\tau_1 = 0.01, \tau_2 = \tau_1^2$ , при нарушении безотказности PIC2S2-CG  $\tau_2 = \theta \tau_1^2$ , где  $\tau_1^{-1} > \theta > 1$ . Использовался сдвиг по диагонали на величину  $2\tau^2$  и  $2\sigma\tau^2$ , где  $\sigma = 1$ . В таблицах 6-13 приведены значения чисел итераций методов PIC2S1-CG и PIC2S2-CG, полученные при решении уравнения  $Ax=b$  с матрицами из Флоридской коллекции. В таблицах 12,13 приведены также значения чисел итераций методов PIC2S4-CG и PIC2S5-CG при оптимальном с точки зрения числа итераций значении  $\alpha$ , полученные при решении уравнения  $Ax=b$  с двумя матрицами из Флоридской коллекции. При этом  $\tau_1 = 0.01, \tau_2 = \tau_1^2$ , использовался сдвиг по диагонали на величину  $2\tau^2$  и  $2\sigma\tau^2$ , где  $\sigma = 1$ . В круглых скобках приведено приближенное значение коэффициента возрастания числа итераций по сравнению с однопроцессорным методом IC2S( $\tau$ )-CG, в квадратных скобках – относительная плотность предобусловливателя  $2 * NZU / (NZA + N)$ , где  $NZU$  – количество ненулевых элементов в матрице  $U$ .

Таблица 6

Значения числа итераций, полученные при решении уравнения  $Ax=b$  для матрицы apache2 для различных значений  $p$ 

Метод	p=1	p=27	p=64	p=125	p=216	p=343
PIC2S1-CG	252[1.69]	506	543	609	750	787
PIC2S2-CG		342 [1.86]	352 (1.39) [1.88]	376 (1.49) [1.90]	392 (1.55) [1.91]	384 (1.52) [1.92]



Таблица 7

Значения числа итераций, полученные при решении уравнения  $Ax = b$  для матрицы `parabolic_fem` для различных значений  $p$

Метод	p=1	p=8	p=27	p=64	p=125	p=216
PIC2S1-CG	184[2.00]	244	280	289	303	323
PIC2S2-CG			273 (1.48) [2.07]	272 (1.48) [2.10]	294 (1.59) [2.11]	301 (1.63) [2.13]

Таблица 8

Значения числа итераций, полученные при решении уравнения  $Ax = b$  для матрицы `tmt_sym` для различных значений  $p$

Метод	p=1	p=27	p=64	p=125	p=216
PIC2S1-CG	362[2.04]	423	426	450	461
PIC2S2-CG		414 (1.14) [2.05]	415 (1.14) [2.05]	435 (1.2) [2.05]	443 (1.22) [2.06]

Таблица 9

Значения числа итераций, полученные при решении уравнения  $Ax = b$  для матрицы `ecology2` для различных значений  $p$

Метод	p=1	p=64	p=125	p=216	p=343
PIC2S1-CG	395[2.65]	568	620	641	
PIC2S2-CG		524 (1.32) [2.48]	552 (1.39) [2.53]	565 (1.43) [2.58]	575 (1.46) [2.62]

Таблица 10

Значения числа итераций, полученные при решении уравнения  $Ax = b$  методом PIC2S2-CG для матрицы `thermal2` для различных значений  $p$

Метод	p=1	p=64	p=125	p=216
PIC2S2-CG	424[2.07]	501 (1.18) [2.10]	504 (1.18) [2.11]	514 (1.21) [2.12]

Обозначим  $n_k(p)$  среднее число узлов сетки, приходящееся на один процессор. При решении задачи с матрицей `apache2`  $n_k(343) \approx 2085$ , при решении задачи с матрицей `parabolic_fem`  $n_k(216) \approx 2434$ , в случае матрицы `tmt_sym`  $n_k(216) \approx 3361$ , при решении задачи с матрицей `ecology2`  $n_k(343) \approx 2915$ . Как видно из таблиц 6-9, при расчетах этих задач на приведенных числах процессоров вплоть до числа узлов сетки, приходящихся на одно ядро, не более 2000-3361 коэффициент возрастания числа итераций по сравнению с расчетом на одном процессоре не превосходил 1.63. Заметим, что уменьшать число узлов сетки, приходящихся на одно ядро, в этих задачах может оказаться нецелесообразным из-за вопроса масштабируемости вычислений. Заметим, что при использовании

метода PIC2S1-CG при решении этих задач число итераций растет с ростом числа процессоров сильнее.

Перейдем теперь к решению задач с более плотными матрицами.

Таблица 11

Значения числа итераций, полученные при решении уравнения  $Ax = b$  для матрицы boneS01 для различных значений  $p$

Метод	$p=1$	$p=27$	$p=64$	$p=125$
PIC2S1-CG	202[1.34]	278	337	369
PIC2S2-CG		318 (1.57) [1.44]	376 (1.86) [1.48]	414 (2.05) [1.49]

При решении задачи с матрицей boneS01  $n_k(125) \approx 1067$ , коэффициент возрастания числа итераций приемлемый.

Заметим, что для всех этих 6 задач метод PIC2S2-CG был безотказным, рост числа итераций с ростом числа процессоров приемлемым, увеличение относительной плотности предобусловливателя с ростом числа процессоров было небольшим, менее 12%.

Таблица 12

Значения числа итераций, полученные при решении уравнения  $Ax = b$  методами PIC2S1-CG, PIC2S2-CG, PIC2S4-CG и PIC2S5-CG для матрицы cfd2 для различных значений  $p$

Метод	$p=1$	$p=27$	$p=64$
PIC2S1-CG	221[1.9]	422	460
PIC2S2-CG ( $\tau_1 = 0.01, \tau_2 = 4\tau_1^2$ )		577	635
PIC2S4-CG $\tau_1 = 0.01, \tau_2 = \tau_1^2$ $\tau_1 = 0.01, \tau_2 = \alpha\tau_1^2$		414 (1.87) $\alpha = 14$ [2.25]	491 (2.22) $\alpha = 12$ [2.4]
PIC2S5-CG $\tau_1 = 0.01, \tau_2 = \tau_1^2$ $\tau_1 = 0.01, \tau_2 = \alpha\tau_1^2$		416 (1.88) $\alpha = 7$ [2.26]	509(2.25) $\alpha = 6$ [2.4]

При  $\tau_1 = 0.01, \tau_2 = \tau_1^2, \tau_2 = 2\tau_1^2, \tau_2 = 3\tau_1^2$  при решении уравнения  $Ax = b$  для матрицы cfd2 метод PIC2S2-CG оказался небезотказным.  $n_k(64) \approx 1928$ , коэффициент возрастания числа итераций в методах PIC2S1-CG, PIC2S4-CG, PIC2S5-CG приемлемый, относительная плотность предобусловливания слабо зависит от  $\alpha$ , возрастает с ростом  $p$  не более, чем на 27%. Заметим, что при использовании метода PIC2S1-CG при решении задач boneS01 и cfd2 число

итераций, как правило, растет медленнее с ростом  $p$ , однако время вычисления предобусловливателя будет существенно больше (см. выше).

Таблица 13

Значения числа итераций, полученные при решении уравнения  $Ax=b$  методами PIC2S1-CG, PIC2S2-CG и PIC2S4-CG, PIC2S5-CG для матрицы offshore для различных значений  $p$

Метод	p=1	p=27	P=64
PIC2S1-CG	88 [1.35]	294	350
PIC2S2-CG ( $\tau_1 = 0.01, \tau_2 = 5\tau_1^2$ )		279	377
PIC2S4-CG $\tau_1 = 0.01, \tau_2 = \tau_1^2$ $\tau_1 = 0.01, \tau_2 = \alpha\tau_1^2$		249 (2.82) $\alpha = 12$ [1.35]	372 (4.22) $\alpha = 12$ [1.34]
PIC2S5-CG $\tau_1 = 0.01, \tau_2 = \tau_1^2$ $\tau_1 = 0.01, \tau_2 = \alpha\tau_1^2$		363 (4.12) $\alpha = 5$ [1.35]	

При  $\tau_1 = 0.01, \tau_2 = \tau_1^2$ ,  $\tau_1 = 0.01, \tau_2 = 2\tau_1^2$  при решении задачи с матрицей offshore метод PIC2S2-CG оказался небезотказным.  $n_k(64) \approx 4059$ , число итераций растет с ростом  $p$  довольно быстро во всех четырех методах. Из-за проблем с памятью, связанных отчасти с недостаточно оптимальной программой, не удалось посчитать число итераций PIC2S5-CG при  $p=64$ .

Заметим, что как показывают результаты расчетов для рассмотренных матриц из Флоридской коллекции, приведенные в работе [23], при использовании метода сопряженных градиентов с предобусловливанием Якоби число итераций получается в 10-30 раз больше, чем при использовании метода IC2S( $\tau$ )-CG.

Рост числа итераций и эффективность рассматриваемых параллельных методов зависит также от способа разбиения на подобласти и способа упорядочения узлов сетки на одном процессоре и на  $p$  процессорах. Можно, например, использовать спроецированное упорядочение узлов сетки, построенное для одного процессора. Можно в каждой подобласти вводить свое новое упорядочение узлов сетки.

Для решения задач с матрицами cfd2 и offshore при построении параллельных методов PIC2S2-CG, PIC2S3-CG, PIC2S4-CG, PIC2S5-CG использовался также инкрементный метод разбиения области расчета [24,25]. Строилось CM или RCM упорядочение узлов всей области расчета, а затем это упорядочение проецировалось на подобласти. В таблицах 14,15 приведены

значения чисел итераций, полученных в расчетах этими методами, коэффициентов возрастания числа итераций по сравнению с однопроцессорным методом и относительной плотности предобусловливателя. Приведены результаты расчетов при оптимальном с точки зрения числа итераций значении  $\alpha$  и при минимальном  $\alpha$ , обеспечивающем безотказность метода (при  $\alpha = \alpha_1$ ).

Таблица 14

Значения числа итераций, полученные при решении уравнения  $Ax=b$  для матрицы cfd2 для различных значений  $p$  при использовании инкрементного метода построения разбиений

Метод	p=1		p=8		p=27		p=64	
	CM	RCM	CM	RCM	CM	RCM	CM	RCM
PIC2S2-CG $\tau_1 = 0.01, \tau_2 = \tau_1^2$	246	221	283	250	412	356	-	-
	[1.96]	[1.9]			[2.29]	[2.21]		
PIC2S3-CG $\tau_1 = 0.01, \tau_2 = \tau_1^2$	246	221	283	244	423	451	-	-
	[1.96]	[1.9]			[2.29]	[2.21]		
PIC2S4-CG $\tau_1 = 0.01, \tau_2 = \alpha\tau_1^2$					387	348	489 (2)	486 (2.2)
					$\alpha = 5$	$\alpha = 3$	$\alpha = 10$ [2.44]	$\alpha = 12$ [2.37]
					[2.28]	[2.21]		
Min $\alpha$ ( $\alpha = \alpha_1$ ) ч.ит.приMin $\alpha$							$\alpha = 8$ [2.44]	$\alpha = 8$ [2.36]
							499	498
PIC2S5-CG $\tau_1 = 0.01, \tau_2 = \alpha\tau_1^2$					396	345	505(2.05)	508 (2.29)
					$\alpha = 6$	$\alpha = 3$	$\alpha = 8$ [2.44]	$\alpha = 8$ [2.37]
					[2.28]	[2.21]		
Min $\alpha$ ( $\alpha = \alpha_1$ ) ч.ит.приMin $\alpha$							$\alpha = 6$ [2.44]	$\alpha = 8$
							515	508

Как показывают расчеты, с ростом  $\alpha \geq \alpha_1$  в методах PIC2S4-CG, PIC2S5-CG сначала происходит уменьшение числа итераций, а затем после точки минимума происходит рост числа итераций. Как видно из таблицы 14, относительная плотность предобусловливания слабо зависит от  $\alpha$ , растет с ростом числа процессоров не более, чем на 25%, коэффициент возрастания числа итераций с ростом  $p$  приемлемый. При оптимальном значении  $\alpha$  число итераций методов PIC2S4-CG, PIC2S5-CG меньше, чем в таблице 12.

Как видно из таблицы 15, наблюдается приемлемый рост числа итераций рассмотренных методов с ростом числа процессоров, значительно меньший, чем в таблице 13, число итераций в таблице 15 при  $p > 1$  примерно в 1.5-2 раза меньше, чем в таблице 13. Относительная плотность предобусловливания слабо

зависит от  $\alpha$ , а для задачи с матрицей offshore слабо зависит и от числа процессоров.

Таблица 15

Значения числа итераций, полученные при решении уравнения  $Ax=b$  для матрицы offshore для различных значений  $p$  при использовании инкрементного метода построения разбиений

Метод	p=1		p=8		p=27		p=64	
	CM	RCM	CM	RCM	CM	RCM	CM	RCM
PIC2S2-CG $\tau_1 = 0.01, \tau_2 = \tau_1^2$	125	88	159	-	-	-	-	-
	[1.4]	[1.34]						
PIC2S3-CG $\tau_1 = 0.01, \tau_2 = \tau_1^2$	125	88	158	-	-	-	-	-
	[1.4]	[1.34]						
PIC2S4-CG $\tau_1 = 0.01, \tau_2 = \alpha\tau_1^2$			108		165(1.3)	161(1.8)	247(1.97)	247(2.8)
			$\alpha = 10$		$\alpha = 3$	$\alpha = 10$	$\alpha = 8$	$\alpha = 14$
					[1.42]	[1.37]	[1.43]	[1.38]
Min $\alpha (\alpha = \alpha_1)$ ч.ит.при Min $\alpha$			$\alpha = 5$		$\alpha = 2$	$\alpha = 7$	$\alpha = 6$	$\alpha = 11$
			128		165[1.41]	172[1.37]	271[1.43]	265[1.37]
PIC2S5-CG $\tau_1 = 0.01, \tau_2 = \alpha\tau_1^2$			112		161(1.29)	161(1.8)	247(1.97)	251(2.85)
			$\alpha = 6$		$\alpha = 3$	$\alpha = 9$	$\alpha = 8$	$\alpha = 11$
					[1.41]	[1.36]	[1.43]	[1.38]
Min $\alpha (\alpha = \alpha_1)$ ч.ит.при Min $\alpha$			$\alpha = 5$		$\alpha = 2$	$\alpha = 6$	$\alpha = 5$	$\alpha = 9$
			121		164[1.42]	181[1.37]	263[1.43]	255[1.38]

Как видно из таблиц 14,15, минимальное значение  $\alpha$ , при котором метод PIC2S5-CG становился безотказным, не превосходило минимальное значение  $\alpha$ , при котором метод PIC2S4-CG становился безотказным. Число итераций в методе PIC2S5-CG, как правило, немного больше, чем в методе PIC2S4-CG. Кроме того, как говорилось выше, время расчета и требуемая машинная память при вычислении предобусловливателя в методе PIC2S5-CG обычно больше, чем в методе PIC2S4-CG. Однако для метода PIC2S5-CG доказана безотказность при достаточно большом значении  $\alpha$ .

Проводилось также исследование влияния значения параметров  $\tau_1$ ,  $\tau_2 = \tau_1^2$  на скорость сходимости и плотность предобусловливателя метода PIC2S2-CG для ряда задач. В таблице 16 приведены значения чисел итераций, относительной плотности предобусловливателя и произведения этих чисел для различных значений  $\tau_1$ , полученных при решении ряда задач из Флоридской коллекции методом PIC2S2-CG на 125 процессорах. Для сравнения в таблице

17 приведены соответствующие значения при расчете на одном процессоре методом IC2S( $\tau$ )-CG.

Таблица 16

Значения числа итераций и относительной плотности предобусловливателя, полученные при решении уравнения  $Ax=b$  методом PIC2S2-CG для некоторых матриц из Флоридской коллекции для  $p=125$

Матрица	$\tau_1 = 0.03$	$\tau_1 = 0.01$	$\tau_1 = 0.002$
APACHE2	$635 \times [0.99] \approx 629$	$376 \times [1.9] \approx 714$	$226 \times [4.62] \approx 963$
Parabolic_ferm	$484 \times [1.27] \approx 615$	$294 \times [2.11] \approx 620$	$145 \times [4.71] \approx 689$
tmt_sym	$718 \times [1.28] \approx 919$	$435 \times [2.05] \approx 891$	$219 \times [4.52] \approx 990$
thermal2	$856 \times [1.38] \approx 1181$	$504 \times [2.11] \approx 1063$	$249 \times [4.43] \approx 1103$

Таблица 17

Значения числа итераций и относительной плотности предобусловливателя, полученные при решении уравнения  $Ax=b$  методом IC2S( $\tau$ )-CG для некоторых матрицы из Флоридской коллекции ( $p=1$ )

Матрица	$\tau_1 = 0.03$	$\tau_1 = 0.01$	$\tau_1 = 0.002$
APACHE2	$444 \times [0.93] \approx 412$	$252 \times [1.69] \approx 426$	$118 \times [4.2] \approx 496$
Parabolic_ferm	$329 \times [1.24] \approx 408$	$184 \times [2.0] = 368$	$90 \times [3.77] \approx 339$
tmt_sym	$651 \times [1.26] \approx 820$	$362 \times [2.04] \approx 738$	$166 \times [4.3] \approx 714$

С уменьшением  $\tau_1$  число итераций уменьшается, однако растет относительная плотность предобусловливателя. При этом произведение этих значений в случае 125 процессоров получается больше, чем в случае одного процессора. Напомним, что на время решения системы уравнений кроме времени построения и обращения предобусловливателя также сказывается время умножения матрицы на вектор, время вычисления скалярных произведений, число итераций, количество используемых процессоров. Параметры  $\tau_1$ ,  $\tau_2$ ,  $\alpha$  надо подбирать экспериментально, минимизируя время решения системы уравнений.

Итак, в настоящей работе рассмотрены методы PIC2S1-CG, PIC2S2-CG [17] и предложены новые параллельные варианты метода IC2S( $\tau$ )-CG – методы PIC2S3-CG, PIC2S4-CG, PIC2S5-CG. Доказана безотказность метода PIC2S5-CG при достаточно больших значениях параметра  $\alpha$  для любых матриц  $A=A^T > 0$ . Проведено численное исследование безотказности и скорости сходимости методов PIC2S1-CG, PIC2S2-CG для ряда задач из Флоридской коллекции. Для 6 из 8 задач из Флоридской коллекции метод PIC2S2-CG оказался безотказным, наблюдался приемлемый рост числа итераций с ростом числа процессоров, небольшой рост относительной плотности предобусловливателя. Для двух задач из Флоридской коллекции (для которых

PIC2S2-CG не являлся безотказным) проведено также исследование безотказности и скорости сходимости методов PIC2S3-CG, PIC2S4-CG, PIC2S5-CG. Начиная с некоторого значения  $\alpha$  при решении этих задач методы PIC2S4-CG, PIC2S5-CG становятся безотказными. Расчеты этих двух задач показали, что использование инкрементного способа разбиения области расчета в методах PIC2S4-CG, PIC2S5-CG предпочтительнее, чем способа из работы [12], возможно из-за меньшего числа узлов на разделителях.

Автор выражает благодарность Капорину Игорю Евгеньевичу за проявленное внимание к работе, ценные советы и замечания, Харченко Сергею Александровичу за проявленное внимание к работе и ценные советы, Якобовскому Михаилу Владимировичу и Головченко Евдокии Николаевне за проявленный интерес к работе и построение разбиения матриц cfd2, offshore инкрементным методом.

## 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 линейный алгебраических уравнений с симметричной положительно определенной матрицей // *Препринты ИПМ им. М.В. Келдыша*. 2013. №13. 17 с. URL: [http://keldysh.ru/papers/2013/prep2013\\_13.pdf](http://keldysh.ru/papers/2013/prep2013_13.pdf)
10. 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.

11. 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.
12. 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.
13. 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.
14. 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.
15. 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.
16. 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.
17. 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://keldysh.ru/papers/2014/prep2014\\_52.pdf](http://keldysh.ru/papers/2014/prep2014_52.pdf)
18. Davis T.A., Hu Y.F. University of Florida sparse matrix collection // ACM Trans. on Math. Software, 2011. V.38 (<http://www.cise.ufl.edu/research/sparse/matrices>)
19. Tismenetsky M. A new preconditioning technique for solving large sparse linear systems// Linear Algebra Appls. 1991. V. 154-156. – P.331-353.
20. 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.



21. Manteuffel T.A. An incomplete factorization technique for positive definite linear systems// Math.Comput. 1980. V. 34. – P.473-497.
22. Jennigs A. Malik G.M. Partial elimination // J.Inst. Math. Appl. 1977. V. 20. – P.307-316.
23. Капорин И.Е. Использование полиномов Чебышева и приближенного обратного треугольного разложения для предобусловливания метода сопряженных градиентов // Ж. вычисл. матем. и матем. физики. 2012. Т. 52. № 2. – С.1-26.
24. Якобовский М.В. Инкрементный алгоритм декомпозиции графов // Вестник Нижегородского университета им. Н.В. Лобачевского. Серия «Математическое моделирование и оптимальное управление». Вып.1(28). Нижний Новгород: Издательство ННГУ. 2005. с.243-250.
25. Головченко Е.Н. Параллельный пакет декомпозиции больших сеток // Математическое моделирование. 2011. Т. 23. № 10. 3-18.

## Оглавление

1. Введение .....	3
2. Алгоритм построения предобусловливателя в методе IC2S( $\tau$ )-CG .....	5
3. Некоторые параллельные варианты метода IC2S( $\tau$ )-CG, основанные на использовании специального переупорядочения матрицы коэффициентов .....	7
4. Безотказность методов PIC2S1-CG, PIC2S2-CG, PIC2S3-CG, PIC2S4-CG, PIC2S5-CG.....	18
5. Результаты расчетов .....	20
Список литературы.....	30