

# Решение системы линейных уравнений методом Холецкого

Лобанова Валерия, группа 310

# Содержание

<b>1</b>	<b>Введение</b>	<b>3</b>
1.1	Постановка задачи. Разложение Холецкого . . . . .	3
1.2	Оценка сложности алгоритма построения верхнетреугольной матрицы в разложении Холецкого . . . . .	4
<b>2</b>	<b>Блочный метод Холецкого</b>	<b>5</b>
2.1	Описание блочного метода Холецкого . . . . .	5
2.2	Описание решения систем $R^T y = b$ и $DRx = y$ . . . . .	5
2.3	Хранение матриц . . . . .	6
2.4	Оценка сложности в алгоритме построения верхнетреугольной матрицы в блочном разложении Холецкого . . . . .	7
<b>3</b>	<b>Параллельный блочный метод Холецкого</b>	<b>9</b>
3.1	Описание параллельного блочного метода Холецкого . . . . .	9
3.2	Описание параллельного решения систем $R^T y = b$ и $DRx = y$ . . . . .	10
3.3	Оценка числа точек синхронизаций . . . . .	11
3.4	Оценка сложности в алгоритме построения верхнетреугольной матрицы в параллельном блочном разложении Холецкого . . . . .	12

# 1 Введение

## 1.1 Постановка задачи. Разложение Холецкого

**Задача.** Найти решение системы линейных уравнений  $Ax = b$ , где  $A$  — симметричная вещественнозначная матрицы размера  $n \times n$ ,  $b$  — известный вектор размера  $n$ ,  $x$  — неизвестный вектор.

*Идея решения.* Поиск решения будет осуществляться с помощью разложения Холецкого матрицы  $A = R^T DR$ , где

$R$  — верхнетреугольная матрица,

$D$  — диагональная матрица с 1 или  $-1$  на диагонали.

Найдем такое  $y$ , что  $R^T y = b$  и затем из условия  $DRx = y$  найдем  $x$ .  $\square$

**Теорема.** Пусть матрица  $A$  — самосопряженная и все ее угловые миноры отличны от нуля. Тогда существует матрица  $R = (r_{ij}) \in RT(n)$  с вещественными положительными элементами на главной диагонали и диагональная матрица  $D$  с вещественными равными по модулю единице диагональными элементами такие, что  $A = R^T DR$ .

*Решение задачи.* Применим точечный метод Холецкого для поиска матрицы  $R$ . Элементы  $d_{ii}, r_{ii}, r_{ij}$  могут быть вычислены по следующим формулам:

$$d_{ii} = \operatorname{sgn}\left(a_{ii} - \sum_{k=1}^{i-1} |r_{ki}|^2 d_{kk}\right), \quad i = 1, \dots, n, \quad (1)$$

$$r_{ii} = \sqrt{\left|a_{ii} - \sum_{k=1}^{i-1} |r_{ki}|^2 d_{kk}\right|}, \quad i = 1, \dots, n,$$

$$r_{ij} = (r_{ii} d_{ii})^{-1} \left(a_{ij} - \sum_{k=1}^{i-1} r_{ki} d_{kk} r_{kj}\right), \quad i < j, \quad i, j = 1, \dots, n,$$

$\square$

## 1.2 Оценка сложности алгоритма построения верхне-треугольной матрицы в разложении Холецкого

Из формул (1) следует, что для вычисления элемента  $d_{ii}$ ,  $i = 1, \dots, n$  требуется  $2(i - 1)$  операций (умножение на  $d_{ii}$  за операцию не считаем). Следовательно, вычисление всех элементов матрицы  $D$  требует

$$\sum_{i=1}^n 2(i - 1) = n(n - 1) = O(n^2), \quad n \rightarrow \infty \quad \text{операций.}$$

Для вычисления элемента  $r_{ii}$  требуется  $2(i - 1) + 1 = 2i - 1$  операций (учитываем 1 операцию извлечения корня).

При фиксированном  $i = 1, \dots, n$  вычисление элементов  $r_{ij}$  для всех  $j = i + 1, \dots, n$  по формулам (1) требует

$$\sum_{j=i+1}^n (2i - 1) = (n - i)(2i - 1) \quad \text{операций.}$$

Таким образом нахождение матрицы  $R$  требует

$$\sum_{i=1}^n (n - i)(2i - 1) + (2i - 1) = \frac{2n^2 + 3n + 1}{6} = \frac{n^3}{3} + O(n^2), \quad n \rightarrow \infty \quad \text{операций.}$$

## 2 Блочный метод Холецкого

### 2.1 Описание блочного метода Холецкого

Разобьем матрицу  $A$  на блоки  $(A_{ij})$  размера  $m \times m$ , где  $m < n$  и в случае когда  $m \nmid n \Rightarrow n = m * k + l, l \neq 0$ , крайние блоки могут иметь размеры  $m \times l, l \times m, l \times l$ . Матрицы  $R$  и  $D$  можно также искать в виде блочных матриц.

Из формул  $A = R^T D R$  и (1) ясно, что формулы для нахождения блоков матрицы  $R$  имеют вид:

$$R_{ii}^T D_i R_{ii} = A_{ii} - \sum_{j=1}^{i-1} R_{ji}^T D_j R_{ji}, \quad i = 1, \dots, k, \quad (2)$$

$$R_{ii}^T D_i R_{is} = A_{is} - \sum_{j=1}^{i-1} R_{ji}^T D_j R_{js}, \quad i, s = 1, \dots, k, \quad i < s$$

$$R_{is} = D_i (R_{ii}^T)^{-1} (A_{is} - \sum_{j=1}^{i-1} R_{ji}^T D_j R_{js}), \quad i, s = 1, \dots, k, \quad i < s \quad (3)$$

Тем самым сначала получаются блоки  $R_{ii}$  и  $D_i$  точечным разложением Холецкого из формулы (2), а затем используя (3) вычисляются  $R_{is}$  для  $s = i + 1, \dots, n$ .

### 2.2 Описание решения систем $R^T y = b$ и $DRx = y$

Для решения  $R^T y = b$  представляем, что  $R^T$  на самом деле не транспонированная и лежит в памяти как  $R$ , но работаем с ней как с транспонированной. Тогда получаем следующий алгоритм:

```
for (j = 0; j < n; j++)
{
    sum = 0;
    for (i = 0; i < j; i++)
        sum += Y[i] * R_{ij};

    Y[j] = (B[j] - sum) / R_{jj};
}
```

Для решения  $DRx = y$  важно учитывать, что умножение на матрицу  $D$  можно производить после подсчета суммы.

```
for (i = n - 1; i >= 0; i--)
{
    sum = 0;
    for (j = n - 1; j > i; j--)
        sum += X[j] * R_{ij};

    X[i] = D[i] * (Y[i] - sum * D[i]) / R_{ii};
}
```

## 2.3 Хранение матриц

Так как матрица  $A$  симметричная, то логично хранить не всю матрицу, а только верхнюю ее часть над главной диагональю и саму диагональ.

$$a_{00} = a[0]; \quad a_{11} = a[n]; \quad a_{22} = a[n + (n - 1)]; \dots$$

$$a_{ii} = a\left[\sum_{j=0}^{i-1} (n - j)\right] = a[i * (2 * n - i + 1) / 2];$$

$$a_{is} = a\left[\sum_{j=0}^{i-1} (n - j)\right] = a[i * (2 * n - i + 1) / 2 + (s - i)], \quad i \leq s$$

У матрицы  $D$  хранить нужно только диагональ в массиве длины  $n$ .

При вычислении матрицы  $R$  элементы  $R_{is}$  можно записывать сразу на место  $A_{is}$ , так как  $A_{is}$  больше не будет использоваться.

## 2.4 Оценка сложности в алгоритме построения верхнетреугольной матрицы в блочном разложении Холецкого

Если известно количество операций в случае  $l = 0$ , то количество операций при  $l \neq 0$  можно оценить сверху, заменив в оценке  $k$  на  $k + 1$ .

Начнём с оценки количества операций для  $R_{ji}^T D_j R_{js}$ , чтобы не путаться в индексах рассмотрим это произведение как  $R^T D R$ , тогда

$$(R^T D R)_{ij} = \sum_{k=1}^{\min(i,j)} r_{ki} d_k r_{kj}$$

здесь  $\min(i, j) - 1$  аддитивных и  $\min(i, j)$  мультипликативных операций, то есть всего  $2\min(i, j) - 1$  операций для одного элемента (по аналогии с неблочным методом умножение на  $d_i$  за операцию не считаем).

Тогда для вычисления  $R_{ji}^T D_j R_{js}$  требуется

$$\begin{aligned} \sum_{i=1}^m \sum_{j=1}^m (2\min(i, j) - 1) &= \sum_{i=1}^m \sum_{j=1}^i (2j - 1) + \sum_{i=1}^m \sum_{j=i+1}^m (2i - 1) = \\ &= \frac{m(m+1)(2m+1)}{6} + \frac{m(2m^2 - 3m + 1)}{6} = \frac{m(2m^2 + 1)}{3} \end{aligned}$$

Обозначим как  $Mult(m) = (2m^3 + m)/3$ .

Для вычисления  $A_{is} - \sum_{k=1}^{i-1} R_{ji}^T D_j R_{js}$  требуется:

$$H(i, m) = (i - 1)(Mult(m) + m^2) = (i - 1)(2m^3 + 3m^2 + m)/3 \quad \text{операций.}$$

Сложность разложения Холецкого  $Chol(m) = m^3/3$ .

Сложность вычисления блока  $R_{ii}$  и  $D_i$ :  $H(i, m) + Chol(m)$

Сложность вычисления всех диагональных блоков:

$$S_1(n, m, k) = \sum_{i=1}^k (H(i, m) + Chol(m)) = n(2mn + 3n + k - 3m - 1)/6$$

Умножение на треугольную матрицу требует  $Y(m) = m^3$  операций. Здесь имеется ввиду умножение на  $(R_{ii}^T)^{-1}$  в формуле (3). Подсчёт обратной к  $R_{ii}^T$  учтём позже, так как это вычисление выполняется 1 раз при подсчете всей строки.

Итак, сложность вычисления недиагонального блока  $R_{ij}$ :

$$R(i, m) = H(i, m) + Y(m) = (i - 1)(2m^3 + 3m^2 + m)/3 + m^3$$

Сложность вычисления всех недиагональных блоков  $R$ :

$$S_2(n, m, k) = \sum_{i=1}^k \sum_{s=i+1}^k R(i, m) = n(k - 1)(2mn + 3n + k + 5m^2 - 6m - 2)/18$$

Для вычисления строки -  $R_{ij}$  при фиксированном  $i$  требуется  $(R_{ii}^T)^{-1}$ , следовательно нужно  $(k - 1)$  раз найти обратную матрицу за  $S_3(n, m) = (k - 1)Chol(m) = (k - 1)m^3/3$  операций.

Итак, нахождение всех блоков  $R_{is}$  требует

$$\begin{aligned} S(n, m) &= S_1 + S_2 + S_3 = \frac{n(2mn + 3n + k - 3m - 1)}{6} + \\ &+ \frac{n(k - 1)(2mn + 3n + k + 5m^2 - 6m - 2)}{18} + \frac{(k - 1)m^3}{3} = \\ &= \frac{n(2n^2 + m^2 + 9mn - 3m - 1 + 3nk + k^2)}{18} - \frac{m^3}{3} = \\ &= \boxed{\frac{n^3}{9} + \frac{nm^2}{18} + \frac{n^2m}{2} - \frac{nm}{6} - \frac{n}{18} + \frac{n^3}{6m} + \frac{n^3}{18m^2} - \frac{m^3}{3}} \\ S(n, n) &= \frac{n^3}{9} + \frac{n^3}{18} + \frac{n^3}{2} - \frac{n^2}{6} - \frac{n}{18} + \frac{n^2}{6} + \frac{n}{18} - \frac{n^3}{3} = \frac{n^3}{3} \\ S(n, 1) &= \frac{n^3}{9} + \frac{n}{18} + \frac{n^2}{3} - \frac{n}{6} - \frac{n}{18} + \frac{n^3}{6} + \frac{n^3}{18} - \frac{1}{3} = \frac{n^3}{3} + \frac{n^2}{3} - \frac{n}{6} \\ &\boxed{S(n, n) = \frac{n^3}{3} \quad S(n, 1) = \frac{n^3}{3} + O(n^2), \quad n \rightarrow \infty} \end{aligned}$$



## 3 Параллельный блочный метод Холецкого

### 3.1 Описание параллельного блочного метода Холецкого

Из формул (2) и (3) видно, что после вычисления диагонального блока и его обращения все блоки вне диагонали можно искать в любом порядке в контексте текущей строки, при чём для вычисления блока  $R_{is}$  требуются блоки из столбцов  $i$  и  $s$ , находящиеся строго в предыдущих строках.

Пусть  $p$  — количество потоков и потоки никак не мешают друг другу. Принадлежность столбца потоку определяется так:  $t$ -ый поток обрабатывает столбцы с номерами  $t + zp$ , где  $z \in \mathbb{Z}$ .

Таким образом, предлагается следующий параллельный алгоритм, в котором каждый поток считает блоки только из своего столбца.

```
for (i = 0; i < block_lim; i++)
{
    R_ii_tmp = A_{ii};
    calculate_diag_block R_{ii};
    inverse_upper_matrix R_{ii};

    if (i % p < t_id)
        j = i - (i % p) + t_id;
    else
        j = i - (i % p) + p + t_id;

    for (; j < block_lim; j += p)
    {
        R_ij_tmp = A_{ij};
        calculate_block R_{ij};
        A_{ij} = R_ij_tmp;
    }

    Barrier ();

    if (i % p == t_id)
        A_{ii} = R_ii_tmp;
}
```

### 3.2 Описание параллельного решения систем $R^T y = b$ и $DRx = y$

Сделаем параллельным описанный выше линейный алгоритм решения системы  $R^T y = b$ .

Каждый поток подсчитывает свой  $y[i]$  по следующему правилу: если  $i = t \pmod{p}$ , то поток с номером  $t$  вычисляет  $y[i]$ .

```
for (j = 0 + t_id; j < n; j += p)
{
    sum = 0;
    for (i = 0; i <= j - p; i++)
        sum += Y[i] * R_{ki};

    pthread_mutex_lock (mut);
    while (*elems_done < i)
        pthread_cond_wait (cond, mut);
    pthread_mutex_unlock (mut);

    for (; i < j; i++)
        sum += Y[i] * R_{ij};

    Y[j] = (B[j] - sum) / R_{jj};

    pthread_mutex_lock (mut);
    (*elems_done)++;
    pthread_cond_broadcast (cond);
    pthread_mutex_unlock (mut);
}
```

Аналогично получается параллельный алгоритм для решения системы  $DRx = y$ . Каждый поток подсчитывает свой  $x[i]$  по следующему правилу: если  $i = t \pmod{p}$ , то поток с номером  $t$  вычисляет  $y[n - 1 - i]$ .

```
for (i = n - 1 - t_id; i >= 0; i -= p)
{
    sum = 0;
    for (j = n - 1; j >= i + p; j--)
        sum += X[j] * R_{ij};

    pthread_mutex_lock (mut);
    while (*elems_done < n - 1 - i)
```

```

    pthread_cond_wait (cond, mut);
pthread_mutex_unlock (mut);

for (; j > i; j--)
    sum += X[j] * R_{ij};

X[i] = D[i] * (B[i] - sum * D[i]) / R_{ii};

pthread_mutex_lock (mut);
(*elems_done)++;
pthread_cond_broadcast (cond);
pthread_mutex_unlock (mut);
}

```

### 3.3 Оценка числа точек синхронизаций

Считаем, что  $n = mk + l$ ,  $l = 0$ .

Разложение матрицы методом Холецкого требует  $k$  барьеров. При решении систем  $R^T y = b$  и  $DRx = y$  используется по  $n$  точек синхронизации. Причем каждый поток использует только свою память, а синхронизация осуществляется за счет условной переменной, где условие - это количество уже вычисленных элементов.

Таким образом весь алгоритм требует  $\boxed{\frac{n}{m} + 2n}$  точек синхронизаций.

### 3.4 Оценка сложности в алгоритме построения верхнетреугольной матрицы в параллельном блочном разложении Холецкого

Из приведенного выше алгоритма видно, что все потоки вычисляют совпадающие диагональные блоки и обратные к ним, поэтому

$$S_1(n, m, p) = S_1(n, m) \quad \text{и} \quad S_3(n, m, p) = S_3(n, m)$$

Эти формулы из оценки блочного алгоритма.

Вычисление внедиагональных блоков происходит полностью параллельно, поэтому  $S_2(n, m, p) = S_2(n, m)/p$

Итак,

$$\begin{aligned} S(n, m, p) &= S_1 + \frac{S_2}{p} + S_3 = \frac{n(2mn + 3n + k - 3m - 1)}{6} + \\ &+ \frac{n(k-1)(2mn + 3n + k + 5m^2 - 6m - 2)}{18p} + \frac{(k-1)m^3}{3} = \\ &= \frac{n^3}{9} + \frac{n^2m}{6p} + \frac{n^2m}{3} + \frac{nm^2}{3} - \frac{5nm^2}{18p} - \frac{m^3}{3} + \frac{n^2}{2} - \frac{n^2}{2p} - \frac{nm}{2} + \frac{nm}{3p} - \\ &\quad - \frac{n}{6} + \frac{n}{9p} + \frac{n^3}{6pm} + \frac{n^3}{18pm^2} + \frac{n^2}{6m} - \frac{n^2}{6pm} \end{aligned}$$

Причем  $S(n, m, 1) = S(n, m)$