

# МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

УДК 519.612+519.613:621.391.1

И.В. Семушин

## ЕЩЕ РАЗ О БОЛЬШИХ ОБРАТНЫХ МАТРИЦАХ: ОТ ФОРМАЛИЗМОВ К РЕАЛИЗАЦИИ



*Семушин Иннокентий Васильевич, доктор технических наук, окончил Ленинградский электротехнический институт (ЛЭТИ) им. В.И. Ульянова (Ленина), профессор кафедры «Информационные технологии» Ульяновского государственного университета. Имеет монографии, статьи, учебные пособия и патенты на изобретения. Область научных интересов: фильтрация и управление в условиях неопределенности. [e-mail: kentvsem@yandex.ru].*

### Аннотация

В практических алгоритмах следует избегать, по возможности, вычисления обратных матриц, даже если формальное решение этих задач выражено с использованием обозначений некоторых обратных матриц. Это делают, заменяя отыскание обратных матриц решением соответствующих систем линейных уравнений, и тем самым снижают вычислительную нагрузку на алгоритм. Однако встречаются проекты, где эта рекомендация оказывается не учтена.

Данная статья более подробно освещает вопрос вычислительной сложности различных алгоритмов: как содержащих обращение данной матрицы, так и избегающих его. Руководители проектов и разработчики алгоритмов могут воспользоваться представленными здесь результатами для принятия эффективных по сложности проектных решений, формально включающих большие обратные матрицы.

Ключевые слова: LU-разложение, компактная схема Краута, метод окаймления, исключение Жордана, элиминативная форма обратной матрицы, метод наименьших квадратов, алгоритм Поттера.

### Abstract

Whenever appropriate, computation of inverse matrices in practical algorithms should be avoided even if the formal problem solution is expressed using inverse matrix notation. One can achieve this by correspondent replacement of matrix inversion by solving a linear system of equations. By that, processor computation load may be reduced. However, there take place projects where this recommendation is left out of consideration.

The paper digs deeper into a problem of computational complexity of different algorithms: both including and avoiding the inversion of a given matrix. Project managers and algorithm designers can take advantage of the results presented here to set forth and develop low-complexity designs involving, as a mere formality, big inverse matrices.

Key words: LU-decomposition, Crout compact scheme, bordering method, Jordan elimination, elimination form of the inverse matrix, least squares method, Potter algorithm.

**ВВЕДЕНИЕ**

В различных разделах математических информационных технологий (МИТ) есть задачи, формальное решение которых выражено в терминах обратных матриц.

**Пример МИТ-1:** безусловная оптимизация целевой функции  $f(x)$  относительно вектора переменных  $x \triangleq [x_1, \dots, x_n]^T$  часто базируется на квадратической модели, получаемой из усеченного ряда Тейлора функции  $f(x)$  около очередной ( $k$ -й) итерационной точки  $x^{(k)}$ ,  $k = 1, 2, \dots$ :

$$f(x^{(k)} + \delta) \approx q^{(k)}(\delta) = f^{(k)} + g^{(k)T} \delta + \frac{1}{2} \delta^T G^{(k)} \delta,$$

где  $\delta \triangleq x - x^{(k)}$  и  $q^{(k)}(\delta)$  есть квадратическая аппроксимация функции для итерации  $k$ , в которой  $g(x) \triangleq \nabla f(x)$  есть вектор градиента,  $\nabla$  – оператор первых производных (его элементы суть  $\partial/\partial x_i$  для  $i = 1, \dots, n$ ) и  $G(x) \triangleq \nabla^2 f(x)$  – матрица Гессе ( $n \times n$  матрица вторых производных  $\partial^2 f / (\partial x_i \partial x_j)$ ). Тогда итерация  $x^{(k+1)}$  в методе Ньютона берется как  $x^{(k)} + \delta^{(k)}$ , где поправка  $\delta^{(k)}$  есть значение аргумента, доставляющее минимум функции  $q^{(k)}(\delta)$ . Метод требует, чтобы нулевая, первые и вторые производные функции  $f(x)$  были доступны в любой точке. Это означает, что числовые значения коэффициентов  $f^{(k)} \triangleq f(x^{(k)})$ , векторов  $g^{(k)} \triangleq g(x^{(k)})$  и матриц  $G^{(k)} \triangleq G(x^{(k)})$  в представлении аппроксимирующей функции  $q^{(k)}(\delta)$  известны. Кроме того, поскольку  $q^{(k)}(\delta)$  имеет единственный минимум, когда матрица Гессе  $G^{(k)}$  положительно определена, то необходимое условие для точки минимума  $\delta^{(k)}$ , т. е. уравнение  $\nabla q^{(k)}(\delta^{(k)}) = 0$ , порождает простой алгоритм этого метода [1, р. 44]:

- (а) решить  $G^{(k)} \delta = -g^{(k)}$  относительно  $\delta = \delta^{(k)}$ ,
- (б) положить  $x^{(k+1)} = x^{(k)} + \delta^{(k)}$ .

Этот алгоритм явно указывает, что популярную запись метода Ньютона в виде формулы

$$x^{(k+1)} = x^{(k)} - [G^{(k)}]^{-1} g^{(k)}$$

нельзя воспринимать как буквальное руководство к реализации: сначала, якобы, вычисление обратной матрицы  $[G^{(k)}]^{-1}$  и затем умножение ее на  $g^{(k)}$ , чтобы таким образом переходить от  $x^{(k)}$  к  $x^{(k+1)}$ .

**Пример МИТ-2:** проект прекодинга концевых перекрестных помех в каналах VDSL – Very High-Speed Digital Subscriber Line. Он основан [2], [3] на декомпозиции матрицы  $H_k = [h_k^{i,j}]$ ,  $i, j = 1, \dots, N$ , передачи канала на  $k$ -й несущей частоте  $f$  от  $i$ -го передатчика к  $j$ -му получателю в произведение двух матриц:

$F_k^{-1} \times A_k$ , где  $F_k^{-1} \triangleq \text{diag} \left[ (h_k^{1,1})^{-1} | \dots | (h_k^{N,N})^{-1} \right]$  есть диагональная матрица прямых коэффициентов передачи,

$F_k \triangleq \text{diag} \left[ (h_k^{1,1})^{-1} | \dots | (h_k^{N,N})^{-1} \right]$  есть инверсия этой

матрицы и  $A_k \triangleq F_k H_k$ . В базовом проекте некоторых сетевых операторов матрицы прекодера  $P_k$  вычисляются напрямую в виде  $P_k \triangleq A_k^{-1}$  [4, р. 34–36], что создает излишнюю вычислительную нагрузку на модемы устройств ONU (Optical Network Unit), от которых исходит многожильный кабель витых пар для разводки по модемам в помещения пользователей – абонентов оператора.

**Пример МИТ-3:** ковариационные алгоритмы фильтрации Калмана, в том числе для линейных задач метода наименьших квадратов (МНК). Имеем известные представления [5, с. 238–248]:

$$z = Ax + v,$$

$z$  есть  $m$ -вектор наблюдений (экспериментальных данных) и  $A$  есть  $m \times n$ -матрица, которую задают до того, как получают  $z$ , и которую в различных приложениях называют либо матрицей регрессоров, либо матрицей наблюдений, либо матрицей плана эксперимента. Критерий МНК:

$$J(x) = (z - Ax)^T (z - Ax) = \sum_{j=1}^m v(j)^2 = v^T v.$$

Требую минимум этого критерия, для искомого МНК-решения  $\bar{x}$  получают так называемые *нормальные уравнения*:

$$A^T A \bar{x} = A^T z.$$

Последовательное решение этих уравнений обеспечивает рекурсия МНК. В стандартной ковариационной форме она имеет вид алгоритма Калмана:

I. *Инициализация.* Начальные значения: МНК-оценка  $x_0$  и ее ковариация  $P_0$ :

$$\tilde{x} := x_0, \quad \tilde{P} := P_0. \tag{1}$$

Если априорные данные отсутствуют, это означает,  $x_0 = 0$  и  $P_0 = \varepsilon^{-1} \text{diag} [1 | \dots | 1]$  при возможно малом  $\varepsilon$ , теоретически  $\varepsilon \rightarrow 0$ .

II. *Обработка порции наблюдений.* Очередная порция наблюдений  $z = Ax + v$ :

$$\left. \begin{aligned} K &= \tilde{P} A^T (A \tilde{P} A^T + I)^{-1}, \\ \hat{P} &= \tilde{P} - K A \tilde{P}, \\ \hat{x} &= \tilde{x} + K (z - A \tilde{x}). \end{aligned} \right\} \tag{2}$$

III. *Экстраполяция между порциями.* Распространение оценки  $\hat{x}$  и ее ковариации  $\hat{P}$  между наблюдениями к моменту повторения этапа II со следующей порцией наблюдений (вплоть до их исчерпания):

$$\tilde{P} := \hat{P}; \quad \tilde{x} := \hat{x}. \tag{3}$$

В первом выражении блока (2) имеем обращение матрицы. Устранение этой нежелательной операции обеспечивают переходом к скаляризованным наблюдениям (с их предварительной декорреляцией, если существует такая необходимость). Для предотвраще-

ния возможной потери положительной определенности матрицы  $\hat{P}$  после вычитания во втором выражении блока (2) применяют различные численно устойчивые модификации этого выражения. Первая реализация этого типа датируется 1963 годом, и она носит имя *квадратно-корневой алгоритм Поттера* [5, с. 250].

Приведенные выше примеры ясно демонстрируют различные практики и необходимость избегать операции обращения матриц в вычислительных алгоритмах. Особенно это важно в случае матриц больших размеров, который типичен для приведенного Примера МИТ-2. Комплекснозначные матрицы для этого приложения достигают размера нескольких сотен для каждой из нескольких тысяч несущих частот (подканалов в выделенном для VDSL частотном диапазоне). Данная статья мотивирована, в первую очередь, именно этим приложением – цифровыми инфокоммуникационными системами, поскольку, как утверждают сотрудники компании Huawei [4, р. 39, 55–70], «тема эффективного обращения как больших, так и малых матриц извечна, поскольку возникает в большом разнообразии решаемых задач». Раскрывая эту тему, статья дает точные сравнительные характеристики сложности различных вычислительных стратегий. Мерой сложности служит необходимое количество операций умножения и деления чисел (в общем случае – комплексных) для решения следующей задачи.

Даны: вектор  $x \in \mathbb{C}^N$  и матрица  $A \in \mathbb{C}^{N \times N}$ .

Требуется: вычислить  $x'$ , удовлетворяющий равенству  $x' = A^{-1}x$ .

Обозначения:

- Строчные латинские буквы обозначают  $N$ -векторы (одной и той же размерности  $N$ ).

- Соответствующие  $N \times N$ -матрицы преобразований обозначены заглавными латинскими буквами.

- Множества комплекснозначных векторов и матриц обозначены  $\mathbb{C}^N$  и  $\mathbb{C}^{N \times N}$ , соответственно.

- Возможные стратегии решения обозначены символами  $\Sigma_n$ . Здесь они пронумерованы от  $n = 1$  до  $n = 5$ .

Одновременно,  $\Sigma_n$  обозначает суммарное количество операций умножения и деления чисел для стратегии  $\Sigma_n$ , и  $n$  совпадает с номером раздела данной статьи.

Таким образом, в основной своей части статья содержит пять разделов.

Раздел 1 характеризует типовое (базовое) решение, часто применяемое в проектах: (а) гауссово  $LU$ -разложение матрицы  $A$  с тем, что матрицы  $L$  и  $U$  замещают  $A$ ; (б) на этой основе вычисление  $P = A^{-1}$ ; (в) умножение  $x' = Px$ . При этом для пункта (а) проанализированы: стандартная схема разложения, компактная схема Краута и схема окаймления.

Раздел 2 характеризует первое альтернативное решение: (а) жорданово  $LU$ -разложение матрицы  $A$  с тем,

что матрицы  $L$  и  $-U^{-1}$  замещают  $A$ ; (б) на этой основе вычисление  $P = A^{-1}$ ; (в) умножение  $x' = Px$ .

Раздел 3 характеризует второе альтернативное решение: (а) гауссово  $LU$ -разложение матрицы  $A$  с тем, что матрицы  $L$  и  $U$  замещают  $A$ ; (б) на этой основе вычисление  $P = A^{-1}$  в элиминативной форме, а именно: отыскание  $L^{-1}$  и  $U^{-1}$  посредством элементарных матриц и затем перемножение:  $P = U^{-1}L^{-1}$ ; (в) умножение  $x' = Px$ .

Раздел 4 характеризует третье альтернативное решение: (а) гауссово  $LU$ -разложение матрицы  $A$  с тем, что матрицы  $L$  и  $U$  замещают  $A$  (как в разделе 1); (б) на этой основе решение системы  $Ax' = x$  относительно  $x'$ .

Раздел 5 оценивает четвертое альтернативное решение, полученное здесь по принципу МНК (1)–(3), при этом выбрана одна из возможных численно устойчивых версий этого метода – вышеупомянутый квадратно-корневой алгоритм Поттера.

## 1 СТРАТЕГИЯ $\Sigma_1$ : ГАУССОВА СХЕМА РЕШЕНИЯ СИСТЕМ ПОСРЕДСТВОМ ОБРАТНОЙ МАТРИЦЫ

Алгоритмы  $LU$ -разложения матрицы  $A$  разнообразны и хорошо известны [5, с. 27–81]. Возьмем для сравнения три типа алгоритмов (рис. 1).

**Шаг 1:**  $A = LU$ . Подсчет показывает, что для всех схем (см. рис. 1), несмотря на различие в порядке действий, их вычислительная сложность одинакова:

$$\Sigma_1^{(1)} = \frac{(N-1)N(N+1)}{3}.$$

**Шаг 2:** (а) Из системы  $LW = I$  находим  $W$ .

Здесь  $\Sigma_1^{2(a)} = \frac{N(N+1)(N+2)}{6}.$

(б) Из системы  $UX = W$  находим  $X \triangleq A^{-1}$ ,  $P := X$ .

Здесь  $\Sigma_1^{2(b)} = \frac{(N-1)N^2}{2}.$

В сумме:

$$\Sigma_1^{(2)} = \Sigma_1^{2(a)} + \Sigma_1^{2(b)} = \frac{N(2N^2 + 1)}{3}.$$

**Шаг 3:**  $x' = Px$ . Здесь  $\Sigma_1^{(3)} = N^2$ . Таким образом, стратегия  $\Sigma_1$  имеет следующую сложность:

$$\Sigma_1 = \Sigma_1^{(1)} + \Sigma_1^{(2)} + \Sigma_1^{(3)} = N^2(N+1). \quad (4)$$

## 2 СТРАТЕГИЯ $\Sigma_2$ : ЖОРДАНОВА СХЕМА РЕШЕНИЯ СИСТЕМ ПОСРЕДСТВОМ ОБРАТНОЙ МАТРИЦЫ

Как известно, метод Жордана предусматривает полное исключение неизвестных (выше и ниже ведущего уравнения). Подсчет количества операций по этому алгоритму [5, с. 37–41] показывает:

**Шаг 1:**  $A = LU$  с получением матрицы  $L$  и матрицы  $-U^{-1}$ :

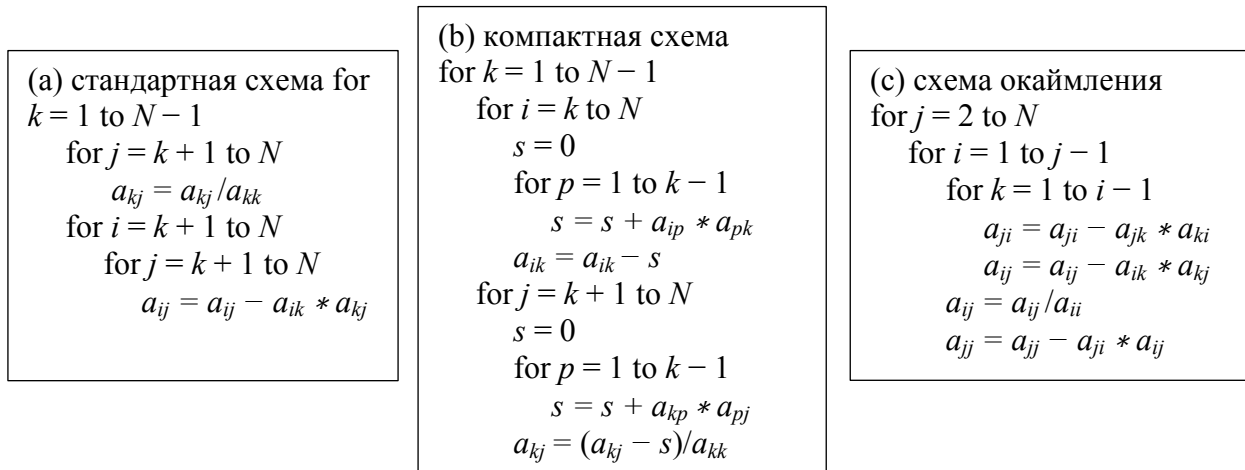


Рис. 1. LU-разложение  $N \times N$ -матрицы  $A$ : (a) гауссово исключение по столбцам; (b) компактная схема Краута; (c) алгоритм окаймления. Матрица  $L$  – нижнетреугольная. Матрица  $U$  – верхнетреугольная с единичной диагональю. Нетривиальные элементы матриц  $L$  и  $U$  записывают поверх данной матрицы  $A$

$$\Sigma_2^{(1)} = \frac{(N-1)N^2}{2}.$$

Шаг 2:  $L^{-1}$  через элементарные матрицы:

$$\Sigma_2^{(2)} = \frac{(N-1)N(N+1)}{6}.$$

Шаг 3:  $P := A^{-1} = U^{-1}L^{-1}$ :

$$\Sigma_2^{(3)} = \frac{(N-1)N(N+1)}{3}.$$

Шаг 4:  $x' = Px$ . Здесь  $\Sigma_2^{(4)} = N^2$ .

Таким образом, стратегия  $\Sigma_2$  имеет следующую сложность:

$$\Sigma_2 = \Sigma_2^{(1)} + \Sigma_2^{(2)} + \Sigma_2^{(3)} + \Sigma_2^{(4)} = \frac{N(N+1)(2N-1)}{2}. \quad (5)$$

### 3 СТРАТЕГИЯ $\Sigma_3$ : ГАУССОВА СХЕМА С ЭЛИМИНАТИВНОЙ ФОРМОЙ ОБРАТНОЙ МАТРИЦЫ

Эта схема отличается методом отыскания матриц  $L^{-1}$  и  $U^{-1}$  после LU-разложения [5, с. 42–46]. Алгоритм состоит из пяти действий со следующими затратами:

Шаг 1:  $A = LU$ , как в стратегии  $\Sigma_1$ :

$$\Sigma_3^{(1)} = \frac{(N-1)N(N+1)}{3}.$$

Шаг 2:  $L^{-1}$  через элементарные матрицы, как в стратегии  $\Sigma_2$ :

$$\Sigma_3^{(2)} = \frac{(N-1)N(N+1)}{6}.$$

Шаг 3:  $U^{-1}$  через элементарные матрицы:

$$\Sigma_3^{(3)} = \frac{(N-2)(N-1)N}{6}.$$

Шаг 4:  $P := A^{-1} = U^{-1}L^{-1}$ , как в стратегии  $\Sigma_2$ :

$$\Sigma_3^{(4)} = \frac{(N-1)N(N+1)}{3}.$$

Шаг 5:  $x' = Px$ . Здесь  $\Sigma_3^{(5)} = N^2$ .

Таким образом, стратегия  $\Sigma_3$  имеет следующую сложность:

$$\Sigma_3 = \Sigma_3^{(1)} + \Sigma_3^{(2)} + \Sigma_3^{(3)} + \Sigma_3^{(4)} + \Sigma_3^{(5)} = \frac{N(N+1)(4N-1)}{6}. \quad (6)$$

### 4 СТРАТЕГИЯ $\Sigma_4$ : ГАУССОВА СХЕМА ДЛЯ РЕШЕНИЯ СИСТЕМ БЕЗ ОБРАТНОЙ МАТРИЦЫ

Теперь перейдем к алгоритмам, избегающим обращения матрицы в решении поставленной задачи.

Шаг 1:  $A = LU$ , как в стратегии  $\Sigma_1$ :

$$\Sigma_4^{(1)} = \frac{(N-1)N(N+1)}{3}.$$

Шаг 2: (a) Из системы  $Lw = I$  находим вектор  $w$ . Здесь  $\Sigma_4^{2(a)} = \frac{N(N+1)}{2}$ . (b) Из системы  $Ux' = w$  на-

ходим  $x'$ . Здесь  $\Sigma_4^{2(b)} = \frac{N(N-1)}{2}$ . В сумме:

$$\Sigma_4^{(2)} = \Sigma_4^{2(a)} + \Sigma_4^{2(b)} = N^2.$$

Таким образом, стратегия  $\Sigma_4$  имеет следующую сложность:

$$\Sigma_4 = \Sigma_4^{(1)} + \Sigma_4^{(2)} = \frac{N(N^2 + 3N - 1)}{3}. \quad (7)$$

### 5 СТРАТЕГИЯ $\Sigma_5$ : КВАДРАТНО-КОРНЕВАЯ МНК-СХЕМА ПОТТЕРА ДЛЯ РЕШЕНИЯ СИСТЕМ

Возьмем в качестве испытываемого МНК-алгоритм Поттера [5, с. 250–251], который ранее для решения

данной задачи  $Ax' = x$  не использовали, а применяли лишь для случая переопределенных систем полного столбцового ранга. Скаляризованная природа этого алгоритма означает, что задачу  $Ax' = x$  надо включать в процесс решения построчно: выполнять цикл по  $n = 1, \dots, N$ . В пределах этого цикла система еще не становится переопределенной. Поэтому погрешность промежуточных МНК-решений  $\tilde{x}$  будет нулевой ( $\nu=0$  и также  $r=0$ , см. [5, с. 250]). Учитывая эти соображения и общую схему (1)–(3), алгоритм за номером (11.35) (см. [5, с. 250]) приводим к виду (8), (9) и (10), который сохраняет аналогию с базовым МНК-алгоритмом (1), (2) и (3), но отличается тем, что, где нужно, транспонирование матриц  $(\cdot)^T$  должно быть заменено на комплексно-сопряженное транспонирование  $(\cdot)^H$ .

I. *Инициализация.* Начальные значения: МНК-оценка  $x_0$  и ее ковариация  $P_0$ :

$$\tilde{x} := x_0, \quad \tilde{P} := P_0. \quad (8)$$

Априорные данные отсутствуют, это означает,  $x_0 = 0$  и  $P_0 = \varepsilon^{-1} \text{diag} [1 \dots 1]$  при возможно малом  $\varepsilon$ , теоретически  $\varepsilon \rightarrow 0$ .

II. *Обработка  $n$ -й порции данных:*  $a \triangleq a_n^T$  есть  $n$ -я строка матрицы  $A$  (записанная в форме столбца) и  $z \triangleq x_n$  есть  $n$ -й элемент вектора  $x$  системы  $Ax' = x$ . Цикл по  $n = 1, \dots, N$ :

$$\left. \begin{aligned} f &= \tilde{S}^H a, \\ \alpha &= f^H f, \\ K &= \tilde{S} f / \alpha, \\ \hat{S} &= \tilde{S} - K f^H, \\ \hat{x} &= \tilde{x} + K(z - A\tilde{x}). \end{aligned} \right\} \quad (9)$$

III. *Экстраполяция между порциями.* Распространение оценки  $\hat{x}$  искомого решения  $x'$  и матрицы  $\hat{S}$  между порциями данных (между вовлекаемыми в процесс уравнениями системы) к моменту повторения этапа II со следующей порцией данных (вплоть до их исчерпания):

$$\tilde{S} := \hat{S}, \quad \tilde{x} := \hat{x}. \quad (10)$$

При  $n = N$  выходим из этапа III и получаем  $\tilde{x}$  в качестве искомого решения  $x'$  системы  $Ax' = x$ . Подсчет показывает, что вычислительная сложность этого алгоритма равна:

$$\Sigma_5 = N(3N^2 + 4N). \quad (11)$$

## 6 СРАВНЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ ПЯТИ СТРАТЕГИЙ: ОТ $\Sigma_1$ ДО $\Sigma_5$

Для пяти рассмотренных стратегий получены выражения (4), (5), (6), (7) и (11), характеризующие их сложность. Чтобы их сравнивать между собой, удобно ввести показатель изменения сложности при переходе от стратегии  $\Sigma_i$  к стратегии  $\Sigma_j$ . Введем его:  $\Delta_{ij} \triangleq \Sigma_i - \Sigma_j$ . Введем также предельные относительные показатели изменения нагрузки на процессор:  $\delta_{ij} \triangleq \lim_{N \rightarrow \infty} \Sigma_i / \Sigma_j$  при переходе от стратегии  $\Sigma_i$  к стратегии  $\Sigma_j$ . Результаты представим в рисунке 2 и таблице 1.

Предельные относительные показатели оказываются такими:

$$\delta_{14} = 3, \delta_{24} = 3, \delta_{34} = 2 \Rightarrow \text{уменьшение нагрузки};$$

$$\delta_{12} = 1, \delta_{23} = 3/2, \delta_{13} = 3/2 \Rightarrow \text{сохранение или уменьшение нагрузки};$$

$$\delta_{15} = 1/3, \delta_{35} = 2/9, \delta_{45} = 1/9 \Rightarrow \text{увеличение нагрузки}.$$

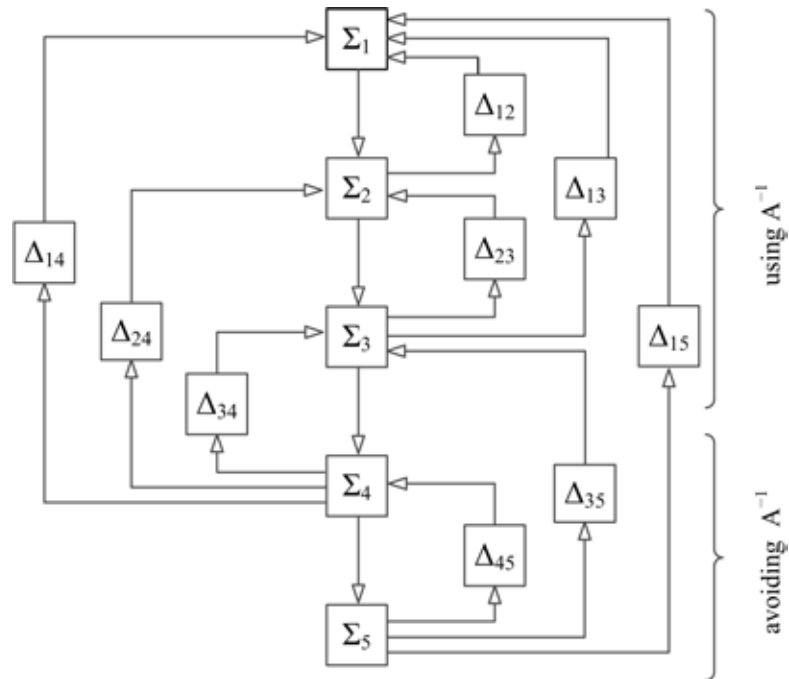


Рис. 2. Вычислительная нагрузка стратегий от  $\Sigma_1$  до  $\Sigma_5$ . Стратегии  $\Sigma_1 - \Sigma_3$  используют обратную матрицу; стратегии  $\Sigma_4$  и  $\Sigma_5$  избегают ее. Снижение нагрузки при переходе  $\Sigma_i \rightarrow \Sigma_j$  равно  $\Delta_{ij}$

Таблица 1  
Снижение нагрузки  $\Delta_{ij}$  при переходе от стратегии  $\Sigma_i$  к стратегии  $\Sigma_j$

$\Delta_{14} = (N/3)(2N^2 + 1)$	$\Delta_{24} = (N/6)(N-1)(4N+1)$	$\Delta_{34} = (N/6)(N-1)(2N-1)$
$\Delta_{12} = (N/2)(N+1)$	$\Delta_{23} = (N/3)(N-1)(N+1)$	$\Delta_{13} = (N/6)(N+1)(2N+1)$
$\Delta_{15} = -N^2(2N+3)$	$\Delta_{35} = -(N/6)(14N^2 + 21N + 1)$	$\Delta_{45} = -(N/3)(8N^2 + 9N + 1)$



### ЗАКЛЮЧЕНИЕ

В статье приведены точные расчетные формулы, выражающие вычислительную сложность решения линейной системы алгебраических уравнений с использованием обратной матрицы системы и без нее. Три стратегии, включающие обратную матрицу, различаются методом ее вычисления: первая ( $\Sigma_1$ ) – с гауссовым  $LU$ -разложением, вторая ( $\Sigma_2$ ) – с жордановым  $LU$ -разложением и третья ( $\Sigma_3$ ) – с элиминативной формой обратной матрицы. Они пронумерованы в порядке уменьшения сложности:  $\Sigma_1 > \Sigma_2 > \Sigma_3$ . Наименьший эффект уменьшения имеет место при переходе  $\Sigma_1 \rightarrow \Sigma_2$ . Он равен  $\Delta_{12} = (1/2)N(N+1)$  и при возрастании  $N$  не представляет интереса. Переход  $\Sigma_2 \rightarrow \Sigma_3$  имеет эффект  $\Delta_{23} = (N/3)(N-1)(N+1)$  и при возрастании  $N$  дает выигрыш в 1,5 раза – такой же, как переход  $\Sigma_1 \rightarrow \Sigma_3$ , при этом  $\Delta_{13} = \Delta_{12} + \Delta_{23}$ . Переход  $\Sigma_1 \rightarrow \Sigma_4$  имеет наибольший эффект:  $\Delta_{14} = \Delta_{13} + \Delta_{34}$ . При возрастании  $N$  он представляет наибольший интерес, поскольку дает выигрыш в 3 раза. Такой заметный выигрыш понятен: он возникает, благодаря отказу от обращения матрицы для решения системы.

Пятая стратегия ( $\Sigma_5$ ) – достаточно «экзотическая». При возрастании  $N$  она создает второе более высокую нагрузку на процессор, чем базовая стратегия ( $\Sigma_1$ ), но она защищена от ситуаций плохой численной обу-

словленности матриц системы. Кроме того, она вдвое эффективнее, чем базовая стратегия, использует разрядную сетку процессора. Поэтому  $\Sigma_5$  может быть рекомендована для приложений с относительно невысокими значениями  $N$  размера матриц.

### СПИСОК ЛИТЕРАТУРЫ

1. Fletcher F. Practical Methods of Optimization: Second Edition. — Chichester | New York | Brisbane | Toronto | Singapore : John Wiley & Sons, 1987.
2. Cendrillon R., Ginis G., Bogaert Etienne Van den, Moonen M. A Near-Optimal Linear Crosstalk Precoder for Downstream VDSL. — IEEE Transactions on Communications, Vol. 55, No. 5, May 2007. — pp. 860–863.
3. Dungen M. Crosstalk Mitigation Techniques for Digital Subscriber Line Systems // Doktor-Ingenieurin (Dr.-Ing.) genehmigte Dissertation. — Germany: Technischen Universität Hamburg-Harburg (TUHH), 2016. — URL: <https://tubdok.tub.tuhh.de/handle/11420/1296>.
4. Semushin I.V. Adaptive State Prediction Methods for Big Data Modeling and Design // Prof. Semushin's talk at Huawei's "The 3rd Professor Day Academic Conference," Huawei Russian Research Center, Moscow, 20–21 December 2016. — URL: <https://www.dropbox.com/s/k2kq7r56dn0rhcd/semushinTalk-20Dec2016.pdf?dl=0&m=>.
5. Семушин И.В. Вычислительные методы алгебры и оценивания. — Ульяновск : УлГТУ, 2011. — URL: <http://venec.ulstu.ru/lib/disk/2013/119.pdf>.