



**Нижегородский государственный университет
им. Н.И.Лобачевского**

Факультет Вычислительной математики и кибернетики

Параллельные численные методы

Метод Холецкого

При поддержке компании Intel

Баркалов К.А.,
Кафедра математического обеспечения ЭВМ

Содержание

- ❑ Постановка задачи
- ❑ Разложение Холецкого
 - Столбцовый алгоритм
 - Строчный алгоритм
 - Оценка трудоемкости
 - Вычислительная погрешность разложения
- ❑ Распараллеливание метода
 - Оценка эффективности
 - Блочный алгоритм
- ❑ Результаты экспериментов



Постановка задачи

- Рассмотрим систему из n линейных алгебраических уравнений вида

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\&\dots \\a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n\end{aligned}$$

- В матричном виде система может быть представлена как

$$Ax=b$$

- $A=(a_{ij})$ есть вещественная матрица размера $n \times n$; b и x – вектора из n элементов.
- Будем искать значения вектора неизвестных x , при которых выполняются все уравнения системы.



Частный случай

- Матрица A – симметричная, если $A=A^T$.
- Матрица A – положительно определенная, если $\forall x > 0 \rightarrow (Ax, x) > 0$.
- Рассмотрим систему уравнений с симметричной положительно определенной матрицей A . Они возникают:
 - При решении задач математической физики
 - методом конечных разностей
 - методом конечных элементов
 - При решении задачи аппроксимации методом наименьших квадратов



Разложение Холецкого

- Основная идея: разложение матрицы A в произведение $A=LL^T$, где L – нижняя треугольная матрица, $l_{ii}>0$.

$$L = \begin{pmatrix} l_{1,1} & 0 & \dots & 0 \\ l_{2,1} & l_{2,2} & \dots & 0 \\ & & \dots & \\ l_{n,1} & l_{n,2} & \dots & l_{n,n} \end{pmatrix}$$

Разложение Холецкого можно получить для любой симметричной положительно определенной матрицы.

- Если разложение получено, то решение системы сводится к последовательному решению двух систем уравнений с треугольными матрицами

$$Ly=b, L^Tx=y.$$



Разложение Холецкого

□ Из условия разложения получаем $a_{ij} = \sum_{k=1}^n l_{ik} l_{kj}^T \quad i, j = \overline{1, n}$

□ Из симметрии матрицы A следует

$$a_{ij} = \sum_{k=1}^{i-1} l_{ik} l_{kj}^T + l_{ii} l_{ij} + \sum_{k=i+1}^n l_{ik} l_{kj}^T = \sum_{k=1}^{i-1} l_{ik} l_{kj}^T + l_{ii} l_{ij}$$

□ В частности, при $i=j$ получаем $a_{ii} = \sum_{k=1}^{i-1} l_{ik}^2 - l_{ii}^2$, и $l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2} (*)$

□ Далее, при $i < j$ получим $l_{ij} = \frac{1}{l_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right) (**)$

□ Таким образом, элементы матрицы L можно вычислить, начиная с ее левого угла, по формулам (*) и (**).



Разложение Холецкого

□ Схема алгоритма

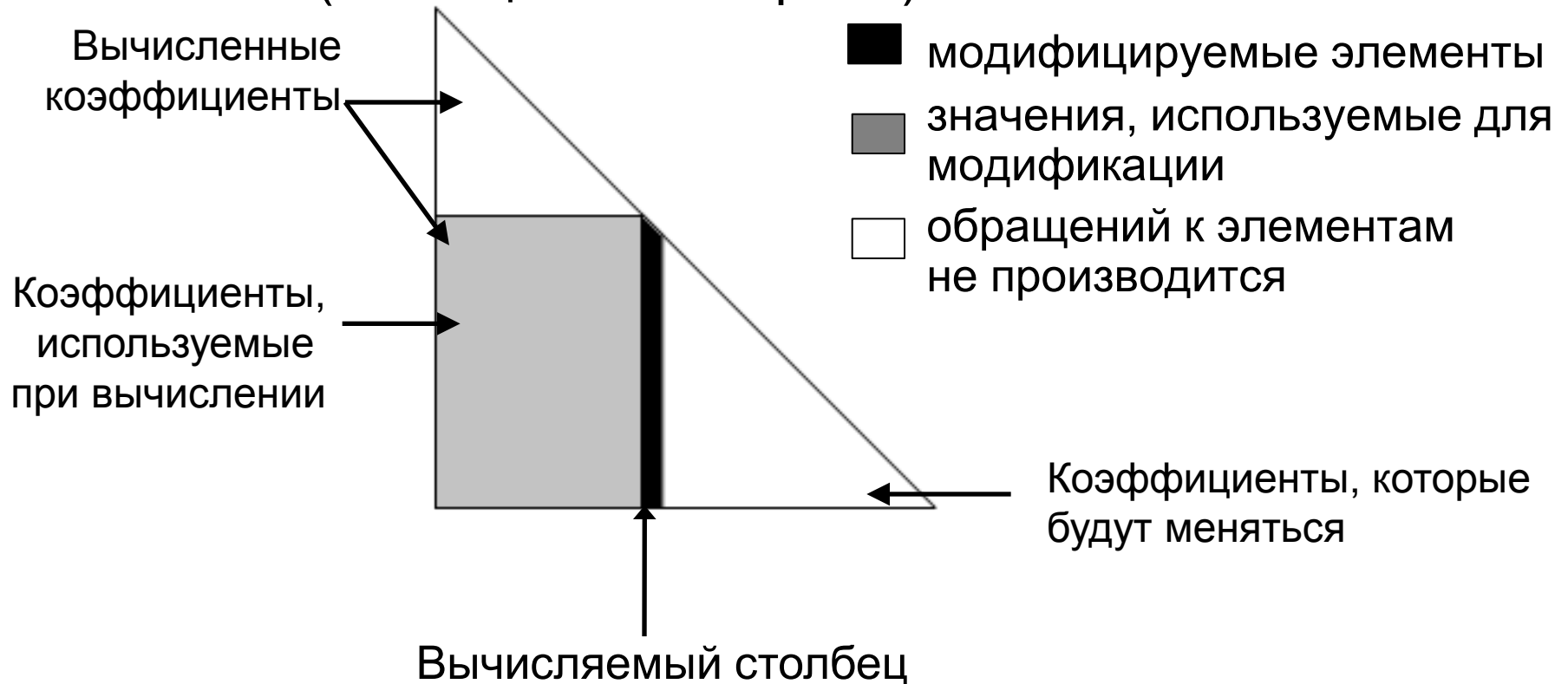
```
for i = 1 to n
  for k = 1 to i - 1
     $a_{ii} = a_{ii} - (a_{ik})^2;$ 
   $a_{ii} = \text{sqrt}(a_{ii});$ 
  for j = i+1 to n
    for k = 1 to i - 1
       $a_{ji} = a_{ji} - a_{ik} a_{jk};$ 
     $a_{ji} = a_{ji} / a_{ii};$ 
  end
end
end
```

$$l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}$$

$$l_{ij} = \frac{1}{l_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right)$$

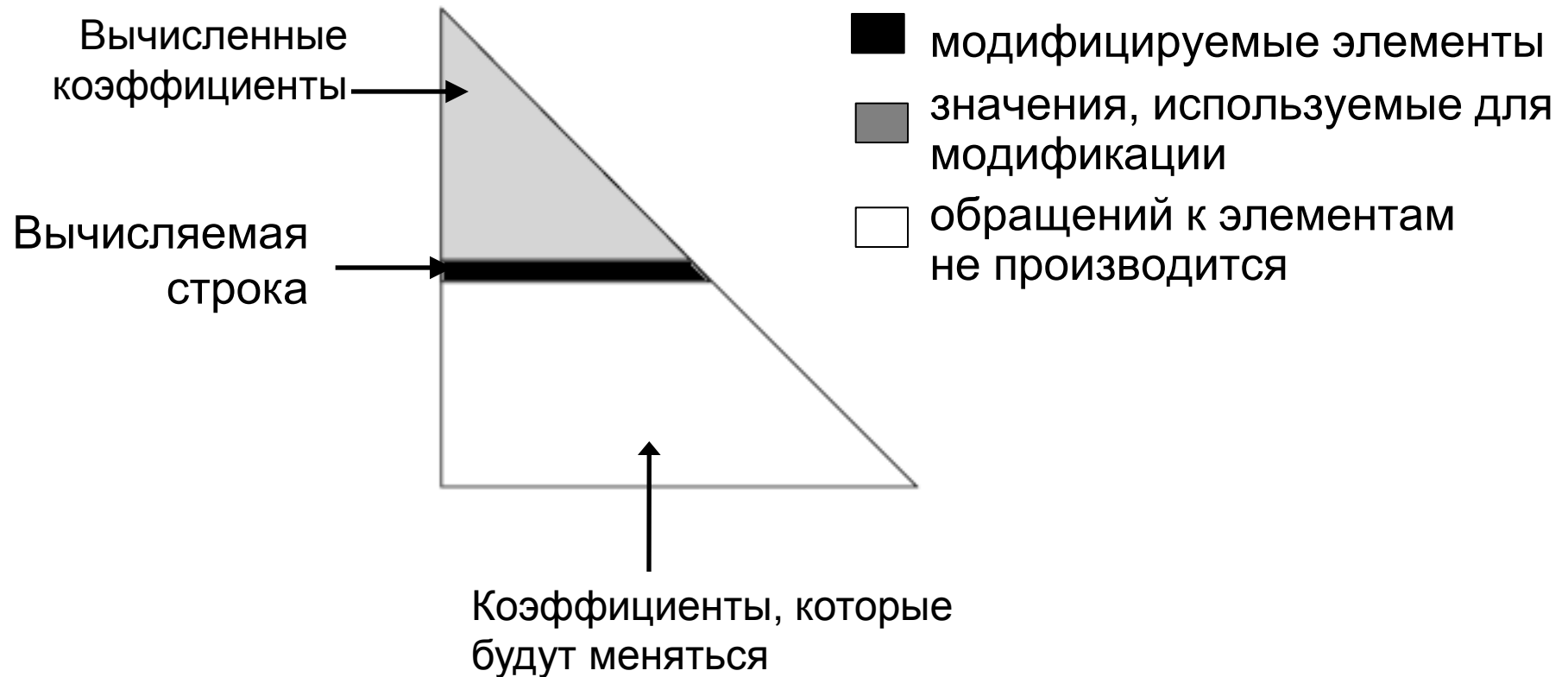
Разложение Холецкого – столбцовый алгоритм

- Общая схема состояния данных на i -ой итерации разложения (столбцовый алгоритм).



Разложение Холецкого – строчный алгоритм

- Общая схема состояния данных на i -ой итерации разложения (строчный алгоритм).



Разложение Холецкого - трудоемкость

□ Трудоемкость алгоритма

□ Вычисления по формуле $l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}$ потребуют

$$\sum_{i=2}^n 2(i-1) = n(n-1) = O(n^2)$$

□ Вычисления по формуле $l_{ij} = \frac{1}{l_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right)$ при $j=\text{const}$ потребуют

$$\sum_{i=2}^{j-1} 2(i-1) = (j-2)(j-1)$$

□ Всего $\sum_{j=2}^n (j-2)(j-1) = \sum_{k=1}^{n-1} k(k-1) = \frac{n(n-1)(n-2)}{3} = \frac{1}{3}n^3 + O(n^2)$



Метод Холецкого – обратный ход

- После разложения матрицы становится возможным определение значений неизвестных:

$$(LL^T)x=b$$

Решение двух треугольных систем

$$Ly=b$$

$$LTx=y$$

Трудоемкость обратного хода $O(n^2)$



Погрешность решения

- Выбор главных элементов не требуется. Известно, что решение, полученное с округлением, удовлетворяет возмущенной системе

$$(A + \Delta A) x = b$$

где $|\Delta A| \leq 3n\varepsilon_m |L| |L^T| + O(\varepsilon_m^2)$

Можно показать, что $\| |L| |L^T| \| \leq n \|A\|$

Откуда следует оценка

$$\|\Delta A\| \leq 3n^2 \varepsilon_m \|A\| + O(\varepsilon_m^2)$$

Сказанное справедливо для норм $\| \cdot \|_\infty$ и $\| \cdot \|_1$.



Параллельный алгоритм

- ❑ Все вычисления сводятся к однотипным вычислительным операциям над строками матрицы L
- ❑ Следовательно, в основу параллельной реализации алгоритма может быть положен принцип распараллеливания по данным,
- ❑ В качестве *базовой подзадачи* примем все вычисления, связанные с обработкой одной строки матрицы A и соответствующего элемента вектора b .



Оценка эффективности

- Время работы последовательного алгоритма

$$T_1 = \frac{1}{3} n^3 \tau$$

где τ – время выполнения одной операции.

- Время работы параллельного алгоритма

$$T_p = \frac{n^3 \tau}{3p}$$

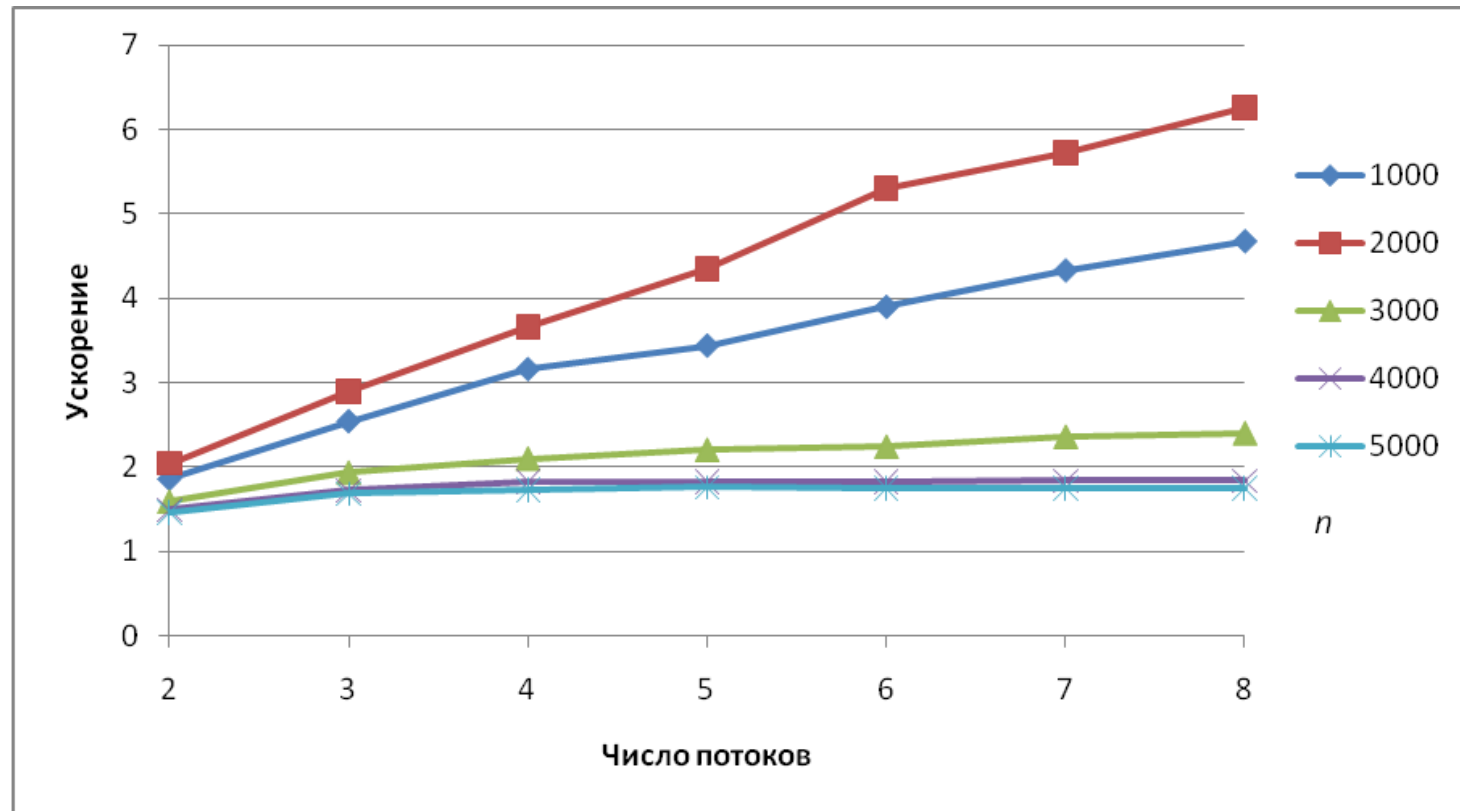
- С учетом накладных расходов δ на создание/закрытие параллельной секции

$$T_p = \frac{2n^3 \tau}{3p} + 3(n-1)\delta$$



Результаты экспериментов

□ Ускорение по отношению к однопоточной версии



□ Наблюдаем неэффективную работу с кэш-памятью

Блочный алгоритм

- ❑ Недостатком изложенного тривиального алгоритма является то, что его схема плохо соответствует правилам использования *кэш-памяти* компьютера.
- ❑ В языке C размещение данных в памяти осуществляется по строкам матрицы A .
- ❑ В рассмотренном нами алгоритме вычисления проводятся по столбцам, и это приводит к низкой эффективности использования кэша.
- ❑ Возможный способ улучшения ситуации – укрупнение вычислительных операций, приводящее к последовательной обработке некоторых прямоугольных подматриц матрицы A .



Блочный алгоритм

- ❑ Разложение осуществляется путем переписывания исходной матрицы элементами искомого фактора L сверху вниз по блокам.
- ❑ Пусть r – размер блока, тогда

$$A = \begin{bmatrix} A_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix} \quad L = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}$$

где A_{11} – подматрица матрицы A размера $r \times r$, A_{21} – размера $(n-r) \times r$, A_{22} – размера $(n-r) \times (n-r)$;

L_{11} , L_{21} , L_{22} – соответствующего размера подматрицы L .



Блочный алгоритм

- Найдем связь между блоками L и A .

$$\begin{bmatrix} A_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \cdot \begin{bmatrix} L_{11}^T & L_{21}^T \\ 0 & L_{22}^T \end{bmatrix} = \begin{bmatrix} L_{11}L_{11}^T & L_{11}L_{21}^T \\ L_{21}L_{11}^T & L_{21}L_{21}^T + L_{22}L_{22}^T \end{bmatrix}$$

Отсюда получаем

$$A_{11} = L_{11}L_{11}^T \quad A_{21} = L_{21}L_{11}^T \quad A_{22} = L_{21}L_{21}^T + L_{22}L_{22}^T$$

- Используя данные матричные равенства, можно найти блоки L_{11}, L_{21} .

Блочный алгоритм

- ❑ Блок L_{11} может быть получен с помощью стандартного алгоритма, т.к. формула $A_{11} = L_{11}L_{11}^T$ соответствует разложению Холецкого для матрицы A_{11} .

- ❑ Блок L_{21} может быть найден из соотношения

$$L_{21}L_{11}^T = A_{21}$$

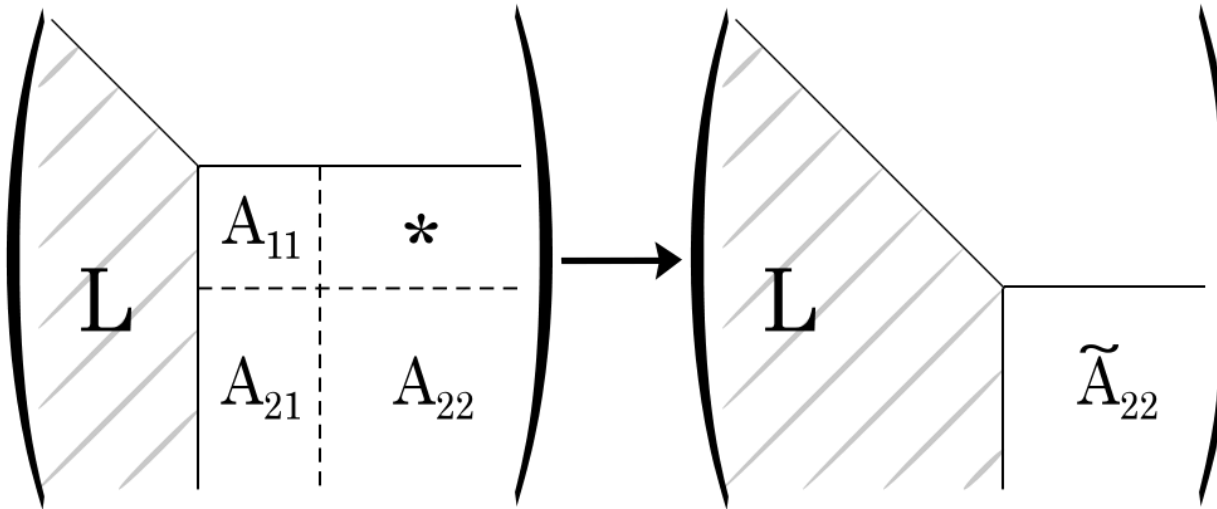
как решение треугольной системы (т.к. L_{11}^T - треугольная матрица) с несколькими правыми частями A_{21} . Для этого также можно применить блочный алгоритм, рассмотренный ранее.

Блочный алгоритм

- Далее вычисляем редуцированную матрицу \tilde{A}_{22} как

$$\tilde{A}_{22} = A_{22} - L_{21}L_{21}^T = L_{22}L_{22}^T$$

- Фактор Холецкого для матрицы \tilde{A}_{22} совпадает с искомым блоком L_{22} , и для его нахождения можно применить описанный алгоритм рекурсивно.



Оценка трудоемкости

- Данная вычислительная процедура включает в себя

$$n^3/3 + O(n^2)$$

операций, как и другие возможные реализации разложения.

- Вклад матричных операций в общее число действий аппроксимируется величиной

$$1 - 1/N^2$$

- При правильном выборе размера блока матричные операции (которые эффективно распараллеливаются) будут составлять большую часть вычислений.



Распараллеливание блочного алгоритма

- Распараллеливание возможно для следующих вычислительных процедур:
 - вычисление блока L_{11} (параллельная версия стандартного алгоритма);
 - вычисление блока L_{21} (параллельное решение систем линейных уравнений с треугольной матрицей и разными правыми частями);
 - выполнение матричного умножения при вычислении редуцированной матрицы \tilde{A}_{22}
- Эффективность параллельного блочного алгоритма будет определяться эффективностью распараллеливания матричных операций.



Результаты экспериментов

□ Сравнение исходного алгоритма с блочным

n	Время работы t , с					
	---	$r=10$	$r=20$	$r=50$	$r=100$	$r=200$
1000	0,11	0,28	0,21	0,16	0,16	0,15
2000	1,05	2,62	1,99	1,39	1,23	1,09
3000	4,16	9,81	7,25	4,84	4,17	3,71
4000	10,00	25,05	18,22	11,74	10,00	9,63
5000	19,59	49,23	35,82	23,24	20,39	20,22

использовался алгоритм умножения матриц по определению

Результаты экспериментов

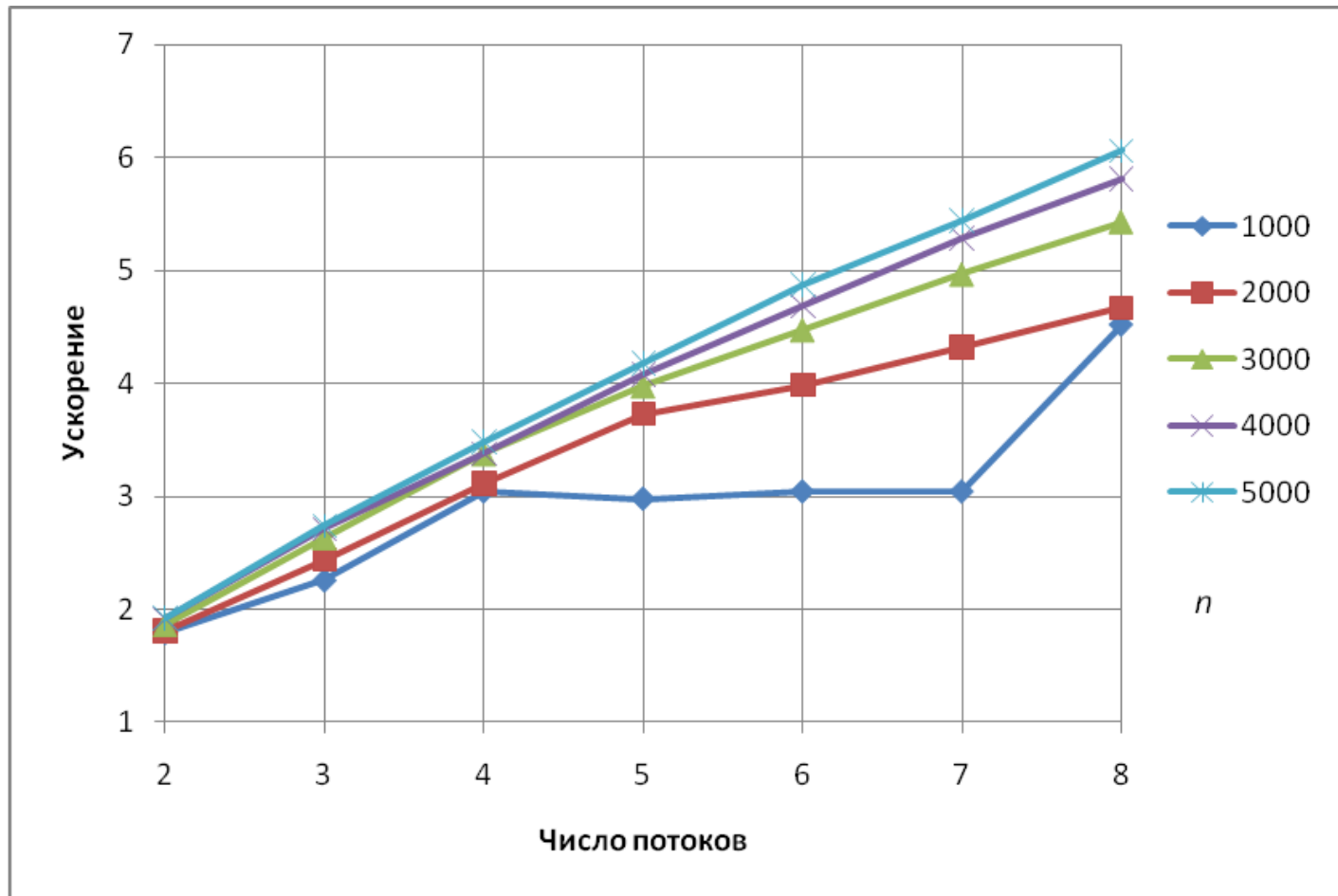
□ Сравнение исходного алгоритма с блочным

n	Время работы t			
	---	$r=50$ (25×50)	$r=100$ (25×50)	$r=200$ (10×200)
1000	0,12	0,11	0,12	0,13
2000	1,07	0,78	0,80	0,83
3000	4,20	2,57	2,61	2,64
4000	10,11	5,98	5,99	6,10
5000	19,75	12,57	11,50	11,84

использовался блочный алгоритм умножения матриц

Результаты экспериментов

□ Ускорение параллельного блочного алгоритма



Заключение

- На занятии рассмотрено:
 - Разложение Холецкого
 - Столбцовый алгоритм
 - Строчный алгоритм
 - Оценка трудоемкости
 - Вычислительная погрешность разложения
 - Распараллеливание метода
 - Оценка эффективности
 - Блочный алгоритм
 - Результаты экспериментов
 - Сравнение блочного и неблочного алгоритмов
 - Ускорение параллельного блочного алгоритма



Литература

1. Самарский А.А., Гулин А.В. Численные методы. – М.: Наука, 1989.
2. Голуб Дж., Ван Лоун Ч. Матричные вычисления. – М.: Мир, 1999.
3. Белов С.А., Золотых Н.Ю. Численные методы линейной алгебры. – Н.Новгород, Изд-во ННГУ, 2005.



Ресурсы сети Интернет

4. Интернет-университет суперкомпьютерных технологий.
[<http://www.hpcu.ru>].
5. Intel Math Kernel Library Reference Manual.
[<http://software.intel.com/sites/products/documentation/hpc/mkl/mklman.pdf>].



Авторский коллектив

- ❑ Баркалов Константин Александрович,
к.ф.-м.н., старший преподаватель кафедры
Математического обеспечения ЭВМ факультета ВМК ННГУ.
barkalov@fup.unn.ru
- ❑ Коды учебных программ разработаны Маловой Анной и
Сафоновой Яной

