

Министерство образования и науки Российской Федерации  
Московский физико-технический институт (государственный университет)

Физтех-школа прикладной математики и информатики  
Кафедра вычислительных технологий и моделирования в геофизике и биоматематике

Выпускная квалификационная работа бакалавра

# Блочные методы типа бисопряжённых градиентов

**Автор:**

Студент 101а группы  
Козлов Николай Андреевич

**Научный руководитель:**

н.с.,к.ф.-м.н.  
Желтков Дмитрий Александрович



Москва 2025

**Аннотация**

Блочный BiCGStab и его друзья  
*Козлов Николай Андреевич*

Краткое описание задачи и основных результатов, мотивирующее  
прочитать весь текст.

**Abstract**

Block BiCGStab and his friends

## Содержание

<b>1</b>	<b>Введение</b>	<b>4</b>
<b>2</b>	<b>Крыловские методы решения систем уравнений</b>	<b>5</b>
2.1	Процедура Арнольди . . . . .	5
2.2	Симметричный алгоритм Ланцоша . . . . .	5
2.3	Метод сопряженных градиентов . . . . .	6
2.4	Процесс биортогонализации Ланцоша . . . . .	8
2.5	Метод бисопряженных градиентов . . . . .	9
2.6	Стабилизированный метод бисопряженных градиентов . . . . .	9
2.7	Блочный метод сопряженных градиентов . . . . .	9
2.8	Блочный метод бисопряженных градиентов . . . . .	9
2.9	Блочный стабилизированный метод бисопряженных градиентов . . . . .	9
2.9.1	Матричнозначные полиномы . . . . .	9
2.9.2	Алгоритм . . . . .	9
2.10	Блочный симметричный метод квазимиимальных невязок . . . . .	9
<b>3</b>	<b>Улучшения блочного метода стабилизированных бисопряженных градиентов</b>	<b>10</b>
3.1	Реортогонализация для поддержания биортогональных соотношений . . . . .	10
3.2	Ортогонализация векторов направлений и проверочных невязок . . . . .	11
3.3	Выбор правых частей . . . . .	12
3.4	Алгоритм . . . . .	12
3.5	Проблемы . . . . .	12
<b>4</b>	<b>Модификация блочного симметричного метода квазимиимальных невязок</b>	<b>14</b>
<b>5</b>	<b>Численные эксперименты</b>	<b>15</b>
<b>6</b>	<b>Заключение</b>	<b>18</b>

## 1 Введение

В ряде приложений возникают большие линейные системы с многими правыми частями. такую задачу можно записать в блочном виде:

$$\begin{array}{c} \overbrace{\hspace{1.5cm}}^N \\ \left[ \begin{array}{c} \boxed{A} \end{array} \right] \begin{array}{c} \overbrace{\hspace{1.5cm}}^s \\ \left[ \begin{array}{c} x_1 \quad \dots \quad x_s \end{array} \right] \end{array} = \begin{array}{c} \overbrace{\hspace{1.5cm}}^s \\ \left[ \begin{array}{c} b_1 \quad \dots \quad b_s \end{array} \right] \end{array} \end{array}$$

$$AX = B,$$

где  $A$  -  $N \times N$  невырожденная разреженная матрица системы;  $B$  -  $N \times s$  невырожденная матрица, столбцы - правые части;  $X$  -  $N \times s$  матрица, столбцы - решения для соответствующих правых частей. Также еще предполагаем, что  $s \ll N$ . Такие задачи можно решать прямыми методами, однако они не подходят для больших задач из-за кубической асимптотики. Так что естественным является использование блочных крыловских методов.

В преимущества блочных крыловских методов входят: высокая производительность на вычислительных системах за счет блочных операций, Более быстрая сходимость, по сравнению с неблочными методами [DIANNE O'LEARY]; в задачах со структурированными системами (например МКЭ) БКМ не разрушают структуру, в отличие от прямых методов. Чрезвычайно большие системы, которые не помещаются целиком в оперативную память можно решать с помощью блочных крыловских методов.

<Рассказ про блочные крыловские методы.>

Для наших целей мы хотим построить крыловские методы, отвечающие следующим требованиям: методы должны находить решения систем общего вида, то есть, которые не обязательно являются эрмитовыми; методы не должны требовать сохранения всего крыловского пространства, то есть должны давать короткие итерационные соотношения; методы должны поддерживать оптимальный размер блока.

## 2 Крыловские методы решения систем уравнений

Как уже было сказано рассматриваемые нами методы базируются на пространстве Крылова, давайте определим его.

**Определение 1.** Пусть  $A$  - матрица порядка  $N$ ,  $v$  - вектор размерности  $N$ . Тогда линейная оболочка вида  $K_m(A, v) \equiv \{v, Av, A^2v, \dots, A^{m-1}v\}$  называется подпространством Крылова, где  $m$  - натуральное число.

Все рассматриваемые в дальнейшем методы являются проекционными процессами. Их суть заключается в том, что приближенное решение ищется в каком-то подпространстве

### 2.1 Процедура Арнольди

Процедура Арнольди - это алгоритм построения ортогонального базиса в крыловском подпространстве  $K_m$ . Алгоритм 1 представляет наиболее простую вариацию такого алгоритма в точной арифметике. По сути алгоритм на каждом шаге ортогонализует

---

#### Алгоритм 1 Алгоритм Арнольди

---

```

1: Выберем  $v_1 = v/\|v\|_2$ , так что  $\|v_1\|_2 = 1$ 

2: for  $j = 1, 2, \dots, m$  do
3:   for  $i = 1, 2, \dots, j$  do
4:      $h_{ij} \leftarrow (Av_j, v_i)$ 
5:   end for
6:    $w_j \leftarrow Av_j - \sum_{i=1}^j h_{ij}v_i$ 
7:    $h_{j+1,j} \leftarrow \|w_j\|_2$ 
8:   if  $h_{j+1,j} = 0$  then
9:     Stop
10:  end if
11:   $v_{j+1} \leftarrow w_j/h_{j+1,j}$ 
12: end for

```

---

$Av_j$  ко всем предыдущим  $v_i$ , применяя процедуру Грама-Шмидта.

Результат работы алгоритма можно записать в матричном виде: обозначим  $V_m$  -  $N \times m$  матрицу со столбцами  $v_1, \dots, v_m$ ;  $\bar{H}_m$  -  $(m+1) \times (m)$  хессенбергова матрица с элементами  $h_{ij}$  из алгоритма 1;  $H_m$  -  $m \times m$  матрица, получающаяся из  $\bar{H}_m$  путем удаления последней строки. Тогда, процедура Арнольди влечет следующие соотношения:

$$AV_m = V_m H_m + w_m e_m^T \quad (1)$$

$$= V_{m+1} \bar{H}_m, \quad (2)$$

$$V_m^T AV_m = H_m \quad (3)$$

### 2.2 Симметричный алгоритм Ланцоша

Симметричный алгоритм Ланцоша - это частный случай процедуры Арнольди, когда матрица  $A$  - симметричная, при таком условии хессенбергова матрица  $H_m$  становится симметричной тридиагональной  $T_m$ . Это позволяет получить короткие итерационные соотношения в теле Алгоритма 2

---

**Алгоритм 2** Симметричный алгоритм Ланцоша

---

```

1: Выберем  $v_1 = v/\|v\|_2$ , так что  $\|v_1\|_2 = 1$ 
2:  $\beta_1 \leftarrow 0, v_0 \leftarrow 0$ 
3: for  $j = 1, 2, \dots, m$  do
4:    $w_j \leftarrow Av_j - \beta_j v_{j-1}$ 
5:    $\alpha_j \leftarrow (w_j, v_j)$ 
6:    $w_j \leftarrow w_j - \alpha_j v_j$ 
7:    $\beta_{j+1} \leftarrow \|w_j\|_2$ 
8:   if  $\beta_{j+1} = 0$  then
9:     Stop
10:  end if
11:   $v_{j+1} \leftarrow w_j/\beta_{j+1}$ 
12: end for

```

---

При этом матрица  $T_m$  имеет вид:

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \ddots & \ddots & \\ & & \ddots & \alpha_{m-1} & \beta_m \\ & & & \beta_m & \alpha_m \end{pmatrix} \quad (4)$$

### 2.3 Метод сопряженных градиентов

Симметричный алгоритм Ланцоша можно использовать для итеративного решения систем линейных уравнений с симметричной положительно определенной матрицей.

Пусть задано начальное приближение  $x_0$ , и векторы направлений из алгоритма Ланцоша  $v_i$ ,  $i = 1, \dots, m$ . На  $m$ -ом шаге алгоритма приближенное решение ищется в аффинном пространстве  $x_0 + K_m$ , где  $K_m(A, r_0) \equiv \{r_0, Ar_0, A^2 r_0, \dots, A^{m-1} r_0\}$ ,  $r_0 = b - Ax_0$ . На невязки при этом налагается условие

$$b - Ax_m \perp K_m. \quad (5)$$

Если взять  $v_1 = r_0/\|r_0\|_2$  и обозначить  $\beta = \|r_0\|_2$ . Тогда  $V_m^T A V_m = T_m$  из (3), а также  $V_m^T r_0 = V_m^T (\beta v_1) = \beta e_1$ . Разложим приближенное решение на  $m$ -ом шаге по базису из векторов  $v_i$ ,  $i = 1, \dots, m$ :

$$x_m = x_0 + V_m y_m. \quad (6)$$

Это выражение эквивалентно равенству:

$$r_m = r_0 - A V_m y_m, \quad (7)$$

домножим слева на  $V_m^T$ :

$$V_m^T r_m = V_m^T r_0 - V_m^T A V_m y_m. \quad (8)$$

Из (5) следует, что  $V_m^T r_m = 0$ , учтём это в (8) и выразим  $y_m$ :

$$y_m = T_m^{-1} \beta e_1. \quad (9)$$

Получим выражение для  $r_m$ :

$$\begin{aligned}
 r_m &= b - A(x_0 + V_m y_m) \\
 &= r_0 - AV_m y_m \\
 &= \beta v_1 - (V_m T_m + h_{m+1,m} v_{m+1} e_m^T) y_m \\
 &= V_m \underbrace{(\beta e_1 - H_m y_m)}_{=0} - h_{m+1,m} e_m^T y_m v_{m+1} \\
 r_m &= -h_{m+1,m} e_m^T y_m v_{m+1}.
 \end{aligned} \tag{10}$$

Из этого выражения следует, что  $r_m \parallel v_{m+1}$ , а значит, что невязки на каждом шаге ортогональны друг другу.

Получим короткие итерационные соотношения для обновления приближенного решения  $x_m$ . LU-разложение матрицы  $T_m$ :

$$T_m = L_m U_m = \begin{pmatrix} 1 & & & & \\ \lambda_2 & 1 & & & \\ & \lambda_3 & \ddots & & \\ & & \ddots & 1 & \\ & & & \lambda_m & 1 \end{pmatrix} \begin{pmatrix} \eta_1 & \beta_2 & & & \\ & \eta_2 & \beta_3 & & \\ & & \ddots & \ddots & \\ & & & \eta_{m-1} & \beta_m \\ & & & & \eta_m \end{pmatrix}$$

Введем обозначения

$$P_m \equiv V_m U_m^{-1}, \tag{11}$$

$$z_m \equiv L_m^{-1} \beta e_1, \tag{12}$$

тогда приближенное решение выражается как

$$x_m = x_0 + P_m z_m. \tag{13}$$

Используя равенство (11) получим формулу для обновления  $p_m$ -последнего столбца  $p_m$  матрицы  $P_m$

$$P_m U_m = V_m \tag{14}$$

$$p_m \eta_m + \beta_m p_{m-1} = v_m \tag{15}$$

$$p_m = \eta_m^{-1} (v_m - \beta_m p_{m-1}) \tag{16}$$

Выразим элементы из последней строчки матрицы  $T_m$  с помощью LU-разложения:

$$\alpha_m = \lambda_m \beta_m + \eta_m \implies \eta_m = \alpha_m - \lambda_m \beta_m$$

$$\beta_m = \lambda_m \eta_{m-1} \implies \lambda_m = \beta_m / \eta_{m-1}$$

В силу вида матрицы  $L_m$ :

$$\begin{aligned}
 z_m &= \begin{pmatrix} z_{m-1} \\ \zeta_m \end{pmatrix}, \\
 \zeta_m &= -\lambda_m \zeta_{m-1}.
 \end{aligned}$$

Как результат получаем формулу для обновления  $x_m$ :

$$x_m = x_{m-1} + \zeta_m p_m$$

Покажем, что столбцы  $P_m$  образуют А-ортогональную систему, т.е, что  $(Ap_i, p_j) = 0$ , для  $i \neq j$ . Для этого нужно показать, что  $P_m^T A P_m$  - диагональная матрица. Подставим определение  $P_m$  в это выражение:

$$P_m^T A P_m = U_m^{-T} V_m^T A V_m U_m^{-1} \quad (17)$$

$$= U_m^{-T} T_m U_m^{-1} \quad (18)$$

$$= U_m^{-T} L_m \quad (19)$$

$U_m^{-T} L_m$  - нижнетреугольная матрица, но она также является и симметричной, так как  $P_m^T A P_m$  - симметричная матрица. Таким образом,  $U_m^{-T} L_m$  - диагональная матрица.

Следствием этого является то, что обновлять приближенное решение можно исходя из поддержания свойств ортогональности невязок и А-ортогональности векторов направлений  $p_i$ . В последующий выкладках вектора  $p_j$  будут нумероваться с нуля, а не с единицы, как это было раньше. А также коэффициенты будут переименованы, чтобы соответствовать общепринятым обозначениям.

$$\begin{aligned} x_{j+1} &= x_j + \alpha_j p_j \implies r_{j+1} = r_j - \alpha_j A p_j \\ \alpha_j &= (r_j, r_j) / (A p_j, r_j) \end{aligned}$$

Из уравнения (16) следует, что

$$\begin{aligned} p_{j+1} &= r_{j+1} + \beta_j p_j \\ \beta_j &= -(r_{j+1}, A p_j) / (p_j, A p_j) = \frac{1}{\alpha_j} (r_{j+1}, (r_{j+1} - r_j)) / (A p_j, p_j) = (r_{j+1}, r_{j+1}) / (r_j, r_j) \end{aligned}$$

Это выражение и А-ортогональность  $p_j$  в свою очередь можно использовать, чтобы преобразовать выражение для  $\alpha_j$ :

$$\begin{aligned} (A p_j, r_j) &= (A p_j, p_j - \beta_{j-1} p_{j-1}) = (A p_j, p_j) \\ \alpha_j &= (r_j, r_j) / (A p_j, p_j) \end{aligned}$$

Теперь у нас есть всё, чтобы записать алгоритм.

---

### Алгоритм 3 Метод сопряженных градиентов

---

```

1:  $r_0 \leftarrow b - A x_0$ ,  $p_0 \leftarrow r_0$ .
2: for  $j = 0, 1, \dots$  do
3:    $\alpha_j \leftarrow (r_j, r_j) / (A p_j, p_j)$ 
4:    $x_{j+1} \leftarrow x_j + \alpha_j p_j$ 
5:    $r_{j+1} \leftarrow r_j - \alpha_j A p_j$ 
6:    $\beta_j \leftarrow (r_{j+1}, r_{j+1}) / (r_j, r_j)$ 
7:    $p_{j+1} \leftarrow r_{j+1} + \beta_j p_j$ 
8: end for
```

---

Этот метод можно адаптировать и для систем общего вида, если домножить обе части уравнения  $Ax = b$  на  $A^T$ , и решать систему с симметричной положительно определенной матрицей  $A^T A$ , однако число обусловленности при этом возрастает в квадрат раз из-за чего данный вариант может давать плохие результаты. [1]

## 2.4 Процесс биортогонализации Ланцоша

Для несимметричных систем можно предъявить алгоритм похожий на симметричный алгоритм Ланцоша, но который будет строить не ортогональный базис в пространстве Крылова, а пару биортогональных базисов в пространствах  $K_m(A, v_1) = \text{span}\{v_1, A v_1, \dots, A^{m-1} v_1\}$  и  $K_m(A^T, v_1) = \text{span}\{v_1, A^T v_1, \dots, (A^T)^{m-1} v_1\}$ .



---

**Алгоритм 4** Процесс биортогонализации Ланцоша

---

```

1: Выберем  $v_1, w_1$  такие что  $(v_1, w_1) = 1$ .
2:  $\beta_1 = \delta_1 \equiv 0, w_0 = v_0 \equiv 0$ 
3: for  $j = 1, 2, \dots, m$  do
4:    $\alpha_j = (Av_j, w_j)$ 
5:    $\hat{v}_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1}$ 
6:    $\hat{w}_{j+1} = A^T w_j - \alpha_j w_j - \delta_j w_{j-1}$ 
7:    $\delta_{j+1} = |(\hat{v}_{j+1}, \hat{w}_{j+1})|^{1/2}$ 
8:   if  $\delta_{j+1} = 0$  then
9:     Stop
10:  end if
11:   $\beta_{j+1} = (\hat{v}_{j+1}, \hat{w}_{j+1})/\delta_{j+1}$ 
12:   $v_{j+1} = \hat{v}_{j+1}/\beta_{j+1}$ 
13:   $w_{j+1} = \hat{w}_{j+1}/\delta_{j+1}$ 
14: end for

```

---

Введём обозначение

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \delta_2 & \alpha_2 & \beta_3 & & \\ & \delta_3 & \ddots & \ddots & \\ & & \ddots & \alpha_{m-1} & \beta_m \\ & & & \delta_m & \alpha_m \end{pmatrix} \quad (20)$$

## 2.5 Метод бисопряженных градиентов

[1]

## 2.6 Стабилизированный метод бисопряженных градиентов

[2]

## 2.7 Блочный метод сопряженных градиентов

[3]

## 2.8 Блочный метод бисопряженных градиентов

[3]

## 2.9 Блочный стабилизированный метод бисопряженных градиентов

[4]

### 2.9.1 Матричнозначные полиномы

Проблемы со сходимостью метода из [4], демонстрация в 4 главе, решение проблемы в 3 главе.

### 2.9.2 Алгоритм

## 2.10 Блочный симметричный метод квазимиимальных невязок

### 3 Улучшения блочного метода стабилизированных бисопряжённых градиентов

В данной главе предложены изменения, направленные на улучшение стабильности блочного метода бисопряжённых градиентов.

#### 3.1 Реортогонализация для поддержания биортогональных соотношений

Для построения базиса в крыловском пространстве и построения невязок алгоритм строится таким образом, чтобы поддерживать следующие соотношения ортогональности:

$$C(\mathbf{Q}_k \mathbf{R}_{k+1}) = 0, \quad (21)$$

$$C^{(1)}(\mathbf{Q}_k \mathbf{P}_k) = 0. \quad (22)$$

Для построения процедуры реортогонализации эти полиномиальные соотношения необходимо перевести в матричный вид. Используя полиномиальное соотношение для  $\mathbf{R}_{k+1}$ , получаем:

$$\mathbf{Q}_k \mathbf{R}_{k+1} = \mathbf{Q}_k \mathbf{R}_k - t \mathbf{Q}_k \mathbf{P}_k \alpha_k \implies S_k = R_k - A P_k \alpha_k$$

Тогда выражение (21) можно представить в виде:

$$\tilde{R}_0^* S_k = 0. \quad (23)$$

В точной арифметике это соотношение выполняется строго, однако при вычислениях на компьютере соотношение (23) выполняется с какой-то погрешностью. Для существенного уменьшения этой погрешности можно произвести ортогонализацию еще раз, взяв  $S_k$  в качестве блока, к которому производится ортогонализация:

$$S_k^r = S_k - A P_k \alpha_k^r. \quad (24)$$

При этом мы стремимся поддерживать соотношение  $\tilde{R}_0^* S_k^r = 0$  с уточненным блоком  $S_k^r$ . Тогда, домножая обе части выражения (24) слева на  $\tilde{R}_0$ , получим уравнение для поправки  $\alpha_k^r$ :

$$(\tilde{R}_0^T A P_k) \alpha_k^r = \tilde{R}_0^T S_k.$$

Поработаем теперь над выражением (22). Используя полиномиальное соотношение для  $P_{k+1}$ , получаем следующее выражение:

$$t \mathbf{Q}_k \mathbf{P}_{k+1} = t \mathbf{Q}_k \mathbf{R}_{k+1} + t \mathbf{Q}_k \mathbf{P}_k. \quad (25)$$

Введем обозначение  $W_k \equiv (t \mathbf{Q}_k \mathbf{R}_{k+1})(A) \circ R_0$ . Тогда выражение (25) можно записать в матричном виде:

$$W_k = A S_k + A P_k \beta_k. \quad (26)$$

Тогда выражение (22) можно представить в виде:

$$\tilde{R}_0^* W_k = 0. \quad (27)$$

Аналогично получаем соотношения реортогонализации для (27):

$$W_k^r = W_k + A P_k \beta_k^r$$

Поправка  $\beta_k$  определяется уравнением:

$$(\tilde{R}_0^* A P_k) \beta_k^r = -\tilde{R}_0^* W_k$$

Следующим шагом получим формулу для вычисления  $P_{k+1}$  с учетом введенных обозначений

$$\begin{aligned} Q_{k+1}P_{k+1} &= Q_{k+1}(R_{k+1} + P_k\beta_k) = \\ &= Q_k R_{k+1} - \omega_k t Q_k R_{k+1} + (1 - \omega_k t) Q_k P_k \beta_k = \\ &= Q_k R_{k+1} - \omega_k t Q_k P_{k+1} + Q_k P_k \beta_k \end{aligned}$$

В матричном виде это выражение записывается как:

$$P_{k+1} = S_{k+1} + P_k \beta_k - \omega_k W_k$$

Для дополнительной минимизации нормы невязки поддерживается следующее соотношение:

$$\langle AS_k, R_{k+1} \rangle_F = 0$$

Для этого выражения также можно выписать процедуру реортогонализации:

$$\begin{aligned} R_{k+1}^r &= R_{k+1} - \omega_k^r T_k \\ \omega_k^r &= \frac{\langle R_{k+1}, T_k \rangle_F}{\langle T_{k+1}, T_k \rangle_F} \end{aligned}$$

### 3.2 Ортогонализация векторов направлений и проверочных невязок

Вспомним вид алгоритма [4], в блоке Алгоритм 5 красным отмечены все места где используется блок векторов направлений  $P_k$ . Легко видеть, что он везде входит в алгоритм вместе матрицей коэффициентов ( $\alpha_k$  и  $\beta_k$ ). Так что если сделать замену  $P_k \leftarrow P_k U$ , где  $U$  -  $s \times s$  матрица, то изменятся лишь сами матрицы коэффициентов, в то время как сами выражения в алгоритме останутся неизменными. Так что можно попробовать

---

#### Алгоритм 5 Блочные стабилизированные бисопряженные градиенты

---

```

 $X_0$  - начальное приближение
 $R_0 = B - AX_0$ 
 $P_0 = R_0$ 
 $\tilde{R}_0$  - произвольная  $N \times s$  матрица
for  $k = 0, 1, 2, \dots$  do
    решить  $\tilde{R}_0^T A P_k \alpha_k = \tilde{R}_0^T R_k$ 
     $S_k = R_k - A P_k \alpha_k$ 
     $T_k = AS_k$ 
     $\omega_k = \frac{\langle T_k, S_k \rangle_F}{\langle T_k, T_k \rangle_F}$ 
     $X_{k+1} = X_k + P_k \alpha_k + \omega_k S_k$ 
     $R_{k+1} = S_k - \omega_k T_k$ 
    решить  $\tilde{R}_0^T A P_k \beta_k = -\tilde{R}_0^T T_k$ 
     $P_{k+1} = R_{k+1} + P_k \beta_k - \omega_k A P_k \beta_k$ 
end for
    
```

---

подобрать такую  $U$ , чтобы вычисления стали более устойчивыми. Например, можно сделать  $QR$ -разложение матрицы  $P_k$ :

$$P_k = Q_P R_P,$$

и в качестве  $U$  взять  $R_P^{-1}$ . Такой выбор  $U$  повлечет ортогонализацию  $P_k$ , что должно улучшить стабильность операций проектирования на вектора направлений.

Как указано в алгоритме 5,  $\tilde{R}_0$  - произвольная матрица, обычно ее выбирают равной  $R_0$ . Аналогично для улучшения стабильности предлагается сделать  $QR$ -разложение матрицы  $R_0$ :

$$R_0 = Q_R R_R,$$

и сделать замену  $\tilde{R}_0 \rightarrow Q_R$ .

### 3.3 Выбор правых частей

Алгоритм перестает сходиться, если блок невязок становится почти вырожденным, поэтому предлагается на этапе инициализации алгоритма сделать  $RRQR$ -разложение блока правых частей, рассмотреть получившуюся перестановку, и выбрать несколько правых частей с номерами, соответствующим первым номерам в перестановке. Благодаря такому выбору рассматривается более линейно-независимый набор столбцов, что положительно сказывается на сходимости.

### 3.4 Алгоритм

---

**Алгоритм 6** Регуляризованный блочный метод стабилизированных бисопряженных градиентов

---

```

 $X_0$  - начальное приближение;
 $R_0 = B - AX_0$ ;
 $P_0 = R_0$ ;
 $R_0 = \tilde{Q}\tilde{U}$  -  $QR$ -разложение  $R_0$ ;
 $\tilde{R}_0 = \tilde{U}$ ;
for  $k = 0, 1, \dots$  do
     $P_k = QU$  -  $QR$ -разложение  $P_k$ ;
     $P_k \rightarrow P_k U^{-1}$ ;
     $V_k = AP_k$ ;
    решить  $(\tilde{R}_0^* V_k) \hat{\alpha}_k = \tilde{R}_0^* R_k$ ;
     $\hat{S}_k = R_k - V_k \hat{\alpha}_k$ ;
    решить  $(\tilde{R}_0^* V_k) \alpha_k = \tilde{R}_0^* \hat{S}_k$ ;
     $S_k = \hat{S}_k - V_k \alpha_k$ ;
     $T_k = AS_k$ ;
     $\hat{\omega}_k = \langle S_k, T_k \rangle_F / \langle T_k, T_k \rangle_F$ ;
     $\hat{R}_{k+1} = S_k - \hat{\omega}_k T_k$ ;
     $\omega_k = \langle \hat{R}_{k+1}, T_k \rangle_F / \langle T_k, T_k \rangle_F$ ;
     $R_{k+1} = \hat{R}_{k+1} - \omega_k T_k$ ;
     $X_{k+1} = X_k + P_k(\hat{\alpha}_k + \alpha_k) + (\hat{\omega}_k + \omega_k) S_k$ ;
    решить  $(\tilde{R}_0^* V_k) \hat{\beta}_k = -\tilde{R}_0^* T_k$ ;
     $\hat{W}_k = T_k + V_k \hat{\beta}_k$ ;
    решить  $(\tilde{R}_0^* V_k) \beta_k = -\tilde{R}_0^* \hat{W}_k$ ;
     $W_k = \hat{W}_k + V_k \beta_k$ ;
     $P_{k+1} = S_k + P_k(\hat{\beta}_k + \beta_k) - (\hat{\omega}_k + \omega_k) W_k$ ;
end for

```

---

### 3.5 Проблемы

В данном алгоритме возможны аварийные остановки, в случаях, когда матрица  $\tilde{R}_0^* AP_k$  становится вырожденной. В такой ситуации авторы [4] предлагают провести рестарт с другой  $\tilde{R}_0$ .

Но главным недостатком алгоритма [4] является выбор  $\omega$  в виде скалярной матрицы, из-за этого чем больше размер блока мы берем для расчета, тем меньше по модулю становится  $\omega$ , что в свою очередь ведет к стагнации алгоритма. Наша модификация алгоритма также страдает от этой проблемы. Была надежда, что получится обобщить метод на случай, когда  $\omega_k$  - произвольная  $s \times s$  матрица, но в ходе исследования выяснилось, что это невозможно из-за некоммутативности матричнозначных полиномов, которая не позволяет получить короткие итерационные формулы.

#### 4 Модификация блочного симметричного метода квазимиимальных невязок

Один из главных элементов блочного симметричного метода квазимиимальных невязок [5] является процесс Грамма-Шмидта с квазискалярным произведением. Далее будет представлена модификация этого алгоритма, использующая процесс Грамма-Шмидта с настоящим скалярным произведением (QR-разложение), но приводящая к тем же рекуррентным формулам. Тогда откроется простор для более устойчивых реализаций QR-разложения и их библиотечным имплементациям, а также к нескольким новым свойствам невязок.

Симметричный процесс Ланцоша приводит к следующему матричному соотношению:

$$A \begin{pmatrix} V_1 & \dots & V_k & V_{k+1} \end{pmatrix} = \begin{pmatrix} V_1 & \dots & V_k & V_{k+1} \end{pmatrix} \begin{pmatrix} \alpha_1 & \delta_1 & & & \\ \beta_2 & \alpha_2 & \delta_2 & & \\ & \beta_3 & \ddots & \ddots & \\ & & \ddots & \alpha_{m-1} & \delta_{m-1} \\ & & & \beta_m & \alpha_m \\ & & & & \beta_{m+1} \end{pmatrix}, \quad (28)$$

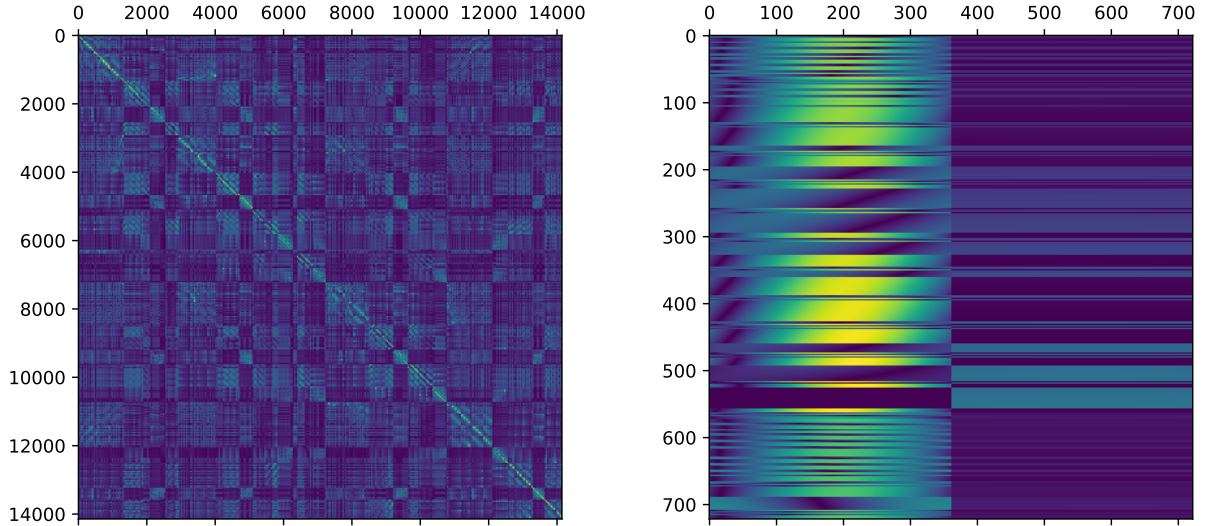
где  $\delta_{i-1} = \beta_i^T$  в версии из статьи [5], в нашей модификации же получится другой вид для этой матрицы коэффициентов. Из соотношения (28), выделяя последний блок, получаем

$$AV_k = V_{k+1} \delta_k \quad (29)$$

В ходе итераций поддерживается соотношение  $V_i^T V_j = 0, i \neq j$ .

## 5 Численные эксперименты

Тесты производились на интересующей нас задаче – линейной системе с многими правыми частями, возникающей при решении задачи электромагнитного рассеяния методом интегральных уравнений [STAVTSEV]. Порядок системы – 14144, всего правых частей – 722. Каждая правая часть соответствует разным углам падения, а также первая половина правых частей отличается от правой типом поляризации.



(a) Матрица системы в логарифмическом масштабе

(b) первые 722 строки правой части в логарифмическом масштабе

Рис. 1

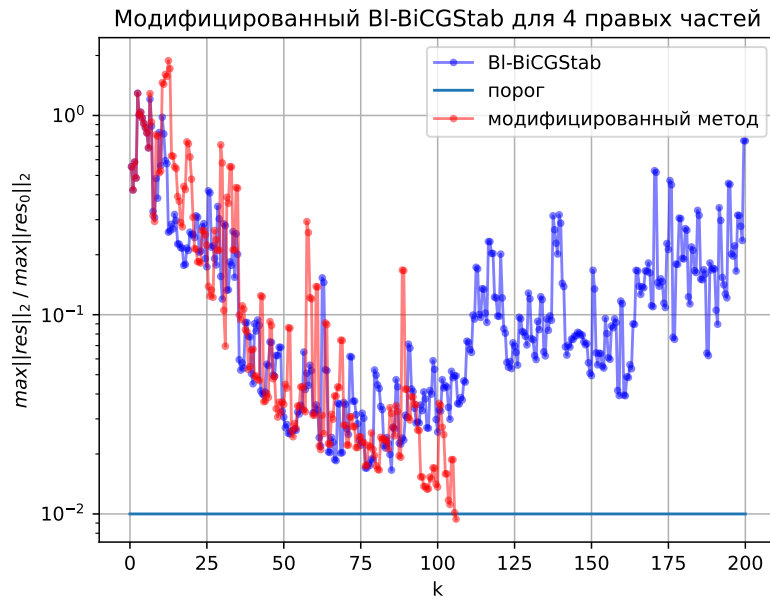


Рис. 2

Первый тест демонстрирует, что метод из статьи [4] не сходится с требуемой точно-

стью, в то время как версия с улучшениями, описанными в главе 3, сходится линейно без проблем. Эксперимент проводился в одинарной точности для четырех правых частей с номерами: 0, 90, 180, 270. Его результаты представлены на рис.2

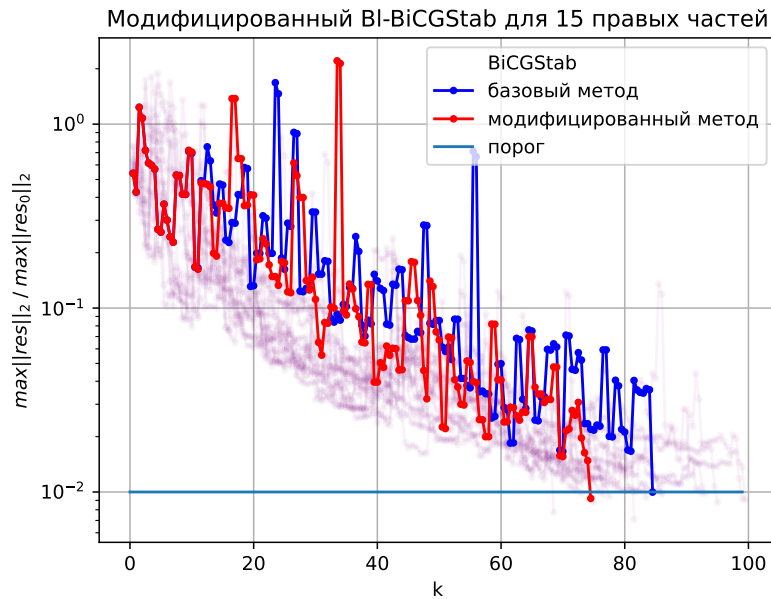


Рис. 3

15 правых частей, уменьшения числа итераций, считаем в двойной точности

Второй тест демонстрирует, что улучшения, описанные в главе 3 позволяют получить выгоду по количеству матвеков, по сравнению с решением систем с каждой правой частью в отдельности. Эксперимент проводился в двойной точности с 15 правыми частями, выбранными с помощью RRQR. Его результаты представлены на рис.3. По оси абсцисс - количество итераций, по оси ординат - относительная максимальная невязка в блоке. Фиолетовым изображено падение невязки при решении задачи с каждой правой частью в отдельности, синим - метод из статьи [4], красным - метод с улучшениями из главы 3. Для решения этой задачи стабилизированными бисопряженными градиентами было потрачено 2525 матвеков, для решения методом из статьи [GUENNOUNI] - 2535, модифицированный метод сошелся за 2235 матвеков, таким образом выгода составляет 12% по сравнению с неблочной версией.

более 30 правых частей, демонстрация отсутствия взрыва невязки



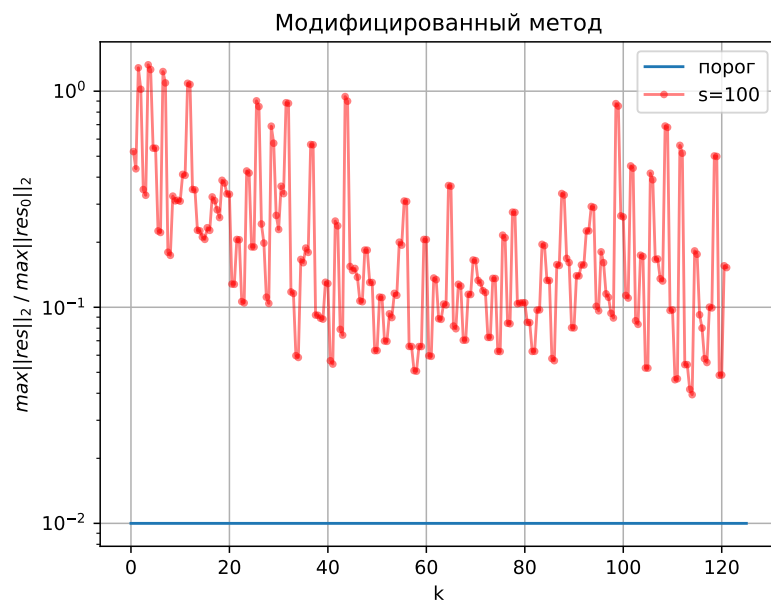


Рис. 4

## 6 Заключение

Результаты

Нерешенные проблемы редукции блока

## Список литературы

- [1] *Saad, Yousef*. Iterative Methods for Sparse Linear Systems / Yousef Saad. — 2nd edition. — Philadelphia, PA: Society for Industrial and Applied Mathematics, 2003.
- [2] *van der Vorst, H. A.* Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems / H. A. van der Vorst // *SIAM Journal on Scientific and Statistical Computing*. — 1992. — Vol. 13, no. 2. — Pp. 631–644. <https://doi.org/10.1137/0913035>.
- [3] *O’Leary, Dianne P.* The block conjugate gradient algorithm and related methods / Dianne P. O’Leary // *Linear Algebra and its Applications*. — 1980. — Vol. 29. — Pp. 293–322. — Special Volume Dedicated to Alson S. Householder. <https://www.sciencedirect.com/science/article/pii/0024379580902475>.
- [4] *el Guennouni A., Jbilou K. Sadok H.* A block version of BiCGSTAB for linear systems with multiple right-hand sides. / Jbilou K. Sadok H. el Guennouni, A. // *ETNA. Electronic Transactions on Numerical Analysis [electronic only]*. — 2003. — Vol. 16. — Pp. 129–142. <http://eudml.org/doc/124803>.
- [5] *Boyse, William E.* A Block QMR Method for Computing Multiple Simultaneous Solutions to Complex Symmetric Systems / William E. Boyse, Andrew A. Seidl // *SIAM Journal on Scientific Computing*. — 1996. — Vol. 17, no. 1. — Pp. 263–274. <https://doi.org/10.1137/0917019>.