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.

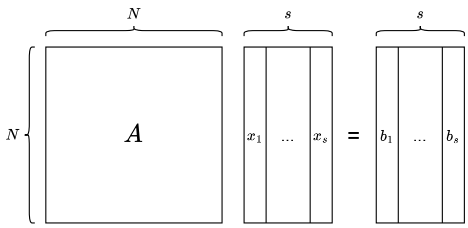
Abstract

Block BiCGStab and his friends

Содержание

1 Введение

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



$$AX = B,$$

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 сложности. 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 [1]; 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;

2 Крыловские методы решения систем уравнений

Ключевым объектом в рассматриваемом классе методов является пространство Крылова, определим его.

Определение 2.1. *Пусть A - матрица порядка N, v - вектор размерности N. Тогда линейная оболочка вида $K_m(A,v) \equiv \{v,Av,A^2v,...,A^{m-1}v\}$ называется подпространством Крылова, где m - натуральное число.*

Все рассматриваемые в дальнейшем методы являются проекционными. В таких методах приближенное решение ищется в крыловском пространстве при этом решение на подпространстве ищется, как правило, на основе некоторого проекционного соотношения (которое и задаёт метод).

2.1 Процедура Арнольди [2]

Процедура Арнольди - это алгоритм построения ортогонального базиса в крыловском подпространстве K_m. Алгоритм 1 представляет наиболее простую вариацию такого алгоритма в точной арифметике.

Алгоритм 1 Алгоритм Арнольди

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

for $j = 1, 2, \ldots, m$ **do**
    for $i = 1, 2, \ldots, j$ **do**
        $h_{ij} \leftarrow (Av_j, v_i)$
    end for
    $w_j \leftarrow Av_j - \sum_{i=1}^{j} h_{ij} v_i$
    $h_{j+1,j} \leftarrow \|w_j\|_2$
    if $h_{j+1,j} = 0$ **then**
        Stop
    end if
    $v_{j+1} \leftarrow w_j / h_{j+1,j}$
end for

Алгоритм на каждом шаге ортогонализует 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,...,v_m; $\overline{H}_m$ - $(m+1) \times (m)$ хессенбергова матрица с элементами h_{ij} из алгоритма 1; H_m - $m \times m$ матрица, получающаяся из $\overline{H}_m$ путем удаления последней строки. Тогда, процедура Арнольди влечет следующие соотношения:

$$AV_m = V_m H_m + w_m e_m^T \tag{1}$$

$$= V_{m+1} \overline{H}_m, \tag{2}$$

$$V_m^T A V_m = H_m \tag{3}$$

2.2 Симметричный алгоритм Ланцоша [2]

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 рекуррентные соотношения, приведённые в Алгоритме 2

Алгоритм 2 Симметричный алгоритм Ланцоша

Выберем $v_1 = v/\|v\|_2$, так что $\|v_1\|_2 = 1$
$\beta_1 \leftarrow 0$, $v_0 \leftarrow 0$
for $j = 1, 2, \ldots, m$ **do**
    $w_j \leftarrow Av_j - \beta_j v_{j-1}$
    $\alpha_j \leftarrow (w_j, v_j)$
    $w_j \leftarrow w_j - \alpha_j v_j$
    $\beta_{j+1} \leftarrow \|w_j\|_2$
    if $\beta_{j+1} = 0$ **then**
        Stop
    end if
    $v_{j+1} \leftarrow w_j/\beta_{j+1}$
end for

При этом матрица T_m имеет вид:

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

2.3 Метод сопряженных градиентов [2]

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, $i = 1,...,m$. На m-ом шаге алгоритма приближенное решение ищется в аффинном пространстве $x_0 + K_m$, где $K_m\,(A,r_0) \equiv \{r_0, Ar_0, A^2r_0, ..., A^{m-1}r_0\}$, $r_0 = b - Ax_0$. На невязки при этом налагается условие

$$b - Ax_m \perp K_m. \tag{5}$$

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, $i = 1,...,m$:

$$x_m = x_0 + V_m y_m. \tag{6}$$

Это выражение эквивалентно равенству:

$$r_m = r_0 - AV_m y_m, \tag{7}$$

домножим слева на V_m^T:

$$V_m^T r_m = V_m^T r_0 - V_m^T A V_m y_m. \tag{8}$$

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

$$y_m = T_m^{-1}\beta e_1. \tag{9}$$

Получим выражение для r_m:

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

$$r_m = -t_{m+1,m} e_m^T y_m v_{m+1}. \tag{10}$$

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:

$$\begin{gathered} \alpha_m = \lambda_m \beta_m + \eta_m \implies \eta_m = \alpha_m - \lambda_m \beta_m \\ \beta_m = \lambda_m \eta_{m-1} \implies \lambda_m = \beta_m / \eta_{m-1} \end{gathered}$$

В силу вида матрицы L_m:

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

Как результат получаем формулу для обновления x_m:

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

Покажем, что столбцы P_m образуют A-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:

$$P_m^T A P_m = U_m^{-T} V_m^T A V_m U_m^{-1} \tag{17}$$

$$= U_m^{-T} T_m U_m^{-1} \tag{18}$$

$$= U_m^{-T} L_m \tag{19}$$

$U_m^{-T} L_m$ - 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 A-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.

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

Из уравнения (16) после перенормировки $p_i, i = 1,...,m$ следует, что

$$p_{j+1} = r_{j+1} + \beta_j p_j$$
$$\beta_j = -(r_{j+1}, Ap_j)/(p_j, Ap_j) = \frac{1}{\alpha_j}(r_{j+1}, (r_{j+1} - r_j))/(Ap_j,p_j) = (r_{j+1}, r_{j+1})/(r_j,r_j)$$

Это выражение и A-ортогональность p_j в свою очередь можно использовать, чтобы преобразовать выражение для α_j:

$$(Ap_j,r_j) = (Ap_j,p_j - \beta_{j-1}p_{j-1}) = (Ap_j,p_j)$$
$$\alpha_j = (r_j,r_j)/(Ap_j, p_j)$$

Теперь у нас есть всё, чтобы записать алгоритм.

Алгоритм 3 Метод сопряженных градиентов

1: $r_0 \leftarrow b - Ax_0$, $p_0 \leftarrow r_0$.
2: **for** $j = 0, 1, \ldots$ **do**
3: $\quad\alpha_j \leftarrow (r_j, r_j)/(Ap_j, p_j)$
4: $\quad x_{j+1} \leftarrow x_j + \alpha_j p_j$
5: $\quad r_{j+1} \leftarrow r_j - \alpha_j A p_j$
6: $\quad\beta_j \leftarrow (r_{j+1}, r_{j+1})/(r_j, r_j)$
7: $\quad p_{j+1} \leftarrow r_{j+1} + \beta_j p_j$
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.

2.4 Процесс биортогонализации Ланцоша [2]

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 $K_m(A, v_1) =$

Алгоритм 4 Процесс биортогонализации Ланцоша

```
Выберем v_1, w_1 такие что (v_1, w_1) = 1.
β_1 = δ_1 ≡ 0, w_0 = v_0 ≡ 0
for j = 1, 2, ..., m do
    α_j = (A v_j, w_j)
    v̂_{j+1} = A v_j − α_j v_j − β_j v_{j−1}
    ŵ_{j+1} = A^T w_j − α_j w_j − δ_j w_{j−1}
    δ_{j+1} = |(v̂_{j+1}, ŵ_{j+1})|^{1/2}
    if δ_{j+1} = 0 then
        Stop
    end if
    β_{j+1} = (v̂_{j+1}, ŵ_{j+1}) / δ_{j+1}
    v_{j+1} = v̂_{j+1} / β_{j+1}
    w_{j+1} = ŵ_{j+1} / δ_{j+1}
end for
```

$span\{v_1,Av_1,...,A^{m-1}v_1\}$ и $K_m(A^T, v_1) = span\{v_1,A^Tv_1,...,(A^T)^{m-1}v_1\}$, то есть такую пару $v_1,...,v_m$ и $w_1,...,w_m$, что $(v_i,w_j) = \delta_{ij}$, $1 \leq i,\, j \leq m$.

Введём обозначения:

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

$$W_m = \begin{bmatrix} w_1 & ... & w_m \end{bmatrix}$$

Тогда легко убедиться, что если на m-ом шаге не произошло аварийной остановки, то алгоритм порождает следующие матричные соотношения:

$$\begin{aligned} AV_m &= V_mT_m + \delta_{m+1}v_{m+1}e_m^T \\ A^TW_m &= W_mT_m^T + \beta_{m+1}w_{m+1}e_m^T \\ W_m^TAV_m &= T_m \end{aligned}$$

2.5 Метод бисопряженных градиентов [2]

Метод бисопряженных градиентов выводится из процесса биортогонализации Ланцоша аналогично тому, как метод сопряженных градиентов выводился из симметричного процесса Ланцоша. Прибилиженное решение на m-ом шаге будет искаться как наилучшее приближение в пространстве $x_0 + K_m$, где $K_m = \{v_1, Av_1,...,A^{m-1}v_1\}$, так, чтобы невязка r_m была ортогональна пространству $L_m = w_1,A^Tw_1,...,(A^T)^{m-1}w_1$. Так же, как и при выводе сопряженных градиентов возьмём $v_1 = r_0/\|r_0\|_2$, а вектор w_1 можно взять произвольным, таким что $(v_1, w_1) \neq 0$, например, v_1. Алгоритм будет решать не только систему $Ax = b$, но и некоторую двойственную систему $A^Tx^* = b^*$ (причём $b^* = b$, если $w_1 = v_1$). Производим LU-разложение для матрицы T_m, полученной в ходе процесса биортогонализации Ланцоша:

$$T_m = L_mU_m,$$

и вводим обозначения для векторов направлений p_i, p_i^*:

$$\begin{aligned} P_m &= V_m U_m^{-1} \\ P_m^* &= W_m L_m^{-T} \\ \begin{bmatrix} p_1 & ... & p_m \end{bmatrix} &= P_m \\ \begin{bmatrix} p_1^* & ... & p_m^* \end{bmatrix} &= P_m^* \end{aligned}$$

Приближенное решение выражается также как и в методе сопряженных градиентов:

$$x_m = x_0 + P_m L_m^{-1}(\beta e_1).$$

И, аналогично методу сопряженных градиентов невязки окажутся сонаправлены векторам из базиса:

$$\begin{gathered} r_j \parallel v_{j+1},\ j = 1,...,m \\ r_j^* \parallel v_{j+1}^*,\ j = 1,...,m \end{gathered}$$

Отсюда следует, что эти наборы невязок биортогональны:

$$(r_j^*, r_i) = 0, \text{ при } 1 \le i,j \le m,\ i \ne j \tag{20}$$

Легко показать, что наборы векторов p_i^*, $i = 1,...,m$ и p_i, $i = 1,...,m$ - А-ортогональны:

$$(P_m^*)^T A P_m = L_m^{-1} W_m^T A V_m U_m^{-1} = L_m^{-1} T_m U_m^{-1} = I.$$

Благодаря полученнным свойствам биортогональности невязок и А-биортогональности векторов направлений, аналогичным же образом можно получить короткие итерационные соотношения для получения новых векторов p_i, r_i , x_i, и записать окончательный алгоритм: К сожалению, данный метод на практике проявляет нерегулярное уменьше-

Алгоритм 5 Метод бисопряженных градиентов

$r_0 \leftarrow b - Ax_0$, r_0^* т.ч. $(r_0^*, r_0) \ne 0$, например $r_0^* = r_0$
$p_0 \leftarrow r_0$, $p_0^* = r_0^*$
for $j = 0, 1, \ldots$ **do**
  $\alpha_j \leftarrow (r_j^*, r_j)/(p_j^*, Ap_j)$
  $x_{j+1} \leftarrow x_j + \alpha_j p_j$
  $r_{j+1} \leftarrow r_j - \alpha_j A p_j$
  $r_{j+1}^* \leftarrow r_j^* - \alpha_j A^T p_j^*$
  $\beta_j \leftarrow (r_{j+1}^*, r_{j+1})/(r_j^*, r_j)$
  $p_{j+1} \leftarrow r_{j+1} + \beta_j p_j$
  $p_{j+1}^* \leftarrow r_{j+1} + \beta_j p_j^*$
end for

ние невязки и тратит вычислительные ресурсы на поиск решения двойственной задачи, которая нас не интересует. Для решения этих проблем был придуман метод стабилизированных бисопряженных градиентов.

2.6 Стабилизированный метод бисопряженных градиентов [3]

Невязки, полученные при помощи метода бисопряженных градиентов r_k и r_k^* лежат в пространствах Крылова $K_{m+1}^r(A, r_0) = \{r_0, Ar_0, ..., A^m r_0\}$ и $K_{m+1}^l(A^T, r_0^*) = \{r_0^*, (A^T)r_0^*, ..., (A^T)^m r_0^*\}$ соответственно, следовательно, их можно выразить с помощью многочлена от матрицы:

$$r_k = \mathcal{R}_k(A)r_0$$
$$r_k^* = \mathcal{Q}_k(A^T)r_0^*$$

Из вида итерационных соотношений легко видеть, что $\mathcal{R}_k \equiv \mathcal{Q}_k$, и $\mathcal{R}_k(0) = 1$.

Как было показано в предыдущем пункте, метод бисопряженных градиентов работает за счёт поддержания ортогонализационных соотношений на невязки и вектора направлений. Преобразуем соотношение для невязок (20):

$$(r_j^*, r_i) = 0$$
$$(\mathcal{Q}_j(A^T)r_0^*, \mathcal{R}_i(A)r_0) = (r_0^*, \mathcal{Q}_j(A)\mathcal{R}_i(A)r_0) = 0$$

$$(r_0^*, \mathcal{Q}_j(A)\mathcal{R}_i(A)r_0) = 0 \tag{21}$$

Из этого же соотношения (20) следует, что $r_j \perp K_j^l(A^T, r_0^*)$, значит, выражение (21) верно для любого многочлена $\mathcal{Q}_j$ порядка j. В частности рассмотрим

$$\mathcal{Q}_j(t) = (1 - \omega_0 t)(1 - \omega_1 t) \cdot ... \cdot (1 - \omega_{j-1} t) \tag{22}$$

И будем выбирать ω_j так, чтобы минимизировать норму r_i. Итерационные соотношения в методе бисопряженных градиентов можно записать в полиномиальном виде:

$$\mathcal{R}_{i+1} = \mathcal{R}_i - t\alpha_i \mathcal{P}_i \tag{23}$$
$$\mathcal{P}_{i+1} = \mathcal{P}_i + \beta_i \mathcal{P}_i, \tag{24}$$

где аналогично невязкам r_i вектора направлений p_i были выражены через полином от матрицы системы как $p_i = \mathcal{P}_i(A)r_0$.

Опираясь на (21) введём обозначение для стабилизированных невязок:

$$r_i = \mathcal{Q}_i(A)\mathcal{R}_i(A)r_0 \tag{25}$$

Получим короткие итерационные соотношения для обновления стабилизированной невязки:

$$\mathcal{Q}_{i+1}(A)\mathcal{R}_{i+1}(A)r_0 = (1 - \omega_i A)\mathcal{Q}_i(A)(\mathcal{R}_i(A) - \alpha_i \mathcal{P}_i(A))r_0 =$$
$$= \{\mathcal{Q}_i(A)\mathcal{R}_i(A) - \alpha_i A\mathcal{Q}_i(A)\mathcal{P}_i(A)\}r_0 - \omega_i A\{\mathcal{Q}_i(A)\mathcal{R}_i(A) - \alpha_i A\mathcal{Q}_i(A)\mathcal{P}_i(A)\ \}r_0$$

Обозначим $s_i = r_i - \alpha_i A p_i$, тогда обновление стабилизированной невязки будет производится по следующему соотношению:

$$r_{i+1} = s_i - \omega_i A s_i \tag{26}$$

Выберем ω_i так, чтобы минимизировать норму невязки r_{i+1}:

$$\omega_i = \frac{(As_i, s_i)}{(As_i, As_i)} \tag{27}$$

Аналогично введем обозначение для новых векторов направлений:

$$p_i = \mathcal{Q}_i(A)\mathcal{P}_i(A)r_0 \tag{28}$$

и получим рекуррентные соотношения для них:

$$\begin{aligned}\mathcal{Q}_{i+1}(A)\mathcal{P}_{i+1}(A)r_0 &= \mathcal{Q}_{i+1}(A)(\mathcal{R}_{i+1}(A) + \beta_i\mathcal{P}_i(A))r_0 \\ &= \mathcal{Q}_{i+1}(A)\mathcal{R}_{i+1}(A)r_0 + \beta_i\{\mathcal{Q}_i(A)\mathcal{P}_i(A)r_0 - \omega_i A\mathcal{Q}_i(A)\mathcal{P}_i(A)\}r_0\end{aligned}$$

$$p_{i+1} = r_{i+1} + \beta_i(p_i - \omega_i A p_i) \tag{29}$$

Теперь необходимо выразить коэффициенты β_i, α_i с помощью новых невязок и направлений.

$$\tilde{\rho}_{i+1} \equiv (r_0^*, r_{i+1}) = (r_0^*, \mathcal{Q}_{i+1}(A)\mathcal{R}_{i+1}(A)r_0) = (\mathcal{Q}_{i+1}(A^T)r_0^*, \mathcal{R}_{i+1}(A)r_0)$$

Как уже говорилось, для невязок, полученных при помощи метода бисопряженных градиентов справедливо, что $\mathcal{R}_j(A)r_0 \perp K_j^l(A^T, r_0^*)$, следовательно от $\mathcal{Q}_{i+1}(A^T)$ останется только $(-1)^i\omega_0...\omega_i(A^T)^i$. В методе бисопряженных градиентов $\mathcal{Q}_i \equiv \mathcal{R}_i$, следовательно,

$$\rho_{i+1} \equiv (\mathcal{R}_{i+1}(A^T)r_0^*, \mathcal{R}_{i+1}(A)r_0) = (-1)^i\alpha_0 \cdot ... \cdot \alpha_i((A^T)^i, \mathcal{R}_{i+1})$$

Таким образом, можно выразить β_i через новые невязки и направления:

$$\beta_i = \rho_{i+1}/\rho_i = (\alpha_i/\omega_i)(\tilde{\rho}_{i+1}/\tilde{\rho}_i) = \frac{(r_0^*, r_{i+1})}{(r_0^*, r_i)}\frac{\alpha_i}{\omega_i} \tag{30}$$

Аналогичным образом можно показать, что

$$\alpha_i = \frac{(r_0^*, r_i)}{(r_0^*, Ap_i)} \tag{31}$$

Наконец, можно записать окончательный вид алгоритма:

Алгоритм 6 Стабилизированные бисопряженные градиенты

$r_0 \leftarrow b - Ax_0$, r_0^* т.ч. $(r_0^*, r_0) \neq 0$, например $r_0^* = r_0$
$p_0 \leftarrow r_0$, $p_0^* = r_0^*$
for $k = 0, 1, \ldots$ **do**
  $v_k \leftarrow Ap_k$
  $\alpha_k \leftarrow (r_0^*, r_k)/(r_0^*, v_k)$
  $s_k \leftarrow r_k - \alpha_k v_k$
  $t_k \leftarrow As_k$
  $\omega_k = (t_k, s_k)/(t_k, t_k)$
  $x_{k+1} \leftarrow x_k + \alpha_j p_j + \omega_k s_k$
  $r_{k+1} \leftarrow r_k - \omega_k A t_k$
  $\beta_k \leftarrow \frac{(r_0^*, r_{k+1})}{(r_0^*, r_k)}\frac{\alpha_k}{\omega_k}$
  $p_{k+1} \leftarrow r_{k+1} + \beta_k(p_k - \omega_k v_k)$
end for

2.7 Блочный метод бисопряженных градиентов [1]

Для решения линейных систем с многими правыми частям $AX = B$, где A - невырожденная $N \times N$ матрица, B - $N \times s$ блок (матрица) правых частей с полным рангом, блочный метод бисопряженных градиентов строит 2 набора блоков направлений: $\{P_0, ..., P_k\}$ и $\{\tilde{P}_0, ..., \tilde{P}_k\}$, чьи столбцы своей линейной оболочкой порождают блочные пространства Крылова $K_{k+1}^r(A, R_0) = \{R_0, AR_0, ..., A^k R_0\}$, где $R_0 = B - AX_0$ и

$K^l_{k+1}(A^T,\tilde{R}_0) = \{\tilde{R}_0,A^T\tilde{R}_0,...,(A^T)^k\tilde{R}_0\}$, где $\tilde{R}_0$ - произвольная $N \times s$ матрица (фигурные скобки обозначают линейную оболочку столбцов матриц в наборе), и два набора блоков невязок: $\{R_0,...,R_k\}$ и $\{\tilde{R}_0,...,\tilde{R}_k\}$ так, чтобы для них выполнялись блочные соотношения ортогональности:

$$\tilde{R}_i^T R_j = 0, \text{ для } i < j \tag{32}$$
$$\tilde{P}_i^T A P_j = 0, \text{ для } i < j \tag{33}$$
$$R_i^T \tilde{R}_j = 0, \text{ для } i < j \tag{34}$$
$$P_i^T A \tilde{P}_j = 0, \text{ для } i < j \tag{35}$$
$$\tag{36}$$

И выглядит он следующим образом: алгоритм останавливается, если хотя бы одна из

Алгоритм 7 Блочный метод биспоряженных градиентов

X_0 - $N \times s$ блок начальных приближений, $R_0 = B - AX_0$
$\tilde{R}_0$ - произвольная $N \times s$ матрица
$P_0 = R_0$, $\tilde{P}_0 = \tilde{R}_0$
for $k = 0,1,...$ **do**
  $\alpha_k \leftarrow (\tilde{P}_k^T A P_k)^{-1} \tilde{R}_k^T R_k$
  $X_{k+1} \leftarrow X_k + P_k \alpha_k$
  $R_{k+1} \leftarrow R_k - AP_k\alpha_k$
  $\tilde{\alpha}_k = (P_k^T A^T \tilde{P}_k)^{-1} R_k^T \tilde{R}_k$
  $\beta_k = (\tilde{R}_k^T R_k)^{-1} \tilde{R}_{k+1}^T R_{k+1}$
  $\tilde{\beta}_k = (R_k^T \tilde{R}_k)^{-1} R_{k+1}^T \tilde{R}_{k+1}$
  $P_{k+1} = R_{k+1} + P_k\beta_k$
  $\tilde{P}_{k+1} = \tilde{R}_{k+1} + \tilde{P}_k\tilde{\beta}_k$
end for

матриц: $\tilde{P}_k^T A P_k$ или $\tilde{R}_k^T R_k$ становится вырожденной.

При $s = 1$ приведённый метод эквивалентен бисопряженным градиентам.

2.8 Блочный метод сопряженных градиентов [1]

Если матрица A - симметричная и положительно определенная и $\tilde{R}_0 = R_0$, то блочный метод бисопряженных градиентов превращается в блочный метод сопряженных градиентов: Важным свойством этого алгоритма является то, что для него существует

Алгоритм 8 Блочный метод сопряженных градиентов

X_0 - $N \times s$ блок начальных приближений, $R_0 = B - AX_0$
$P_0 = R_0$
for $k = 0,1,...$ **do**
  $\alpha_k \leftarrow (P_k^T A P_k)^{-1} R_k^T R_k$
  $X_{k+1} \leftarrow X_k + P_k \alpha_k$
  $R_{k+1} \leftarrow R_k - AP_k\alpha_k$
  $\beta_k = (R_k^T R_k)^{-1} R_{k+1}^T R_{k+1}$
  $P_{k+1} = R_{k+1} + P_k\beta_k$
end for

теорема сходимости:

Теорема 2.1. *После k шагов блочного метода сопряженных градиентов, ошибка дял i-ой правой части $e_i^{(k)} = x_i^{(k)} - x_i^*$ ограничена как:*

$$e_i^{(k)T} A e_i^{(k)} \leq \left(\frac{1 - \sqrt{\kappa^{-1}}}{1 + \sqrt{\kappa^{-1}}} \right)^{2k} c, \tag{37}$$

где $\kappa = \lambda_N / \lambda_s$, c - некоторая константа.

Если решать линейную систему с многими правыми частями для каждой правой части в отдельности методом сопряженных градиентов, то скорость сходимости будем определяться числом обучловленности системы: $\kappa = \lambda_N/\lambda_1$. Если же решать блочным методом сопряженных градиентов, то скорость сходимости определяется не нижней границей спектра λ_1, а s-ой снизу компонентой λ_s, что может существенно повысить скорость сходимости.

2.9 Блочный стабилизированный метод бисопряженных градиентов [4]

Блочное обобщение стабилизированных бисопряженных градиентов производится при помощи матричнозначных полиномов. Так что следует рассмотреть их определение и некоторые свойства.

2.9.1 Матричнозначные полиномы

Определение 2.2. *Матричнозначным полиномом $\mathcal{P}$ порядка k называется полином вида:*

$$\mathcal{P}(t) = \sum_{i=0}^{k} t^i \Omega_i,$$

где Ω_i - $s \times s$ матрица, а t - число.

Определение 2.3 (Операции с матричнозначными полиномами)**.**

$$\mathcal{P}(A) \circ Y = \sum_{i=0}^{k} A^i Y \Omega_i, \text{ где } Y \text{ - } N \times s \text{ матрица}$$

$$(\mathcal{P}\Theta)(t) = \sum_{i=0}^{k} t^i \Omega_i \Theta, \text{ где } \Theta \text{ - } s \times s \text{ матрица}$$

Утверждение 2.1 (Свойства операций с матричнозначными полиномами)**.**

$$(\mathcal{P}(A) \circ Y)\Theta = (\mathcal{P}\Theta)(A) \circ Y$$
$$(\mathcal{P} + \mathcal{Q})(A) \circ Y = \mathcal{P}(A) \circ Y + \mathcal{Q}(A) \circ Y$$

2.9.2 Алгоритм

Пользуясь свойствами из утверждения 2.1, аналогично методу бисопряженных градиентов можно выразить блок невязок и блок направлений с помощью матричнозначного полинома от матрицы системы:

$$R_k = \mathcal{R}_k(A) \circ R_0 \tag{38}$$
$$P_k = \mathcal{P}_k(A) \circ R_0 \tag{39}$$
$$\mathcal{R}_{k+1}(t) = \mathcal{R}_k(t) - t\mathcal{P}(t)\alpha_k \tag{40}$$
$$\mathcal{P}_{k+1}(t) = \mathcal{R}_{k+1}(t) + \mathcal{P}(t)\beta_k \tag{41}$$
$$\mathcal{P}_0(t) = \mathcal{R}_0(t) = I_s \tag{42}$$

Для того, чтобы записать свойства ортогональности в терминах матричнозначных полиномов полезно ввести определения двух функционалов:

Определение 2.4.

$$C(\mathcal{P}) \equiv \tilde{R}_0^T(\mathcal{P}(A) \circ R_0) \tag{43}$$

$$C^{(1)}(\mathcal{P}) \equiv C(t\mathcal{P}) \tag{44}$$

Свойства ортогональности для блока невязок и блока направлений:

Утверждение 2.2.

$$C(\mathcal{R}_k\mathcal{T}_i) = 0, \text{ } \textit{для } i < k \tag{45}$$

$$C^{(1)}(\mathcal{P}_k\mathcal{T}_i) = 0, \text{ } \textit{для } i < k \tag{46}$$

Доказательство. Из свойства ортогональности для блочного метода бисопряженных градиентов (32) следует, что

$$R_k \perp K_k(A^T,\tilde{R}_0),$$

следовательно, для $i < k$

$$\tilde{R}_0^T A^i R_k = 0,$$

и воспользуемся определением полиномов $\mathcal{R}_k$ (38):

$$\tilde{R}_0^T(A^i\mathcal{R}_k(A)) \circ R_0 = 0,$$

откуда сразу же следует первое утверждение, которое требовалось доказать. По рекурентной формуле (40):

$$AP_k = (R_k - R_{k+1})\alpha_k^{-1}.$$

Применив это выражение, первое доказанное утверждение и определение (39), получаем для $i < k$:

$$C^{(1)}(t^i\mathcal{P}_k) = \tilde{R}_0^T(A^{i+1}\mathcal{P}_k(A) \circ R_0) = \tilde{R}_0^T A^{i+1}P_k = \tilde{R}_0^T A^i(R_k - R_{k+1})\alpha_k^{-1} = 0,$$

откуда сразу следует второе утверждение, которое требовалось доказать. □

Рассмотрим матричнозначные полиномы $\mathcal{Q}_k$, которые задаются рекуретной формулой:

$$\mathcal{Q}_{k+1} = (1 - \omega_k t)\mathcal{Q}_k(t),$$

где ω_k - скалярные матрица. И будем выбирать ω_k такой, чтобы минимизировать норму Фробениуса блока стабилизированных невязок, для которых введём переобозначение $R_k \equiv (\mathcal{Q}_k\mathcal{R}_k(A)) \circ R_0$, и для которых, аналогично стабилизированному методу бисопряженных градиентов, справедлива рекуррентная формула:

$$R_{k+1} = S_k - \omega_k AS_k,$$

из которой следует, что ω_k выражается как

$$\omega_k = \frac{< AS_k,S_k >_F}{< AS_k,AS_k >_F}.$$

Также введем переобозначение для стабилизированных направлений $P_k \equiv (\mathcal{Q}_k\mathcal{P}_k(A)) \circ R_0$.

Используя утверждение 2.2 и рекуретное соотношение (40), найдём выражения для α_k:

$$C(\mathcal{Q}_k\mathcal{R}_k) = C^{(1)}(\mathcal{Q}_k\mathcal{P}_k)\alpha_k$$

Переводя в матричную запись, получаем линейную систему на α_k:

$$(\tilde{R}_0^T A P_k)\alpha_k = \tilde{R}_0^T R_k$$

Аналагично, используем утверждение 2.2 и рекурретное соотношение (41), чтобы получить линейную систему на матрицу коэффициентов β_k:

$$C^{(1)}(\mathcal{Q}_k\mathcal{R}_{k+1}) = -C^{(1)}(\mathcal{Q}_k\mathcal{P}_k)\beta_k,$$

в матричном виде это выражение имеет вид:

$$(\tilde{R}_0^T A P_k)\beta_k = -\tilde{R}_0^T A S_k$$

.

Итак, итоговый вид алгоритма:

Алгоритм 9 Блочный стабилизированный метод бисопряженных градиентов

X_0 - блок начальных приближений, $R_0 = B - AX_0$ - блок начальных невязок
$P_0 = R_0$
$\tilde{R}_0$ - произвольная $N \times s$ матрица
for $k = 0,1,...$ **do**
    $V_k = AP_k$
    $\alpha_k = (\tilde{R}_0^T V_k)^{-1}(\tilde{R}_0^T R_k)$
    $S_k = R_k - V_k\alpha_k$
    $T_k = AS_k$
    $\omega_k = \frac{<T_k,S_k>_F}{<T_k,T_k>_F}$
    $X_{k+1} = X_k + P_k\alpha_k + \omega_k S_k$
    $R_{k+1} = S_k - \omega_k T_k$
    $\beta_k = (\tilde{R}_0^T V_k)^{-1}(-\tilde{R}_0^T T_k)$
    $P_{k+1} = R_{k+1} + (P_k - \omega_k V_k)\beta_k$
end for

Однако этот метод, как будет показано в разделе 5, данный метод не сходится с большим количеством правых частей, и показывает слабое уменьшение числа итераций с увеличением размера блока в задаче электромагнитного рассеяния [5]. В разделе 3 будут представлены шаги для повышения устойчивости вычислений.

2.10 Блочный симметричный метод квазиминимальных невязок [6]

Определим квазискалярное произведение двух векторов над полем комплексных чисел x и y как $\langle x,y\rangle_Q = \sum_k x_k y_k$. По отношению к такому произведению комплексные симметричные матрицы будут самосопряженными. Рассматриваемый в этом подразделе метод итеративно ищет решение линейной системы порядка N с s правыми частями $AX = B$, в случае, когда A - симметричная комплесная матрица.

2.10.1 Блочный симметричный алгоритм Ланцоша

Естественным образом можно получить блочное обобщение симметричного метода Ланцоша, аналогично тому, как блочный метод бисопряженных градиентов следовал

Алгоритм 10 Блочный симметричный алгоритм Ланцоша

1: X_0 - блок начальных приближений
2: $R_0 = B - AX_0$
3: $V_0 = 0$
4: $\tilde{V}_1 = R_0$
5: **for** $k = 1{,}2{,}...$ **do**
6: $\quad V_k, \beta_k \xleftarrow{\text{квази-}QR} \tilde{V}_k$
7: $\quad \tilde{V}_{k+1} = AV_k - V_{k-1}\beta_k^T$
8: $\quad \alpha_k = V_k^T \tilde{V}_{k+1}$
9: $\quad \tilde{V}_{k+1} \leftarrow \tilde{V}_{k+1} - V_k \alpha_k$
10: **end for**

из бисопряженных градиентов. Далее α_k,β_k - $s \times s$ матрицы, V_k и $\tilde{V}_k$ - $N \times s$ матрицы, В строке номер 6 производится квази-QR разложение, которое можно выполнить с помощью модернизированного процесса Грамма-Шмидта, но нужно все скалярные произведения заменить на квазискалярные произведения.

Результат работы блочного симметричного алгоритма Ланцоша можно записать в матричном виде:

$$A\mathcal{V}_k = \mathcal{V}_{k+1}\tilde{T}_k, \tag{47}$$

где были введены обозначения:

$$\mathcal{V}_k = \left[V_1, ..., V_k\right]$$

$$\tilde{T}_k = \begin{bmatrix} \alpha_1 & \beta_2^T & & & \\ \beta_2 & \alpha_2 & \beta_3^T & & \\ & \beta_3 & \ddots & \ddots & \\ & & \ddots & \alpha_{k-1} & \beta_k^T \\ & & & \beta_k & \alpha_k \\ & & & & \beta_{k+1} \end{bmatrix}.$$

2.10.2 Алгоритм

Приближенное решение будем искать в виде $X_k = X_0 + \mathcal{V}_k Z_k$, тогда для блока невязок $R_k = B - AX_k$ после применения формулы (47) получаем выражение:

$$R_k = V_1\beta_1 - \mathcal{V}_{k+1}\tilde{T}_k Z_k$$

. Введём обозначение $\Omega_k = Diag(\omega_1,...,\omega_k)$, где $\omega_i = Diag(\|col_i(V_k)\|)$.

Вставим Ω_k в формулу для R_k:

$$R_k = \mathcal{V}_{k+1}\Omega_{k+1}^{-1}\left[\omega_1 e_1 \beta_1 - \Omega_{k+1}\tilde{T}_k Z_k\right],$$

где e_1 - $N \times s$ матрица, которая состоит из первых s строк единичной матрицы.

Если бы s была равна 1, и если бы $\mathcal{V}$ была бы матрицей с ортогональными столбцами, то по унитарной инвариантности нормы Фробениуса, можно было бы выразить норму невязки как:

$$\|\omega_1 e_1 \beta_1 - \Omega_{k+1}\tilde{T}_k Z_k\|,$$

и дальше можно было бы минимизировать эту норму. Но предположения, в которых полученно данное выражение в нашем случае не имеют места. Однако все равно можно построить алгоритм, который бы подбирал Z_k такой, чтобы минимизировать эту норму, в этом и заключается идея "квазиминимальных невязок а сама эта норма называется *квазиневязкой.*

Таким образом, будем искать такую Z_k, чтобы минимизировать норму

$$\|col_i(\omega_1 e_1 \beta_1 - \Omega_{k+1} \tilde{T}_k Z_k)\|$$

для каждой правой части $i = 1,..,m$ независимо и одновременно. Минимизация осуществляется путём QR разложения, где матрица Q_{k+1} выбирается так, чтобы

$$Q_{k+1}\Omega_{k+1}\tilde{T}_k = \begin{bmatrix} U_k \\ 0 \end{bmatrix} = \begin{bmatrix} \zeta_1 & \eta_2 & \theta_3 & & \\ & \zeta_2 & \eta_3 & \ddots & \\ & & \ddots & \ddots & \theta_k \\ & & & \zeta_{k-1} & \eta_k \\ & & & & \zeta_k \\ & & & & 0 \end{bmatrix}$$

Рекуретное обновление матрицы Q_{k+1}:

$$Q_{k+1} = \begin{bmatrix} I_{(k-1)s} & 0 \\ 0 & Q(a_k,b_k,c_k,d_k) \end{bmatrix} \begin{bmatrix} Q_k & 0 \\ 0 & I_s \end{bmatrix}, \text{ где } Q(a_k, b_k, c_k, d_k) = \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix}$$

По унитарной инвариантности нормы Фробениуса, преобразуем квазиневязку:

$$\|col_i \left(Q_{k+1}\omega_1 e_1 \beta_1 - \begin{bmatrix} R_k \\ 0 \end{bmatrix} Z_k \right) \|, \text{ для } i = 1,...,m$$

Чтобы минимизировать квазиневязку в таком виде, определим

$$\tilde{t}_{k+1} = Q_{k+1}\omega_1 e_1 \beta_1 = \begin{bmatrix} t_k \\ \tilde{\tau}_{k+1} \end{bmatrix}, \text{ где } \tilde{\tau}_{k+1} \text{ - это } s \times s \text{ матрица.}$$

и выбирем Z_k как $Z_k = R_k^{-1} t_k$.

Тогда квазиневязку можно вычислить с помощью $\tilde{\tau}_{k+1}$:

$$QRES_k = max_i \|col_i(\tilde{\tau}_{k+1})\|,$$

а настоящая невязка определяется, как и в других рассмотренных блочных методах:

$$RES_k = max_i \|col_i(AX_k - B)\|.$$

Итоговый вид алгоритма:

Однако, как будет показано в разделе 5 у этого метода есть ряд проблем:

1. Возможен большой взлёт невязки на первых итерациях алгоритма.

2. Поддерживается квази-ортогональность внутри блока $V_i^T V_i$ - диагональная матрица. Вместо этого свойства было бы лучше поддерживать другое, например, обычную ортогональность внутри блока.

3. Нет сходимости в арифметике с одинарной точностью в задаче электромагнитного рассеяния [5], если брать все 722 правые части в блоке.

Алгоритм 11 Блочный симметричный метод квазиминимальных невязок

$V_0 = P_0 = P_{-1} = 0_{N\times s}$, N - размер матрицы A, s - количество правых частей.
$c_0 = b_{-1} = b_0 = 0_{s\times s}$
$a_0 = d_{-1} = d_0 = I_{s\times s}$
$R_0 = B - AX_0$
$\tilde{V}_1 = R_0$
$V_1,\, \beta_1 \xleftarrow{\text{квази}-QR} \tilde{V}_1$
$\omega_0 = 0_{s\times s}$
$\omega_1 = Diag\{\|col_i(v_i)\|\}$
$\tilde{\tau}_1 = \omega_1\beta_1$
for $k = 1, \ldots$ **do**
    $\tilde{V}_{k+1} = AV_k - V_{k-1}\beta_k^T$
    $\alpha_k = V_k^T\tilde{V}_{k+1}$
    $\tilde{V}_{k+1} = \tilde{V}_{k+1} - V_k\alpha_k$
    $V_{k+1},\, \beta_{k+1} \xleftarrow{\text{квази}QR} \tilde{V}_{k+1}$
    $\omega_{k+1} = Diag\{\|col_i(v_{k+1})\|\}$
    $\theta_k = b_{k-2}\omega_{k-1}\beta_k^T$
    $\eta_k = a_{k-1}d_{k-2}\omega_{k-1}\beta_k^T + b_{k-1}\omega_k\alpha_k$
    $\tilde{\zeta}_k = c_{k-1}d_{k-2}\omega_{k-1}\beta_k^T + d_{k-1}\omega_k\alpha_k$
    $Q_k,\ \begin{bmatrix}\zeta_k\\ 0_{s\times s}\end{bmatrix} \xleftarrow{QR} \begin{bmatrix}\tilde{\zeta}_k\\ \omega_{k+1}\beta_{k+1}\end{bmatrix}$
    $\begin{bmatrix}a_k & b_k\\ c_k & d_k\end{bmatrix} \leftarrow Q_k^*$
    $P_k = (V_k - P_{k-1}\eta_k - P_{k-2}\theta_k)\zeta_k^{-1}$
    $\tau_k = a_k\tilde{\tau}_k$
    $X_k = X_{k-1} + P_k\tau_k$
    $\tilde{\tau}_{k+1} = c_k\tilde{\tau}_k$
end for

3 Модификация блочного метода стабилизированных бисопряженных градиентов

В данной главе предложены изменения, направленные на улучшение стабильности блочного метода бисопряженных 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:

$$C(\mathbf{Q}_k\mathbf{R}_{k+1}) = 0, \tag{48}$$

$$C^{(1)}(\mathbf{Q}_k\mathbf{P}_k) = 0. \tag{49}$$

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

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

Тогда выражение (48) можно представить в виде:

$$\tilde{R}_0^* S_k = 0. \tag{50}$$

В точной арифметике это соотношение выполняется строго, однако при вычислениях на компьютере соотношение (50) 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:

$$S_k^r = S_k - AP_k\alpha_k^r. \tag{51}$$

При этом мы стремимся поддерживать соотношение $\tilde{R}_0^* S_k^r = 0$ с уточненным блоком S_k^r. Тогда, домножая обе части выражения (51) слева на $\tilde{R}_0$, получим уравнение для поправки α_k^r:

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

Аналогичным образом рассмотрим (49). Используя полиномиальное соотношение для P_{K+1}, получаем следующее выражение:

$$t\mathbf{Q}_k\mathbf{P}_{k+1} = t\mathbf{Q}_k\mathbf{R}_{k+1} + t\mathbf{Q}_k\mathbf{P}_k. \tag{52}$$

Введем обозначение $W_k \equiv (t\mathbf{Q}_k\mathbf{R}_{k+1})(A) \circ R_0$. Тогда выражение (52) можно записать в матричном виде:

$$W_k = AS_k + AP_k\beta_k. \tag{53}$$

Тогда выражение (49) можно представить в виде:

$$\tilde{R}_0^* W_k = 0. \tag{54}$$

Аналогично получаем соотношения реортогонализации для (54):

$$W_k^r = W_k + AP_k\beta_k^r$$

Поправка β_k определяется уравнением:

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

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

$$\begin{aligned}\mathbf{Q}_{k+1}\mathbf{P}_{k+1} &= \mathbf{Q}_{k+1}(\mathbf{R}_{k+1} + \mathbf{P}_k\beta_k) = \\ &= \mathbf{Q}_k\mathbf{R}_{k+1} - \omega_k t\mathbf{Q}_k\mathbf{R}_{k+1} + (1-\omega_k t)\mathbf{Q}_k\mathbf{P}_k\beta_k = \\ &= \mathbf{Q}_k\mathbf{R}_{k+1} - \omega_k t\mathbf{Q}_k\mathbf{P}_{k+1} + \mathbf{Q}_k\mathbf{P}_k\beta_k\end{aligned}$$

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:

$$R^r_{k+1} = R_{k+1} - \omega^r_k T_k$$

$$\omega^r_k = \frac{\langle R_{k+1}, T_k\rangle_F}{\langle T_{k+1}, T_k\rangle_F}$$

3.2 Ортогонализация векторов направлений и проверочных невязок

В Алгоритме 12 приведён метод, предложенный в статье [4]. Красным отмечены все места где используется блок векторов направлений P_k. Легко видеть, что он везде входит в алгоритм вместе матрицей коэффициентов (α_k и β_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

Алгоритм 12 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 $k = 0, 1, 2, ...$ **do**
  решить $\tilde{R}_0^T A P_k \alpha_k = \tilde{R}_0^T R_k$
  $S_k = R_k - AP_k\alpha_k$
  $T_k = AS_k$
  $\omega_k = \frac{\langle T_k, S_k\rangle_F}{\langle T_k, T_k\rangle_F}$
  $X_{k+1} = X_k + P_k\alpha_k + \omega_k S_k$
  $R_{k+1} = S_k - \omega_k T_k$
  решить $\tilde{R}_0^T A P_k \beta_k = -\tilde{R}_0^T T_k$
  $P_{k+1} = R_{k+1} + P_k\beta_k - \omega_k AP_k\beta_k$
end for

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

$$P_k = Q_{P_k} R_{P_k},$$

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

Как указано в алгоритме 12, $\tilde{R}_0$ - 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 $\tilde{R}_0 \to Q_R$.

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 13 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 $k = 0, 1, ...$ **do**
  $P_k = QU$ - QR-разложение P_k;
  $P_k \to P_k U^{-1}$;
  $V_k = AP_k$;
  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 $(\tilde{R}_0^* V_k)\alpha_k = \tilde{R}_0^* \hat{S}_k$;
  $S_k = \hat{S}_k - V_k\alpha_k$;
  $T_k = AS_k$;
  $\hat{\omega}_k =< S_k{,}T_k >_F / < T_k{,}T_k >_F$;
  $\hat{R}_{k+1} = S_k - \hat{\omega}_k T_k$;
  $\omega_k =< \hat{R}_{k+1}, T_k >_F / < T_k, T_k >_F$;
  $R_{k+1} = \hat{R}_{k+1} - \omega_k T_k$;
  $X_{k+1} = X_k + P_k(\hat{\alpha}_k + \alpha_k) + (\hat{\omega}_k + \omega_k)S_k$;
  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 ω в виде скалярной матрицы, из-за этого чем больше размер блока мы берем для расчета, тем меньше по модулю становится ω, 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 ω_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 квазиминимальных невязок

Один из ключевых элементов блочного симметричного метода квазиминимальных невязок [6] является процесс Грамма-Шмидта с квазискалярным произведением. Далее будет представлена модификация этого алгоритма, использующая настоящее QR-разложение. Благодаря этому невязка на шаге алгоритма окажется ближе к настоящей невязке. Немаловажжно и то, что квази-QR в некотором роде эквивалентно LL^T разложению матрицы V^TV, причём это разложение выполняется без выбора ведущего элемента. В отличие от разложения Холецкого, для которого из-за положительной определенности матрицы следует, что все ведущие миноры положительно определены и обусловлены не хуже, чем вся матрица, и поэтому для него выбор ведущего элемента не так существенен, здесь это является проблемой, поэтому важно модифицировать алгоритм, не использующим квази-QR-разложение. В дополнение ко всему вышесказанному, при этом становится возможным использование устойчивых реализаций QR-разложения и применение их библиотечных реализаций.

Блочный симметричный процесс Ланцоша приводит к следующему матричному соотношению:

$$A\begin{bmatrix}V_1 & \dots & V_k & V_{k+1}\end{bmatrix} = \begin{bmatrix}V_1 & \dots & V_k & V_{k+1}\end{bmatrix}\begin{bmatrix}\alpha_1 & \delta_1 & & & \\ \beta_2 & \alpha_2 & \delta_2 & & \\ & \beta_3 & \ddots & \ddots & \\ & & \ddots & \alpha_{k-1} & \delta_{k-1} \\ & & & \beta_k & \alpha_k \\ & & & & \beta_{k+1}\end{bmatrix}, \tag{55}$$

где $\delta_{i-1} = \beta_i^T$ в версии из статьи [6], в нашей модификации же получится другой вид для этой матрицы коэффициентов. Из (55) для k-го блока следует:

$$AV_k = V_{k-1}\delta_{k-1} + V_k\alpha_k + V_{k+1}\beta_{k+1} \tag{56}$$

При построении базиса в блочном крыловском пространстве, требуется выпонение следующего свойства:

$$V_i^TV_j = 0,\ i \neq j \tag{57}$$

Домножая слева выражение (56) на V_{k-1}^T и используя соотношение (57) получаем системы линейных уравнений на матрицу δ_{k-1}:

$$V_{k-1}^TV_{k-1}\delta_{k-1} = V_{k-1}^TAV_k. \tag{58}$$

Сделав замену в (56) вида $k \to k-1$ и учтя выражение (58) выразим δ_{k-1} через β_k:

$$V_{k-1}^TV_{k-1}\delta_{k-1} = \beta_k^TV_k^TV_k.$$

Введем обозначение $\gamma_k = V_k^TV_k$.

Тогда окончательный вид для δ_{k-1}:

$$\delta_{k-1} = \gamma_{k-1}^{-1}\beta_k^T\gamma_k. \tag{59}$$

Аналогично δ_{k-1} из (56) получим системы линейных уравнений на α_k:

$$\gamma_k\alpha_k = V_k^TAV_k.$$

И воспользовавшись свойством (57) преобразуем выражение для α_k:

$$\alpha_k = \gamma_k^{-1} V_k^T (AV_k - V_{k-1}\delta_{k-1}). \tag{60}$$

Выбор β_{k+1} является произвольным и определяется целями исследователя, в предлагаемой модификации β_{k+1} выбрано таким, чтобы выполнялось соотношение $V_{k+1}^* V_{k+1} = I$, где I - единичная $s \times s$ матрица. Этого можно достичь с помощью QR-разложения:

$$V_{k+1}, \beta_{k+1} \xleftarrow{QR} AV_k - V_{k-1}\delta_{k-1} - V_k\alpha_k. \tag{61}$$

Этот выбор обладает рядом преимуществ:

1. получение QR-разложения в сравнении с квази-QR-разложением является более устойчивой операцией,

2. на первой итерации алгоритм ведёт себя как обобщённый метод минимальных невязок, что обеспечивает на первой итерации достижение точного минимума невязки в построенном к этому моменту пространстве Крылова, что в свою очередь предотвращает большие скачки невязки на первых итерациях, как это наблюдается в алгоритме из статьи [6].

Однако с этими изменениями метод все еще не сходится в задаче электромагнитного рассеяния [5] в одинарной точности, поэтому необходимо получить более устойчивые формулы для рекуррентных соотношений.

Для этого можно производить квази-реортогонализацию для поддержания соотношения (57). Поправка к V_{k+1}:

$$V_{k+1} = V_{k-1}\tilde{\delta}_{k-1} + V_k\tilde{\alpha}_k + \tilde{V}_{k+1}\tilde{\beta}_{k+1}, \tag{62}$$

где $\tilde{V}_{k+1}$ - более точно вычисленный блок V_{k+1}. Используя (57), получим формулы для поправок:

$$\tilde{\delta}_{k-1} = \gamma_{k-1}^{-1} V_{k-1}^T V_{k+1} \tag{63}$$
$$\tilde{\alpha}_k = \gamma_k^{-1} V_k^T V_{k+1} \tag{64}$$
$$\tilde{V}_{k+1}\tilde{\beta}_{k+1} \xleftarrow{QR} V_{k+1} - V_{k-1}\tilde{\delta}_{k-1} - V_k\tilde{\alpha}_k \tag{65}$$

Подставим (62) в (56):

$$AV_k = V_{k-1}(\delta_{k-1} + \tilde{\delta}_{k-1}\beta_{k+1}) + V_k(\alpha_k + \tilde{\alpha}_k\beta_{k+1}) + \tilde{V}_{k+1}\beta_{k+1}\tilde{\beta}_{k+1} \tag{66}$$

Таким образом, матрицы коэффициентов после реортогонализации имеют вид:

$$\delta_{k-1}^r = \delta_{k-1} + \tilde{\delta}_{k-1}\beta_{k+1} \tag{67}$$
$$\alpha_k^r = \alpha_k + \tilde{\alpha}_k\beta_{k+1} \tag{68}$$
$$\beta_{k+1}^r = \beta_{k+1}\tilde{\beta}_{k+1} \tag{69}$$

Также предлагается перед рассмотренной квази-реортогонализацией провести реортогонализацию для QR-разложения (61) стандартным образом:

$$V_{k+1}, \beta_{k+1} \xleftarrow{QR} AV_k - V_{k-1}\delta_{k-1} - V_k\alpha_k \tag{70}$$
$$V_{k+1}^r, \tilde{\beta}_{k+1} \xleftarrow{QR} V_{k+1} \tag{71}$$
$$\beta_{k+1}^r = \tilde{\beta}_{k+1}\beta_{k+1}. \tag{72}$$

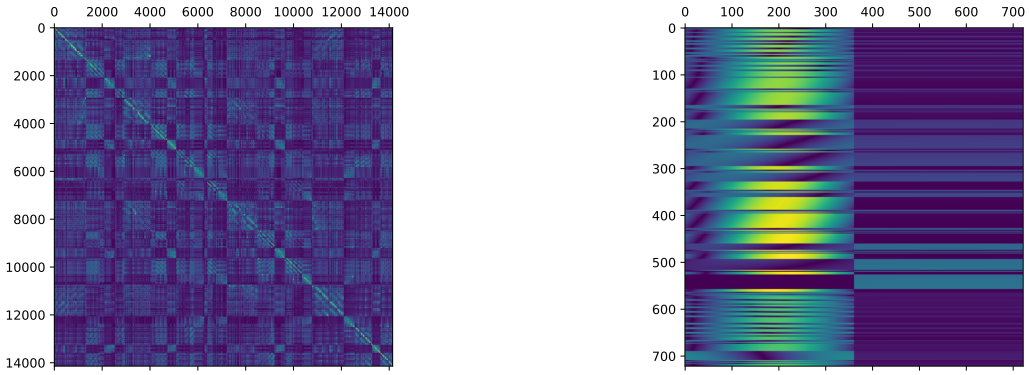
Окончательный вид алгоритма, красным отмечена процедура реортогонализации:

Алгоритм 14 Модифицированный блочный симметричный метод квазиминимальных невязок

$V_0 = P_0 = P_{-1} = 0_{N\times s}$, N - размер матрицы A, s - количество правых частей.
$c_0 = b_{-1} = b_0 = 0_{s\times s}$
$a_0 = d_{-1} = d_0 = I_{s\times s}$
$R_0 = B - AX_0$
$V_1,\, \beta_1 \xleftarrow{QR} R_0$
$\gamma_0 = I_{s\times s}$
$\gamma_1 = V_1^T V_1$
$\tilde{\tau}_1 = \beta_1$
for $k = 1, \ldots$ **do**
$\delta_{k-1} = \gamma_{k-1}^{-1}\beta_k^T\gamma_k$
$\tilde{V}_{k+1} = AV_k - V_{k-1}\delta_{k-1}$
$\alpha_k = \gamma_k^{-1}V_k^T\tilde{V}_{k+1}$
$\tilde{V}_{k+1} = \tilde{V}_{k+1} - V_k\alpha_k$
$V_{k+1},\, \beta_{k+1} \xleftarrow{QR} \tilde{V}_{k+1}$
$V_{k+1},\, \tilde{\beta}_{k+1} \xleftarrow{QR} V_{k+1}$
$\beta_{k+1} \leftarrow \tilde{\beta}_{k+1}\beta_{k+1}$
$\tilde{\alpha}_k = \gamma_k^{-1}V_k^T V_{k+1}$
$\alpha_k \leftarrow \alpha_k + \tilde{\alpha}_k\beta_{k+1}$
$V_{k+1} \leftarrow V_{k+1} - V_k\tilde{\alpha}_k$
$\tilde{\delta}_{k-1} = \gamma_{k-1}^{-1}V_{k-1}^T V_{k+1}$
$\delta_{k-1} \leftarrow \delta_{k-1} + \tilde{\delta}_{k-1}\beta_{k+1}$
$V_{k+1} \leftarrow V_{k+1} - V_{k-1}\tilde{\delta}_{k-1}$
$V_{k+1},\, \tilde{\beta}_{k+1} \xleftarrow{QR} V_{k+1}$
$\beta_{k+1} \leftarrow \tilde{\beta}_{k+1}\beta_{k+1}$
$\gamma_{k+1} = V_{k+1}^T V_{k+1}$
$\theta_k = b_{k-2}\delta_{k-1}$
$\eta_k = a_{k-1}d_{k-2}\delta_{k-1} + b_{k-1}\alpha_k$
$\zeta_k = c_{k-1}d_{k-2}\delta_{k-1} + d_{k-1}\alpha_k$
$Q_k,\, \begin{bmatrix} \zeta_k \\ 0_{s\times s} \end{bmatrix} \xleftarrow{QR} \begin{bmatrix} \tilde{\zeta}_k \\ \beta_{k+1} \end{bmatrix}$
$\begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \leftarrow Q_k^*$
$P_k = (V_k - P_{k-1}\eta_k - P_{k-2}\theta_k)\zeta_k^{-1}$
$\tau_k = a_k\tilde{\tau}_k$
$X_k = X_{k-1} + P_k\tau_k$
$\tilde{\tau}_{k+1} = c_k\tilde{\tau}_k$
end for

5 Численные эксперименты

Тесты производились на интересующей нас задаче – линейной системе с многими правыми частями, возникающей при решении задачи электромагнитного рассеяния методом интегральных уравнений [5]. Порядок системы - 14144, количество правых частей - 722. Каждая правая часть соответствует разным углам падения, а также первая половина правых частей отличается от второй типом поляризации. При этом матрица системы является комплексной и симметричной.



(a) Матрица системы в логарифмическом масштабе (отображены модули элементов матрицы).

(b) первые 722 строки блока правых частей в логарифмическом масштабе (отображены модули элементов матрицы).

Рис. 1

На рис.1 представлен вид матрицы системы и первые 722 строки блока правых частей. Из-за разных поляризаций первая половина правых частей сильно отличается от второй половины, и если применять блочные крыловские алгоритмы к блоку правых частей, содержащему правые части из первой и второй половины одновременно, то алгоритмы могут очень плохо себя показывать в таком сценарии.

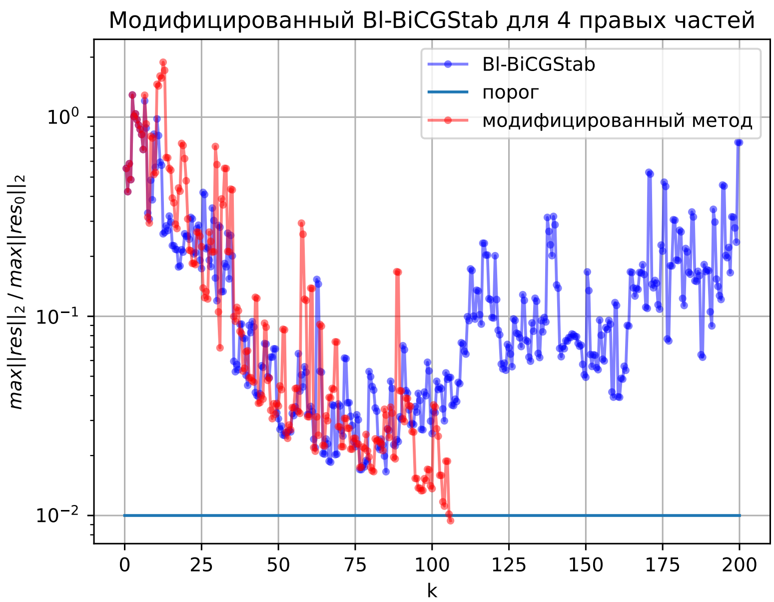


Рис. 2

Первый тест демонстрирует, что метод из статьи [4] не достигает требуемой точности, в то время как версия с улучшениями, описанными в главе 3, сходится. Эксперимент проводился в одинарной точности для четырех правых частей с номерами: 0, 90, 180, 270. Его результаты представлены на рис.2

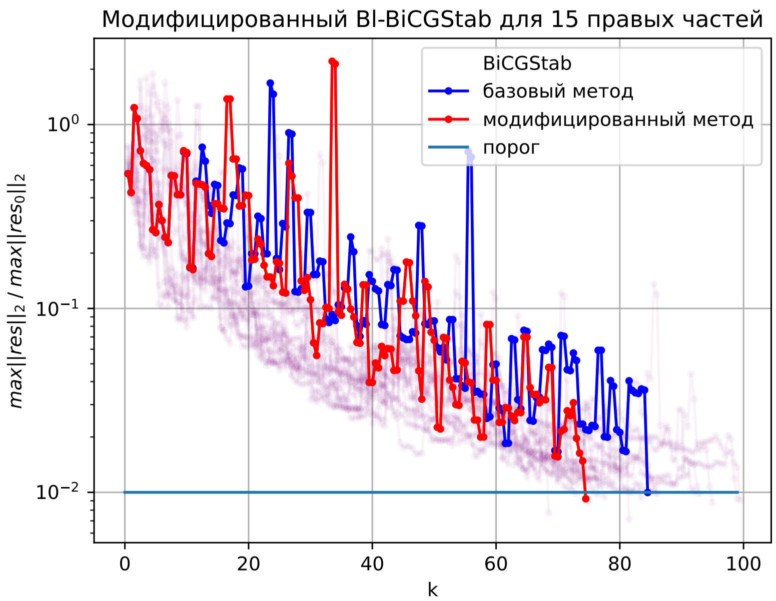


Рис. 3

Второй тест демонстрирует, что улучшения, описанные в главе 3 позволяют получить выгоду по количеству умножений матрицы системы на вектор, 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 матрично-векторных умножений (МВУ), для решения методом из статьи [4] - 2535, модифицированный метод сошелся за 2235 МВУ, таким образом выгода составил 12% по сравнению с неблочной версией.

Однако количество итераций даже при всех предложенных улучшениях все равно велико, так что было принято решение рассмотреть алгоритм на основе метода квазиминимальных невязок, описанный в главе 4.

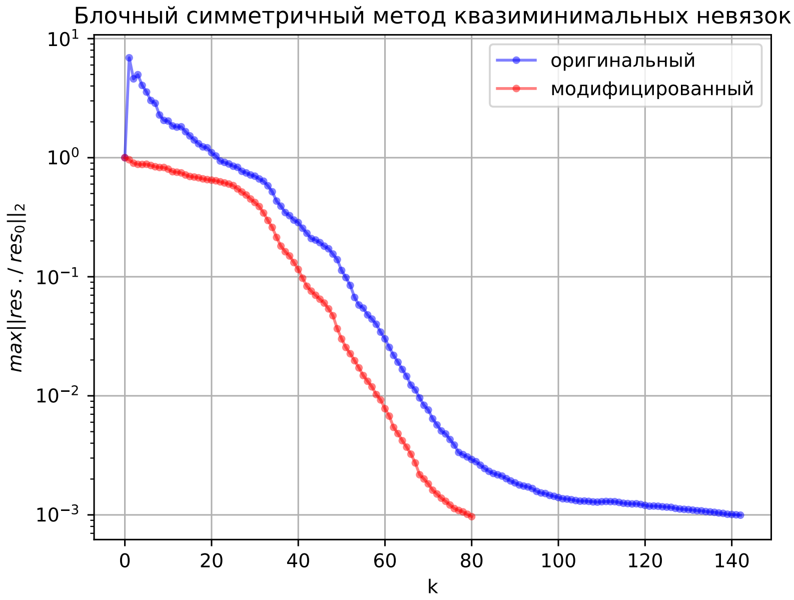


Рис. 4

На рис.4 показано уменьшение максимальной относительной невязки в блоке в зависимости от числа итераций, синяя кривая соответствует алгоритму из статьи [6], красная - модифицированному алгоритму из главы 4. Эксперимент производился в двойной точности с 45 правыми частями, остановка происходила при достижении порога 10^{-3} для демонстрации различия в сходимости двух методов. Примечательным моментом является скачок на первой итерации для метода из статьи [6], которого нет в модифицированной версии по причинам, освещенным в конце главы 4.

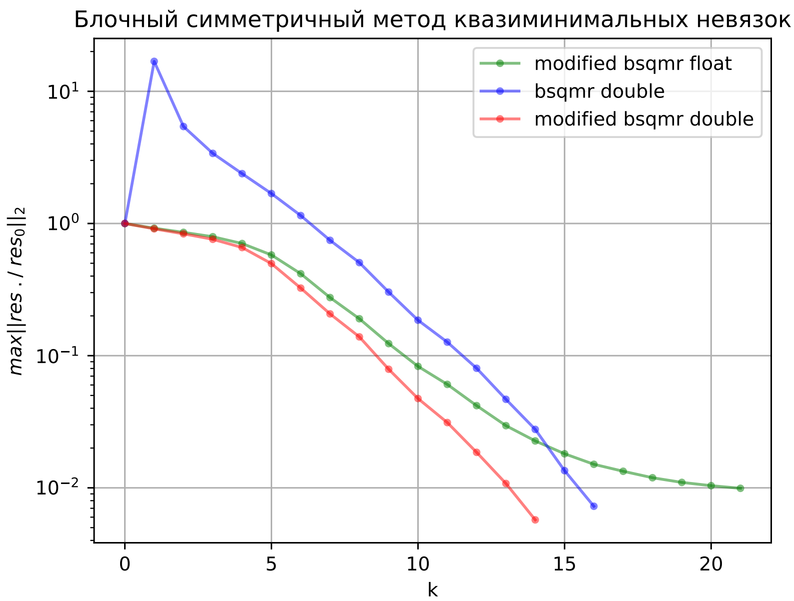


Рис. 5

На рис.5 представлены результаты расчета со всеми 722 правыми частями одновременно. Синий график представляет алгоритм из статьи [6], красный и зеленый графики - его модификацию из главы 4. Причем расчёты для синей и красной кривых выполнены в арифметике с двойной точностью, а для зелёной - с одинарной. Немодифицированный алгоритм в одинарной точности уже на первых итерациях показывает сильную расходимость, поэтому эти расчёты не включены на график.

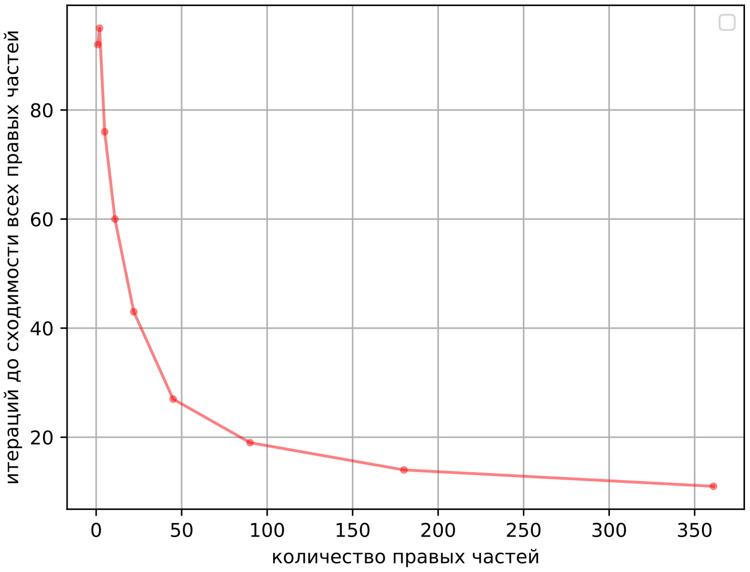


Рис. 6: Количество итераций модифицированного блочого симметричного метода квазиминимальных невязок в зависимости от количества задействованных правых частей

На рис.6 изображена зависимость количества итераций блочого симметричного метода квазиминимальных невязок в зависимости от количества задействованных правых частей для первых 361 правой части в задаче электромагнитного рассеяния [5]. Видно, что количество итераций уменьшается с увеличением размера блока, следовательно, уменьшается и размер задействованного для решения крыловского пространства.

6 Заключение

Результаты

Нерешенные проблемы редукции блока

Список литературы

[1] *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.

[2] *Saad, Yousef.* Iterative Methods for Sparse Linear Systems / Yousef Saad. — 2nd edition. — Philadelphia, PA: Society for Industrial and Applied Mathematics, 2003.

[3] *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.

[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] *Stavtsev, S. L.* Application of Mosaic-Skeleton Approximations for Solving EFIE / S. L. Stavtsev, E. E. Tyrtyshnikov // Progress in Electromagnetics Research Symposium (PIERS) 2009 Proceedings. — PIERS Proceedings. — Moscow, Russia: The Electromagnetics Academy, 2009. — Abstracts published in PIERS 2009 Moscow (ISBN 978-1-934142-09-7). https://piers.org/proceedings/piers2009proc.html.

[6] *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.