УДК 519.612

*А. О. Синюхин, А. А. Буздин*

# МЕТОД ПОСТРОЕНИЯ АЛГОРИТМОВ ПОЛНОЙ РЕДУКЦИИ ДЛЯ РЕШЕНИЯ СИСТЕМ УРАВНЕНИЙ С БЛОЧНОЙ ТРЕХДИАГОНАЛЬНОЙ МАТРИЦЕЙ



*Рассмотрен подход к реализации метода полной редукции для системы уравнений с блочной трехдиагональной матрицей произвольной размерности. Выведены формулы, позволяющие производить исключение неизвестных в произвольном порядке. Изложены алгоритмы для первой и второй краевой задачи.*

*An approach to the method of cyclic reduction for the system of equations with block-tridiagonal matrix of arbitrary dimension is presented. Formulas for elimination of unknowns in arbitrary order are obtained. Algorithms for the first and second boundary problems are described.*

**Ключевые слова:** система линейных уравнений, блочная трехдиагональная матрица, метод полной редукции.

**Key words:** system of linear equations, block-tridiagonal matrix, method of cyclic reduction.

## Введение

Метод редукции является одним из эффективных прямых методов решения СЛАУ с блочной трехдиагональной матрицей

$$A = \text{blocktridiag}\{-I, C_i, -I\}, \quad (1)$$

где $C_i$ — линейные функции от одной матрицы. Такие СЛАУ возникают, например, в результате конечно-разностной дискретизации краевых задач для уравнения вида

$$\frac{\partial}{\partial x}\left(\phi(x)\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\psi(y)\frac{\partial u}{\partial y}\right)+q(x)u = f(x, y).$$

Ранние алгоритмы метода полной редукции имели следующие недостатки: они были неустойчивы и количество блоков системы не могло быть произвольным. От недостатков удавалось избавиться путем использования искусственных приемов, затрудняющих понимание и реализацию алгоритмов [1; 2]. В данной работе предложен подход к методу редукции, позволяющий с единой точки зрения естественным образом получить легко реализуемый и устойчивый алгоритм для произвольного числа узлов и различных краевых задач.

## 1. Вычисление обратной матрицы

Определим полином $\Delta_q^p(x)$ следующим образом:

$$\Delta_q^p(x)=\begin{vmatrix} C_q(x) & -1 & 0 & \cdots & 0 \\ -1 & C_{q+1}(x) & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & C_{p-1}(x) & -1 \\ 0 & \cdots & 0 & -1 & C_p(x) \end{vmatrix}. \quad (2)$$

Тогда через $\Delta_q^p(C)$ обозначим соответствующий матричный полином, где $C\in\mathbb{R}^{m\times m}$. При этом будем полагать $\Delta_{p+1}^p(C)\equiv I$.

Матричные полиномы $\Delta_q^p(C)$ обладают свойствами, которые легко получить из разложения определителя (2) по строке с номером *i*:

1) $\Delta_i^N = C_i\Delta_{i+1}^N - \Delta_{i+2}^N$,
2) $\Delta_1^N = -\Delta_1^{i-2}\Delta_{i+1}^N + C_i\Delta_1^{i-1}\Delta_{i+1}^N - \Delta_1^{i-1}\Delta_{i+2}^N$.

Сформулируем и докажем лемму, которая будет использоваться для дальнейшего вывода расчетных формул. Для сокращения количество индексов в лемме и в последующих рассуждениях обозначим через

$$\Delta_k^l / \Delta_n^m = \Delta_k^l\left(\Delta_n^m\right)^{-1} = \left(\Delta_n^m\right)^{-1}\Delta_k^l.$$

**Лемма 1.** *Пусть A – матрица вида (1). Тогда элементы обратной матрицы равны*

$$\tilde{A}_{i,j} = \left(\Delta_1^N\right)^{-1}\Delta_1^{i-1}\Delta_{j+1}^N,\ i\leqslant j,\ \tilde{A}_{i,j} = \left(\Delta_1^N\right)^{-1}\Delta_1^{j-1}\Delta_{i+1}^N,\ i\geqslant j. \quad (3)$$

*Доказательство.* Покажем, что $AA^{-1}$ – единичная матрица. Используя свойство 2), получаем: $\left(AA^{-1}\right)_{i,i} = -\tilde{A}_{i-1,i} + C_i\tilde{A}_{i,i} - \tilde{A}_{i+1,i} = I$.

Для случая $i > j$, применяя свойство 1), получим

$$\left(AA^{-1}\right)_{i,j} = -\tilde{A}_{i-1,j} + C_i\tilde{A}_{i,j} - \tilde{A}_{i+1,j} = \frac{1}{\Delta_1^N}\left(-\Delta_1^{j-1}\Delta_i^N + C_i\Delta_1^{j-1}\Delta_{i+1}^N - \Delta_1^{j-1}\Delta_{i+2}^N\right) =$$
$$= \frac{\Delta_1^{j-1}}{\Delta_1^N}\left(-\Delta_i^N + C_i\Delta_{i+1}^N - \Delta_{i+2}^N\right) = 0.$$

Для оставшегося случая $i < j$ равенство нулю проверяется совершенно аналогично. □

Отметим, что формулы (3) являются обобщением выражений, которые приведены в [3, с. 61–62] и [4] для первой краевой задачи.

## 2. Метод исключения неизвестных в произвольном порядке

Выведем формулы, позволяющие вычислять коэффициенты уравнений и правую часть после исключения произвольной группы неизвестных ($u_{l+1}\ldots u_{r-1}$) из системы (1). Перепишем ее преобразуемую часть:

$$\begin{aligned} &-u_{l-1} + C_l u_l - \begin{pmatrix} I & 0 & \ldots & 0 \end{pmatrix}\bar{U}_c = f_l, \\ &-\begin{pmatrix} I & 0 & \ldots & 0 \end{pmatrix}^T u_l + A_c\bar{U}_c - \begin{pmatrix} 0 & \ldots & 0 & I \end{pmatrix}^T u_r = \bar{F}_c, \\ &-\begin{pmatrix} 0 & \ldots & 0 & I \end{pmatrix}\bar{U}_c + C_r u_r - u_{r+1} = f_r. \end{aligned} \quad (4)$$

В этих формулах $\bar{U}_c = (u_{l+1} \quad \ldots \quad u_{r-1})$, $\bar{F}_c = (f_{l+1} \quad \ldots \quad f_{r-1})$. Выразим блок $\bar{U}_c$ из 2-го уравнения системы (4) и поставим в 1-е и 3-е, получим

$$-u_{l-1} + C_l^{(2)} u_l - R_l^{(2)} u_r = f_l^{(2)}, -L_r^{(2)} u_l + C_r^{(2)} u_r - u_{r+1} = f_r^{(2)}, \tag{5}$$

где коэффициенты уравнений (5) рассчитываются по следующим формулам:

$$\begin{aligned} R_l^{(2)} &= -(I \quad 0 \quad \ldots \quad 0) A_c^{-1} (0 \quad \ldots \quad 0 \quad I)^T, L_r^{(2)} = -(0 \quad \ldots \quad 0 \quad I) A_c^{-1} (I \quad 0 \quad \ldots \quad 0)^T, \\ C_l^{(2)} &= C_l - (I \quad 0 \quad \ldots \quad 0) A_c^{-1} (I \quad 0 \quad \ldots \quad 0)^T, C_r^{(2)} = C_r - (0 \quad \ldots \quad 0 \quad I) A_c^{-1} (0 \quad \ldots \quad 0 \quad I)^T, \end{aligned} \tag{6}$$

а правые части — по формулам

$$f_l^{(2)} = f_l + (I \quad 0 \quad \ldots \quad 0) A_c^{-1} \bar{F}_c, f_r^{(2)} = f_r + (0 \quad \ldots \quad 0 \quad I) A_c^{-1} \bar{F}_c. \tag{7}$$



В случае, когда исключаемый блок расположен в начале или конце, можно формально полагать $l = 0$ или $r = N + 1$ соответственно.

**Лемма 2. (Общий вид коэффициентов уравнения и правой части при исключении произвольной группы неизвестных).** *Пусть матрица СЛАУ имеет вид (1). Тогда в результате исключения неизвестных с номерами $l+1, \ldots, r-1$ коэффициенты в (5) выражаются формулами*

$$R_l^{(2)} = L_r^{(2)} = -\frac{1}{\Delta_{l+1}^{r-1}}, \quad C_l^{(2)} = \frac{\Delta_l^{r-1}}{\Delta_{l+1}^{r-1}}, \quad C_r^{(2)} = \frac{\Delta_{l+1}^{r}}{\Delta_{l+1}^{r-1}}, \tag{8}$$

а правые части в (5) — формулами

$$f_l^{(2)} = f_l + \sum_{i=l+1}^{r-1} \frac{\Delta_{i+1}^{r-1}}{\Delta_{l+1}^{r-1}} f_i, \quad f_r^{(2)} = f_r + \sum_{i=l+1}^{r-1} \frac{\Delta_{l+1}^{i-1}}{\Delta_{l+1}^{r-1}} f_i. \tag{9}$$

При этом, поскольку результат не зависит от способа исключения, (8) и (9) не зависят от того, в каком порядке это исключение производилось.

*Доказательство.* Выражения для $L_r^{(2)}$, $R_l^{(2)}$ а также правой части можно получить, если подставить необходимые элементы матрицы $A_c^{-1}$ в формулы (6), (7). Для расчета $C_l^{(2)}$, $C_r^{(2)}$ требуется кроме этого применить свойство 1) после приведения к общему знаменателю:

$$C_l^{(2)} = C_l - \Delta_{l+2}^{r-1} / \Delta_{l+1}^{r-1} = \Delta_l^{r-1} / \Delta_{l+1}^{r-1}, C_r^{(2)} = C_r - \Delta_{l+1}^{r-2} / \Delta_{l+1}^{r-1} = \Delta_{l+1}^{r} / \Delta_{l+1}^{r-1}. \ \square$$

**Теорема.** *Уравнение, связывающее неизвестные $u_l$, $u_c$, $u_r$ (1), имеет вид*

$$-\frac{1}{\Delta_{l+1}^{c-1}} u_l + \frac{\Delta_{l+1}^{r-1}}{\Delta_{l+1}^{c-1} \Delta_{c+1}^{r-1}} u_c - \frac{1}{\Delta_{c+1}^{r-1}} u_r = f_c^{(p)}. \tag{9.1}$$

*В случае, когда исключается блоки ($u_1 \ \ldots \ u_{c-1}$) и ($u_{c+1} \ \ldots \ u_{r-1}$), имеем*

$$\frac{\Delta_1^{r-1}}{\Delta_1^{c-1} \Delta_{c+1}^{r-1}} u_c - \frac{1}{\Delta_{c+1}^{r-1}} u_r = f_c^{(p)}, \tag{9.2}$$

*а когда блоки ($u_{l+1} \ \ldots \ u_{c-1}$) и ($u_{c+1} \ \ldots \ u_N$), то*

$$-\frac{1}{\Delta_{l+1}^{c-1}} u_l + \frac{\Delta_{l+1}^{N}}{\Delta_{l+1}^{c-1} \Delta_{c+1}^{N}} u_c = f_c^{(p)}. \tag{9.3}$$

*Доказательство.* Для вывода формул (9.1)—(9.3) достаточно дважды воспользоваться леммой 2, исключив блоки неизвестных ($u_{l+1} \ldots u_{c-1}$) и ($u_{c+1} \ \ldots \ u_{r-1}$), а затем применить свойство 1). □

Вычисление правой части по формуле (9) нецелесообразно, поскольку требует большое количество арифметических действий. Вместо этого построим процесс преобразования правых частей следующим образом. Пусть к определенному моменту оказались исключенными блоки ($u_{l+1} \dots u_{c-1}$) и ($u_{c+1} \dots u_{r-1}$), а правые части $f_l^{(p)}, f_c^{(p)}, f_r^{(p)}$ известны. Исключим теперь неизвестную $u_c$, выразив ее из формулы (9.1) и подставив в формулы для соседних уравнений, что повлечет преобразование правой части этих уравнений по формулам

$$f_l^{(p+1)} = f_l^{(p)} + \frac{\Delta_{c+1}^{r-1}}{\Delta_{l+1}^{r-1}} f_c^{(p)}, \quad f_r^{(p+1)} = f_r^{(p)} + \frac{\Delta_{l+1}^{c-1}}{\Delta_{l+1}^{r-1}} f_c^{(p)} \tag{10}$$



Выражения (10) позволяют описать эволюцию правой части при исключении одной произвольной неизвестной. Таким образом, *прямой ход* сводится к заданию порядка исключения и применению для пересчета правых частей формул (10). Процесс исключения повторяется до тех пор, пока в СЛАУ не останется одно уравнение с одной неизвестной

$$\frac{\Delta_1^N}{\Delta_1^{c-1}\Delta_{c+1}^N} u_c = f_c^{(n)},$$

решая которое, получаем $u_c$.

*Обратный ход* метода реализуется по формулам (9.1) , (9.2) и (9.3) как рекуррентный процесс. Если известны $u_l$, $u_r$ и правая часть $f_c^{(p)}$, то из выражения (9.1) неизвестная $u_c$ может быть вычислена следующим образом:

$$u_c = \frac{\Delta_{l+1}^{c-1}\Delta_{c+1}^{r-1}}{\Delta_{l+1}^{r-1}} f_c^{(p)} + \frac{\Delta_{l+1}^{c-1}}{\Delta_{l+1}^{r-1}} u_r + \frac{\Delta_{c+1}^{r-1}}{\Delta_{l+1}^{r-1}} u_l. \tag{11}$$

Для реализации алгоритма по (10), (11) необходимо находить произведения отношений полиномов на некоторый вектор: $\Delta_k^l / \Delta_n^m f$ . Как и в случае скалярного аргумента, отношение матричных полиномов $\Delta_k^l / \Delta_n^m$ можно разложить на элементарные дроби, если полином в знаменателе не имеет кратных корней и степень числителя меньше степени знаменателя. Разложение определяется следующими формулами:

$$\frac{\Delta_k^l}{\Delta_n^m} = \sum_i \alpha_i \left(C - \lambda_i I\right)^{-1}, \tag{12}$$

$$\alpha_i = \frac{\Delta_k^l(\lambda_i)}{(\Delta_n^m)'(\lambda_i)} = \frac{\Delta_k^l(\lambda_i)}{\prod_{i\neq j}(\lambda_i - \lambda_j)}, \tag{13}$$

где $\lambda_i$ — корни полинома $\Delta_n^m(x)$. Значение $\Delta_k^l(\lambda_i)$ можно посчитать по рекуррентной формуле, которая вытекает из свойства 1):

$$\Delta_k^l(x) = C_l(x)\Delta_k^{l-1}(x) - \Delta_k^{l-2}(x),$$

а корни $\lambda_i$ могут быть эффективно посчитаны методом бисекции.

Выражения (12), (13) могут быть рассмотрены как следующий вычислительный процесс: для каждого корня $\lambda_i$ считаем $\alpha_i$ и находим $\left(C - \lambda_i I\right)^{-1} \alpha_i f$ методом прогонки. Если числитель на некотором корне $\lambda_i$ оказывается равным нулю, то $\lambda_i = 0$ и соответствующее слагаемое в (12) обнуляется.

В ряде случаев корни полинома известны заранее, например для первой и второй краевой задачи, что упрощает алгоритм.

## 3. Первая краевая задача для эллиптического уравнения

Матрица СЛАУ для первой краевой задачи имеет следующий вид:

$$A = \text{blocktridiag}\{-I, C, -I\}.$$

В этом случае легко проверить, что $\Delta_q^p = \mathrm{U}_{p-q+1}(C/2)$, где U — полином Чебышева второго рода. Тогда уравнение, содержащее неизвестные $u_l$, $u_c$, $u_r$, будет иметь вид

$$-\frac{1}{\mathrm{U}_{c-l-1}}u_l + \frac{\mathrm{U}_{r-l-1}}{\mathrm{U}_{c-l-1}\mathrm{U}_{r-c-1}}u_c - \frac{1}{\mathrm{U}_{r-c-1}}u_r = f_c^{(p)}. \quad (14)$$

При этом правая часть при исключении произвольной неизвестной преобразуется по формулам

$$f_l^{(p+1)} = f_l^{(p)} + \frac{\mathrm{U}_{r-c-1}}{\mathrm{U}_{r-l-1}}f_c^{(p)},\quad f_r^{(p+1)} = f_r^{(p)} + \frac{\mathrm{U}_{c-l-1}}{\mathrm{U}_{r-l-1}}f_c^{(p)}. \quad (15)$$

Для пересчета правой части требуется вычислять $\mathrm{U}_k(C/2)/\mathrm{U}_n(C/2)f$, $k < n$, а для применения формулы (14) — $\mathrm{U}_k(C/2)\,\mathrm{U}_l(C/2)/\mathrm{U}_n(C/2)f$.

Корни полинома $\mathrm{U}_n(x)$ известны: $\lambda_s = \cos(s\pi/(n+1))$. Обозначим $\theta_{s,n} = \pi s/(n+1)$. Значение производной $\mathrm{U}'_n(x)$ на этих корнях известно:

$$\mathrm{U}'_n(\lambda_s) = (-1)^{s-1}(n+1)/\sin^2(\theta_{s,n}).$$

Известно и значение полинома

$$\mathrm{U}_k(\lambda_s):\ \mathrm{U}_k(\lambda_s) = \sin((k+1)\theta_{s,n})/\sin(\theta_{s,n}).$$

Таким образом, согласно выражениям (12), (13) можно записать коэффициенты разложения в явном виде:

$$\alpha_s = (-1)^{s-1}\sin((k+1)\theta_{s,n})\sin(\theta_{s,n})/(n+1).$$

Так же на элементарные дроби раскладывается отношение $\mathrm{U}_k(x/2)\mathrm{U}_l(x/2)(\mathrm{U}_n(x/2))^{-1}$, при этом коэффициенты разложения

$$\alpha_s = (-1)^{s-1}\sin((k+1)\theta_{s,n})\sin((l+1)\theta_{s,n})/(n+1).$$

Отметим, что если $(k+1)s \bmod (n+1) = 0$ или $(l+1)s \bmod (n+1) = 0$, то в разложении соответствующая элементарная дробь отсутствует.

Алгоритм прямого хода можно построить следующим образом: сначала исключим неизвестные с нечетными номерами, затем с номерами, кратными двум и не кратными четырем, затем не кратными восьми и т. д., пока не останется одно уравнение с одной неизвестной. Далее обратный ход метода реализуется согласно формуле (14).

**Алгоритм 1.** В массиве $P$ задается правая часть СЛАУ. В результате работы алгоритма полученное решение записывается в этот же массив.

1. Прямой ход.

```
for (k = 0; k < [log2(N)]; k+=1 ) {
  for (i = 2^k; i ≤ N; i = i + 2^(k+1)) {
    l := i – 2^k; r := min(i + 2^k, N + 1);
    if (l ≥ 1)  P[l] := P[l] + UoverU(r – i – 1, r – l – 1, P[i]);
    if (r ≤ N) P[r] := P[r] + UoverU(i – l – 1, r – l – 1, P[i]);
  }
}
```

2. Обратный ход.

```
for (k = [log₂(N)]; k ≥ 0; k–=1) {
   for (i = 2^k; i ≤ N; i = i + 2^(k+1)) {
      l := i – 2^k; r := min(i + 2^k, N + 1);
      P[i] := UUoverU(i – l – 1, r – i – 1, r – l – 1, P[i]);
      if (l ≥ 1) P[i] := P[i] + UoverU(r – i – 1, r – l – 1, P[l]);
      if (r ≤ N) P[i] := P[i] + UoverU(i – l – 1, r – l – 1, P[r]);
   }
}
function UoverU(k, n, f)
   R := 0;
   for (s = 1; s ≤ n; s += 1) {
      if (k + 1)s mod (n + 1) == 0  continue;
      α_s = 2(–1)^(s–1) sin((k+1)πs/(n+1)) sin(πs/(n+1))/(n+1);
      R := R + Sweeping(C – 2cos(πs/(n+1))·I, α_s f);
   }
   return R;
function UUoverU(k, l, n, f)
   R := 0;
   for (s = 1; s ≤ n; s += 1)  {
      if ((k + 1)s mod (n + 1) == 0) or ((l + 1)s mod (n + 1) == 0)  continue;
      α_s = 2(–1)^(s–1) sin((k+1)πs/(n+1))sin((l+1)πs/(n+1))/(n+1);
      R := R + Sweeping(C – 2cos(πs/(n+1))·I, α_s f);
   }
   return R;
```

Здесь Sweeping($M$, $f$) — процедура решения СЛАУ с трехдиагональной матрицей $M$ и правой частью $f$ методом прогонки.

## 4. Вторая краевая задача для эллиптического уравнения

Матрица СЛАУ для второй краевой задачи для эллиптического уравнения с постоянными коэффициентами имеет вид

$$A = \begin{pmatrix} C/2 & -I & 0 & \cdots & 0 \\ -I & C & -I & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & C & -I \\ 0 & \cdots & 0 & -I & C/2 \end{pmatrix}$$

Отметим, что $\Delta_q^p = \mathrm{U}_{p-q+1}(C/2)$ для $q > 1$ и $p < N$.

В силу одного из свойств, связывающих полиномы Чебышева первого и второго рода, $\Delta_1^p = C/2 \cdot \mathrm{U}_{p-1}(C/2) - \mathrm{U}_{p-2}(C/2) = \mathrm{T}_p(C/2)$, где T — полином Чебышева первого рода. Аналогично $\Delta_q^N = \mathrm{T}_{N-q+1}(C/2)$.

В результате разложения $\Delta_1^N$ по первой и последней строке определителя получим

$$\Delta_1^N = C/2\Delta_2^N - \Delta_3^N = C/2\left(C/2\Delta_2^{N-1} - \Delta_2^{N-2}\right) - C/2\Delta_3^{N-1} + \Delta_3^{N-2} =$$
$$= C^2/4\,\mathrm{U}_{N-2} - \left(C\,\mathrm{U}_{N-3} - \mathrm{U}_{N-4}\right) = \left(C^2/4 - I\right)\mathrm{U}_{N-2}(C/2).$$

Таким образом, все $\Delta_q^p$ представляют собой полиномы Чебышева первого и второго рода от одной матрицы.

Построим процесс следующим образом: будем исключать только «внутренние» неизвестные, то есть на первом шаге имеющие номера 2, 4, 6, 8, 10, 12, …, на втором — 3, 7, 11 и т. д. При таком порядке исключения на прямом ходу необходимо вычислять только $\mathrm{U}_k/\mathrm{U}_n f$. В результате останется два уравнения, содержащие неизвестные $u_1$ и $u_N$:

$$\frac{\mathrm{T}_{N-1}}{\mathrm{U}_{N-2}}u_1 - \frac{1}{\mathrm{U}_{N-2}}u_N = f_1^{(p)},$$
$$-\frac{1}{\mathrm{U}_{N-2}}u_1 + \frac{\mathrm{T}_{N-1}}{\mathrm{U}_{N-2}}u_N = f_N^{(p)}.$$

Произведем еще одно исключение любой из оставшихся неизвестных, например $u_N$, что повлечет пересчет $f_1$:

$$f_1^{(p+1)} = f_1^{(p)} + \frac{1}{\mathrm{T}_{N-1}} f_N^{(p)}.$$

Из единственного уравнения с одной неизвестной получим $u_1$

$$u_1 = \frac{T_{N-1}}{\left(C^2/4 - I\right)U_{N-2}} f_1^{(p+1)}.$$

Затем вычислим $u_N = \dfrac{\mathrm{U}_{N-2}}{\mathrm{T}_{N-1}} f_N^{(p)} + \dfrac{1}{\mathrm{T}_{N-1}} u_1$.

Далее аналогично обратному ходу в первой краевой задаче по формуле (14) вычисляем $u_c$, используя найденные значения $u_l$, $u_r$ и $f_c^{(p)}$.

Таким образом, кроме отношений $\mathrm{U}_k/\mathrm{U}_n$ и $\mathrm{U}_k\mathrm{U}_l/\mathrm{U}_n$ необходимо по одному разу найти $\mathrm{U}_k/\mathrm{T}_n$, $\mathrm{T}_k/\left(\left(x^2/4-1\right)\mathrm{U}_n\right)$ и два раза $1/\mathrm{T}_n$. Корни полинома $\mathrm{T}_n$ известны: $x_s = \cos\left((2s-1)\pi/(2n)\right)$.

Обозначим $\eta_{s,n} = (2s-1)\pi/(2n)$. Выпишем значения числителя и производной знаменателя на этих корнях:

$$\mathrm{U}_k(x_s) = \sin\left((k+1)\eta_{s,n}\right)/\sin\left(\eta_{s,n}\right),$$
$$\mathrm{T}_n'(x_s) = n\sin\left(n\eta_{s,n}\right)/\sin\left(\eta_{s,n}\right) = (-1)^{s-1} n/\sin\left(\eta_{s,n}\right).$$

Для отношения $\mathrm{U}_k/\mathrm{T}_n$ получаем коэффициенты разложения:

$$\alpha_s = (-1)^{s-1}\sin\left((k+1)\eta_{s,n}\right)/n.$$

Обратимся к проблеме разложения на элементарные дроби выражения

$$\mathrm{T}_{n+1}(x/2)/\left(\left(x^2/4-1\right)\mathrm{U}_n(x/2)\right).$$

Производная знаменателя на его корнях принимает следующие значения:

$$\left(\left(x^2/4-1\right)\mathrm{U}_n\right)' = \begin{cases}\left(x^2/4-1\right)\mathrm{U}_n'(x/2), & x = 2\lambda_s,\\ x/2\,\mathrm{U}_n(x/2), & x = \pm 2,\end{cases}$$

где $\lambda_s = \cos\left(\pi s/(n+1)\right)$ — корни полинома $\mathrm{U}_n(x)$. Значение $\mathrm{U}_n'(\lambda_s)$ приводилось ранее. В силу одного из свойств полиномов Чебышева

$$T_k(\cos\theta) = \cos(k\theta),\text{ имеем: } T_{n+1}(\lambda_s) = \cos(\pi s) = (-1)^s.$$

Итак, получаем следующие коэффициенты разложения на элементарные дроби:

$$\alpha_s = \begin{cases} 2/(n+1), x = 2\lambda_s, \\ 1/(n+1), x = \pm 2. \end{cases}$$

**Алгоритм 2.**

1. Прямой ход.

**for** ($k = 0$; $k < [\log_2(N)]$; $k$+=1 ) {
  **for** ($i = 2^k + 1$; $i < N$; $i = i + 2^{k+1}$) {
    $l := i - 2^k$; $r := \min(i + 2^k, N)$;
    $P[l] := P[l]$ + UoverU($r - i - 1$, $r - l - 1$, $P[i]$);
    $P[r] := P[r]$ + UoverU($i - l - 1$, $r - l - 1$, $P[i]$);
  }
}
$P[1] := P[1]$ + UoverT(0, $N - 1$, $P[N]$);

2. Обратный ход.

$P[1]$ := SolveFirstLine($N - 2$, $P[1]$);
$P[N]$ := UoverT($N - 2$, $N - 1$, $P[N]$) + UoverT(0, $N - 1$, $P[1]$);
**for** ($k = [\log_2(N)]$; $k >= 0$; $k$ –= 1) {
  **for** ($i = 2^k + 1$; $i < N$; $i = i + 2^{k+1}$) {
    $l := i - 2^k$; $r := \min(i + 2^k, N)$;
    $P[i]$ := UUoverU($r - i - 1$, $i - l - 1$, $r - l - 1$, $P[i]$);
    $P[i] := P[i]$ + UoverU($r - i - 1$, $r - l - 1$, $P[l]$);
    $P[i] := P[i]$ + UoverU($i - l - 1$, $r - l - 1$, $P[r]$);
  }
}

**function** UoverT(k, n, $f$)
  $R := 0$;
  **for** ($s = 1$; $s \le n$; $s$ += 1) {
    $\eta_s = (2s-1)\pi/(2n);\quad \alpha_s = 2(-1)^{s-1}\sin((k+1)\eta_s)/n;$
    $R := R + \text{Sweeping}(C - 2\cos(\eta_s)\cdot I, \alpha_s f);$
  }
  **return** $R$;

**function** SolveFirstLine($n$, $f$)
  $R := 0$; $\alpha = 2/(n+1)$;
  **for** (s = 1; s ≤ n; s += 1) {
    $\eta_s = s\pi/(n+1)$; $R := R + \text{Sweeping}(C - 2\cos(\eta_s)\cdot I, \alpha f)$;
  }
  $\alpha = 1/(n+1)$;
  $R := R + \text{Sweeping}(C + 2I, \alpha f) + \text{Sweeping}(C - 2I, \alpha f)$;
  **return** $R$;

## 5. Третья краевая задача для эллиптического уравнения

Граничные условия третьего рода приводят к следующей СЛАУ:

$$(C+2\alpha I)u_1 - 2u_2 = f_1,$$
$$-u_{i-1} + Cu_i - u_{i+1} = f_i,$$
$$-2u_{N-1} + (C+2\beta I)u_N = f_N.$$

Чтобы привести матрицу данной системы к виду (1), можно разделить первое и последнее уравнения на 2. При $q > 1$, $p < N$ полиномы $\Delta_q^p$ по-прежнему равны $\mathrm{U}_{p-q+1}(C/2)$.

Для решения третьей краевой задачи можно использовать алгоритм, аналогичный тому, что изложен для второй краевой задачи, однако необходимо вычислить корни полиномов

$$\Delta_1^k(x),\ \Delta_1^N(x),\ \Delta_l^N(x),$$

например, методом бисекции.

## Заключение

Метод исключения, исследованный в данной работе, позволяет достаточно легко реализовать алгоритмы для решения краевых задач для уравнения с постоянными коэффициентами. Алгоритмы являются устойчивыми, когда матрица $C-2I$ положительно определена. Исключение неизвестных возможно в любом удобном порядке по единообразным формулам (10), (11).

Если количество блоков системы равно $2^n - 1$, то рассмотренный алгоритм совпадает с алгоритмом, который описан в [3, с. 136].

Изложенный метод допускает обобщение на случай, когда матрица СЛАУ имеет вид $A = \text{blocktridiag}\{-\beta_i L, C_i, -\gamma_i L\}$.

## Список литературы

1. *Buzbee B. L., Golub G. H., Nielson C. W.* On direct methods for solving Poisson's equation // SIAM Journal of Numerical Analysis. 1970. № 7. P. 627–656.

2. *Sweet R.* A Cyclic Reduction Algorithm for Solving Block Tridiagonal Systems of Arbitrary Dimensions // SIAM J. Number. Anal. 1977. Vol. 14, № 4. P. 706-720.

3. *Самарский А. А, Николаев Е. С.* Методы решения сеточных уравнений. М., 1978.

4. *Bank R. E., Rose D. J.* Marching Algorithms For Elliptic Boundary Value Problems. The Constant Coefficient Case // SIAM J. Number. Anal. 1977. Vol. 14, № 5.

## Об авторах

Александр Олегович Синюхин — асп., Балтийский федеральный университет им. И. Канта, Калининград.

E-mail: asinyukhin@inbox.ru

Алексей Алексеевич Буздин — канд. физ.-мат. наук, доц., Балтийский федеральный университет им. И. Канта, Калининград.

E-mail: buzdin@mail.ru

## About the authors

Alexander Sinyukhin — PhD student, I. Kant Baltic Federal University, Kaliningrad.
E-mail: asinyukhin@inbox.ru

Dr Alexey Buzdin — Ass. Prof., I. Kant Baltic Federal University, Kaliningrad.
E-mail: buzdin@mail.ru