Нижегородский государственный университет
им. Н.И.Лобачевского

Факультет Вычислительной математики и кибернетики

Параллельные численные методы

Метод Гаусса

При поддержке компании Intel

Баркалов К.А.,
Кафедра математического обеспечения ЭВМ

Содержание

- Постановка задачи
- Нормы векторов и матриц
- Решение треугольных систем
- Метод Гаусса для систем общего вида
- Связь метода Гаусса и LU-разложения
- Вычислительная погрешность
- Распараллеливание метода
- Оценка эффективности
- Результаты экспериментов

Постановка задачи

- Рассмотрим систему из n линейных алгебраических уравнений вида

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

- В матричном виде система может быть представлена как

$$Ax=b$$

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

Нормы векторов и матриц

- Линейное пространство называют *нормированным*, если каждому его вектору x поставлено в соответствие число, называемое *нормой* и обозначаемое как $\|x\|$.
- *Аксиомы* нормированного пространства:

1. $\|x\| \geq 0$, причем $\|x\| = 0 \leftrightarrow x = 0$.
2. $\|x + y\| \leq \|x\| + \|y\|$
3. $\|\alpha x\| \leq |\alpha| \, \|x\|$

- Пример нормы вектора

$$\|x\|_p = \left(|x_1|^p + |x_2|^p + \ldots + |x_n|^p\right)^{1/p}$$

где $p \geq 1$.

Нормы векторов и матриц

- При p=1,2 и при $p \to \infty$ получаем соответственно манхеттенскую $\|x\|_1$, евклидову $\|x\|_2$ и чебышеву $\|x\|_\infty$ нормы

$$\|x\|_1 = |x_1| + |x_2| + \ldots + |x_n|$$

$$\|x\|_2 = \sqrt{|x_1|^2 + |x_2|^2 + \ldots + |x_n|^2} = \sqrt{x^T x}$$

$$\|x\|_\infty = \max\{|x_1|, |x_2|, \ldots, |x_n|\}$$

- Для нормы в пространстве матриц требуют также выполнения свойства

4. $\|AB\| \le \|A\| \, \|B\|$

Нормы векторов и матриц

- Пусть A – матрица, x – вектор. Нормы – согласованы, если $\|Ax\| \le \|A\| \cdot \|x\|$
- Матричная норма – *подчиненная* норме вектора, если

$$\|A\| = \sup_{x \ne 0} \frac{\|Ax\|}{\|x\|}$$

- Подчиненная норма всегда согласована со своей векторной нормой.
- Матричная норма, подчиненная векторной норме $\|x\|_p$, обозначается $\|A\|_p$

Нормы векторов и матриц

- Нормы $\|A\|_1$, $\|A\|_2$, $\|A\|_\infty$ называются манхэттенской, спектральной и чебышевской нормами. Известно, что

$$\|A\|_1 = \max_j \left\{ \sum_{i=1}^{n} | a_{ij} | \right\} \qquad \|A\|_2 = \max_j \left\{ \sqrt{\lambda_j (A^T A)} \right\} \qquad \|A\|_\infty = \max_i \left\{ \sum_{j=1}^{n} | a_{ij} | \right\}$$

- Для симметричной матрицы $\|A\|_2 = \max_j \left\{ \lambda_j (A) \right\}$
- Норма $\|A\|_F$ называется фробениусовской нормой

$$\|A\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^2}$$

- Нормы $\|x\|_2$ и $\|A\|_F$ согласованы.

Число обусловленности матрицы

- *Число обусловленности* невырожденной матрицы A

$$\operatorname{cond} A = \|A\|\,\|A^{-1}\|$$

- Если матричная норма подчинена векторной, то

$$\operatorname{cond} A = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} \Big/ \inf_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

- Для спектрального числа обусловленности справедливо

$$\operatorname{cond}_2 A = \sqrt{\frac{\max_j \lambda_j (A^T A)}{\min_j \lambda_j (A^T A)}}$$

- Для симметричной матрицы A получаем $\operatorname{cond}_2 A = \dfrac{\max_j |\lambda_j (A)|}{\min_j |\lambda_j (A)|}$

Треугольные матрицы

- Частный случай: верхняя (нижняя) треугольная матрица.

$$
\begin{aligned}
a_{11}x_1 \quad + a_{12}x_2 \quad &+ \ldots + a_{1n}x_n &= b_1 \\
a_{22}x_2 \quad &+ \ldots + a_{2n}x_n &= b_2 \\
\ldots \\
a_{nn}x_n &&= b_n
\end{aligned}
\qquad\qquad Ux=b
$$

- Найдем решение системы *обратной подстановкой*

$$
x_n = b_n / a_{nn} \qquad x_i = \left(c_i - \sum_{j=i+1}^{n} a_{ij}x_j \right) \Big/ a_{ii}\ , \quad i=n-1,\ldots,1.
$$

- Обратная подстановка требует $n^2+O(n)$ операций.

Случай нескольких правых частей

- $UX=B$, где $U \in R^{n\times n}$, $B \in R^{n\times m}$, $X \in R^{n\times m}$, $n>m$.
- Разобьем систему на блоки размера $m \times m$.
- Т.к. U_{NN} – треугольная матрица, то решим систему $U_{NN}X_N=B_N$ для всех неизвестных из X_N.
- Исключим X_N из всех блочных уравнений от N–1 до 1
- и т.д.

$$\begin{bmatrix} U_{11} & U_{12} & \dots & U_{1N} \\ 0 & U_{22} & \dots & U_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & U_{NN} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix}$$

$$\begin{bmatrix} U_{11} & U_{12} & \dots & U_{1N-1} \\ 0 & U_{22} & \dots & U_{2N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & U_{N-1N-1} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{N-1} \end{bmatrix} = \begin{bmatrix} B_1 - U_{1N}X_N \\ B_2 - U_{2N}X_N \\ \vdots \\ B_{N-1} - U_{N-1N}X_N \end{bmatrix}$$

Случай нескольких правых частей

- Общее число операций – n^2.
- Определим долю матричных умножений в общей трудоемкости алгоритма
- Пусть $n=rN$;
- N обратных подстановок для решения треугольных систем размера $m \times m$ – Nr^2 операций;
- Долю матричных операций можно оценить как

$$1-\frac{Nr^2}{n^2}=1-\frac{1}{N}$$

Погрешность решения

- Известно, что решение, полученное на компьютере с машинной точностью ε_m, будет являться точным решением возмущенной системы

$$(U+\Delta U)x = b,$$

причем норма матрицы возмущения будет

$$||\Delta U|| \le n\varepsilon_m||U||+O(\varepsilon_m{}^2)$$

- Далее рассмотрим алгоритм Гаусса для систем общего вида

$$Ax=b$$

Метод Гаусса

- Основная идея: приведение матрицы A к верхнему треугольному виду с помощью эквивалентных преобразований

$$Ux=c \qquad U=\begin{pmatrix} u_{1,1} & u_{1,2} & \dots & u_{1,n} \\ 0 & u_{2,2} & \dots & u_{2,n} \\ & & \dots & \\ 0 & 0 & \dots & u_{n,n} \end{pmatrix}$$

- Эквивалентные преобразования:
  – умножение уравнения на константу;
  – прибавление к уравнению другого уравнения.

Метод Гаусса – прямой ход

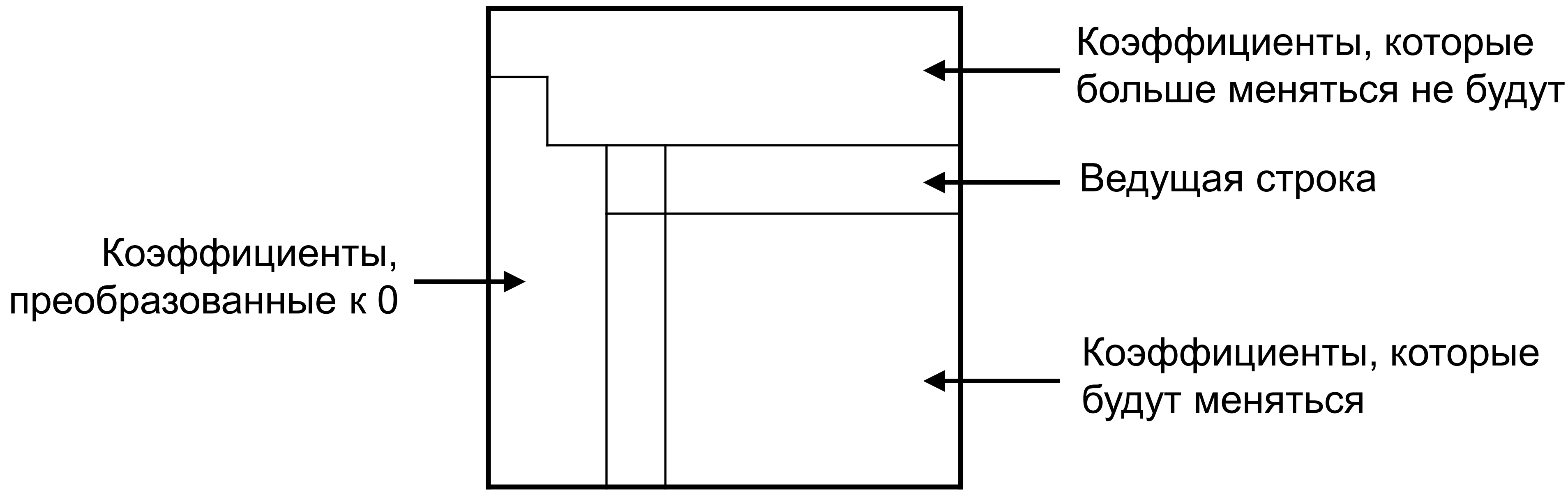
- На итерации i, $1 \le i \le n$, метода производится исключение неизвестной i для всех уравнений с номерами k, $i<k\le n$. Для этого из этих уравнений осуществляется вычитание строки i, умноженной на константу (a_{ki}/a_{ii}), чтобы результирующий коэффициент при неизвестной x_i в строках оказался нулевым.
- Все необходимые вычисления определяются при помощи соотношений:

$$\begin{aligned} a'_{kj} &= a_{kj} - \mu_{ki} \cdot a_{ij}, \\ b'_k &= b_k - \mu_{ki} \cdot b_i, \end{aligned} \qquad i \le j \le n,\ i < k \le n,\ 1 \le i < n$$

где $\mu_{ki} = a_{ki} / a_{ii}$ - множители Гаусса.

Метод Гаусса

- Общая схема состояния данных на i-ой итерации прямого хода алгоритма.



Метод Гаусса – обратный ход

После приведения матрицы коэффициентов к треугольному виду становится возможным определение значений неизвестных:

- Из последнего уравнения преобразованной системы может быть вычислено значение переменной x_n,
- Из предпоследнего уравнения становится возможным определение переменной x_{n-1}, и т.д.

В общем виде, выполняемые вычисления при обратном ходе метода Гаусса могут быть представлены при помощи соотношений:

$$x_n = c_n / u_{nn}\,, \qquad x_i = \left(c_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \Big/ u_{ii}\,, \quad i=n-1,\ldots,1$$

LU-разложение матрицы

- *LU*-разложение – представление матрицы *A* в виде

$$A=LU,$$

- где *L* – нижняя треугольная матрица с диагональными элементами, равными единице, а *U* – верхняя треугольная матрица с ненулевыми диагональными элементами.
- Известно, что *LU*-разложение существует и единственно, если главные миноры матрицы *A* отличны от нуля.
- Алгоритм *LU*-разложения тесно связан с методом исключения Гаусса

Связь LU-разложения и метода Гаусса

- Преобразования *k*-го шага метода Гаусса равносильны домножению системы слева на матрицу

$$M^{(k)} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & -\mu_{k+1,k} & 1 & & & \\ & & -\mu_{k+2,k} & & 1 & & \\ & & & & & \ddots & \\ & & -\mu_{n,k} & & & & 1 \end{bmatrix}$$

- Зная матрицы $M^{(i)}$, можно записать матрицу U и вектор c как

$$U = M^{(n-1)} M^{(n-2)} \ldots M^{(1)} A,$$

$$c = M^{(n-1)} M^{(n-2)} \ldots M^{(1)} b.$$

Связь LU-разложения и метода Гаусса

- Обозначим $L^{-1}=M^{(n-1)}M^{(n-2)}\ldots M^{(1)}$. Можно непосредственно проверить, что

$$L = \begin{bmatrix} 1 & & & & \\ \mu_{21} & 1 & & & \\ \vdots & \vdots & \ddots & & \\ \mu_{n-1,1} & \mu_{n-1,2} & \cdots & 1 & \\ \mu_{n,1} & \mu_{n,2} & \cdots & \mu_{n,n-1} & 1 \end{bmatrix}$$

- Отсюда получаем $A=LU$ (за $\frac{2}{3}n^3 + O(n^2)$ операций)
- Обратный ход – решаем две треугольные системы

$$Ly=b,\ Ux=y,$$

за $O(n^2)$ операций.

Выбор ведущего элемента

- Описанный алгоритм применим, только если ведущие элементы отличны от нуля, т.е.

$$a_{ii} \neq 0$$

- Рассмотрим k-й шаг алгоритма. Пусть

$$s = \arg\max \{ | a_{kk} |, | a_{k+1,k} |, \ldots, | a_{n,k} | \}$$

- Тогда переставим s-ю и k-ю строки матрицы (*выбор ведущего элемента по столбцу*).
- В итоге получаем систему

$$PAx = Pb,$$

где P – *матрица перестановки*.

Погрешность решения

Пусть выполнено LU-разложение матрицы на компьютере

$$A+\delta A=LU,$$

где δA – эквивалентное возмущение.

Известно, что выполняется неравенство

$$||\delta A||\le n\varepsilon_m \, ||L|| \, ||U||+O(\varepsilon_m{}^2)$$

При выполнении обратного хода получаем еще одно возмущение

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

Можно показать, что

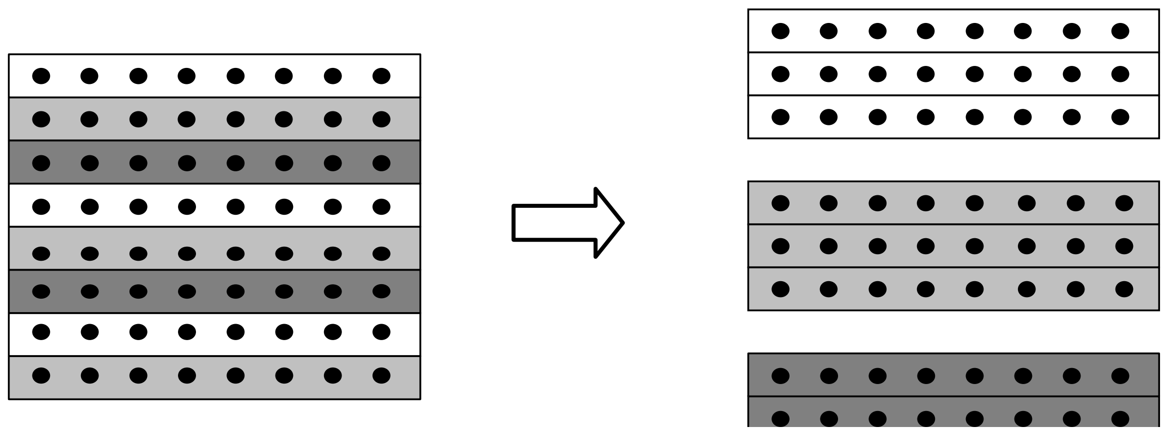
$$||\delta A+\Delta A||\le 3n\varepsilon_m \, ||L|| \, ||U||+O(\varepsilon_m{}^2)$$

Параллельный алгоритм

- Все вычисления сводятся к однотипным вычислительным операциям над строками матрицы коэффициентов системы линейных уравнений.
- 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*.

Параллельный алгоритм

- Размер матрицы больше, чем число ядер (n>>p)
- Базовые подзадачи можно укрупнить, объединив в рамках одной подзадачи несколько строк матрицы.



- Использование **циклического способа** формирования полос позволяет обеспечить лучшую балансировку вычислительной нагрузки между подзадачами

Оценка эффективности

- Время работы последовательного алгоритма

$$T_1 = \frac{2}{3} n^3 \tau$$

где τ – время выполнения одно операции.

- Время работы параллельного алгоритма

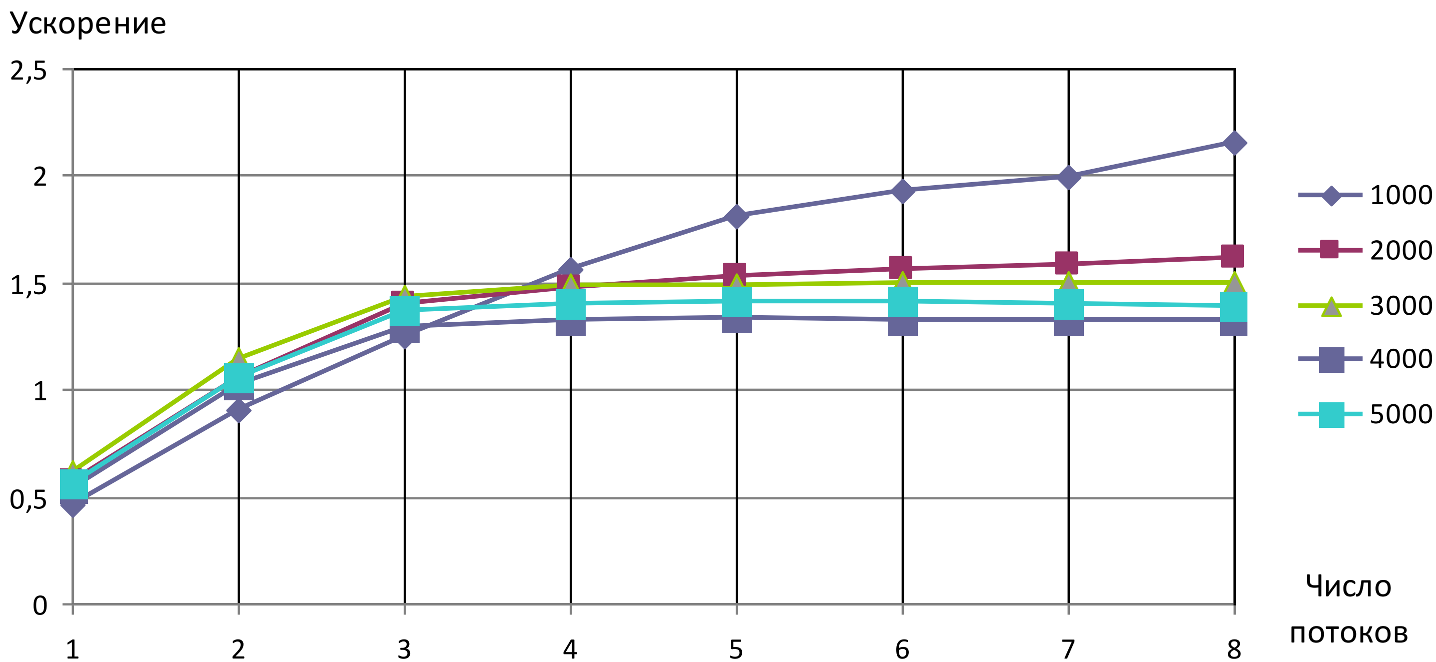
$$T_p = \frac{2n^3\tau}{3p}$$

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

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

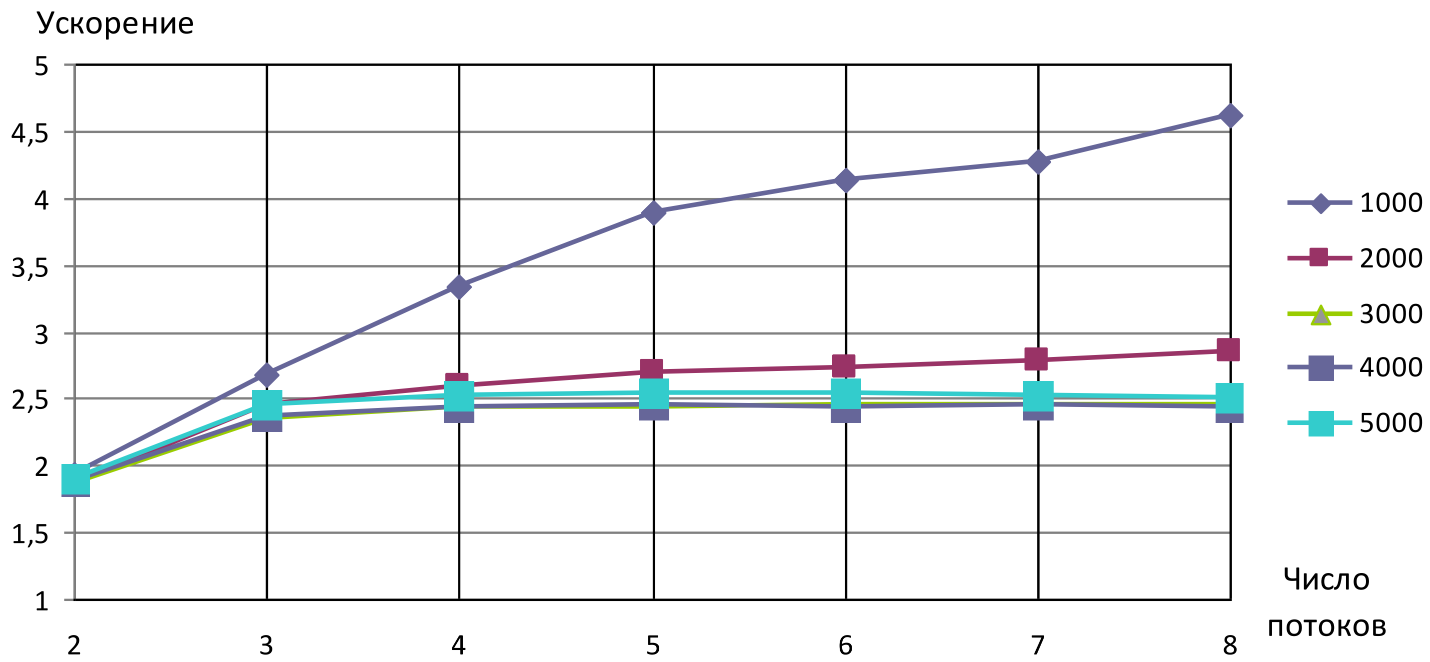
Результаты экспериментов

- Ускорение по отношению к последовательной версии



Результаты экспериментов

- Ускорение по отношению к однопоточной версии



Наблюдаем «эффект кэш-памяти»

Блочное LU-разложение

- 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 замещения исходной матрицы *A* на искомых компоненты *L* и *U* по блокам.
- Пусть r – размер блока, тогда

$$A = \begin{array}{cc} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} & \begin{array}{c} r \\ n-r \end{array} \\ \begin{array}{cc} r & n-r \end{array} & \end{array} \qquad L = \begin{array}{cc} \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} & \begin{array}{c} r \\ n-r \end{array} \\ \begin{array}{cc} r & n-r \end{array} & \end{array} \qquad U = \begin{array}{cc} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} & \begin{array}{c} r \\ n-r \end{array} \\ \begin{array}{cc} r & n-r \end{array} & \end{array}$$

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \cdot \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} = \begin{bmatrix} L_{11}U_{11} & L_{11}U_{12} \\ L_{21}U_{11} & L_{21}U_{12} + L_{22}U_{22} \end{bmatrix}$$

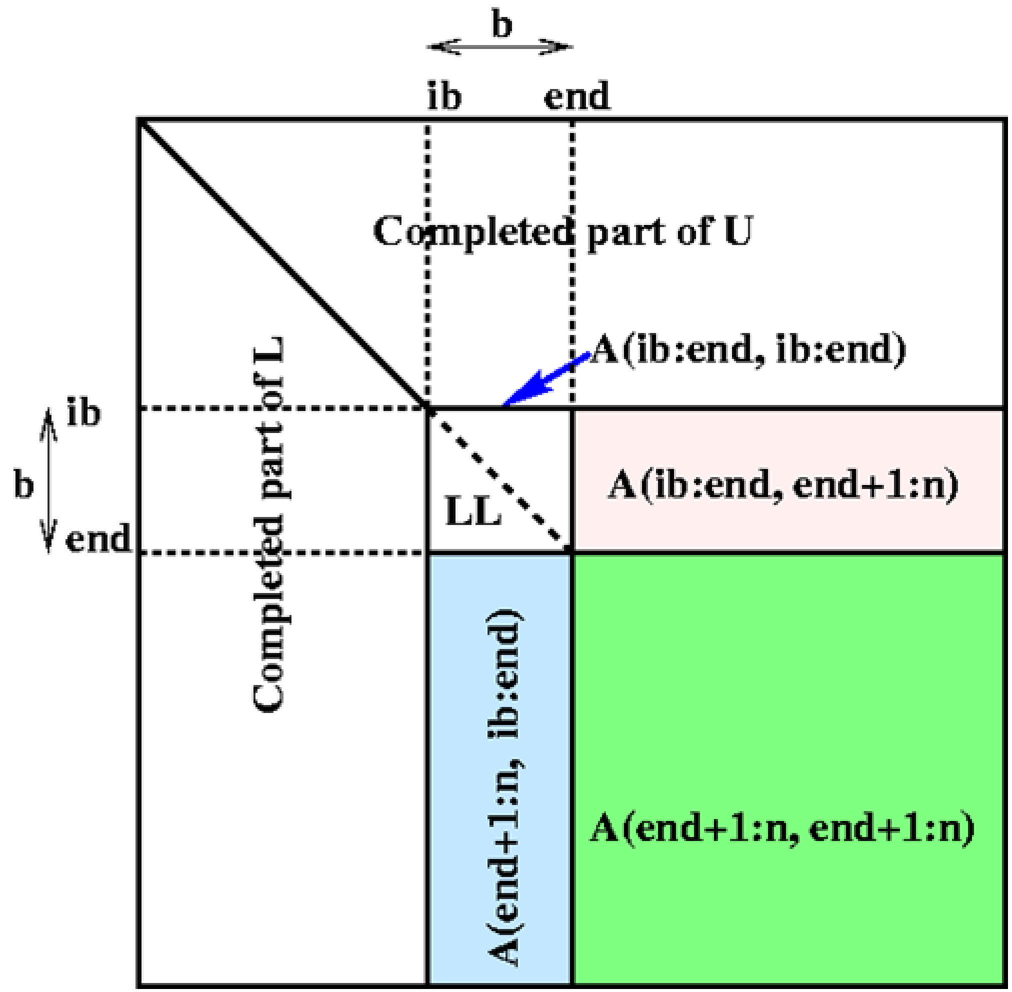
Блочный алгоритм

- Получаем $A_{11} = L_{11}U_{11}$ $L_{11}U_{12} = A_{12}$ $L_{21}U_{11} = A_{21}$
- Разложение L_{11}, U_{11} может быть получено с помощью стандартного алгоритма.
- Блоки U_{12} и L_{21} могут быть найдены решением треугольных систем с несколькими правыми частями.
- Далее вычисляем редуцированную матрицу $\tilde{A}_{22}$ как

$$\tilde{A}_{22} = A_{22} - L_{21}U_{12} = L_{22}U_{22}$$

- LU-разложение матрицы $\tilde{A}_{22}$ совпадает с искомыми блоками L_{22}, U_{22} для исходной матрицы *A*, и для его нахождения можно применить описанный алгоритм рекурсивно.

Блочный алгоритм



Applications of Parallel Computers,
UC Berkeley , Computer Science Division, www.cs.berkeley.edu/~demmel/cs267_Spr10

Оценка трудоемкости

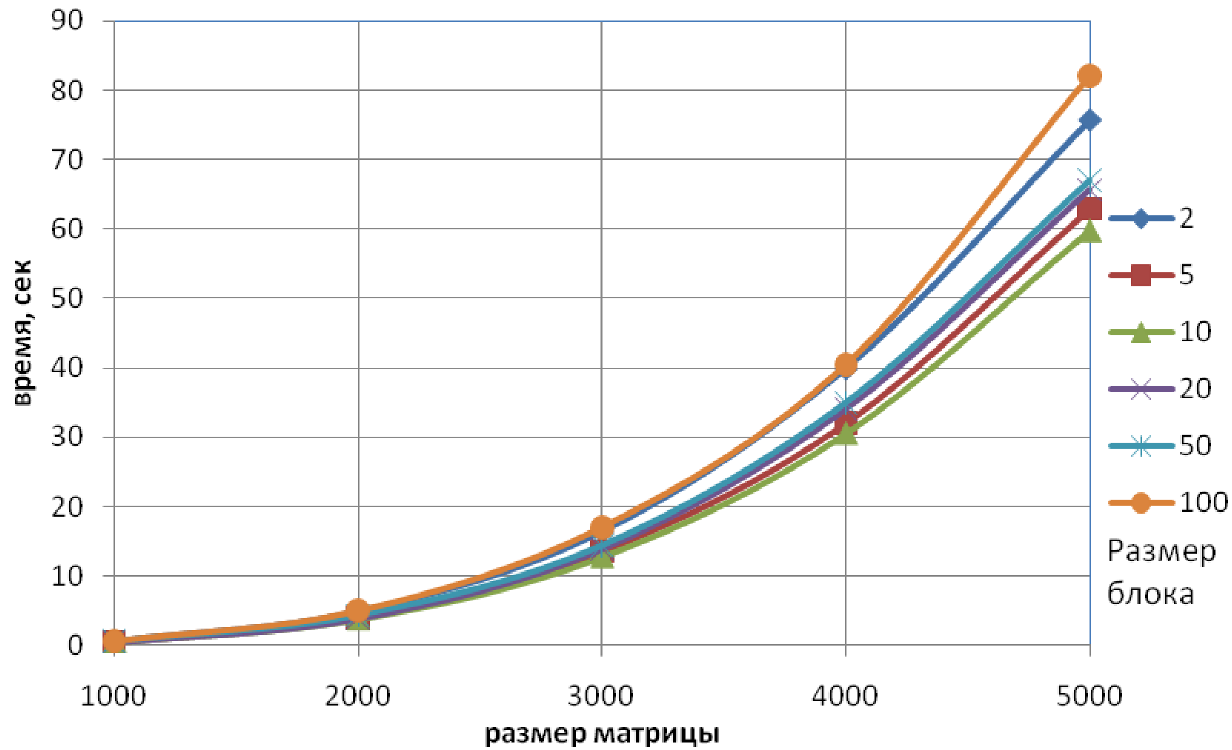
- Данная вычислительная процедура включает в себя
$$2n^3/3+O\left(n^2\right)$$
операций, как и другие возможные реализации разложения.
- Вклад матричных операций в общее число действий аппроксимируется величиной
$$1-1/N^2$$
- При правильном выборе размера блока матричные операции (которые эффективно распараллеливаются) будут составлять большую часть вычислений.

Распараллеливание блочного алгоритма

- Распараллеливание возможно для следующих вычислительных процедур:
  - вычисление блоков L_{11} U_{11}(параллельная версия стандартного алгоритма);
  - вычисление блоков L_{21} U_{12}(параллельное решение систем линейных уравнений с треугольной матрицей и разными правыми частями);
  - выполнение матричного умножения при вычислении редуцированной матрицы $\widetilde{A}_{22}$
- Эффективность параллельного алгоритма будет определяться эффективностью распараллеливания матричных операций.

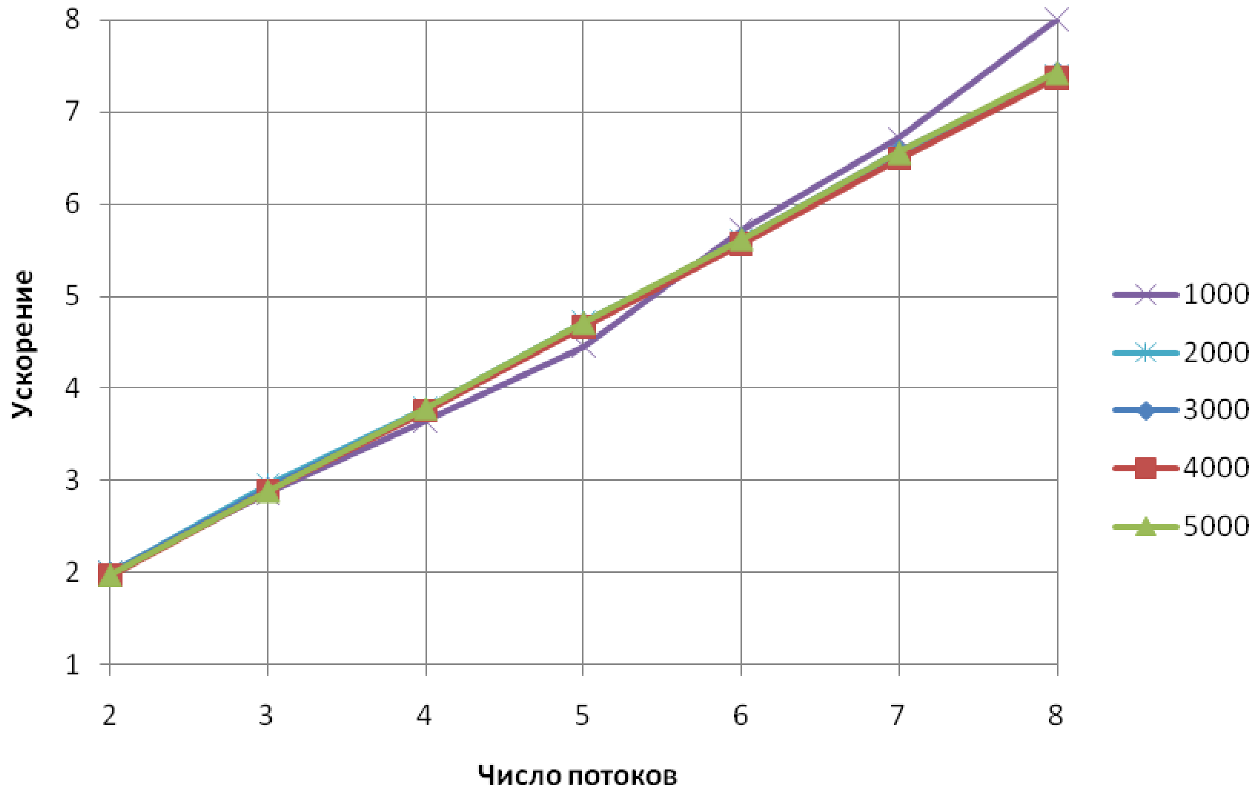
Результаты экспериментов

- Время работы при разных размерах блока



Результаты экспериментов

- Ускорение по отношению к однопоточной программе



- Линейная масштабируемость!

Литература

1. Самарский А.А., Гулин А.В. Численные методы. – М.: Наука, 1989.
2. Воеводин В.В. Вычислительные основы линейной алгебры. – М.: Наука, 1977.
3. Голуб Дж., Ван Лоун Ч. Матричные вычисления. – М.: Мир, 1999.
4. Белов С.А., Золотых Н.Ю. Численные методы линейной алгебры. – Н.Новгород, Изд-во ННГУ, 2005.

Ресурсы сети Интернет

5. Интернет-университет суперкомпьютерных технологий. [http://www.hpcu.ru].
6. 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