Курс “Методы решения СЛАУ”

ФОНД
ЦЕЛЕВОГО
КАПИТАЛА
МФТИ

Кузнецов А.А.
Петров Д.А.

Системы уравнений. Прямые методы

Кузнецов Александр
kuznetsov.aa@mipt.ru

Системы уравнений

Опр. Системой линейных алгебраических уравнений назовем матричное равенство вида:

$$A\,x = b$$

где *A* - матрица размеров *(m, n)*,
b - известный столбец размера *m*,
x - неизвестный столбец размера *n*,
m, n суть натуральные числа

Здесь и далее, если не оговорено обратное, положим:

1) ***n = m***
2) ***det(A) != 0* - существует и притом единственное решение**

Виды решателей СЛАУ

- Прямые методы
  - Методы Гаусса
  - Разложения (LU, QR, SVD и пр.)
  - Метод прогонки
- Итерационные методы
  - Простейшие итерационные
  - Минимизация функционала
  - Проекционные методы
  - Многосеточные методы*

Метод Гаусса. Прямой ход

$$\left(\begin{array}{cccc:c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array}\right) \begin{array}{l} \\ -(1)\cdot {}^{a_{21}}\!/a_{11} \\ \\ -(1)\cdot {}^{a_{n1}}\!/a_{11} \end{array} \quad \left(\begin{array}{cccc:c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ 0 & \tilde{a}_{22} & \dots & \tilde{a}_{2n} & \tilde{b}_2 \\ 0 & \dots & \dots & \dots & \dots \\ 0 & \tilde{a}_{n2} & \dots & \tilde{a}_{nn} & \tilde{b}_n \end{array}\right)$$

(N + 1) операций умножения + *(N + 1)* операций сложения для каждой строки

2 * (N + 1) * (N - 1) **операций для первого столбца**

Метод Гаусса. Прямой ход

$$
\left(\begin{array}{cccc:c}
a_{11} & a_{12} & \cdots & \cdots & \cdots \\
0 & a_{22} & \cdots & \cdots & \cdots \\
0 & 0 & \boxed{0} & a_{34} & \cdots \\
0 & 0 & a_{43} & a_{44} & \cdots
\end{array}\right)
\quad \text{swap} \quad
\left(\begin{array}{cccc:c}
a_{11} & a_{12} & \cdots & \cdots & \cdots \\
0 & a_{22} & \cdots & \cdots & \cdots \\
0 & 0 & a_{43} & a_{44} & \cdots \\
0 & 0 & 0 & a_{34} & \cdots
\end{array}\right)
$$

Если на диагонали *(i, i)* встречается нуль, то под ним обязательно есть ненулевой элемент *(j, i)*, причем *j > i*,
иначе *i* -ый столбец можно разложить по первым *i - 1*
Поменяем местами строки *i* и *j*

Метод Гаусса. Прямой ход

$$\left(\begin{array}{cccc:c} a_{11} & a_{12} & \dots & \dots & \tilde{b}_1 \\ 0 & \tilde{a}_{22} & \dots & \dots & \tilde{b}_2 \\ 0 & 0 & \dots & \dots & \dots \\ 0 & 0 & 0 & \tilde{a}_{nn} & \tilde{b}_n \end{array}\right)$$

Арифметических операция для прямого хода:

$$2(N+1)(N-1) + 2(N)(N-2)$$

$$+ \dots + 2 \cdot 3 \cdot 1 = {}^{2}\!/_{3}N^3 + O\left(N^2\right)$$

Метод Гаусса. Обратный ход

$$\left(\begin{array}{cccc:c} a_{11} & a_{12} & \dots & \dots & \tilde{b}_1 \\ 0 & \tilde{a}_{22} & \dots & \dots & \tilde{b}_2 \\ 0 & 0 & \dots & \dots & \dots \\ 0 & 0 & 0 & \tilde{a}_{nn} & \tilde{b}_n \end{array}\right) \begin{array}{l} \\ -(n)\cdot a_{2n}/a_{nn} \\ -(n)\cdot a_{n-1\,n}/a_{nn} \\ \\ \end{array} \left(\begin{array}{cccc:c} \hat{a}_{11} & \hat{a}_{12} & \dots & 0 & \hat{b}_1 \\ 0 & \hat{a}_{22} & \dots & 0 & \hat{b}_2 \\ 0 & 0 & \dots & 0 & \dots \\ 0 & 0 & 0 & \tilde{a}_{nn} & \hat{b}_n \end{array}\right)$$

$4N$ операций на последний столбец

Метод Гаусса. Обратный ход

$$\left(\begin{array}{cccc:c} 1 & 0 & 0 & 0 & \tilde{b}_1 \\ 0 & 1 & 0 & 0 & \tilde{b}_2 \\ 0 & 0 & \dots & \dots & \dots \\ 0 & 0 & 0 & 1 & \tilde{b}_n \end{array}\right)$$

Решение СЛАУ завершено

Не более $O(N^2)$ на весь обратный ход

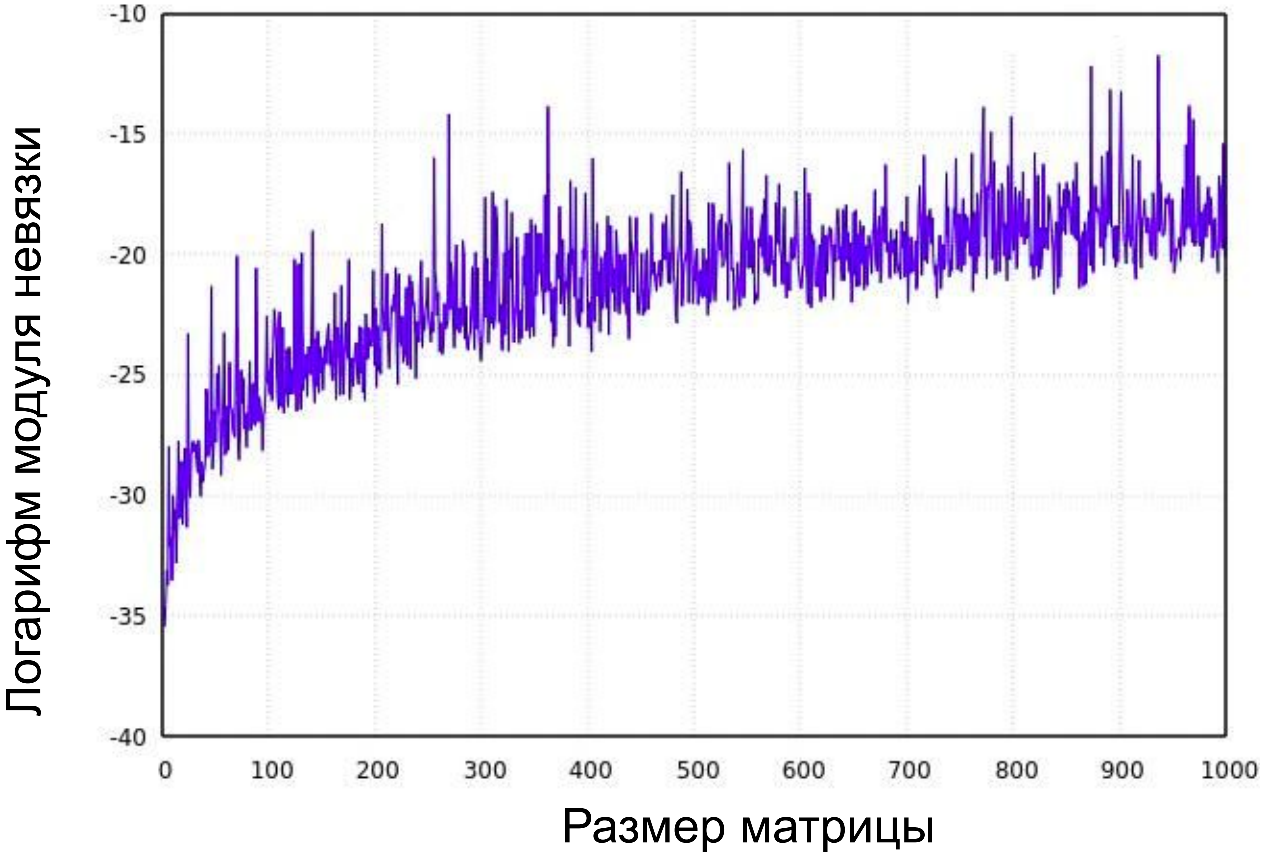
Метод Гаусса. Обсуждение

Сложность алгоритма: $^2/_3\, N^3 + O\left(N^2\right)$

Использование: СЛАУ небольших размеров *N* < *10*

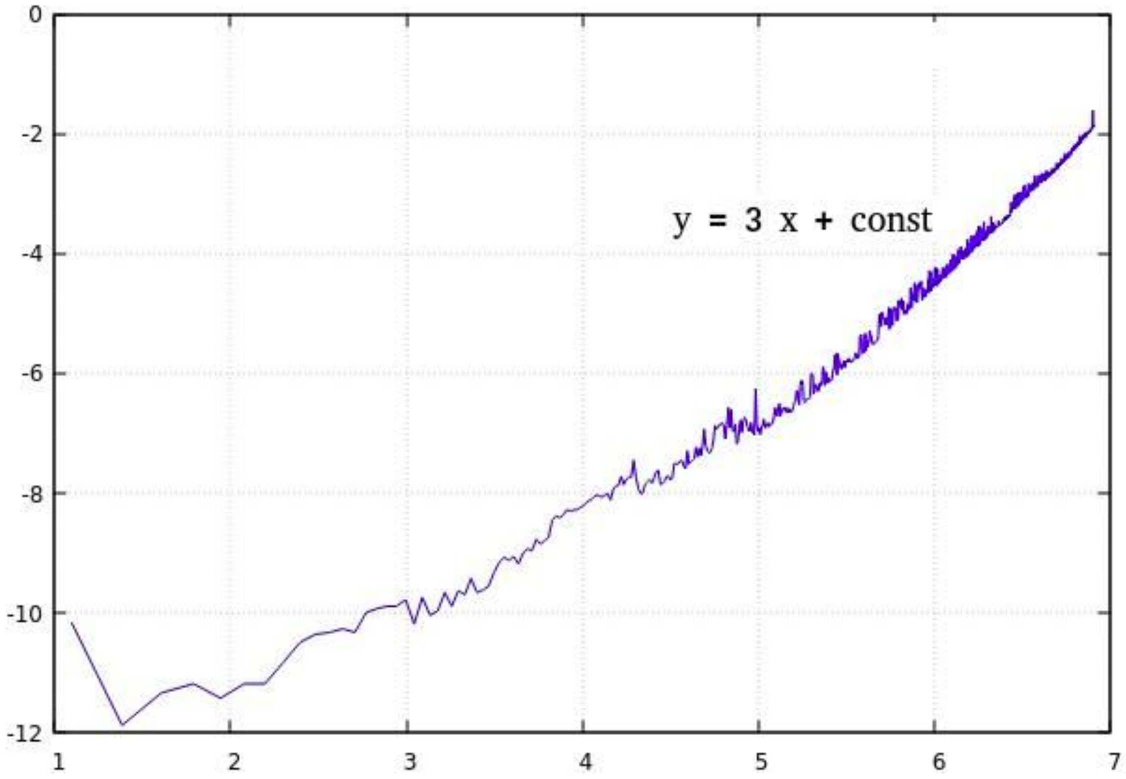
Решение точное? - Ха-Ха! Как бы не так ...

Метод Гаусса. Обсуждение



Невязка $r = b - Ax$

Метод Гаусса. Обсуждение



Логарифм времени исполнения

Логарифм размера

Сложность ~ N^3

ln(Сложность) ~ 3 ln(N)

Ошибки округления

Опр. Ошибкой округления **ε** назовем такое наибольшее положительное число, что в памяти компьютера

$$1 + \varepsilon = 1$$

32 - битное число в памяти компьютера:

s - знак

f - мантисса

e - экспонента

s	f	e
1	23	8

Ошибки округления

32 - битное число в памяти компьютера:

s	f	e
1	23	8

Формула для перевода:

$$(-1)^s \, (1 + f \cdot 2^{-23}) \, 2^{(e - 127)}$$

s - знак

f - мантисса

e - экспонента

Число 2

0 0 00000000000000000000000 1000000

Число 2^{-23}

0 0 00000000000000000000000 0110100

Ошибки округления

32 - битное число в памяти компьютера:

s	f	e
1	23	8

Число 2 + 2^{-23}

0 00000000000000000000000 1000000

+

0 00000000000000000000000 0110100

=

0 00000000000000000000001 1000000 - **достигнут предел точности!**

Ошибки округления

Замечания:

1) **ε** - относительная величина! Чило 1000 имеет ошибку 1000 **ε**

2) При складывании чисел одного порядка ошибка удваивается:

$$1 + 1 = 1 + \boldsymbol{\varepsilon} + 1 + \boldsymbol{\varepsilon} = 2 + 2\,\boldsymbol{\varepsilon}$$

3) **При складывании чисел разных порядков ошибка меняется мало**:

$$1000 + 1 = 1000 + 1000\,\boldsymbol{\varepsilon} + 1 + \boldsymbol{\varepsilon} = 1001 + 1001\,\boldsymbol{\varepsilon}$$

Ошибки округления

Пример (округление в большую сторону)**:**

$(2 + 2^{-24}) + (2 + 2^{-24})$:

0 0000000000000000000000**1** 1000000
+
0 0000000000000000000000**1** 1000000
=
0 0000000000000000000000**2** 1000000

Погрешность удвоилась!

Ошибки округления

Пример (округление в большую сторону)**:**

(1024 + 2^{-14}) + (2 + 2^{-24}):

Погрешность не изменилась!

0 0000000000000000000000**1** 1001010
+
0 0000000000000000000000**1** 1000000
=
0 0000000000000000000000**1** 1001010
+
0 0000000000000000000000**0** 1001010
=
0 0000000000000000000000**1** 1001010

Мысленно привели к одной экспоненте!

Метод Гаусса с выбором главного элемента

$$\left(\begin{array}{cccc:c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array}\right)$$

swap

Перед занулением *i*-ого столбца находим максимальный по модулю элемент в столбце *(j, i)*. Меняем строку *i* со строкой *j* местами.

Арифметические операции производятся с числами, больше различающимися по порядку величины!

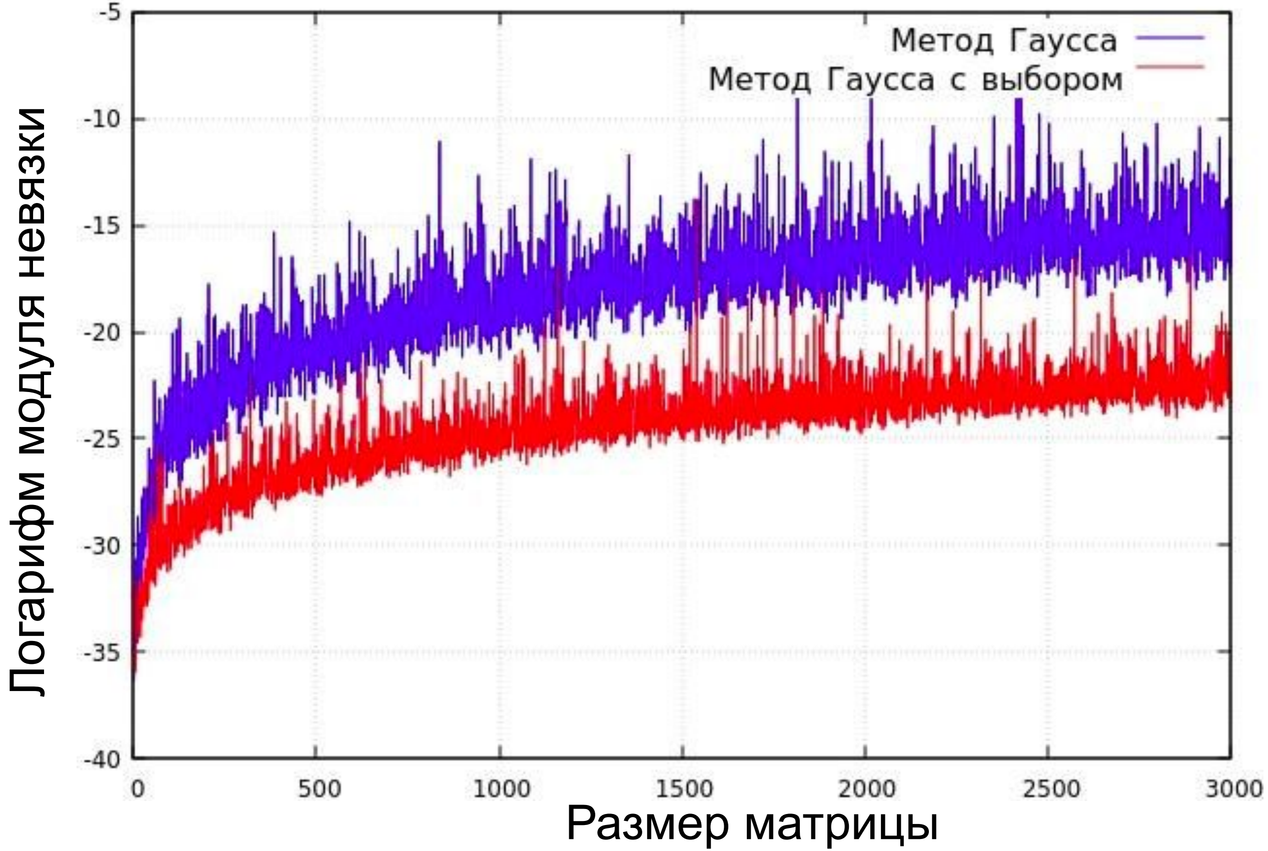
Метод Гаусса с выбором главного элемента. Обсуждение

Сложность алгоритма: $\frac{2}{3} N^3 + O\left(N^2\right)$

Использование: СЛАУ небольших размеров *N < 10*
СЛАУ с заполненными матрицами больших размеров - спорно

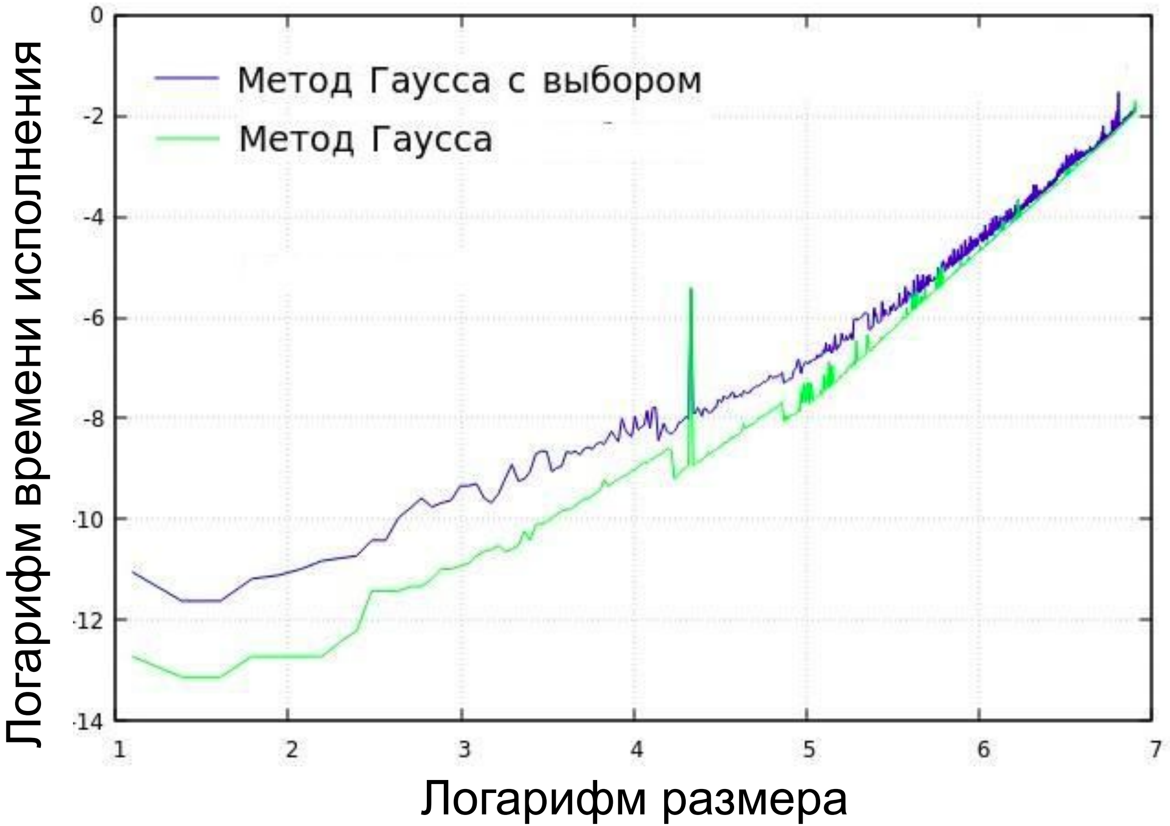
Решение точное? - на порядки точнее, чем у классического метода Гаусса

Метод Гаусса с выбором главного элемента. Обсуждение



Невязка $r = b - Ax$

Метод Гаусса с выбором главного элемента. Обсуждение



Сложность ~ N^3

ln(Сложность) ~ 3 ln(N)

LU разложение

Опр. Главным минором матрицы *A* порядка *k* <= *n* называется детерминант следующей подматрицы:

$$det\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \dots & \dots & \dots & \dots \\ a_{k1} & a_{k2} & \dots & a_{kk} \end{pmatrix}$$

$$\left(\begin{array}{cccc:c} a_{11} & a_{12} & \dots & a_{1k} & \dots \\ a_{21} & a_{22} & \dots & a_{2k} & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{k1} & a_{k2} & \dots & a_{kk} & \dots \\ \hdashline \dots & \dots & \dots & \dots & \dots \end{array}\right)$$

LU разложение

Утв. Если все главные миноры ненулевые, то операция **swap** не встречается в методе Гаусса

Док-во.

1) Элемент (1, 1) ненулевой. При работе с первым столбцом операция **swap** не происходит
2) Если до работы со столбцом i swap не производился, то i-ый главный минор не изменился (к нему прибавлялись или вычитались строки умноженные на число из него самого)

LU разложение

Док-во.

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & \cdots & \cdots \\ 0 & a_{22} & \cdots & \cdots & \cdots \\ 0 & 0 & \boxed{0} & a_{34} & \cdots \\ 0 & 0 & a_{43} & a_{44} & \cdots \end{pmatrix}$$

Элемент *(i, i)* не может быть нулевым, так как иначе *i*-ый главный минор равен нулю (разложение минора по последней строке).

Операция **swap** на i-ом шаге не выполнятся

Доказано

LU разложение

Утв. Для матрицы *A* сложение *j*-ой строки, умноженной на *d* != 0, с *i*-ой строкой можно записать в виде:

$$A_{new} = S \cdot A$$

$$S = \begin{array}{cc} \begin{array}{c} \\ \\ \\ i \\ \\ \end{array} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & d & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \\ & \begin{array}{ccccc} & j & & & \end{array} \end{array}$$

LU разложение

Аналогично можно произвести обратные преобразования строк с верхнедиагональной матрицей *U*, получаемой при прямом ходе Гаусса, и получить исходную матрицу:

$$S_1 \cdot S_2 \cdot \ \ldots \cdot U = A$$

Матрицы S_1, S_2 … при этом нижнедиагональные, а на главной диагонали имеют единицы, и при их умножении друг на друга они остаются таковыми. Их произведение обозначим *L*

LU разложение

Таким образом доказана

Теорема. Пусть у матрицы *A* все главные миноры ненулевые, тогда она представима в виде произведения матриц *L* и *U* единственным образом, где:

$$A = L \cdot U;\ L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 & 0 \\ l_{31} & \dots & 1 & 0 & 0 \\ \dots & \dots & \dots & 1 & 0 \\ l_{n1} & l_{n2} & \dots & \dots & 1 \end{pmatrix};\ U = \begin{pmatrix} u_{11} & u_{12} & \dots & \dots & u_{1n} \\ 0 & u_{22} & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & u_{n-1\,n} \\ 0 & 0 & 0 & 0 & u_{nn} \end{pmatrix}$$

LU разложение

Единственность разложения следует из явных формул для элементов L и U, которые получаются из равенства:

$$\begin{pmatrix} a_{11} & a_{22} & \dots & \dots & a_{n1} \\ a_{21} & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & \dots & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 & 0 \\ l_{31} & \dots & 1 & 0 & 0 \\ \dots & \dots & \dots & 1 & 0 \\ l_{n1} & l_{n2} & \dots & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} u_{11} & u_{12} & \dots & \dots & u_{1n} \\ 0 & u_{22} & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & u_{n-1\,n} \\ 0 & 0 & 0 & 0 & u_{nn} \end{pmatrix}$$

$$u_{ij} = a_{ij} - \left(\sum_{k=1}^{j-1} l_{ik} u_{kj} \right),\ i \leqslant j;$$

$$l_{ij} = \frac{1}{u_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \right),\ i > j;$$

LU разложение

Теперь рассмотрим СЛАУ:

$$A\,x = b$$

Выполним для матрицы *A LU* разложение:

$$L(\,U\,x) \;= b$$

Заменим задачу эквивалентной:

$$L\,z = b;\;\; Ux = z$$

Обе системы решаются обратным ходом Гаусса за $O(N^2)$

LU разложение. Обсуждение

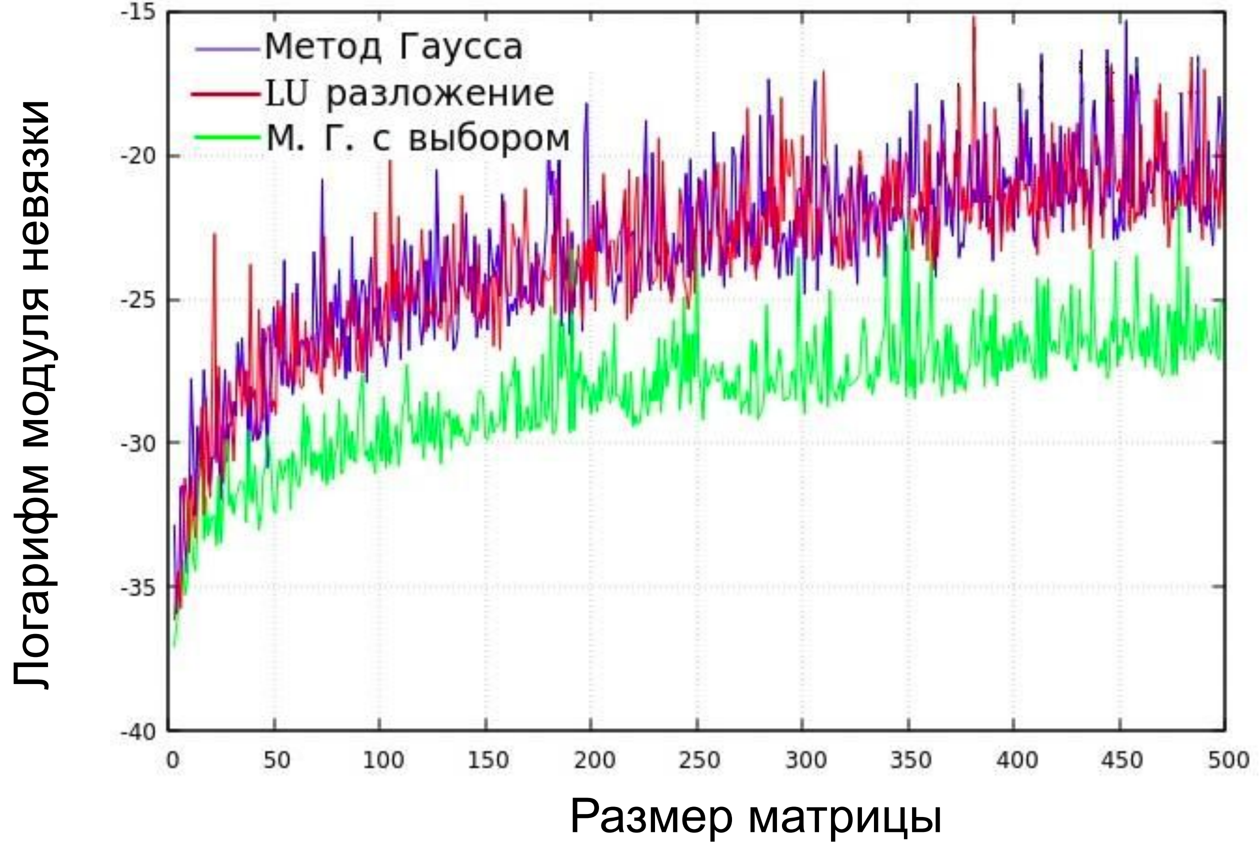
Сложность алгоритма: $^2/_3\, N^3 + O\left(N^2\right)$

Использование: Аналогично методу Гаусса, **предобуславливание**

Решение точное? - точность совпадает с обычным методом Гаусса

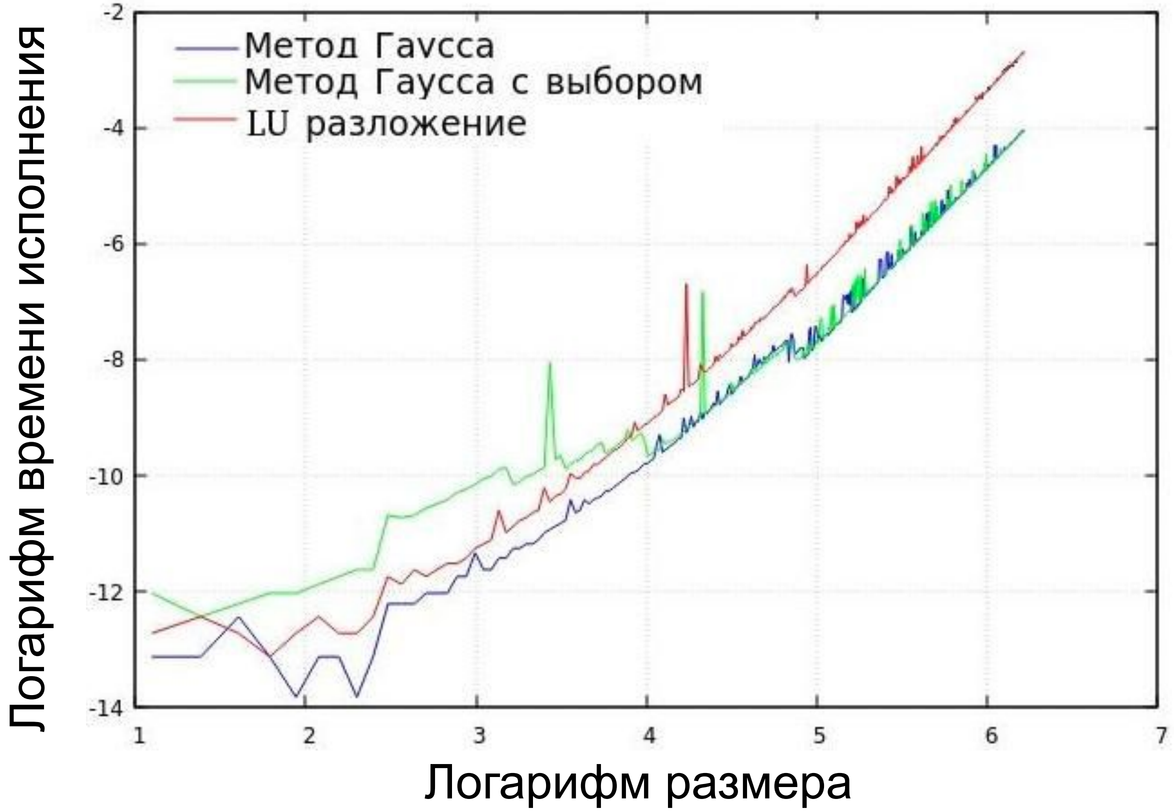
В отличие от метода Гаусса практически обращает матрицу - **решение для новой правой части можно получить за O(N^2)** при готовом разложении.

LU разложение. Обсуждение



Невязка $r = b - Ax$

Метод Гаусса с выбором главного элемента. Обсуждение



Сложность ~ N^3

ln(Сложность) ~ 3 ln(N)

Разложение Холецкого

Пусть матрица системы A имеет положительные главные миноры и является симметричной. Тогда:

$$A = L \cdot U = A^T = (L \cdot U)^T = U^T \cdot L^T$$

Из единственности *LU* разложения следует, что:

$$U = L^T$$

А значит, справедливо разложение Холецкого:

$$A = L \cdot L^T$$

Разложение Холецкого

Из равенства

$$A = L \cdot L^T$$

можно выписать явные формулы для элементов l_{ij} матрицы L

$$l_{11} = \sqrt{a_{11}};$$

$$l_{i1} = a_{i1} / l_{11}, \; i = 2, \ldots, n$$

$$l_{22} = \sqrt{a_{22} - l_{21}^{\;2}}$$

$$l_{i2} = \left(a_{i2} - l_{i1} l_{21} \right) / l_{22}, \; i = 3, \ldots, n$$

$$l_{kk} = \sqrt{a_{kk} - l_{k1}^{\;2} - l_{k2}^{\;2} - \ldots - l_{kk-1}^{\;2}}$$

$$l_{ik} = \left(a_{ik} - l_{ik} l_{k1} - l_{i2} l_{k2} - \ldots - l_{i\,k-1} l_{k\,k-1} \right) / l_{kk}, \; i = k+1, \ldots, n$$

Разложение Холецкого. Обсуждение

Сложность алгоритма: $1/_3\, N^3 + O\left(N^2\right)$

Использование: Для плотных симметричных матриц, **предобуславливание**

Решение точное? - точность совпадает с обычным методом Гаусса

В отличие от метода Гаусса практически обращает матрицу - **решение для новой правой части можно получить за O(N^2)** при готовом разложении.

QR разложение

Опр. Матрицей перехода от старого базиса $\{\mathbf{e}_i\}^N_{i=1}$ к новому базису $\{\mathbf{g}_i\}^N_{i=1}$ называется матрица, в столбцах которой стоят координаты векторов нового базиса в старом базисе

$$\begin{pmatrix} g_1 & g_2 & \cdots & g_n \end{pmatrix} = \begin{pmatrix} e_1 & e_2 & \cdots & e_n \end{pmatrix} \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ s_{n1} & \cdots & \cdots & s_{nn} \end{bmatrix}$$

QR разложение

Алгоритм ортогонализации (Грам - Шмидт)

При помощи алгоритма:

$$\boldsymbol{h}_i = \boldsymbol{e}_i - \sum_{j=1}^{i-1} \frac{\left(\boldsymbol{e}_i \cdot \boldsymbol{h}_j\right)}{\left(\boldsymbol{h}_j \cdot \boldsymbol{h}_j\right)} \boldsymbol{h}_j,\ i = 1,\ \ldots,\ n$$

можно получить ортогональный базис $\{\boldsymbol{h}_i\}^{N}_{i=1}$

QR разложение

$$\boldsymbol{h}_i = \boldsymbol{e}_i - \sum_{j=1}^{i-1} \frac{\left(\boldsymbol{e}_i \cdot \boldsymbol{h}_j\right)}{\left(\boldsymbol{h}_j \cdot \boldsymbol{h}_j\right)} \boldsymbol{h}_j,\ i = 1,\ \ldots,\ n$$

Вектор $\boldsymbol{h}_i$ является линейной комбинацией векторов $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_i\}$. Поэтому в *i*-ом столбце матрицы перехода *(i+1)*-ый и далее элементы нулевые.

Матрица перехода *S* от $\{e_i\}^N_{i=1}$ к $\{h_i\}^N_{i=1}$ *верхнетреугольная.*

$$S = \begin{pmatrix} 1 & \ldots & \ldots & \ldots \\ 0 & 1 & \ldots & \ldots \\ 0 & 0 & 1 & \ldots \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

QR разложение

Для получения ортонормированного базиса каждый вектор $\{h_i\}^N_{i=1}$ стоит разделить на свою норму (длину).

Полученный ОНБ $\{g_i\}^N_{i=1}$ связан с $\{h_i\}^N_{i=1}$ при помощи диагональной матрицы.

Матрица перехода от $\{e\}^N_{i=1}$ к ОНБ $\{\mathbf{g}_i\}^N_{i=1}$ - верхнетреугольная

QR разложение

Опр. Матрица Q называется ортогональной, если:

$$Q^T = Q^{-1}$$

Утв. Ортогональные матрицы и только они являются матрицами перехода между ОНБ

QR разложение

Теорема. Любую невырожденную матрицу *A* можно представить в виде произведения ортогональной *Q* и верхнетреугольной *R*

Док-во.

Пусть столбцы A - координатные столбцы системы базиса $\{w\}^N_{i=1}$ в ОНБ $\{e\}^N_{i=1}$:

$$\begin{pmatrix} w_1 & w_2 & \dots & w_n \end{pmatrix} = \begin{pmatrix} e_1 & e_2 & \dots & e_n \end{pmatrix} \cdot A$$

Теперь ортогонализуем систему векторов $\{w\}^N_{i=1}$, *R* - верхнетреугольная, $\{g\}^N_{i=1}$ - ОНБ:

$$\begin{pmatrix} w_1 & w_2 & \dots & w_n \end{pmatrix} = \begin{pmatrix} g_1 & g_2 & \dots & g_n \end{pmatrix} \cdot R$$

QR разложение

Теорема. Любую невырожденную матрицу *A* можно представить в виде произведения ортогональной *Q* и верхнетреугольной *R*

$\{g\}_{i=1}^{N}$ и $\{e\}_{i=1}^{N}$ связаны ортогональной матрицей Q

$$\begin{pmatrix} e_1 & e_2 & \dots & e_n \end{pmatrix} \cdot Q \cdot R = \begin{pmatrix} e_1 & e_2 & \dots & e_n \end{pmatrix} \cdot A$$

Откуда $A = Q\,R$

Доказано

QR разложение

Для решения СЛАУ нужно при помощи обратного хода Гаусса разрешить:

$$R \cdot x = Q^T b$$

QR разложение. Обсуждение

Сложность алгоритма: $\frac{5}{2} N^3 + O(N^2)$ Ортогонализация

$O(N^3)$ Разложение векторов по базису

Использование: НИКОГДА!!!

Точность: Алгоритм ортогонализации дает неточную ортогональную матрицу из-за ошибок округления

Решение - другой алгоритм!

Алгоритм Хаусхолдера (QR разложение)

Оператор отражения, относительно плоскости с нормалью ***v***:

$$P = I - \frac{2}{v^T \cdot v} v \cdot v^T$$

$$P \cdot x = I \cdot x - \frac{2}{v^T \cdot v} v \cdot v^T \cdot x = x - 2\frac{(v \cdot x)}{(v \cdot v)} v$$

P - симметричный и сам себе обратный (по смыслу), поэтому ортогональный.

Алгоритм Хаусхолдера (QR разложение)

Пусть $\boldsymbol{e_1}$, - первый базисный вектор. Для ненулевого x найдем такой оператор P, что:

$$P \cdot x = \alpha e_1$$

Решение*:*

$$P = I - \frac{2}{v^T \cdot v} v \cdot v^T$$

$$v = x \pm |x| \, e_1$$

Для вычислительной устойчивости берут знак, совпадающий с x_1 (первая координата) - чтобы в случае коллинеарности не занулить v

Алгоритм Хаусхолдера (QR разложение)

Пусть матрица A состоит из столбцов {$\boldsymbol{a}_1$, $\boldsymbol{a}_2$, …, $\boldsymbol{a}_n$}. Найдем оператор P_{a1}. Он действует на матрицу *A* следующим образом:

$$P_{a_1} \cdot A = \begin{pmatrix} r_{11} & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots \end{pmatrix}$$

Алгоритм Хаусхолдера (QR разложение)

Теперь рассмотрим оператор P_{a2}. Первую координату векторов он не изменяет, а на остальные действует подобно предыдущему оператору.

$$P_{a_2} = \begin{bmatrix} 1 & 0 \\ 0 & P_{n-1} \end{bmatrix}$$

P_{n-1} действует в *(n-1)*-мерном пространстве таким образом, что отрезок столбца $\boldsymbol{a_2}$ в *(n-1)*-мерном пространстве (без первой координаты) преобразуется в вектор, коллинеарный $\boldsymbol{e_2}$.
($\boldsymbol{e_2}$ = *(1, 0,...,0)* в *(n-1)*-мерном пространстве)

Алгоритм Хаусхолдера (QR разложение)

Таким образом:

$$P_{a_2} \cdot P_{a_1} \cdot A = \begin{pmatrix} r_{11} & \dots & \dots & \dots \\ 0 & r_{22} & \dots & \dots \\ 0 & 0 & \dots & \dots \\ 0 & 0 & \dots & \dots \end{pmatrix}$$

После *n-1* таких операций, получаем:

$$P_{a_{n-1}} \cdot \ \dots \ \cdot P_{a_1} \cdot A = \begin{pmatrix} r_{11} & \dots & \dots & \dots \\ 0 & r_{22} & \dots & \dots \\ 0 & 0 & \dots & \dots \\ 0 & 0 & 0 & r_{nn} \end{pmatrix} = R$$

Алгоритм Хаусхолдера (QR разложение)

Операторы P сами себе обратны. Это значит, что:

$$A = P_{a_1} \cdot \ \dots \cdot \ P_{a_{n-1}} \cdot R = Q \cdot R$$

Замечания:

1) При реализации алгоритма нужно хранить не матрицы *P*, а соответствующие векторы ***v*** - искомую матрицу Q можно найти по ним
2) При восстановлении нужно не перемножать матрицы P вслепую, а использовать их специфику.

Алгоритм Хаусхолдера. Обсуждение

Сложность алгоритма: $\frac{4}{3} N^3 + O\left(N^2\right)$

Использование: Для решения СЛАУ - НИКОГДА!
МНК для переопределенных систем.
Построение ортогональных матриц

Точность: Лучше, чем в оригинальном QR разложении (меньшее количество арифметических операций)

Алгоритм Гивенса (QR разложение)

Сложность алгоритма: $2\,N^3 + O\left(N^2\right)$

Использование: Для решения СЛАУ - НИКОГДА!
МНК для переопределенных систем.
Построение ортогональных матриц

Точность: Лучше, чем в оригинальном QR разложении (меньшее количество арифметических операций)

Замечание: Применяется в солвере GMRES как вспомогательный алгоритм. Будет рассмотрен далее

Метод трехдиагональной прогонки

Рассмотрим СЛАУ со специфической (трехдиагональной) матрицей:

$$\begin{pmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & 0 & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & 0 & 0 & 0 & a_n & b_n \end{pmatrix}$$

Метод трехдиагональной прогонки

Рассмотрим СЛАУ со специфической (трехдиагональной) матрицей:

$$\begin{cases} b_1 x_1 + c_1 x_2 = d_1 \\ a_2 x_1 + b_2 x_2 + c_2 x_3 = d_2 \\ \ldots \\ a_{n-1} x_{n-2} + b_{n-1} x_{n-1} + c_{n-1} x_n = d_{n-1} \\ a_n x_{n-1} + b_n x_n = d_n \end{cases}$$

Метод трехдиагональной прогонки

x_i переменную можно вычислить через x_{i+1} , $i = 1, \ldots, n-1$

$$x_i = \gamma_i \, x_{i+1} + \beta_i$$

$$\gamma_1 = {-c_1}/{b_1} ; \; \beta_1 = {d_1}/{b_1}$$

$$\gamma_i = \frac{-c_i}{a_i \gamma_{i-1} + b_i} ; \; \beta_i = \frac{d_i - a_i \beta_{i-1}}{a_i \gamma_{i-1} + b_i}$$

Метод трехдиагональной прогонки

Для последней переменной:

$$a_n \cdot \left(\gamma_{n-1} x_n + \beta_{n-1} \right) + b_n \cdot x_n = d_n$$

Вычисляем x_n . Далее находим переменные в обратном порядке через рекурсивные соотношения

Достаточное условие устойчивости - **диагональное преобладание**

$$|b_i| \geq |a_i| + |c_i|, \; i = 1, \ldots, n$$

для одной строки - неравенство обязано быть строгим!

Метод трехдиагональной прогонки. Обсуждение

Сложность алгоритма: $O(N)$

Использование: Трехдиагональные СЛАУ - ВСЕГДА !!!

Точность: Превосходно

Замечания:
1) Если матрица близка к трехдиагональной, можно провести элементарные преобразования
2) Аналогичные алгоритмы есть для 5 -, 7 -, … - диагональной прогонки