

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ»

Б. И. Адамов, А. Н. Маслов, Н. В. Осадченко

**ПРИМЕНЕНИЕ ОСНОВНЫХ МАТРИЧНЫХ
РАЗЛОЖЕНИЙ В ЗАДАЧАХ
МЕХАНИКИ И РОБОТОТЕХНИКИ**

Учебное пособие

по курсам

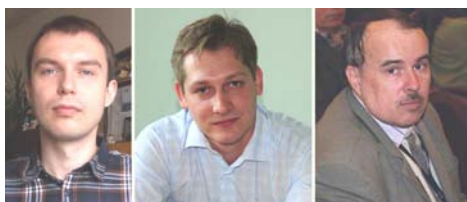
«Вычислительная механика», «Численные методы в робототехнике»,
«Вычислительные методы компьютерного моделирования в механике»,
«Основы мехатроники и робототехники»

для студентов, обучающихся по направлению
«Мехатроника и робототехника»

Москва

Издательство МЭИ

2019



УДК 519
ББК 22.193
А281

*Утверждено учебным управлением НИУ «МЭИ»
в качестве учебного издания*

Подготовлено на кафедре робототехники, мехатроники,
динамики и прочности машин (РМДиПМ)

Рецензенты: М.Н. Кирсанов, д-р. физ.-мат. наук, проф. каф. РМДиПМ
Е.С. Сбытова, канд. физ.-мат. наук, вед. инж. ФГУП МОКБ «Марс»

Адамов Б.И.

А281 Применение основных матричных разложений в задачах механики и робототехники /
Б.И.Адамов, А.Н.Маслов, Н.В.Осадченко. — М.: Издательство МЭИ, 2019. — 84 с.

ISBN 978-5-7046-2150-8

Учебное пособие посвящено методам решения систем линейных алгебраических уравнений (СЛАУ) и других задач вычислительной линейной алгебры с помощью основных матричных разложений. Рассмотрены разложения на треугольные и ортогональные множители, разложение по сингулярным числам. Рассмотрены также прикладные задачи механики и робототехники, в которых возникает необходимость применения обсуждаемых методов: обратная задача о скоростях манипулятора, задачи планирования траектории летательного аппарата с помощью кубических сплайнов, исследования свободных колебаний механических систем, идентификации параметров вынужденных колебаний таких систем. Пособие снабжено подробными примерами решения задач в среде Scilab.

Для студентов, обучающихся по направлению «Мехатроника и робототехника» (15.03.06, 15.04.06).

**УДК 519
ББК 22.193**

Учебное издание

Адамов Борис Игоревич, **Маслов** Антон Николаевич, **Осадченко** Николай Владимирович

**ПРИМЕНЕНИЕ ОСНОВНЫХ МАТРИЧНЫХ РАЗЛОЖЕНИЙ
В ЗАДАЧАХ МЕХАНИКИ И РОБОТОТЕХНИКИ**

Учебное пособие

Редактор Д. Р. Чернова

Компьютерная вёрстка З. Х. Айнетдиновой

Подписано в печать 27.08.2019 г
Печ. л. 5,25 Тираж 75 экз.

Печать цифровая
Изд. № 19у-004

Формат 60×84/16
Заказ № 253

Оригинал-макет подготовлен в РИО НИУ «МЭИ».
111250, г. Москва, ул. Красноказарменная, д. 14.
Отпечатано в типографии НИУ «МЭИ».
111250, г. Москва, ул. Красноказарменная, д. 13.

ISBN 978-5-7046-2150-8

© Национальный исследовательский
университет «МЭИ», 2019

Содержание

ВВЕДЕНИЕ	4
1. ТРЕУГОЛЬНЫЕ РАЗЛОЖЕНИЯ	5
1.1. Матрицы специального вида	5
1.2. Разложение Холецкого	13
1.3. LU -разложение без выбора ведущего элемента	16
1.4. LU -разложение с частичным выбором ведущего элемента	18
2. ОРТОГОНАЛЬНЫЕ РАЗЛОЖЕНИЯ	28
2.1. QR -разложение: метод вращений	28
2.2. QR -разложение: метод отражений	34
3. СИНГУЛЯРНОЕ РАЗЛОЖЕНИЕ И МЕТОД НАИМЕНЬШИХ КВАДРАТОВ	45
3.1. SVD -разложение	45
3.2. Применение матричных разложений для исследования сво- бодных колебаний устойчивой линейной консервативной си- стемы	47
3.3. Нормальные псевдорешения СЛАУ	52
3.4. Применение матричных разложений для решения обратной задачи о скоростях избыточного манипулятора	56
3.5. Метод наименьших квадратов	61
3.6. Определение характеристик колебаний с помощью метода наименьших квадратов	62
4. СИСТЕМЫ УРАВНЕНИЙ С ТРЁХДИАГОНАЛЬНЫМИ МАТРИЦАМИ	66
4.1. Метод прогонки	66
4.2. Кусочные многочлены и сплайны	68
4.3. Интерполяция кубическими сплайнами	72
4.4. Применение кубических сплайнов в задаче планирования траектории автономного летательного аппарата	74
Заключение	81
Список рекомендованной литературы	82

ВВЕДЕНИЕ

Решение многих задач механики и робототехники требует проведения численных расчётов с применением компьютера. В таких расчётах находят применение методы вычислительной линейной алгебры, значительная часть которых основана на получении различного характера разложений используемых матриц, т.е. представления их в виде произведений матриц специального вида.

В пособии приведены необходимые определения и представлены алгоритмы, предназначенные для получения основных матричных разложений и решения с их помощью систем линейных алгебраических уравнений, а также других задач. Рассмотрено также применение метода прогонки в задачах сплайн-интерполяции.

Изложение рассматриваемых методов сопровождается примерами их реализации в системе компьютерной математики Scilab – свободно распространяемом пакете, предназначенном для выполнения инженерных и научных вычислений [5]. По многим параметрам система Scilab не уступает широко распространённому математическому пакету MatLab; однако последний – коммерческая система, не имеющая открытого программного кода, и лицензия на его использование является весьма дорогостоящей, в то время как среда разработки Scilab – бесплатный для конечного пользователя продукт с открытым исходным кодом [6,7].

Общедоступность этой многофункциональной программной системы – её весомое преимущество по сравнению с пакетом MatLab, особенно важное для применения в учебном процессе вузов [8]. По интерфейсу оба пакета достаточно сходны; при этом пакет Scilab слабее поддерживает аналитические операции, но обеспечивает широкий набор инструментов для численных расчётов и может рассматриваться как облегчённый вариант MatLab, сохраняющий большинство возможностей последнего [6].

Помимо примеров чисто вычислительного характера, в пособии рассмотрено использование методов для решения ряда важных для практики задач механики и робототехники.

В пособии рассматриваются только матрицы с действительными элементами. Векторы и линейные операторы обозначены прямым полужирным шрифтом, соответствующие им столбцы и матрицы – обычным курсивным шрифтом (как и скаляры; подобные обозначения широко используют как в линейной алгебре [9], так и в механике [10] и робототехнике [11]). Компоненты векторов и линейных операторов (соответственно, элементы столбцов и матриц) обозначены буквами с индексами, причём нумерация индексов ведётся с нуля.

1. ТРЕУГОЛЬНЫЕ РАЗЛОЖЕНИЯ

1.1. Матрицы специального вида

Различные классы матриц выделяются в соответствии с теми или иными условиями, налагаемыми на их элементы. В ряде важнейших случаев такие же условия могут быть наложены и на линейные операторы, соответствующие этим матрицам. Рассмотрим примеры.

Пусть $C: X \rightarrow Y$ – линейный оператор, действующий из n -мерного векторного пространства X в m -мерное векторное пространство Y . Пусть, далее, $\{e_j\}_{j=0}^{j < n}$ и $\{e'_i\}_{i=0}^{i < m}$ – базисы в X и Y .

Коэффициенты c_{ij} в разложении

$$C e_j = \sum_{i=0}^{i < m} c_{ij} e'_i$$

называются **компонентами** оператора C , а матрица $C \in \mathbb{R}^{m \times n}$ с этими элементами – **матрицей** оператора C [1].

Таким образом, для нахождения компонент линейного оператора достаточно умножить его на векторы базиса в пространстве аргументов, а затем разложить полученные произведения по векторам базиса в пространстве значений.

В дальнейшем все векторные пространства предполагаем евклидовыми, а базисы – ортонормированными. В этих условиях для компонент линейного оператора справедлива явная формула

$$c_{ij} = (e'_i, C e_j)$$

(круглые скобки обозначают скалярное произведение).

Рассмотрим теперь важный частный класс операторов и матриц.

Оператор $S: X \rightarrow X$ – **симметричный** [10], если

$$S^T = S ; \tag{1.1}$$

существенно, что в этом определении пространство аргументов и пространство значений оператора совпадают.

Знак T в (1.1) обозначает операцию транспонирования линейного оператора. В общем случае для оператора $C: X \rightarrow Y$ **транспонированным**

оператором называется [9,11] оператор $C^T : Y \rightarrow X$, однозначно определяемый условием: для любых векторов $v \in X$, $w \in Y$ выполнено тождество

$$(v, C^T w) = (Cv, w).$$

Матрица S симметричного оператора удовлетворяет аналогичному условию:

$$S^T = S, \quad (1.2)$$

в котором транспонирование понимается в обычном смысле [12]: строки и столбцы меняются местами. Вообще же соотношение (1.2) служит определением [2] **симметричной матрицы** (она заведомо – квадратная).

В механике примером симметричной матрицы служит *матрица жесткостей*, фигурирующая в уравнениях малых колебаний консервативных систем [13].

Среди симметричных операторов и матриц выделяют два подкласса.

Симметричный оператор $S : X \rightarrow X$ – **положительно полуопределённый**, если

$$(v, Sv) \geq 0 \quad \forall v.$$

Симметричный оператор $S : X \rightarrow X$ – **положительно определённый**, если

$$(v, Sv) > 0 \quad \forall v \neq 0.$$

Для симметричной матрицы условия положительной полуопределённости и положительной определённости записываются так:

$$(v, Sv) \geq 0 \quad \forall v \quad \text{и} \quad (v, Sv) > 0 \quad \forall v \neq 0.$$

В механике примером симметричной положительной определённой матрицы служит *инерционная матрица*, входящая в уравнения малых колебаний консервативных систем [13].

Оператор $A : X \rightarrow X$ – **антисимметричный** [10], если

$$A^T = -A; \quad (1.3)$$

его матрица также является **антисимметричной**, удовлетворяя условию

$$A^T = -A. \quad (1.4)$$

В механике примером антисимметричного линейного оператора служит [11] *оператор угловой скорости* абсолютно твёрдого тела.

Переходим к следующему важному частному классу операторов и матриц.

Оператор $Q: X \rightarrow Y$ – **ортогональный** [10], если

$$Q^T = Q^{-1}; \quad (1.5)$$

его матрица также является **ортогональной**, удовлетворяя условию

$$Q^T = Q^{-1}. \quad (1.6)$$

Из этих определений вытекает, что ортогональные операторы заведомо являются обратимыми (и действуют между пространствами одинаковой размерности), а ортогональные матрицы являются невырожденными.

Множество всех ортогональных операторов вида $Q: X \rightarrow Y$ обозначают $O(X, Y)$. Если оба пространства совпадают, то вместо $O(X, X)$ пишут $O(X)$; данное множество является мультипликативной группой (**ортогональная группа** пространства X). Группой является и множество $O(n, \mathbb{R})$ всех ортогональных матриц порядка n (**ортогональная группа степени n**).

При умножении сомножителей скалярного произведения на ортогональный оператор это произведение не изменяется (нередко это свойство сохранения скалярных произведений принимают [14] за определение ортогонального оператора). Отсюда следует, что умножение на ортогональный оператор не меняет 2-норму вектора или оператора; аналогично, умножение на ортогональную матрицу не меняет 2-норму столбца или матрицы. Столбцы ортогональной матрицы ортонормированы и попарно ортогональны.

Далее, определитель ортогональной матрицы может быть равен либо 1, либо -1 . В случае $\det Q = 1$ ортогональная матрица называется собственной. Множество всех собственных ортогональных матриц порядка n обозначается $SO(n, \mathbb{R})$ и является группой (**собственная ортогональная группа степени n**) [10].

Весьма частный случай ортогональных матриц представляют собой матрицы перестановок. **Матрица перестановок** – это такая квадратная матрица, у которой в каждой строке и в каждом столбце ровно один элемент отличен от нуля и равен единице [15].

Пусть в j -м столбце матрицы перестановок P порядка n ненулевой элемент стоит в строке с номером $k = k(j)$. Сама матрица будет однозначно задана, если указать для каждого столбца номер той строки, где

расположен ненулевой элемент столбца, т.е. задать функцию $k = k(j)$. Например, для матрицы

$$P = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

имеем: $k(0) = 1$, $k(1) = 2$, $k(2) = 0$. В общем случае элемент матрицы P задаётся таким условным выражением (нотация взята из языка Си):

$$p_{ij} = (i = k(j) ? 1 : 0) .$$

Умножение столбца на матрицу перестановок сводится к перестановке его элементов по правилу: если $b \in \mathbb{R}^n$ и $\beta = Pb$, где P – матрица перестановок, то $\beta_{k(j)} = b_j$ (*основное свойство* матрицы перестановок).

Аналогично, умножение матрицы A на матрицу P слева сводится к перестановке строк матрицы A .

Простейшей перестановкой является *транспозиция* – такая перестановка, при которой два элемента являются местами, а все остальные остаются на своих местах [16]. Ей соответствует **матрица транспозиции** – такая матрица перестановок, которая отличается от единичной матрицы I перестановкой двух её строк (не обязательно различных).

Оговорка в скобках означает, что единичная матрица также рассматривается как матрица транспозиции. Любая матрица транспозиции является не только ортогональной, но и симметричной (а значит, сама для себя служит обратной матрицей).

Справедлива теорема о *факторизованном представлении* матрицы перестановок, по которой любая матрица перестановок P порядка n однозначно представима как произведение n матриц транспозиций вида

$$P = P_{n-1} P_{n-2} \dots P_0 ;$$

здесь j -й сомножитель (нумерация – справа) – матрица транспозиции, отвечающая за перестановку элементов с номерами j и $t = t(j)$, $t \geq j$.

Любая из матриц в данном представлении однозначно определяется заданием числа $t(j)$. Поэтому в компьютерных программах матрицу перестановок можно задать при помощи целочисленного массива перестановок \mathbf{iP} , в котором $\mathbf{iP}[j] = t(j)$.

Задача. Пусть в массиве \mathbf{B} размещены элементы столбца $b = (0 \ 1 \ 2 \ 3)^T$. Требуется получить в том же массиве элементы столбца Pb , если матрица перестановок P представлена своим массивом перестановок $iP = \{2 \ 1 \ 3 \ 3\}$.

Решение. Произведение Pb вычисляем как $P_3P_2P_1P_0b$. Для проведения требуемой перестановки элементов массива \mathbf{B} воспользуемся следующей **Scilab**-программой:

```
--> B = [0; 1; 2; 3];
--> iP = [2; 1; 3; 3] + 1;
--> n = size(B,1);
--> for j = 1:n;
-->     t = iP(j);
-->     s = B(t); B(t) = B(j); B(j) = s;
--> end
--> B'
ans =
      2.      1.      3.      0.
```

Здесь необходимо отметить, что индексация элементов массивов в **Scilab** ведётся не с нуля, а с единицы. По этой причине значения счётчика j и все элементы массива перестановок iP в рассмотренной программе увеличены на единицу. Также заметим, что символы `-->` являются частью интерфейса командного окна **Scilab**, а не программного кода.

Переходим теперь к таким определениям, которые имеют смысл только для матриц, но не для операторов. Если матрица оператора в каком-либо базисе удовлетворяет одному из них, то в другом базисе она удовлетворять такому определению не обязана.

Матрица D – **диагональная**, если

$$i \neq j \Rightarrow d_{ij} = 0$$

(в этом определении, в отличие от последующих, матрица D не обязательно предполагается квадратной).

Матрица L – **нижняя треугольная (левая треугольная)**, если

$$i < j \Rightarrow l_{ij} = 0.$$

Матрица U – **верхняя треугольная (правая треугольная)**, если

$$i > j \Rightarrow u_{ij} = 0.$$

Таким образом, у нижней треугольной матрицы равны нулю все элементы над главной диагональю, а у верхней треугольной матрицы

нулевыми являются все элементы под главной диагональю [2]. Квадратная диагональная матрица – частный случай как нижней, так и верхней треугольной матрицы.

Треугольную матрицу, у которой все диагональные элементы равны 1, называют **унитреугольной** [16].

Заметим, что множество $T(n, \mathbb{R})$ всех невырожденных верхних треугольных матриц порядка n и множество $UT(n, \mathbb{R})$ всех верхних унитреугольных матриц того же порядка являются мультипликативными группами (соответственно, **полная треугольная группа степени n** и **полная унипотентная группа степени n**).

Значимость треугольных матриц для вычислительной линейной алгебры определяется тем, что решение системы линейных алгебраических уравнений (СЛАУ) с треугольной матрицей коэффициентов не вызывает трудностей.

В самом деле, пусть дана СЛАУ $Ly = b$, где L – невырожденная нижняя треугольная матрица порядка n , а b – столбец свободных членов, и требуется вычислить столбец $y = b \setminus L$ (обратной дробной чертой обозначена операция левого деления столбца на матрицу, т.е. операция вычисления решения СЛАУ). Тогда вычисления требуют n^2 арифметических операций и выполняются в соответствии со следующей расчётной формулой [2]:

$$y_j = \left(b_j - \sum_{k=0}^{k < j} l_{jk} y_k \right) / l_{jj}, \quad j = 0, \dots, n-1. \quad (1.7)$$

Аналогично, вычисление решения $x = y \setminus U$ для СЛАУ $Ux = y$, где U – невырожденная верхняя треугольная матрица, достигается применением следующей расчётной формулы [2]:

$$x_j = \left(y_j - \sum_{k=j+1}^{k < n} u_{jk} x_k \right) / u_{jj}, \quad j = n-1, \dots, 0. \quad (1.8)$$

Формулы (1.7) и (1.8) несколько упрощаются в случае унитреугольных матриц: операция деления тогда отсутствует.

Приведём **примеры** пользовательских **Scilab**-функций, реализующих арифметические операции левого матричного деления на треугольные матрицы.

Ниже представлен листинг функции **lmdl** для выполнения левого матричного деления матрицы **b** на нижнюю треугольную матрицу **L** в соответствии с (1.7):

```

function y = lmdl(L, b)
    [n, m] = size(b);
    y = zeros(n, m);
    for j = 1:n
        y(j, :) = (b(j, :) - L(j, 1:(j-1)) * y(1:(j-1), :)) / L(j, j);
    end
endfunction

```

Рассмотрим пример решения СЛАУ $Ly = b$ с нижней треугольной матрицей в среде **Scilab** с использованием функции **lmdl**. Исходные данные:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 3 & 0 \\ 3 & -1 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 4 \\ 1 \end{pmatrix}.$$

Зададим соответствующие матрицы в **Scilab**:

```

--> L = [1 0 0; -2 3 0; 3 -1 -1];
--> b = [1; 4; 1];

```

Решение СЛАУ $Ly = b$:

```

--> y = lmdl(L, b); y'
ans =
    1.    2.    0.

```

Заметим, что штрихом (') в пакете **Scilab** обозначается операция транспонирования матрицы; здесь она использована, чтобы вывести ответ в виде строки, а не столбца.

Точность полученного решения характеризует ∞ -норма невязки $Ly - b$:

```

--> norm(L*y - b, %inf)
ans =
    0.

```

В данном конкретном примере решение **y** получено настолько точно, насколько это возможно на компьютере.

В пакете **Scilab** реализована стандартная функция для выполнения левого матричного деления (\backslash) на невырожденную квадратную матрицу общего вида (обратите внимание, что операнды данной операции записывают в обратном порядке). Для сравнения, решим рассматриваемую систему с помощью неё:

```

--> y_ = L\b; y_'
ans =
    1.    2.    2.591D-16

```

Отметим, что **2.591D-16** – это экспоненциальная запись числа $2,591 \cdot 10^{-16}$.

В рассматриваемом примере решение **y_** – менее точное, чем полученное функцией **lmdl**. Об этом свидетельствует и ∞ -норма невязки:

```
--> norm(L*y_ - b, %inf)
ans =
2.220D-16
```

Рассмотрим теперь функцию **lmdu** для выполнения левого матричного деления матрицы **y** на верхнюю треугольную матрицу **U** по формуле (1.8):

```
function x = lmdu(U, y)
[n, m] = size(y);
x = zeros(n, m);
for j = n:-1:1
    x(j, :) = (y(j, :) - U(j, (j+1):n) * x((j+1):n, :)) / U(j, j);
end
endfunction
```

Приведём пример решения СЛАУ $Ux = y$ с верхней треугольной матрицей в среде **Scilab** с использованием функции **lmdu**. Исходные данные:

$$U = \begin{pmatrix} 1 & -2 & 4 \\ 0 & -2 & 3 \\ 0 & 0 & 1 \end{pmatrix}, \quad y = \begin{pmatrix} 9 \\ 6 \\ 2 \end{pmatrix}.$$

Зададим соответствующие матрицы в **Scilab**:

```
--> U = [1 -2 4; 0 -2 3; 0 0 1];
--> y = [9; 6; 2];
```

Решение СЛАУ $Ux = y$:

```
--> x = lmdu(U, y); x'
ans =
1.    0.    2.
```

Точность полученного решения характеризует норма невязки разности $Ux - y$:

```
--> norm(U*x - y, %inf)
ans =
0.
```

В этом конкретном примере решение **x** получено настолько точно, насколько это возможно на компьютере.

Решение рассматриваемой СЛАУ с помощью операции левого матричного деления (\backslash) даёт такой же результат, что и функция **lindu**:

```
--> x_ = U\y; x_ '
ans =
    1.    0.    2.
```

1.2. Разложение Холецкого

Разложение Холецкого имеет вид

$$A = LL^T, \quad (1.9)$$

где L – нижняя треугольная матрица с положительными диагональными элементами; оно существует и единственно для любой симметричной положительно определённой матрицы A порядка n [3,15].

Алгоритм вычисления матрицы L состоит из n однотипных шагов; на j -м шаге, где $j = 0, \dots, n-1$, вычисляется j -й столбец данной матрицы. В свою очередь, шаг включает два этапа.

Этап 1: вычислить диагональный элемент l_{jj} по формуле

$$l_{jj} = \sqrt{a_{jj} - \sum_{k=0}^{k < j} l_{jk} l_{jk}}$$

(наддиагональные элементы матрицы L – нули, так что их вообще не нужно вычислять).

Этап 2: при $i = j+1, \dots, n-1$ вычислить:

$$l_{ij} = \left(a_{ij} - \sum_{k=0}^{k < j} l_{ik} l_{jk} \right) / l_{jj}.$$

Число арифметических операций, требуемых изложенным алгоритмом для получения разложения (1.9): $\frac{1}{3}n^3 + O(n^2)$ [15].

После того, как разложение Холецкого для матрицы A получено, решение СЛАУ $Ax = b$ не составляет трудностей: оно осуществляется по схеме $y = b \backslash L$, $x = y \backslash L^T$ в соответствии с формулами (1.7), (1.8) (в последней из них следует u_{jk} заменить на l_{kj}).

Метод Холецкого – самый быстрый из методов решения СЛАУ, основанных на получении матричных разложений; к тому же он отличается гарантированной вычислительной устойчивостью [3]. Однако он предъявляет к матрице системы требования симметрии и положительной определённости.

Рассмотрим **пример** решения СЛАУ $Ax = b$ с помощью матричного разложения Холецкого. Дано:

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 7 & 3 \\ 3 & 7 & 14 & 1 \\ 4 & 3 & 1 & 59 \end{pmatrix}, \quad b = \begin{pmatrix} 30 \\ 45 \\ 63 \\ 249 \end{pmatrix}.$$

Построим разложение Холецкого для матрицы A .

Полагаем $j = 0$:

$$l_{00} = \sqrt{a_{00}} = 1;$$

$$l_{10} = a_{10} / l_{00} = 2, \quad l_{20} = a_{20} / l_{00} = 3, \quad l_{30} = a_{30} / l_{00} = 4.$$

При $j = 1$ получаем:

$$l_{11} = \sqrt{a_{11} - l_{10}^2} = \sqrt{5 - 4} = 1;$$

$$l_{21} = (a_{21} - l_{20}l_{10}) / l_{11} = 7 - 3 \cdot 2 = 1;$$

$$l_{31} = (a_{31} - l_{30}l_{10}) / l_{11} = 3 - 4 \cdot 2 = -5.$$

При $j = 2$ имеем:

$$l_{22} = \sqrt{a_{22} - l_{20}^2 - l_{21}^2} = \sqrt{14 - 9 - 1} = 2;$$

$$l_{32} = (a_{32} - l_{30}l_{20} - l_{31}l_{21}) / l_{22} = (14 - 4 \cdot 3 - (-5) \cdot 1) / 2 = -5.$$

Наконец, при $j = 3$ получаем:

$$l_{33} = \sqrt{a_{33} - l_{30}^2 - l_{31}^2 - l_{32}^2} = \sqrt{59 - 16 - 25 - 9} = 3.$$

Разложение (1.9) построено:

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 1 & 2 & 0 \\ 4 & -5 & -3 & 3 \end{pmatrix}.$$

Решим систему $Ly = b$, используя формулу (1.7):

$$\begin{cases} y_0 = 30, \\ 2y_0 + y_1 = 45, \\ 3y_0 + y_1 + 2y_2 = 63, \\ 4y_0 - 5y_1 - 3y_2 + 3y_3 = 249; \end{cases} \Rightarrow y = \begin{pmatrix} 30 \\ -15 \\ -6 \\ 12 \end{pmatrix}.$$

Решим систему $L^T x = y$, используя формулу (1.8):

$$\begin{cases} x_0 + 2x_1 + 3x_2 + 4x_3 = 30, \\ x_1 + x_2 - 5x_3 = -15, \\ 2x_2 - 3x_3 = -6, \\ 3x_3 = 12; \end{cases} \Rightarrow x = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}.$$

Проверим полученные результаты, воспользовавшись стандартными функциями пакета **Scilab**. Для построения разложения Холецкого воспользуемся функцией **chol**, которая возвращает верхнюю треугольную матрицу $R = L^T$, $A = R^T R$.

Зададим матрицу A и построим матрицу L :

```
--> A = [1 2 3 4; 2 5 7 3; 3 7 14 1; 4 3 1 59];
--> R = chol(A); L = R'
```

```
L =
 1.    0.    0.    0.
 2.    1.    0.    0.
 3.    1.    2.    0.
 4.   -5.   -3.    3.
```

Найдём погрешность **err** построения разложения (1.9), вычислив, например, 1-норму разности $A - LL^T$:

```
--> err = norm(A-L*L', 1)
err =
 0.
```

Для решения системы $Ly = b$ воспользуемся функцией **lmdl** (см. раздел 1.1):

```
--> b = [30; 45; 63; 249];
--> y = lmdl(L, b); y'
ans =
 30.  -15.  -6.   12.
```

Для решения системы $L^T x = y$ воспользуемся функцией **lmdu** (см. раздел 1.1):

```
--> x = lmdu(L', y); x'
ans =
 1.    2.    3.    4.
```

Такое же решение СЛАУ даёт операция левого матричного деления (\backslash):

```
--> x = A\b; x'
ans =
 1.    2.    3.    4.
```

Результаты решения задачи, полученные двумя различными способами, совпадают.

1.3. LU -разложение без выбора ведущего элемента

Будем теперь предполагать, что A – произвольная невырожденная квадратная матрица. Существует несколько вариантов метода, основанного на LU -разложении. В методе LU -разложения **без выбора ведущего элемента** используют следующее разложение матрицы A :

$$A = LU, \quad (1.10)$$

где L – нижняя унитреугольная, а U – верхняя треугольная матрицы.

Условие унитреугольности преследует цель обеспечения единственности LU -разложения. Однако существует разложение вида (1.10) не для всякой невырожденной матрицы; условием его существования служит отличие от нуля всех главных миноров матрицы A [15].

Предполагая, что это так, изложим алгоритм вычисления матриц L и U . Он состоит из n однотипных шагов; на j -м шаге, где $j = 0, \dots, n-1$, формируются столбцы матриц L и U с номером j . Один шаг включает три этапа.

Этап 1: при $i = 0, \dots, j-1$ вычислить элементы u_{ij} :

$$u_{ij} = a_{ij} - \sum_{k=0}^{k < i} l_{ik} u_{kj}.$$

Этап 2: по аналогичной формуле вычислить элемент u_{jj} :

$$u_{jj} = a_{jj} - \sum_{k=0}^{k < j} l_{jk} u_{kj}.$$

Этап 3: при $i = j+1, \dots, n-1$ вычислить элементы l_{ij} :

$$l_{ij} = \left(a_{ij} - \sum_{k=0}^{k < j} l_{ik} u_{kj} \right) / u_{jj}.$$

Изложенный алгоритм для нахождения факторов разложения (1.10) требует в 2 раза больше арифметических операций, чем в методе Холецкого: $\frac{2}{3} n^3 + O(n^2)$ [15].

Заметим, что на этапе 3 алгоритма происходит деление на очередной диагональный элемент u_{jj} матрицы U , называемый **ведущим элементом** для j -го шага алгоритма. Требование отличия всех главных миноров матрицы A от нуля гарантирует, что эти элементы – ненулевые (фактически для каждого j ведущий элемент равен отношению главных миноров порядка $j+1$ и j). Однако некоторые ведущие элементы могут оказаться близкими к нулю, и тогда возможен неконтролируемый рост вычислительной погрешности.

После получения LU -разложения матрицы системы $Ax = b$ решение последней осуществляется по схеме $y = b \setminus L$, $x = y \setminus U$ в соответствии с формулами (1.7), (1.8) (в первой из них операция деления отсутствует).

Существует важный класс матриц, для которых метод LU -разложения без выбора ведущего элемента заведомо работоспособен и вычислительно устойчив. Введём соответствующее определение.

Матрица A , для любой строки которой

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}| + \delta, \quad \delta > 0$$

(i – номер строки), называется матрицей с **диагональным преобладанием** величины δ .

Справедлива [17] *теорема Леви – Деспланка*: матрица A с диагональным преобладанием величины δ всегда обратима, причём

$$\|A^{-1}\|_{\infty} \leq \frac{1}{\delta};$$

из данной теоремы немедленно вытекает такое *следствие*:

$$\text{cond}_{\infty} A \leq \frac{\|A\|_{\infty}}{\delta}$$

(в левой части последнего неравенства записано **число обусловленности** матрицы A ; по определению оно для невырожденной матрицы равно произведению норм данной матрицы и обратной к ней, а для вырожденной матрицы равно ∞ [2,3]). Таким образом, матрица A оказывается хорошо обусловленной, если только δ не слишком мало.

Применяя теорему Леви – Деспланка к подматрицам матрицы A , соответствующим её главным минорам, немедленно получаем, что все они отличны от нуля.

В качестве примера построим разложение (1.10) для матрицы

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -4 & 6 & 1 \\ 2 & 7 & 5 \end{pmatrix}.$$

Найдём элементы столбцов с индексом $j = 0$:

$$u_{00} = a_{00} = 2, \quad l_{10} = a_{10} / u_{00} = -2, \quad l_{20} = a_{20} / u_{00} = 1.$$

Теперь примем $j = 1$:

$$u_{01} = a_{01} = -1, \quad u_{11} = a_{11} - l_{10}u_{01} = 4, \quad l_{21} = (a_{21} - l_{20}u_{01}) / u_{11} = 2.$$

Сформируем последний столбец матрицы U ; $j = 2$:

$$u_{02} = a_{02} = 0, \quad u_{12} = a_{12} - l_{10}u_{02} = 1, \quad u_{22} = a_{22} - l_{20}u_{02} - l_{21}u_{12} = 3.$$

Разложение (1.10) построено:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 2 & -1 & 0 \\ 0 & 4 & 1 \\ 0 & 0 & 3 \end{pmatrix}.$$

1.4. LU -разложение с частичным выбором ведущего элемента

Метод LU -разложения с **частичным выбором ведущего элемента** уже не накладывает на матрицу A каких-либо дополнительных условий, кроме невырожденности. Он основывается на том, что надлежащей *перестановкой строк* невырожденной матрицы всегда можно сделать её главные миноры отличными от нуля; LU -разложение здесь строится не для самой матрицы A , а для матрицы PA , где P – матрица перестановок:

$$PA = LU. \quad (1.11)$$

При этом матрица P (а значит, и матрица PA) заранее неизвестна и получается в ходе преобразований, имеющих пошаговый характер. Серию этих преобразований можно задать схемами:

$$A \equiv A_{-1} \rightarrow A_0 \rightarrow \dots \rightarrow A_{n-1} \equiv PA,$$

$$A \equiv U_{-1} \rightarrow U_0 \rightarrow \dots \rightarrow U_{n-1} \equiv U,$$

$$I \equiv L_{-1} \rightarrow L_0 \rightarrow \dots \rightarrow L_{n-1} \equiv L.$$

Для любого j должно выполняться равенство $A_j = L_j U_j$.

В рассмотренных цепочках преобразований переход от предыдущей матрицы к последующей задаётся рекуррентными формулами

$$A_j = P_j U_{j-1}, \quad (1.12)$$

$$U_j = N_j P_j U_{j-1}, \quad (1.13)$$

$$L_j = P_j L_{j-1} P_j N_j^{-1}. \quad (1.14)$$

Матрица P_j – j -я матрица транспозиции в факторизованном представлении матрицы перестановок P . Задающее её число $t(j)$ – это номер той строки матрицы U_{j-1} , в которой находится *максимальный по модулю* элемент j -го столбца этой матрицы, причём поиск такого элемента ведут в строках с номерами $j, \dots, n-1$. В соответствии с (1.12) он станет ведущим элементом для матрицы A_j и последующих матриц соответствующей цепочки преобразований, включая и матрицу U .

Матрица N_j – элементарная нижняя треугольная матрица, т.е. [15] такая нижняя унитреугольная матрица, у которой среди поддиагональных элементов могут быть отличными от нуля лишь элементы j -го столбца: $v_{j+1,j}, \dots, v_{n-1,j}$. Выбирают их так.

Пусть $\tilde{U} = P_j U_{j-1}$; тогда $v_{ij} = -\tilde{u}_{ij}/\tilde{u}_{jj}$. В соответствии с (1.13), такой выбор обеспечивает обнуление поддиагональных элементов j -го столбца матрицы U_j и последующих матриц соответствующей цепочки преобразований, включая и матрицу U .

Проведение вычислений по формуле (1.14) ведётся с учётом того, что обращение элементарной нижней треугольной матрицы сводится к изменению знаков её ненулевых поддиагональных элементов.

Практическая реализация алгоритма LU -разложения с частичным выбором ведущего элемента отличается от изложенной выше пошаговой процедуры получения членов разложения (1.11) порядком выполнения арифметических операций и способом хранения необходимой информации в памяти компьютера. Именно, элементы матриц A_j, U_j, L_j хранятся в одном двумерном массиве \mathbf{LU} , а вместо матриц транспозиций P_j используется массив перестановок \mathbf{iP} .

Перед началом выполнения шагов алгоритма матрица A копируется в массив \mathbf{LU} . Алгоритм состоит из n однотипных шагов; по завершении j -го его шага, где $j = 0, \dots, n-1$, в массиве \mathbf{LU} находятся:

- ненулевые элементы столбцов матрицы U_j с номерами $0, \dots, j$;
 - поддиагональные элементы столбцов матрицы L_j с номерами $0, \dots, j$;
 - элементы столбцов матрицы A_j с номерами $j+1, \dots, n-1$;
- в начальной части массива \mathbf{iP} находятся числа $t(0), \dots, t(j)$.

j -й шаг алгоритма включает четыре этапа. Заметим, что к началу шага – в отличие от пошаговой процедуры – элементы u_{ij} j -го столбца матрицы U_{j-1} еще не вычислены, и первые два этапа посвящены их нахождению.

Этап 1: при $i = 0, \dots, j-1$ вычислить элементы u_{ij} :

$$u_{ij} = a_{ij} - \sum_{k=0}^{k < i} l_{ik} u_{kj}$$

(они совпадают с соответствующими элементами матрицы U_j).

Этап 2: при $i = j, \dots, n-1$ вычислить элементы u_{ij} :

$$u_{ij} = a_{ij} - \sum_{k=0}^{k < j} l_{ik} u_{kj}$$

(эти формулы отличаются от предыдущих верхним пределом в сумме).

Этап 3: найти максимальный по модулю элемент среди всех u_{ij} , где $i \geq j$; если $t = t(j)$ – номер строки, содержащей его, то при $t \neq j$ меняем строки массива \mathbf{LU} с номерами t и j местами; полагаем $\mathbf{iP}[j] = \mathbf{t}$.

Заметим, что после завершения данного этапа j -й столбец массива \mathbf{LU} содержит элементы j -го столбца матрицы \tilde{U} .

Этап 4: при $i = j+1, \dots, n-1$ вычислить элементы l_{ij} (теперь так мы обозначаем уже элементы матрицы L_j):

$$l_{ij} = -v_{ij} = \tilde{u}_{ij} / \tilde{u}_{jj}.$$

После построения разложения (1.11) для матрицы системы $Ax = b$ решение последней осуществляется по схеме $y = (Pb) \setminus L$, $x = y \setminus U$.

Реализовать операцию $y = (Pb) \setminus L$ можно, объединив формулы (1.7) с решением задачи о перестановке элементов столбца b . Это делается копированием значений b_j в массив \mathbf{x} , после чего для $j = 0, \dots, n-1$ полагаем:

а) $\mathbf{t} = \mathbf{iP}[j]$, $\mathbf{s} = \mathbf{x}[\mathbf{t}]$, $\mathbf{x}[\mathbf{t}] = \mathbf{x}[j]$;

б) $\mathbf{x}[j] = \mathbf{s} - \sum_{k=0}^{k < j} l_{jk} \mathbf{x}[k]$.

Операцию $x = y \setminus U$ реализуем, как и раньше, посредством формул (1.8). Подчеркнём, что и значения y_j , и значения x_j размещаются в одном и том же массиве \mathbf{x} .

Рассмотрим **пример** решения СЛАУ $Ax = b$ методом LU -разложения с частичным выбором ведущего элемента. Дано:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{pmatrix}, \quad b = \begin{pmatrix} 10 \\ 28 \\ 47 \end{pmatrix}.$$

Построим разложение (1.11). Скопируем матрицу A в массив \mathbf{LU} .

Пусть $j = 0$. Вычисления на *этапе 1* не проводятся, перейдём сразу к *этапу 2*:

$$i = 0, \quad u_{00} = a_{00} = 1;$$

$$i = 1, \quad u_{10} = a_{10} = 4;$$

$$i = 2, \quad u_{20} = a_{20} = 7.$$

Здесь массив \mathbf{LU} не претерпевает изменений.

Этап 3. Выберем ведущий элемент u_{t0} :

$$|u_{t0}| = \max_{i \geq 0} |u_{i0}| = 7, \quad t = 2.$$

Таким образом, $u_{20} = 7$ – ведущий элемент; номер строки, содержащей ведущий элемент, запишем в массив \mathbf{iP} : $\mathbf{iP}[0] = t = 2$. Так как $t \neq j$, меняем местами строки под номерами $j = 0$ и $t = 2$:

$$\mathbf{LU} = \begin{pmatrix} 7 & 8 & 10 \\ 4 & 5 & 6 \\ 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} \tilde{u}_{00} & a_{01} & a_{02} \\ \tilde{u}_{10} & a_{11} & a_{12} \\ \tilde{u}_{20} & a_{21} & a_{22} \end{pmatrix}.$$

Этап 4. Вычислим поддиагональные элементы:

$$i = 1, \quad l_{10} = \tilde{u}_{10} / \tilde{u}_{00} = 4 / 7;$$

$$i = 2, \quad l_{20} = \tilde{u}_{20} / \tilde{u}_{00} = 1 / 7.$$

Массив \mathbf{LU} принимает вид:

$$\mathbf{LU} = \begin{pmatrix} 7 & 8 & 10 \\ 4/7 & 5 & 6 \\ 1/7 & 2 & 3 \end{pmatrix} = \begin{pmatrix} u_{00} & a_{01} & a_{02} \\ l_{10} & a_{11} & a_{12} \\ l_{20} & a_{21} & a_{22} \end{pmatrix}.$$

Переходим к следующему шагу алгоритма. Теперь $j = 1$.

Этап 1:

$$i = 0, \quad u_{01} = a_{01} = 8.$$

Этап 2:

$$i = 1, \quad u_{11} = a_{11} - l_{10}u_{01} = 5 - \frac{4}{7} \cdot 8 = \frac{3}{7};$$

$$i = 2, \quad u_{21} = a_{21} - l_{20}u_{01} = 2 - \frac{1}{7} \cdot 8 = \frac{6}{7}.$$

Массив \mathbf{LU} принимает вид:

$$\mathbf{LU} = \begin{pmatrix} 7 & 8 & 10 \\ 4/7 & 3/7 & 6 \\ 1/7 & 6/7 & 3 \end{pmatrix} = \begin{pmatrix} u_{00} & u_{01} & a_{02} \\ l_{10} & u_{11} & a_{12} \\ l_{20} & u_{21} & a_{22} \end{pmatrix}.$$

Этап 3. Выберем ведущий элемент u_{t1} :

$$|u_{t1}| = \max_{i \geq 1} |u_{i1}| = \frac{6}{7}, \quad t = 2.$$

Таким образом, $u_{21} = 6/7$ – ведущий элемент; номер содержащей его строки запишем в массив \mathbf{iP} : $\mathbf{iP}[1] = t = 2$. Так как $t \neq j$, меняем местами строки \mathbf{LU} с $j = 1$ и $t = 2$:

$$\mathbf{LU} = \begin{pmatrix} 7 & 8 & 10 \\ 1/7 & 6/7 & 3 \\ 4/7 & 3/7 & 6 \end{pmatrix} = \begin{pmatrix} u_{00} & u_{01} & a_{02} \\ l_{10} & \tilde{u}_{11} & a_{12} \\ l_{20} & \tilde{u}_{21} & a_{22} \end{pmatrix}.$$

Этап 4. Вычислим поддиагональный элемент рассматриваемого столбца:

$$i = 2, \quad l_{21} = \tilde{u}_{21} / \tilde{u}_{11} = \frac{3}{7} \cdot \frac{7}{6} = \frac{1}{2}.$$

Массив \mathbf{LU} принимает вид:

$$\mathbf{LU} = \begin{pmatrix} 7 & 8 & 10 \\ 1/7 & 6/7 & 3 \\ 4/7 & 1/2 & 6 \end{pmatrix} = \begin{pmatrix} u_{00} & u_{01} & a_{02} \\ l_{10} & u_{11} & a_{12} \\ l_{20} & l_{21} & a_{22} \end{pmatrix}.$$

Полагаем теперь $j = 2$; формируем последний столбец массива \mathbf{LU} .

Этап 1:

$$i = 0, \quad u_{02} = a_{02} = 10;$$

$$i = 1, \quad u_{12} = a_{12} - l_{20}u_{02} = 3 - \frac{1}{7} \cdot 10 = \frac{11}{7}.$$

Этап 2:

$$i = 2, \quad u_{22} = a_{22} - l_{20}u_{02} - l_{21}u_{12} = 6 - \frac{4}{7} \cdot 10 - \frac{1}{2} \cdot \frac{11}{7} = -\frac{1}{2}.$$

Массив \mathbf{LU} принимает вид:

$$\mathbf{LU} = \begin{pmatrix} 7 & 8 & 10 \\ 4/7 & 3/7 & 11/7 \\ 1/7 & 6/7 & -1/2 \end{pmatrix} = \begin{pmatrix} u_{00} & u_{01} & u_{02} \\ l_{10} & u_{11} & u_{12} \\ l_{20} & l_{21} & u_{22} \end{pmatrix}.$$

Этап 3. Выбор ведущего элемента u_{t2} :

$$|u_{t2}| = \max_{i \geq 2} |u_{i2}| = \frac{1}{2}, \quad t = 2.$$

Таким образом $u_{22} = -1/2$ – ведущий элемент; номер содержащей его строки запишем в массив \mathbf{iP} : $\mathbf{iP}[2] = t = 2$. Так как $t = j = 2$, перестановку строк массива \mathbf{LU} не производим. Отметим, что на завершающем шаге алгоритма выбор ведущего элемента тривиален, так как он сводится к сортировке массива из одного элемента, из-за чего перестановка строк \mathbf{LU} не производится.

Вычисления *этапа 4* не выполняем.

Разложение (1.11) построено:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 4/7 & 1 & 0 \\ 1/7 & 6/7 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 7 & 8 & 10 \\ 0 & 3/7 & 11/7 \\ 0 & 0 & -1/2 \end{pmatrix}.$$

Массив перестановок имеет вид:

$$\mathbf{iP} = \{2 \ 2 \ 2\}.$$

Перейдём к решению СЛАУ $Ly = Pb$.

Скопируем элементы столбца b в массив \mathbf{X} :

$$\mathbf{x} = \{10 \ 28 \ 47\}.$$

Полагаем $j = 0$:

$$\begin{aligned} \mathbf{t} &= \mathbf{iP}[0] = 2; \\ \mathbf{s} &= \mathbf{x}[\mathbf{t}] = 47; \\ \mathbf{x}[\mathbf{t}] &= \mathbf{x}[0] = 10; \\ \mathbf{x}[0] &= \mathbf{s} = 47. \end{aligned}$$

Массив \mathbf{X} принимает вид:

$$\mathbf{x} = \{47 \ 28 \ 10\}.$$

Пусть теперь $j = 1$:

$$\begin{aligned} \mathbf{t} &= \mathbf{iP}[1] = 2; \\ \mathbf{s} &= \mathbf{x}[\mathbf{t}] = 10; \\ \mathbf{x}[\mathbf{t}] &= \mathbf{x}[1] = 28; \\ \mathbf{x}[1] &= \mathbf{s} - l_{10} \cdot \mathbf{x}[0] = 10 - \frac{1}{7} \cdot 47 = \frac{23}{7}. \end{aligned}$$

Массив \mathbf{x} принимает вид:

$$\mathbf{x} = \{47 \quad 23/7 \quad 28\}.$$

Пусть, наконец, $j = 2$:

$$\mathbf{t} = \mathbf{iP}[2] = 2;$$

$$\mathbf{s} = \mathbf{x}[\mathbf{t}] = 28;$$

$$\mathbf{x}[\mathbf{t}] = \mathbf{x}[2] = 28;$$

$$\mathbf{x}[2] = \mathbf{s} - l_{20} \cdot \mathbf{x}[0] - l_{21} \cdot \mathbf{x}[1] = 28 - \frac{4}{7} \cdot 47 - \frac{1}{2} \cdot \frac{23}{7} = -\frac{1}{2}.$$

Массив \mathbf{x} принимает вид:

$$\mathbf{x} = \{47 \quad 23/7 \quad -1/2\}.$$

Таким образом, $y = (47 \quad 23/7 \quad -1/2)^T$.

Решим систему $Ux = y$, используя формулу (1.8):

$$\begin{cases} 7x_0 + 8x_1 + 10x_2 = 47, \\ \frac{6}{7}x_1 + \frac{11}{7}x_2 = \frac{23}{7}, \\ -\frac{1}{2}x_2 = -\frac{1}{2}; \end{cases} \Rightarrow x = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}.$$

Сравним решение задачи, полученное с помощью изложенного алгоритма, с результатами вычислений в среде **Scilab**.

Введём исходные данные и основные результаты решения задачи, полученные выше:

```
--> A = [1 2 3; 4 5 6; 7 8 10];
```

```
--> b = [10; 28; 47];
```

```
--> L_ = [1 0 0; 1/7 1 0; 4/7 1/2 1];
```

```
--> U_ = [7 8 10; 0 6/7 11/7; 0 0 -1/2];
```

```
--> y_ = [47; 23/7; -1/2];
```

```
--> x_ = [3; 2; 1];
```

В системе **Scilab** имеется [5] стандартная функция **lu**, которая предназначена для построения LU -разложения с частичным выбором ведущего элемента и возвращает матрицы L , U и P . Воспользуемся ей:


```

--> [L, U, P] = lu(A)
P =
    0.    0.    1.
    1.    0.    0.
    0.    1.    0.
U =
    7.    8.          10.
    0.    0.8571429   1.5714286
    0.    0.          - 0.5
L =
    1.          0.    0.
    0.1428571   1.    0.
    0.5714286   0.5   1.

```

Проверим точность построения разложения (1.11), вычислив норму разности $PA-LU$:

```

--> norm(P*A - L*U, %inf)
ans =
    0.

```

Сравним теперь элементы матриц L , U , вычисленных по изложенному выше алгоритму (массивы $L_$ и $U_$) и функцией lu (массивы L и U). Вычислим разности массивов L и $L_$, U и $U_$:

```

--> norm(L - L_, %inf)
ans =
    2.220D-16
--> norm(U - U_, %inf)
ans =
    3.331D-16

```

Указанные отличия элементов массивов L и $L_$, U и $U_$ вызваны ошибками округления при выполнении арифметических операций на компьютере.

Для решения системы $Ly=b$ воспользуемся функцией $lmd1$ (см. раздел 1.1):

```

--> y = lmd1(L, P*b); y'
ans =
    47.    3.2857143   - 0.5

```

Это решение с точностью до погрешностей, вызванных ошибками округления, совпадает с полученным ранее столбцом y (массив y). Об этом свидетельствует норма разности y и y_* :

```
--> norm(y - y_*, %inf)
ans =
    3.553D-15
```

Для решения системы $L^T x = y$ воспользуемся функцией `lmdu` (см. раздел 1.1):

```
--> x = lmdu(U, y); x'
ans =
    3.    2.    1.
```

Элементы массива x , выведенные на экран в среде **Scilab**, округлены и на самом деле отличаются от элементов массива x_* , в котором хранится точное решение СЛАУ. В этом можно убедиться, вычислив норму их разности `err`:

```
--> err = norm(x - x_*, %inf)
err =
    1.243D-14
```

Следует отметить, что операция левого матричного деления (`\`) в **Scilab** реализована именно методом LU -разложения с частичным выбором ведущего элемента (если число обусловленности матрицы меньше $4,5 \cdot 10^{14}$). Построим решение рассматриваемой СЛАУ (массив x_{bs}), используя эту операцию, и определим его точность:

```
--> x_bs = A\b; x_bs'
ans =
    3.    2.    1.
--> norm(x_bs - x_*, %inf)
ans =
    8.882D-16
```

Таким образом, результаты решения СЛАУ, полученные тремя разными способами, совпадают с точностью до погрешностей, вызванных ошибками округления.

В завершении данного раздела отметим, что и метод Холецкого, и метод LU -разложения эксплуатировали одну и ту же идею: системы линейных уравнений с треугольными матрицами решаются просто. Но есть и другие просто решаемые системы; о них будет рассказано в следующей главе.

Контрольные вопросы и задания

1. Сформулируйте определения симметричного, антисимметричного и ортогонального операторов.

2. Сформулируйте определение симметричной, диагональной, ортогональной, верхней и нижней треугольных матриц.

3. Выпишите: а) все антисимметричные матрицы 1-го порядка; б) все ортогональные матрицы 1-го порядка.

4. Выпишите все матрицы 2-го порядка, которые одновременно антисимметричны и ортогональны.

5. Сформулируйте определение матрицы перестановок. Что такое матрица транспозиции?

6. Выпишите все матрицы перестановок 1-го и 2-го порядка.

7. Объясните, почему в факторизованном представлении матрицы перестановок P порядка n сомножитель P_{n-1} всегда равен I .

8. Что такое матричное разложение Холецкого? Запишите расчётные соотношения для построения этого разложения. Каков алгоритм решения СЛАУ методом Холецкого? Каковы условия применимости этого метода?

9. Проверьте, являются ли матрицы

$$\frac{1}{3} \begin{pmatrix} 2 & 2 & -1 \\ 2 & -1 & 2 \\ -1 & 2 & 2 \end{pmatrix}, \quad \frac{1}{5} \begin{pmatrix} 4 & 3 & 0 \\ 0 & 0 & -5 \\ -3 & 4 & 0 \end{pmatrix}, \quad \frac{1}{5} \begin{pmatrix} 4 & 0 & 3 \\ 0 & 5 & 0 \\ 3 & 0 & 4 \end{pmatrix}$$

ортогональными. Есть ли среди них собственные ортогональные матрицы? Можно ли построить разложение Холецкого каждой из перечисленных матриц?

10. Что такое LU -разложение квадратной матрицы? Запишите расчётные соотношения для построения этого разложения. Каков алгоритм решения СЛАУ методом LU -разложения? Каковы условия применимости этого метода?

11. Что такое матрица с диагональным преобладанием? Сформулируйте теорему Леви – Деспланка.

12. Покажите, что при умножении произвольной матрицы A на любое ненулевое число λ число обусловленности матрицы не меняется.

13. Опишите алгоритм решения СЛАУ методом LU -разложения с частичным выбором ведущего элемента. Каков алгоритм построения указанного матричного разложения? Каковы условия применимости метода LU -разложения с частичным выбором ведущего элемента?

14. Решите СЛАУ $Ax = b$, где

$$A = \begin{pmatrix} 4 & -4 & 8 & -4 \\ -4 & 5 & -9 & 5 \\ 8 & -9 & 33 & 7 \\ -4 & 5 & 7 & 22 \end{pmatrix}, \quad b = \begin{pmatrix} -4 \\ 5 \\ 7 \\ 23 \end{pmatrix},$$

методами Холецкого, LU -разложения без выбора и с частичным выбором ведущего элемента. Почему второй из перечисленных методов можно применять в рассматриваемой задаче? Является ли матрица этой системы матрицей с диагональным преобладанием?

15. Решите СЛАУ $Ax = b$, где

$$A = \begin{pmatrix} 100 & 20 & -10 & 10 \\ -20 & 116 & 22 & 8 \\ 10 & 14 & 101 & -8 \\ 20 & 16 & -10 & 104 \end{pmatrix}, \quad b = \begin{pmatrix} -110 \\ 172 \\ 214 \\ -128 \end{pmatrix},$$

методами LU -разложения без выбора и с частичным выбором ведущего элемента. Является ли матрица этой системы матрицей с диагональным преобладанием? Почему первый метод из перечисленных можно применять в рассматриваемой задаче?

2. ОРТОГОНАЛЬНЫЕ РАЗЛОЖЕНИЯ

2.1. QR -разложение: метод вращений

Метод QR -разложения основан [1] на следующем разложении квадратной матрицы A :

$$A = QR, \quad (2.1)$$

где Q – ортогональная, а R – правая треугольная матрицы (последняя будет невырожденной, если невырождена сама матрица A , что мы далее и предполагаем).

При наличии такого разложения для матрицы системы $Ax = b$ последнюю решают по схеме $y = b \setminus Q$, $x = y \setminus R$. В силу (1.6) первая опе-

рация реализуется умножением столбца b на матрицу Q^T ; вторая же операция, как обычно, реализуется посредством формул (1.8), только роль матрицы U здесь играет матрица R .

Обсудим коротко порядок получения разложения (2.1) (подробности можно найти в [3]). Здесь обычно используют [1] два подхода: *метод вращений* и *метод отражений*.

В методе вращений процесс получения матрицы R можно описать как следующую цепочку преобразований (здесь $R_{00} = A$, $R_{n-1, n-2} = R$):

$$R_{00} \rightarrow R_{10} \rightarrow R_{20} \rightarrow \dots \rightarrow R_{n-1,0} \rightarrow R_{21} \rightarrow \dots \rightarrow R_{ij} \rightarrow \dots \rightarrow R_{n-1, n-2},$$

где:

- 1) j меняется от 0 до $n-2$;
- 2) i меняется от $j+1$ до $n-1$;
- 3) каждая последующая матрица $R_{ij} \equiv R^{\text{нов}}$ получается из предыдущей $R^{\text{стар}}$ умножением её на матрицу Гивенса G_{ij} :

$$R^{\text{нов}} = G_{ij} R^{\text{стар}}. \quad (1.16)$$

При этом **матрицей Гивенса** (или **матрицей плоского вращения**) для заданной пары индексов ij ($i > j$) называется ортогональная матрица $G_{ij} \in SO(n, \mathbb{R})$, которая отличается от единичной матрицы I следующими особенностями: 1) значения диагональных элементов с индексами jj и ii равны не 1, а некоторому числу c ; 2) значения внедиагональных элементов с индексами ji и ij равны не 0, а соответственно $(-s)$ и s , где число s взято таким, что $c^2 + s^2 = 1$ (так что c и s можно принять за косинус и синус некоторого угла φ – *угла вращения*).

Для каждого преобразования (2.2) в цепочке угол вращения выбирают так, чтобы $r_{ij}^{\text{нов}} = 0$. Для этого достаточно вычислить значение

$$r = (d \geq 0 ? \sqrt{d^2 + p^2} : -\sqrt{d^2 + p^2}),$$

где d и p – элементы матрицы $R^{\text{стар}}$ с индексами jj и ij , после чего принять

$$c = d/r, \quad s = -p/r$$

(в случае $r = 0$ полагают $c=1$, $s=0$; тогда $G_{ij} = I$).

Матрица Q при этом совпадает с произведением

$$G_{10}^T \cdots G_{n-1, n-2}^T;$$

на практике либо её вычисляют по рекуррентным формулам, либо вообще не используют, работая непосредственно с наборами чисел c и s , задающими матрицы Гивенса.

Рассмотрим **пример**. Требуется решить СЛАУ $Ax = b$, используя QR -разложение, построенное методом вращений. Дано:

$$A = \begin{pmatrix} 0 & 16 & -14 \\ -5\sqrt{2} & 9\sqrt{2} & -6\sqrt{2} \\ -5\sqrt{2} & 21\sqrt{2} & -4\sqrt{2} \end{pmatrix}, \quad b = \begin{pmatrix} -12 \\ 2\sqrt{2} \\ 18\sqrt{2} \end{pmatrix}.$$

Построим разложение (2.1) методом вращений. Матрица R вычисляется по столбцам путём обнуления поддиагональных элементов умножениями на матрицы плоских вращений G_{ij} .

1) Пусть $j=0, i=1$. Положим $R^{\text{стар}} = A$.

Вычисляем матрицу G_{10} . Найдём вспомогательные величины:

$$d = r_{jj}^{\text{стар}} = r_{00}^{\text{стар}} = 0,$$

$$p = r_{ij}^{\text{стар}} = r_{10}^{\text{стар}} = -5\sqrt{2}.$$

Поскольку $d=0$, то

$$r = \sqrt{d^2 + p^2} = \sqrt{25 \cdot 2} = 5\sqrt{2}.$$

Найдём косинус и синус угла вращения:

$$c = \frac{d}{r} = 0, \quad s = -\frac{p}{r} = -\frac{-5\sqrt{2}}{5\sqrt{2}} = 1.$$

Соответствующая матрица Гивенса имеет вид

$$G_{10} = \begin{pmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Вычислим матрицу $R^{\text{нов}}$:

$$R^{\text{нов}} = G_{10} R^{\text{стар}} = \begin{pmatrix} 5\sqrt{2} & -9\sqrt{2} & 6\sqrt{2} \\ 0 & 16 & -14 \\ -5\sqrt{2} & 21\sqrt{2} & -4\sqrt{2} \end{pmatrix}.$$

Элемент этой матрицы в позиции $i=1, j=0$ обнулён.

2) Пусть $j=0, i=2$. Положим $R^{\text{стар}} = R^{\text{нов}}$.

Вычисляем матрицу G_{20} . Найдём вспомогательные величины

$$d = r_{jj}^{\text{стар}} = r_{00}^{\text{стар}} = 5\sqrt{2},$$

$$p = r_{ij}^{\text{стар}} = r_{20}^{\text{стар}} = -5\sqrt{2}.$$

Поскольку $d > 0$, то

$$r = \sqrt{d^2 + p^2} = \sqrt{25 \cdot 2 + 25 \cdot 2} = 10.$$

Найдём косинус и синус угла вращения:

$$c = \frac{d}{r} = \frac{5\sqrt{2}}{10} = \frac{\sqrt{2}}{2}, \quad s = -\frac{p}{r} = -\frac{-5\sqrt{2}}{10} = \frac{\sqrt{2}}{2}.$$

Матрица Гивенса имеет вид

$$G_{20} = \begin{pmatrix} c & 0 & -s \\ 0 & 1 & 0 \\ s & 0 & c \end{pmatrix} = \begin{pmatrix} \sqrt{2}/2 & 0 & -\sqrt{2}/2 \\ 0 & 1 & 0 \\ \sqrt{2}/2 & 0 & \sqrt{2}/2 \end{pmatrix}.$$

Вычислим матрицу $R^{\text{нов}}$:

$$R^{\text{нов}} = G_{20}R^{\text{стар}} = \begin{pmatrix} 10 & 30 & 10 \\ 0 & 16 & -14 \\ 0 & 12 & 2 \end{pmatrix}.$$

Все поддиагональные элементы этой матрицы в столбце с индексом $j=0$ обнулены. Перейдём к обнулению поддиагонального элемента следующего столбца.

3) Пусть $j=1, i=2$. Положим $R^{\text{стар}} = R^{\text{нов}}$.

Построим матрицу G_{21} . Найдём вспомогательные величины

$$d = r_{jj}^{\text{стар}} = r_{11}^{\text{стар}} = 16,$$

$$p = r_{ij}^{\text{стар}} = r_{21}^{\text{стар}} = 12.$$

Поскольку $d > 0$, то

$$r = \sqrt{d^2 + p^2} = \sqrt{16^2 + 12^2} = 20.$$

Найдём косинус и синус угла вращения:

$$c = \frac{d}{r} = \frac{16}{20} = \frac{4}{5}, \quad s = -\frac{p}{r} = -\frac{12}{20} = -\frac{3}{4}.$$

Матрица Гивенса имеет вид

$$G_{21} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4/5 & 3/5 \\ 0 & -3/5 & 4/5 \end{pmatrix}.$$

Вычислим матрицу $R^{\text{нов}}$:

$$R^{\text{нов}} = G_{21}R^{\text{стар}} = \begin{pmatrix} 10 & 30 & 10 \\ 0 & 20 & -10 \\ 0 & 0 & 10 \end{pmatrix}.$$

Все поддиагональные элементы этой матрицы обнулены, так что разложение (2.1) построено, причём

$$R = R^{\text{нов}} = \begin{pmatrix} 10 & 30 & 10 \\ 0 & 20 & -10 \\ 0 & 0 & 10 \end{pmatrix}, \quad Q^T = G_{21}G_{20}G_{10}.$$

Вычислять матрицу Q не будем.

Решим СЛАУ $Qy = b$. В силу ортогональности Q ,

$$y = Q^T b = G_{21}G_{20}G_{10}b = G_{21}G_{20} \cdot \begin{pmatrix} -2\sqrt{2} \\ -12 \\ 18\sqrt{2} \end{pmatrix} = G_{21} \cdot \begin{pmatrix} -20 \\ -12 \\ 16 \end{pmatrix} = \begin{pmatrix} -20 \\ 0 \\ 20 \end{pmatrix}.$$

Решим систему $Rx = y$ с верхней треугольной матрицей, используя формулу (1.8):

$$\begin{cases} 10x_0 - 30x_1 + 10x_2 = -20, \\ 20x_1 - 10x_2 = 0, \\ 10x_2 = 20; \end{cases} \Rightarrow x = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}.$$

Завершим рассмотрение данного примера, построив в среде **Scilab** QR -разложение и сравнив его с полученными выше результатами.

Зададим матрицу СЛАУ:

```
--> s2 = sqrt(2);
--> A=[0 16 -14;-5*s2 9*s2 -6*s2;-5*s2 21*s2 -4*s2];
```

Результаты построения разложения (1.15), полученные выше методом вращений:

```
--> R_ = [10 -30 10; 0 20 -10; 0 0 10];
--> G10 = [0 -1 0; 1 0 0; 0 0 1];
--> G20 = [s2/2 0 -s2/2; 0 1 0; s2/2 0 s2/2];
```


Восстановим матрицу Q по известным матрицам Гивенса:

```
--> Q_ = (G21*G20*G10) '  
Q_ =  
  0.          0.8          - 0.6  
 - 0.7071068  - 0.4242641  - 0.5656854  
 - 0.7071068   0.4242641   0.5656854
```

Проверим точность построенного разложения, вычислив норму разности $A - QR$:

```
--> norm(A-Q_*R_, 1)  
ans =  
 3.553D-15
```

В системе **Scilab** имеется [5] стандартная функция **qr**, предназначенная для построения QR -разложения матрицы. Воспользуемся ей:

```
--> [Q, R] = qr(A)  
R =  
 - 10.    30.   - 10.  
  0.    - 20.    10.  
  0.     0.    10.  
Q =  
  0.          - 0.8          - 0.6  
  0.7071068   0.4242641  - 0.5656854  
  0.7071068  - 0.4242641   0.5656854
```

Проверим точность построенного функцией **qr** разложения, вычислив норму разности $A - QR$:

```
--> norm(A-Q*R, 1)  
ans =  
 2.132D-14
```

Обратите внимание, что столбцы Q и $Q_$, а также строки матриц R и $R_$ совпадают с точностью до знака (вычислительные погрешности мы не учитываем). Это – следствие неединственности QR -разложения (в алгоритме, изложенном в начале раздела, единственность достигалась за счёт определённого выбора угла вращения).

2.2. QR-разложение: метод отражений

Переходим к описанию метода отражений. Описанию предпослём несколько определений.

Пусть X – евклидово векторное пространство. **Тензорным произведением** двух векторов $\mathbf{a}, \mathbf{b} \in X$ называется линейный оператор $\mathbf{a} \otimes \mathbf{b}: X \rightarrow X$, задаваемый [10,18] формулой

$$\forall \mathbf{x} \in X \quad (\mathbf{a} \otimes \mathbf{b}) \mathbf{x} = (\mathbf{b}, \mathbf{x}) \mathbf{a} . \quad (2.3)$$

Не всякий линейный оператор можно представить в виде тензорного произведения векторов. Операторы, представимые таким образом, называют [19] **мультипликативными**. Если оба сомножителя произведения – ненулевые векторы, то ранг оператора равен 1, а иначе произведение оказывается нулевым оператором.

Заметим, что в определении пространство X предполагалось евклидовым. В действительности операцию тензорного произведения двух векторов можно определить и без такого предположения, но тогда произведение уже не будет линейным оператором; это будет объект другого типа – контравариантный тензор степени 2 [20].

При помощи операции тензорного произведения можно, в частности, записать явную формулу для оператора поворота абсолютно твёрдого тела, находящую многочисленные применения в механике. Пусть Γ и Γ° – операторы ориентации этого тела соответственно в текущей и некоторой фиксированной (*отсчётной*) его конфигурациях [11] (это – собственные ортогональные операторы). **Оператор поворота \mathbf{Q}** данного тела определяется равенством $\Gamma = \mathbf{Q} \Gamma^\circ$.

Предположим, что поворот происходит на угол φ вокруг оси с единичным вектором \mathbf{e} . Тогда для оператора поворота справедливо [18] следующее явное выражение:

$$\mathbf{Q} = \cos \varphi \cdot \mathbf{I} + (1 - \cos \varphi) \mathbf{e} \otimes \mathbf{e} + \sin \varphi \check{\mathbf{e}} . \quad (2.4)$$

Здесь \mathbf{I} – единичный оператор, а $\check{\mathbf{e}}$ – **присоединённый оператор** для вектора \mathbf{e} , т.е. антисимметричный линейный оператор, взаимно однозначно сопоставляемый данному вектору по формуле

$$\forall \mathbf{x} \quad \check{\mathbf{e}} \mathbf{x} = [\mathbf{e}, \mathbf{x}] . \quad (2.5)$$

Заметим, что операция перехода от вектора к присоединённому оператору представляет собой [20] результат применения к данному вектору отображения ad – *присоединённого представления* алгебры вектор-

ов трёхмерного евклидова векторного пространства, которое является изоморфизмом этой алгебры Ли и алгебры Ли действующих в данном пространстве антисимметричных операторов.

В силу тождества $\mathbf{e} \otimes \mathbf{e} = \mathbf{I} + \check{\mathbf{e}}\check{\mathbf{e}}$ для тензорного квадрата единичного вектора формулу (2.4) можно также записать и в виде

$$\mathbf{Q} = \mathbf{I} + (1 - \cos\varphi)\check{\mathbf{e}}\check{\mathbf{e}} + \sin\varphi\check{\mathbf{e}}. \quad (2.6)$$

Вернёмся к пространствам произвольной размерности n . Если в таком пространстве X задать ортонормированный базис, то матрицей оператора $\mathbf{a} \otimes \mathbf{b}$ в этом базисе будет матрица $a \otimes b \equiv ab^T$ – тензорное произведение столбцов a и b . Её элемент с индексами ij равен произведению $a_i b_j$.

Оператор $\mathbf{P}: X \rightarrow X$ называется **проектором**, если $\mathbf{P}^2 \equiv \mathbf{P}\mathbf{P} = \mathbf{P}$; говорят, что \mathbf{P} есть проектор **на** подпространство $L \subset X$, если образ этого оператора совпадает с L . Если при этом L ортогонально ядру оператора, то такой оператор называется **ортогональным проектором** [15].

Нужно подчеркнуть, что ортогональный проектор *не является* ортогональным оператором (если только он не совпадает с \mathbf{I}), зато это – всегда симметричный оператор.

Пусть теперь $\mathbf{u} \in X$ – произвольный ненулевой вектор, направленный вдоль прямой D и ортогональный гиперплоскости P , $D \perp P$; тогда ортогональные проекторы на D и P даются формулами

$$\mathbf{P}_1 = \frac{\mathbf{u} \otimes \mathbf{u}}{(\mathbf{u}, \mathbf{u})} \quad \text{и} \quad \mathbf{P}_2 = \mathbf{I} - \frac{\mathbf{u} \otimes \mathbf{u}}{(\mathbf{u}, \mathbf{u})}. \quad (2.7)$$

Из формул (2.7) видно, что в сумме оба проектора составляют единичный оператор. Рассмотрим их разность $\mathbf{P}_2 - \mathbf{P}_1$.

Оператор

$$\mathbf{H} = \mathbf{I} - \frac{2}{(\mathbf{u}, \mathbf{u})} \mathbf{u} \otimes \mathbf{u} \quad (2.9)$$

называется **оператором отражения** относительно гиперплоскости P , а вектор \mathbf{u} – **вектором отражения**.

Название связано с тем, что при умножении на \mathbf{H} произвольного вектора \mathbf{x} та составляющая последнего, которая пропорциональна \mathbf{u} , меняет знак, а составляющая, ортогональная \mathbf{u} , не изменяется.

Матрица оператора \mathbf{H} в ортонормированном базисе

$$H = I - \frac{2}{(u, u)} u \otimes u \equiv I - \frac{2}{(u, u)} u u^T \quad (2.9)$$

называется **матрицей отражения** относительно гиперплоскости P , или [3] **матрицей Хаусхолдера** (столбец u компонент вектора \mathbf{u} обычно также именуют «вектором отражения»). Оператор \mathbf{H} и матрица H одновременно ортогональны и симметричны, причём определитель матрицы H равен -1 .

Рациональный порядок действий при умножении столбца a на матрицу Хаусхолдера сводится к последовательному выполнению трёх операций:

$$K = \frac{(u, u)}{2}, \quad \lambda = \frac{(u, a)}{K}, \quad a^* = a - \lambda u \quad (2.10)$$

(это быстрее, чем действовать по обычному правилу умножения матрицы на столбец; если же столбцов несколько, то число K достаточно вычислить один раз). Через a^* здесь обозначен столбец Ha . В силу последней из формул (2.10) i -й элемент столбца a после умножения на H не изменяется, если i -й элемент вектора отражения u равен нулю.

Решим теперь следующую задачу. Для столбца $a \in \mathbb{R}^n$ надо найти такую матрицу Хаусхолдера H , что при умножении на неё столбца a останутся неизменными несколько его элементов, стоящих на заранее заданных позициях, а остальные элементы (кроме одного с заранее заданным номером j) станут равными нулю.

Фактически задача сводится к нахождению подходящего вектора отражения u . Положим его элементы с номерами, соответствующими оставляемым не изменившимися элементами столбца a , равными нулю, и вычислим сумму квадратов S' тех элементов столбца a , которые должны быть обнулены.

Теперь рассмотрим два случая. Первый из них (*исключительный*) характеризуется тем, что одновременно $S' = 0$ и $a_j = 0$.

В данной ситуации при выборе вектора отражения имеется значительный произвол. Для определённости полагают равными нулю все его элементы, кроме j -го, а последний полагают равным 1.

В противном случае (*основной* случай) вычисляем:

$$\sigma = (a_j \geq 0 ? 1 : -1), \quad S = \sigma \sqrt{a_j^2 + S'}$$

после чего элементы вектора отражения u с номерами k , соответствующими обнуляемым элементам столбца a , полагаем равными их исходным значениям: $u_k = a_k$, а элемент u_j полагаем равным $a_j + S$.

Число σ здесь – знак j -го элемента столбца a (впрочем, если он равен нулю, то σ всё равно полагают отличным от нуля). Знак у S выбран так, чтобы избежать роста вычислительной погрешности в случае вычитания близких чисел при вычислении u_j .

Вектор отражения найден, причём его конкретный выбор позволяет упростить вычисления по формулам (2.10). Именно, нетрудно показать, что в рассматриваемом случае $\lambda = 1$, а формулы принимают вид

$$K = S(S + a_j), \quad a^* = a - u. \quad (2.11)$$

(число K следует вычислять, если предстоит умножать на H и другие столбцы, для которых, разумеется, уже $\lambda \neq 1$).

Проиллюстрируем решение задачи об обнулении элементов столбца на конкретном **примере**. Пусть требуется найти вектор отражения u такой, что при умножении на матрицу Хаусхолдера H обнуляются последние два элемента столбца $a = (1 \ 5 \ -1 \ 2 \ -2)^T$, а первые два остаются без изменения. Нужно найти произведения $a^* = Ha$ и $b^* = Hb$, где $b = (7 \ 3 \ 4 \ -2 \ 8)^T$.

Найдём сумму квадратов обнуляемых элементов столбца a :

$$S' = a_3^2 + a_4^2 = 2^2 + 2^2 = 8.$$

Постольку $a_2 < 0$, то $\sigma = -1$ и

$$S = \sigma \sqrt{a_2^2 + S'} = -\sqrt{1 + 8} = -3.$$

Сформируем вектор отражения u . Поскольку элементы a_0 и a_1 не подлежат изменению, то $u_0 = 0$, $u_1 = 0$. Другие элементы равны:

$$u_2 = a_2 + S = -1 - 3 = -4,$$

$$u_3 = a_3 = 2, \quad u_4 = a_4 = -2.$$

Найдём произведение $a^* = Ha$, используя формулы (2.11). Сначала вычисли вспомогательную величину

$$K = S(S + a_2) = S u_2 = 12.$$

Теперь применим последнюю из формул (2.11). Первые два элемента не претерпевают изменений (так как $u_0 = 0$, $u_1 = 0$):

$$a_0^* = a_0 - u_0 = a_0 = 1, \quad a_1^* = a_1 - u_1 = a_1 = 5.$$

Другие компоненты столбца a^* получаются такими:

$$a_2^* = a_2 - u_2 \equiv -S = 3,$$

$$a_3^* = a_3 - u_3 \equiv 0,$$

$$a_4^* = a_4 - u_4 \equiv 0.$$

Найдём теперь произведение $b^* = Hb$, используя формулы (2.10). Сначала вычислим вспомогательную величину

$$\lambda = \frac{(u, b)}{K} = \frac{u_2 b_2 + u_3 b_3 + u_4 b_4}{K} = \frac{-36}{12} = -3.$$

Применим теперь последнюю из формул (2.10). Первые два элемента столбца b^* не претерпевают изменений (так как $u_0 = 0$, $u_1 = 0$):

$$b_0^* = b_0 - \lambda u_0 = b_0 = 7, \quad b_1^* = b_1 - \lambda u_1 = b_1 = 3.$$

Другие компоненты столбца b^* получаются следующими:

$$b_2^* = b_2 - \lambda u_2 = 4 + 3 \cdot (-4) = -8,$$

$$b_3^* = b_3 - \lambda u_3 = -2 + 3 \cdot 2 = 4,$$

$$b_4^* = b_4 - \lambda u_4 = 8 + 3 \cdot (-2) = 2.$$

Решение примера завершено.

Переходим к рассмотрению процесса получения QR -разложения матрицы A методом отражений. Процесс получения матрицы R этим методом можно описать как следующую цепочку преобразований (здесь $R_{-1} = A$, $R_{n-2} = R$):

$$R_{-1} \rightarrow R_0 \rightarrow \dots \rightarrow R_{n-2},$$

где

$$R_j = H_j R_{j-1},$$

а матрицу Хаусхолдера H_j выбирают так же, как и в рассмотренной выше задаче об обнулении элементов столбца – чтобы обнулить поддиагональные элементы j -го столбца матрицы R_{j-1} .

Матрица Q при этом совпадает с произведением

$$H_0 \cdot \dots \cdot H_{n-2};$$

на практике либо её вычисляют по рекуррентным формулам, либо вообще не используют, работая непосредственно с векторами отражения.

Рассмотрим **пример**. Построим разложение (2.1) для матрицы

$$A = \begin{pmatrix} 10 & -170 & 60 \\ -40 & 104 & 174 \\ 80 & -28 & 282 \end{pmatrix}.$$

В процессе решения задачи построим матрицу $V = Q^T$, используя рекуррентную формулу

$$V_j = H_j V_{j-1}, \quad j = 0, 1;$$

$$V = V_1, \quad V_{-1} = I.$$

Далее обозначаем: $R^{\text{стар}} = R_{j-1}$, $R^{\text{нов}} = R_j$, $V^{\text{стар}} = V_{j-1}$, $V^{\text{нов}} = V_j$.

1) Обнулیم поддиагональные элементы столбца с индексом $j = 0$. Положим $R^{\text{стар}} = A$, $V^{\text{стар}} = I$. На этом шаге изменениям подвергнутся все элементы матриц $R^{\text{стар}}$ и $V^{\text{стар}}$.

Найдём сумму квадратов обнуляемых элементов рассматриваемого столбца:

$$S' = (r_{10}^{\text{стар}})^2 + (r_{20}^{\text{стар}})^2 = (-40)^2 + (80)^2 = 8000.$$

Постольку $r_{00} > 0$, то $\sigma = 1$ и

$$S = \sigma \sqrt{(r_{00}^{\text{стар}})^2 + S'} = \sqrt{100 + 8000} = 90.$$

Сформируем вектор отражения u :

$$u_0 = r_{00}^{\text{стар}} + S = 10 + 90 = 100,$$

$$u_1 = r_{10}^{\text{стар}} = -40, \quad u_2 = r_{20}^{\text{стар}} = 80.$$

По формуле (2.11) найдём вспомогательную величину

$$K = S (S + r_{00}^{\text{стар}}) = S u_0 = 9000.$$

Вычислим матрицу $R^{\text{нов}} = H_0 R^{\text{стар}}$. Столбец с индексом 0 формируется согласно последней из формул (2.11):

$$r_{00}^{\text{нов}} = r_{00}^{\text{стар}} - u_0 \equiv -S = -90,$$

$$r_{10}^{\text{нов}} = r_{10}^{\text{стар}} - u_1 \equiv 0, \quad r_{20}^{\text{нов}} = r_{20}^{\text{стар}} - u_2 \equiv 0.$$

Поддиагональные элементы рассматриваемого столбца обнулены.

Перейдём к расчёту оставшихся элементов по формуле (2.10).

Сформируем столбец $R^{\text{нов}}$ с индексом 1. Вычислим вспомогательную величину

$$\lambda = \frac{r_{01}^{\text{стар}} u_0 + r_{11}^{\text{стар}} u_1 + r_{21}^{\text{стар}} u_2}{K} = \frac{(-170) \cdot 100 + 104 \cdot (-40) + (-28) \cdot 80}{9000} = -\frac{26}{10}.$$

В соответствии с (2.10), имеем:

$$r_{01}^{\text{нов}} = r_{01}^{\text{стар}} - \lambda u_0 = -170 - \frac{-26}{10} \cdot 100 = 90,$$

$$r_{11}^{\text{нов}} = r_{11}^{\text{стар}} - \lambda u_1 = 104 - \frac{-26}{10} \cdot (-40) = 0,$$

$$r_{21}^{\text{нов}} = r_{21}^{\text{стар}} - \lambda u_2 = -28 - \frac{-26}{10} \cdot 80 = 180.$$

Теперь сформируем столбец $R^{\text{нов}}$ с индексом 2:

$$\lambda = \frac{r_{02}^{\text{стар}} u_0 + r_{12}^{\text{стар}} u_1 + r_{22}^{\text{стар}} u_2}{K} = \frac{60 \cdot 100 + 174 \cdot (-40) + 282 \cdot 80}{9000} = \frac{24}{10};$$

$$r_{02}^{\text{нов}} = r_{02}^{\text{стар}} - \lambda u_0 = 60 - \frac{24}{10} \cdot 100 = -180,$$

$$r_{12}^{\text{нов}} = r_{12}^{\text{стар}} - \lambda u_1 = 174 - \frac{24}{10} \cdot (-40) = 270,$$

$$r_{22}^{\text{нов}} = r_{22}^{\text{стар}} - \lambda u_2 = 282 - \frac{24}{10} \cdot 80 = 90.$$

Матрица $R^{\text{нов}}$ сформирована. Перейдём к построению матрицы $V^{\text{нов}}$. На данном шаге алгоритма эта матрица совпадает с матрицей Хаусхолдера H_0 и, следовательно, является симметричной:

$$V^{\text{нов}} = H_0 V^{\text{стар}} = H_0.$$

Ограничимся вычислением её диагональных и поддиагональных элементов.

Сформируем столбец $V^{\text{нов}}$ с индексом 0 в соответствии с (2.10):

$$\lambda = \frac{v_{00}^{\text{стар}} u_0 + v_{10}^{\text{стар}} u_1 + v_{20}^{\text{стар}} u_2}{K} = \frac{1 \cdot 100 + 0 + 0}{9000} = \frac{1}{90};$$

$$v_{00}^{\text{нов}} = v_{00}^{\text{стар}} - \lambda u_0 = 1 - \frac{1}{90} \cdot 100 = -\frac{1}{9},$$

$$v_{10}^{\text{нов}} = v_{10}^{\text{стар}} - \lambda u_1 = 0 - \frac{1}{90} \cdot (-40) = \frac{4}{9},$$

$$v_{20}^{\text{нов}} = v_{20}^{\text{стар}} - \lambda u_2 = 0 - \frac{1}{90} \cdot 80 = -\frac{8}{9}.$$

Аналогично найдём поддиагональные элементы столбца $V^{\text{нов}}$ с индексом 1:

$$\lambda = \frac{v_{01}^{\text{стар}} u_0 + v_{11}^{\text{стар}} u_1 + v_{21}^{\text{стар}} u_2}{K} = \frac{0 - 1 \cdot 40 + 0}{9000} = -\frac{4}{900};$$

$$v_{11}^{\text{нов}} = v_{11}^{\text{стар}} - \lambda u_1 = 1 - \frac{-4}{900} \cdot (-40) = \frac{37}{45},$$

$$v_{21}^{\text{нов}} = v_{21}^{\text{стар}} - \lambda u_2 = 0 - \frac{-4}{900} \cdot 80 = \frac{16}{45}.$$

Завершим построение матрицы $V^{\text{нов}}$, вычислив диагональный элемент последнего столбца:

$$\lambda = \frac{v_{02}^{\text{стар}} u_0 + v_{12}^{\text{стар}} u_1 + v_{22}^{\text{стар}} u_2}{K} = \frac{0 + 0 + 1 \cdot 80}{9000} = \frac{8}{900};$$

$$v_{22}^{\text{нов}} = v_{22}^{\text{стар}} - \lambda u_2 = 1 - \frac{8}{900} \cdot 80 = \frac{13}{45}.$$

Итак, один шаг алгоритма выполнен,

$$R^{\text{нов}} = \begin{pmatrix} -90 & 90 & -180 \\ 0 & 0 & 270 \\ 0 & 180 & 90 \end{pmatrix}, \quad V^{\text{нов}} = \begin{pmatrix} -1/9 & 4/9 & -8/9 \\ 4/9 & 37/45 & 16/45 \\ -8/9 & 16/45 & 13/45 \end{pmatrix}.$$

2) Положим $R^{\text{стар}} = R^{\text{нов}}$, $V^{\text{стар}} = V^{\text{нов}}$. Обнулим поддиагональный элемент столбца $R^{\text{стар}}$ с индексом $j=1$. На данном шаге алгоритма наддиагональный элемент $r_{01}^{\text{стар}}$ не подвергнется изменению, следовательно, компонента вектора отражения $u_0=0$. Таким образом, все элементы матриц $R^{\text{стар}}$ и $V^{\text{стар}}$, находящиеся в строках с индексом 0, также не изменятся.

Найдём сумму квадратов обнуляемых элементов рассматриваемого столбца:

$$S' = (r_{21}^{\text{стар}})^2 = 180^2.$$

Постольку $r_{11}=0$, то $\sigma=1$ и

$$S = \sigma \sqrt{(r_{00}^{\text{стар}})^2 + S'} = \sqrt{0 + 180^2} = 180.$$

Сформируем вектор отражения u :

$$u_0 = 0,$$

$$u_1 = r_{11}^{\text{стар}} + S = 180,$$

$$u_2 = r_{21}^{\text{стар}} = 180.$$

По первой из формул (2.11) найдём вспомогательную величину

$$K = S(S + r_{11}^{\text{стар}}) = S u_1 = 180^2 = 32400.$$

Вычислим матрицу $R^{\text{нов}} = H_1 R^{\text{стар}}$. При выполнении умножения строка $R^{\text{стар}}$ с индексом 0 не изменится. Далее вычисляем элементы строк 1 и 2 матрицы $R^{\text{нов}}$.

Поскольку скалярное произведение вектора отражения на нулевой столбец $R^{\text{стар}}$ равно нулю, то указанный столбец не изменится.

Элементы столбца с индексом 1 формируются согласно второй из формул (2.11):

$$r_{11}^{\text{нов}} = r_{11}^{\text{стар}} - u_1 \equiv -S = -180,$$

$$r_{21}^{\text{нов}} = r_{21}^{\text{стар}} - u_2 \equiv 0.$$

Поддиагональный элемент рассматриваемого столбца обнулён.

Перейдём к расчёту оставшихся элементов по формулам (2.10).

Вычислим элементы столбца $R^{\text{нов}}$ с индексом 2. Найдём вспомогательную величину

$$\lambda = \frac{r_{12}^{\text{стар}} u_1 + r_{22}^{\text{стар}}}{K} = \frac{180 \cdot 270 + 180 \cdot 90}{32400} = 2;$$

$$r_{12}^{\text{нов}} = r_{12}^{\text{стар}} - \lambda u_1 = 270 - 2 \cdot 180 = -90,$$

$$r_{22}^{\text{нов}} = r_{22}^{\text{стар}} - \lambda u_2 = 90 - 2 \cdot 180 = -270.$$

Матрица $R^{\text{нов}}$ сформирована. Перейдём к построению $V^{\text{нов}}$:

$$V^{\text{нов}} = H_1 V^{\text{стар}}.$$

Здесь также только вычисляем элементы строк 1 и 2 матрицы $V^{\text{нов}}$.

Сформируем столбец $V^{\text{нов}}$ с индексом 0 в соответствии с (2.10):

$$\lambda = \frac{v_{10}^{\text{стар}} u_1 + v_{20}^{\text{стар}} u_2}{K} = \frac{1}{32400} \left(\frac{4}{9} \cdot 180 - \frac{8}{9} \cdot 180 \right) = -\frac{1}{405};$$

$$v_{10}^{\text{нов}} = v_{10}^{\text{стар}} - \lambda u_1 = \frac{4}{9} + \frac{1}{405} \cdot 180 = \frac{8}{9},$$

$$v_{20}^{\text{нов}} = v_{20}^{\text{стар}} - \lambda u_2 = -\frac{8}{9} + \frac{1}{405} \cdot 180 = -\frac{4}{9}.$$

Вычислим столбец $V^{\text{нов}}$ с индексом 1 в соответствии с (2.11):

$$\lambda = \frac{v_{11}^{\text{стар}} u_1 + v_{21}^{\text{стар}} u_2}{K} = \frac{1}{32400} \left(\frac{37}{45} \cdot 180 + \frac{16}{45} \cdot 180 \right) = \frac{53}{8100};$$

$$v_{11}^{\text{нов}} = v_{11}^{\text{стар}} - \lambda u_1 = \frac{37}{45} - \frac{53}{8100} \cdot 180 = -\frac{16}{45},$$

$$v_{21}^{\text{нов}} = v_{21}^{\text{стар}} - \lambda u_2 = \frac{16}{45} - \frac{53}{8100} \cdot 180 = -\frac{37}{45}.$$

Завершим построение матрицы $V^{\text{нов}}$ и найдём элементы столбца с индексом 2:

$$\lambda = \frac{v_{02}^{\text{стар}} u_0 + v_{12}^{\text{стар}} u_1 + v_{22}^{\text{стар}} u_2}{K} = \frac{1}{32400} \left(\frac{16}{45} \cdot 180 + \frac{13}{45} \cdot 180 \right) = \frac{29}{8100};$$

$$v_{12}^{\text{нов}} = v_{12}^{\text{стар}} - \lambda u_1 = \frac{16}{45} - \frac{29}{8100} \cdot 180 = -\frac{13}{45},$$

$$v_{22}^{\text{нов}} = v_{22}^{\text{стар}} - \lambda u_2 = \frac{13}{45} - \frac{29}{8100} \cdot 80 = -\frac{16}{45}.$$

Последний шаг алгоритма выполнен, и разложение (2.1) построено:

$$R = R^{\text{нов}} = \begin{pmatrix} -90 & 90 & -180 \\ 0 & -180 & -90 \\ 0 & 0 & -270 \end{pmatrix}, \quad Q^T = V^{\text{нов}} = \begin{pmatrix} -1/9 & 4/9 & -8/9 \\ 8/9 & -16/45 & -13/45 \\ -4/9 & -37/45 & -16/45 \end{pmatrix}.$$

Упражнение. Самостоятельно проверьте правильность построения разложения, вычислив в среде **Scilab** разность $A - QR$. Постройте разложение (2.1), используя функцию **qr**.

Для алгоритма QR -разложения число используемых арифметических операций составляет [15] $2n^3 + O(n^2)$ при использовании матриц Гивенса и $4/3 n^3 + O(n^2)$ при использовании матриц Хаусхолдера.

Таким образом, алгоритм QR -разложения на основе вращений Гивенса работает в 3 раза медленнее, чем алгоритм LU -разложения, а алгоритм на основе отражений Хаусхолдера – в 2 раза медленнее.

В то же время при решении СЛАУ высокой размерности с помощью метода LU -разложения иногда (как показывает практика, крайне редко) возможна [3] существенная потеря точности, связанная с сильным ростом элементов промежуточных матриц. Применение же QR -разложения гарантированно предотвращает появление численной неустойчивости (это его достоинство основано на сохранении 2-нормы матрицы при умножении на ортогональные матрицы) [21].

Контрольные вопросы и задания

1. Что такое QR -разложение квадратной матрицы? Какова схема решения СЛАУ методом QR -разложения?

2. Выпишите все варианты QR -разложения для матрицы 1-го порядка $A = (a)$.

3. Как выглядит матрица плоского вращения (матрица Гивенса)?

4. В чём состоит основная идея метода вращений? Приведите основные расчётные соотношения этого метода.

5. Что собой представляет тензорное произведение двух векторов в евклидовом пространстве?

6. Что такое оператор поворота? Запишите явные формулы для оператора поворота в трёхмерном пространстве.

7. На основе формул (2.4) и (2.6) покажите, что матрица оператора поворота вокруг вектора e в трёхмерном пространстве является матрицей Гивенса, если столбцы компонент вектора в ортонормированном базисе равны (рассмотреть три варианта):

$$e = (1 \ 0 \ 0)^T, \quad e = (0 \ -1 \ 0)^T, \quad e = (0 \ 0 \ 1)^T.$$

Замечание. Матрица присоединённого оператора \check{e} имеет вид:

$$\check{e} = \begin{pmatrix} 0 & -e_2 & e_1 \\ e_2 & 0 & -e_0 \\ -e_1 & e_0 & 0 \end{pmatrix}.$$

8. Можно ли единичную матрицу 2-го порядка представить в виде тензорного произведения двух столбцов?

9. Какие линейные операторы называют проекторами, ортогональными проекторами и операторами отражения? Что такое матрица отражения (матрица Хаусхолдера)?

10. Пусть \mathbf{H} – оператор отражения относительно плоскости в трёхмерном пространстве, ортогональному единичному вектору e – вектору отражения. Докажите, что $\mathbf{H} = -\mathbf{Q}$, где \mathbf{Q} – оператор поворота вокруг вектора e на угол $\varphi = \pi$.

11. Какова основная идея метода отражений? Приведите основные расчётные соотношения этого метода.

12. Используя методы вращений и отражений, постройте QR -разложения матрицы

$$A = \begin{pmatrix} -6 & 6 & -36 \\ -8 & 8 & 2 \\ 0 & -20 & -10 \end{pmatrix}.$$

13. Решите СЛАУ $Ax = b$, где

$$A = \begin{pmatrix} 15 & -60 & 54 & -39 \\ -15 & -20 & -8 & -17 \\ 15 & -40 & -34 & 29 \\ -15 & 40 & -92 & 7 \end{pmatrix}, \quad b = \begin{pmatrix} -39 \\ -7 \\ 49 \\ 77 \end{pmatrix},$$

методом QR -разложения. Постройте последнее методами вращений и отражений.

3. СИНГУЛЯРНОЕ РАЗЛОЖЕНИЕ И МЕТОД НАИМЕНЬШИХ КВАДРАТОВ

3.1. SVD -разложение

Пусть $A \in \mathbb{R}^{m \times n}$. Метод SVD -разложения (аббревиатура происходит от английского названия *singular value decomposition* ‘разложение по сингулярным числам’) основан на следующем разложении матрицы:

$$A = USV^T, \quad (3.1)$$

где $U \in O(m, \mathbb{R})$, $V \in O(n, \mathbb{R})$ – ортогональные матрицы, $S \in \mathbb{R}^{m \times n}$ – диагональная матрица, диагональные элементы s_{jj} неотрицательны и далее обозначаются σ_j [22].

Числа σ_j – **сингулярные числа** матрицы A .

Нумеруются они так: $\sigma_0, \sigma_1, \dots, \sigma_{p-1}$, где $p = \min(m, n)$.

Можно показать, что для любой матрицы SVD -разложение (называемое также [1] **сингулярным разложением**) существует. Единственности нет; однако сами сингулярные числа определяются однозначно (с точностью до нумерации).

Действительно, матрицы $AA^T \in \mathbb{R}^{m \times m}$ и $A^T A \in \mathbb{R}^{n \times n}$ симметричны и положительно полуопределены; из представления (3.1) вытекает, что $AA^T = U(SS^T)U^{-1}$ и $A^T A = V(S^T S)V^{-1}$.

Но матрицы $SS^T \in \mathbb{R}^{m \times m}$ и $S^T S \in \mathbb{R}^{n \times n}$ – диагональные, а их диагональные элементы есть квадраты сингулярных чисел, дополненные, быть может, нулями (для той из этих матриц, у которой порядок больше). Полученные для матриц AA^T и $A^T A$ равенства означают, что они приводятся преобразованиями подобия к диагональному виду, а диагональные элементы матриц SS^T и $S^T S$ есть собственные значения матриц AA^T и $A^T A$. Значит, сингулярные числа матрицы A – квадратные корни из этих значений [21].

Набор сингулярных чисел матрицы содержит важную информацию о ней. Так, её ранг равен количеству ненулевых сингулярных чисел, а 2-норма равна наибольшему из сингулярных чисел.

Если матрица A – квадратная порядка n и невырожденная, то при наличии разложения (3.1) для матрицы системы $Ax = b$ последнюю решают по схеме $y = b \setminus U$, $z = y \setminus S$, $x = z \setminus V^T$. Первая и третья операции реализуются умножением столбцов b и z на матрицы U^T и V ; вторая же операция сводится к серии делений: $z_j = y_j / \sigma_j$.

Рассмотрим **пример** построения сингулярного разложения в среде **Scilab**. Исходная матрица:

$$A = \begin{pmatrix} 30 & -36 & -48 \\ 40 & 27 & 36 \end{pmatrix}.$$

Разложение (3.1) выполним, используя стандартную функцию **svd** [5]:

```
--> A = [30 -36 -48; 40 27 36];
--> [U, S, V] = svd(A)
V =
- 1.665D-16      1.          0.
  0.6            3.331D-16   - 0.8
  0.8            3.331D-16    0.6
S =
 75.    0.    0.
  0.    50.    0.
U =
- 0.8    0.6
  0.6    0.8
```

Таким образом, сингулярные числа матрицы A оказываются такими: $\sigma_0 = 75$, $\sigma_1 = 50$. Отметим, что функция **svd** располагает сингулярные числа в порядке убывания.

Проверим точность построения разложения, вычислив норму $A - USV^T$:

```
--> norm(A - U*S*V', 1)
ans =
    2.132D-14
```

Алгоритм получения разложения (3.1) достаточно сложен и здесь не приводится (его можно найти в [22]). Вычислительная трудоёмкость алгоритма *SVD*-разложения значительно выше, чем у других обсуждавшихся в данной главе алгоритмов. В качестве метода решения СЛАУ его применяют редко; однако построение *SVD*-разложения бывает нужно при решении некоторых других задач вычислительной линейной алгебры (о некоторых из них речь пойдёт в следующей главе).

В заключение подчеркнём следующую общую особенность методов решения СЛАУ, основанных на построении матричных разложений. Распространены задачи, в которых нужно решить несколько СЛАУ с различными правыми частями, но одной матрицей A . В таких задачах матричное разложение строится лишь один раз.

3.2. Применение матричных разложений для исследования свободных колебаний устойчивой линейной консервативной системы

Уравнения малых колебаний механической системы с n степенями свободы в окрестности изолированного устойчивого положения равновесия имеют вид [13]

$$A\ddot{x} + Cx = 0, \quad (3.2)$$

где $x = (x_0 \ \dots \ x_{n-1})^T$ – вектор отклонений обобщённых координат от положения равновесия, A – симметричная положительно определённая матрица инерционных коэффициентов, C – симметричная положительно определённая матрица жесткостей. Кинетическая и потенциальная энергии системы (3.2) равны, соответственно, $T = \dot{x}^T A \dot{x} / 2$, $\Pi = x^T C x / 2$.

Требуется построить решение $x(t)$ системы (3.2) с заданными начальными условиями $x(0)$ и $\dot{x}(0)$ методом нормальных координат [13]. Последние удовлетворяют системе независимых дифференциальных уравнений

$$\ddot{z}_i + \omega_i^2 z_i = 0, \quad i = 0, \dots, n-1, \quad (3.3)$$

где z_i – нормальная координата, $\omega_i > 0$ – i -я собственная частота (здесь их нумерация отличается от принятой в механике, где наименьшая собственная частота называется первой [13]).

Нормальные координаты совершают одночастотные колебания по закону

$$z(t) = z_i(0) \cos \omega_i t + \frac{\dot{z}_i(0)}{\omega_i} \sin \omega_i t. \quad (3.4)$$

Преобразование нормальных координат в исходные осуществляется с помощью нормированной модальной матрицы M по формуле

$$x = M z. \quad (3.5)$$

В рамках данного раздела мы построим модальную матрицу M , используя разложения (1.9) и (3.1).

Выполним разложение Холецкого инерционной матрицы $A = LL^T$ и сделаем замену переменных $y = L^T x$. Кинетическая и потенциальная энергии примут вид $T = \dot{y}^T \dot{y} / 2$, $\Pi = y^T C_y y / 2$, где матрица жесткостей C_y совпадает с $L^{-1} C (L^{-1})^T$. В силу симметрии C_y , она может быть найдена по формуле $C_y = (C \setminus L)^T \setminus L$.

Теперь выполним сингулярное разложение матрицы жесткостей C_y . Поскольку она симметрична и положительно определена, разложение (3.1) имеет вид $C_y = U S U^T$ (т.е. $U = V$). Сингулярные числа матрицы C_y совпадают с квадратами собственных частот системы (3.2):

$$S = \text{diag}(\omega_i^2).$$

Столбец нормальных координат получается с помощью преобразования $z = U^T y = U^T L^T x$. Действительно, в новых переменных, в силу ортогональности матрицы U , кинетическая и потенциальная энергии имеют вид $T = \dot{z}^T \dot{z} / 2$, $\Pi = z^T S z / 2$, а колебания описываются уравнением

$$\ddot{z} + S z = 0,$$

эквивалентным, из-за диагональности матрицы S , системе (3.4).

Таким образом, построено преобразование координат, обратное к (3.5); его матрица M^{-1} совпадает с $M_{\text{inv}} = U^T L^T$. В соответствии с этим, найдём нормированную модальную матрицу:

$$M = U \setminus L^T.$$

Отметим, что начальные условия для нормальных координат, используемые в (3.4), имеют вид:

$$z(0) = M_{\text{inv}} x(0), \quad \dot{z}(0) = M_{\text{inv}} \dot{x}(0). \quad (3.6)$$

Рассмотрим пример исследования собственных колебаний консервативной системы (3.2) с использованием средств **Scilab**. Приведённый ниже текст программы необходимо дополнить определением функций **lmdl** и **lmdu** (см. раздел 1.1) для выполнения левых матричных делений на треугольные матрицы. Далее предполагаем, что все рассматриваемые величины являются безразмерными.

Зададим инерционную матрицу и матрицу жесткостей:

$$\mathbf{A} = [14/3 \ 3/2 \ 0 \ ; \ 3/2 \ 1 \ 0 \ ; \ 0 \ 0 \ 4];$$

$$\mathbf{C} = [55.955 \ 2 \ -2 \ ; \ 2 \ 16.715 \ -2 \ ; \ -2 \ -2 \ 4];$$

Столбцы начальных условий по координатам и скоростям:

$$\mathbf{x}_0 = [0; \ 0.2; \ -0.3]; \ // \ x(0)$$

$$\mathbf{Dx}_0 = [-0.01; \ 0; \ 0.03]; \ // \ \dot{x}(0)$$

Приступим к нахождению нормированной модальной матрицы. Построим разложение Холецкого матрицы инерционной $A = LL^T$:

$$\mathbf{L} = \mathbf{chol}(\mathbf{A})';$$

Построим матрицу жесткостей $C_y = (C \setminus L)^T \setminus L$:

$$\mathbf{C}_y = \mathbf{lmdl}(\mathbf{L}, \mathbf{C}); \ \mathbf{C}_y = \mathbf{lmdl}(\mathbf{L}, \mathbf{C}_y');$$

Выполним сингулярное разложение матрицы C_y :

$$[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \mathbf{svd}(\mathbf{C}_y)$$

Результат разложения:

$$\mathbf{V} =$$

-	0.3034236	-	0.9492113	0.0832588
	0.9526946	-	0.3006050	0.0448281
-	0.0175233	0.0929221		0.9955192

$$\mathbf{S} =$$

44.250287	0.	0.
0.	8.779165	0.
0.	0.	0.9188241

$$\mathbf{U} =$$

-	0.3034236	-	0.9492113	0.0832588
	0.9526946	-	0.3006050	0.0448281
-	0.0175233	0.0929221		0.9955192

Обратите внимание, что матрицы V и U совпадают (насколько это возможно при выполнении расчёта на компьютере).

Квадраты собственных частот системы (3.2) совпадают с сингулярными числами матрицы C_y :

$$\omega_0^2 = 44,250287, \quad \omega_1^2 = 8,779165, \quad \omega_2^2 = 0,918841.$$

Найдём массив собственных частот:

```
omega = diag(S).^0.5
```

Вычислим нормированную модальную матрицу $M = U \setminus L^T$, воспользовавшись функцией `lmdu` (см. раздел 1.1), а также обратную модальную матрицу $M_{inv} = U^T L^T$:

```
M = lmdu(L',U); M_inv=U'*L';
```

Результат вычисления модальной матрицы M :

```
M =
- 0.5659908 - 0.3051304 0.0185183
1.3238803 - 0.4177258 0.0622939
- 0.0087617 0.0464610 0.4977596
```

Построим решение системы (3.2) при $t \in [0, T_1]$, где T_1 – период колебаний с наименьшей собственной частотой. Сформируем массив \mathbf{t} из 100 значений времени, равномерно распределённых на $[0, T_1]$:

```
T1 = 2*pi/omega(3,1); t = linspace(0, T1, 100);
```

Найдём начальные условия для нормальных координат и скоростей (3.6):

```
z_0 = M_inv*x_0; // z(0)
Dz_0 = M_inv*Dx_0; // z'(0)
```

Вычислим массивы значений нормальных координат z_0, z_1, z_2 на рассматриваемой временной сетке \mathbf{t} по формуле (3.4):

```
om = omega(1,1);
z0 = z_0(1,1)*cos(om*t) + Dz_0(1,1)/om*sin(om*t);
om = omega(2,1);
z1 = z_0(2,1)*cos(om*t) + Dz_0(2,1)/om*sin(om*t);
om = omega(3,1);
z2 = z_0(3,1)*cos(om*t) + Dz_0(3,1)/om*sin(om*t);
```

Вычислим значения координат x_0, x_1, x_2 :

```
x = M*[z0;z1;z2];
```

Массивы значений перечисленных координат составляют строки матрицы \mathbf{x} :

$$x_0(\mathbf{t}) = \mathbf{x}(1, :), \quad x_1(\mathbf{t}) = \mathbf{x}(2, :), \quad x_2(\mathbf{t}) = \mathbf{x}(3, :).$$

Построим графики полученных решений. Очистим графические окна 1 и 2:

```
clf([1,2]);
```

В окне 1 построим графики $x_0(t)$, $x_1(t)$, $x_2(t)$ (рис. 3.1):

```
scf(1);
plot(t, x(1,:), 'k-d');
plot(t, x(2,:), 'm^-');
plot(t, x(3,:), 'k-');
legend(['$x_0(t)$', '$x_1(t)$', '$x_2(t)$'], 2);
```

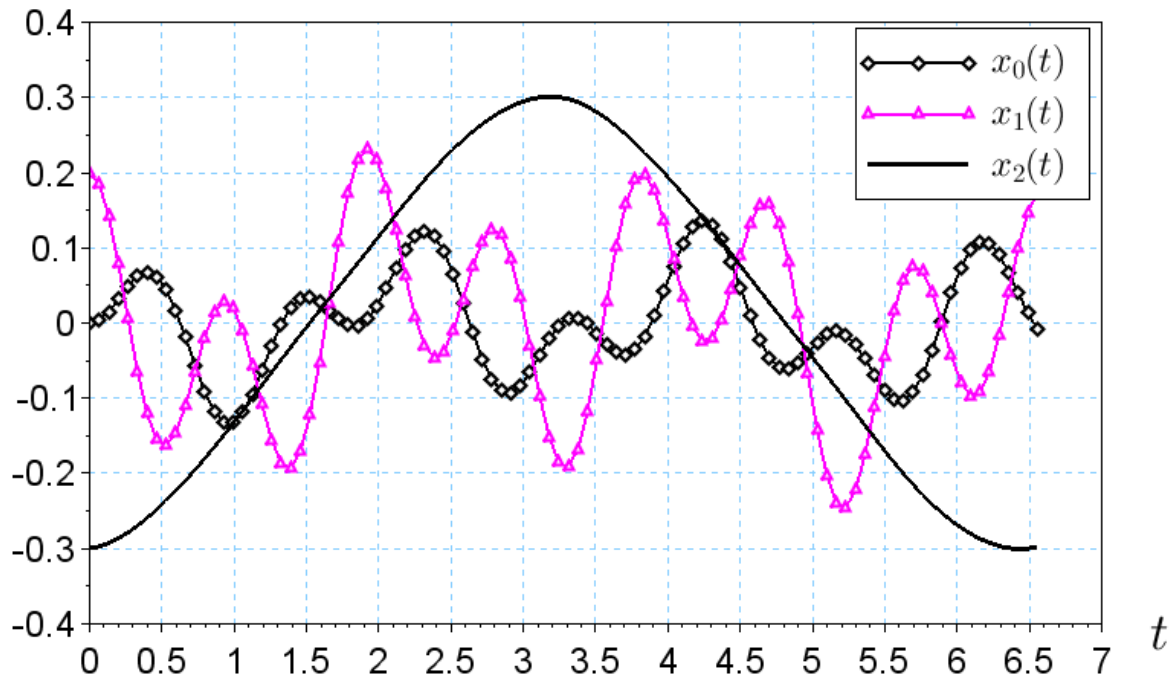


Рис. 3.1. Графики собственных колебаний консервативной системы

Исследуем точность полученных решений $x_0(t)$, $x_1(t)$, $x_2(t)$, подставив их в систему (3.2). Погрешность решения задачи характеризует столбец невязок правой и левой частей (3.2)

$$\varepsilon = A\ddot{x} + Cx.$$

Для вычисления столбца ускорений \ddot{x} (матрица Wx) найдём ускорения нормальных координат как $\ddot{z} = -Sz$ и умножим их на элементы модальной матрицы M :

```
Wz = -S*[z0;z1;z2]; Wx = M*Wz;
```

Вычислим значения невязок и найдём значения 2-норм столбца ε (массив `eps_norm`):

```
eps = A*Wx + C*x;
eps_norm = sqrt(eps(1,:) .^2 + eps(2,:) .^2 + eps(3,:) .^2);
```

Построим график $\|\varepsilon(t)\|$ в окне 2 (рис. 3.2):

```
scf(2); plot(t,eps_norm,'k-^');
legend(['\|\varepsilon(t)\|'], 1);
```

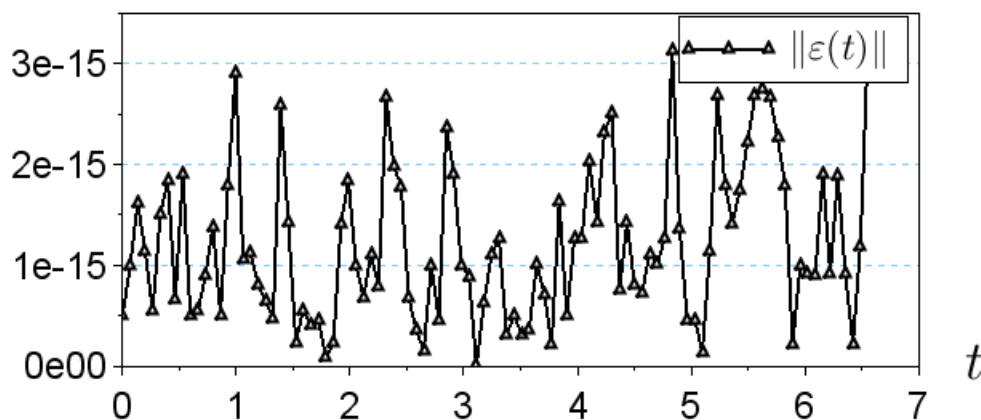


Рис. 3.2. График нормы невязки, характеризующий точность решений

Величины невязок (см. рис. 3.2) подтверждает правильность решения с точностью до вычислительных погрешностей ($\max \|\varepsilon\| < 4 \cdot 10^{-15}$).

3.3. Нормальные псевдорешения СЛАУ

Рассмотрим СЛАУ

$$Ax = b, \quad (3.7)$$

где $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Заметьте, что сейчас матрица A – не обязательно квадратная; а если она и квадратная, то предположений о её невырожденности не делается.

В этих условиях решение СЛАУ (3.7) может не существовать или же не быть единственным. Во многих задачах, однако, имеет смысл иное, обобщённое, понятие решения системы линейных уравнений. Приведём ряд определений, следуя [1,15].

Для любого столбца $x \in \mathbb{R}^n$ столбец $r = Ax - b \in \mathbb{R}^m$ называется **невязкой**.

Псевдорешением системы $Ax = b$ называют всякий столбец x , для которого 2-норма невязки минимальна:

$$\|Ax - b\|_2 \rightarrow \min,$$

т.е.

$$\|Ax - b\|_2^2 \equiv (Ax - b, Ax - b) \rightarrow \min.$$

В отличие от решений, псевдорешения всегда существуют, но их может быть много: они могут заполнять целую прямую, плоскость и так далее (в случае нулевой матрицы – всё пространство). Можно, однако, сформулировать простое условие, которое выделит из всего множества псевдорешений одно.

Нормальное псевдорешение системы $Ax = b$ – псевдорешение минимальной 2-нормы.

В случае совместности системы $Ax = b$ невязка псевдорешений равна нулю, и тогда это – обычные *решения* данной системы; нормальное псевдорешение в данном случае называют **нормальным решением**.

Справедлив следующий результат [15,22]: если $A = USV^T$ – *SVD*-разложение матрицы СЛАУ (см. раздел 3.1), σ_j – сингулярные числа этой матрицы, а $S^+ \in \mathbb{R}^{n \times m}$ – диагональная матрица, на диагонали которой стоят элементы

$$s_{jj}^+ = (\sigma_j > 0 ? 1/\sigma_j : 0) ,$$

то нормальное псевдорешение СЛАУ (3.7) можно найти по формулам

$$y = U^T b, \quad z = S^+ y, \quad x = Vz . \quad (3.8)$$

В соответствии с (3.8), для столбца x получаем такое выражение:

$$x = VS^+ U^T b . \quad (3.9)$$

Фигурирующая в (3.9) матрица $A^+ = VS^+ U^T$ называется **псевдообратной матрицей** для матрицы A . Таким образом, нормальное псевдорешение СЛАУ $Ax = b$ даётся [21] формулой $x = A^+ b$.

Для невырожденной квадратной матрицы псевдообратная матрица совпадает с обычной обратной матрицей; при этом рассмотренная выше матрица S^+ есть псевдообратная матрица для матрицы S .

При нахождении нормального псевдорешения нет необходимости вычислять псевдообратную матрицу в явном виде – удобнее использовать формулы (3.8). Однако нередко эта матрица и сама по себе представляет интерес.

Можно показать [15], что псевдообратная матрица A^+ – это единственная матрица X , удовлетворяющая следующим **уравнениям Пенроуза**:

$$(AX)^T = AX, \quad (XA)^T = XA, \quad AXA = A, \quad XAX = X ;$$

отсюда следует, что псевдообратная матрица определена *однозначно*.

Известно, что если СЛАУ $Ax = b$ совместна, то для её нормального решения при неточном задании исходных данных (вместо b задан столбец $b + \Delta b$) для относительной погрешности результата имеет место [22] оценка

$$\frac{\|\Delta x\|}{\|x\|} \leq \text{cond}^+ A \frac{\|\Delta b\|}{\|b\|};$$

здесь $\text{cond}^+ A = \|A\|_2 \cdot \|A^+\|_2$ – **нормальное число обусловленности**.

В случае несовместной СЛАУ подобная оценка (уже для псевдорешения) тоже существует, но имеет более сложный вид: в правую часть добавляется слагаемое, характеризующее меру несовместности системы.

Поскольку $\|A\|_2 = \sigma_{\max}$ и $\|A^+\|_2 = 1 / \sigma_{\min}$, где σ_{\max} и σ_{\min} – соответственно максимальное и минимальное из ненулевых сингулярных чисел, то $\text{cond}^+ A = \sigma_{\max} / \sigma_{\min}$.

Для *любой* (совместной или несовместной) СЛАУ $Ax = b$ невязка $r = Ax - b$ непрерывным образом зависит от элементов матрицы A ; ранг же её (а значит, и псевдорешение) есть разрывная функция элементов.

Данное обстоятельство следует учитывать при вычислении псевдорешений. Так, если среди сингулярных чисел есть нулевые, то за счёт неточного задания элементов матрицы A или за счёт ошибок округления вычисленные значения сингулярных чисел могут оказаться отличными от нуля. А тогда матрица S^+ и вместе с ней псевдорешение изменятся кардинальным образом.

С возникшим затруднением можно справиться так. Предположим, что от получаемого приближённого решения (или псевдорешения) СЛАУ требуется лишь малость невязки (соответственно, близость её к минимальному значению). Тогда разумно, найдя SVD -разложение, выделить среди ненулевых сингулярных чисел столь малые, что их отличие от нуля можно считать следствием различных погрешностей, и положить их равными нулю.

На практике задают [22] некоторый допуск τ и обнуляют те σ_j , которые меньше τ . Это соответствует переходу к матрицам меньшего ранга: от S к S_τ и от A – к $A_\tau \equiv US_\tau V^T$. При таком переходе невязка получаемого приближённого псевдорешения будет мало отличаться от невязки точного псевдорешения, но обусловленность задачи улучшится, так как будет выполнено неравенство: $\text{cond}^+ A_\tau \leq \sigma_{\max} / \tau$.

В результате по крайней мере те составляющие псевдорешения плохо обусловленной СЛАУ, которые можно найти с приличной точностью, будут с неплохой точностью найдены.

Помимо применения алгоритма *SVD*-разложения, известен другой способ нахождения псевдорешения – *метод нормальных уравнений*. **Нормальной системой уравнений** для СЛАУ $Ax = b$ называют СЛАУ

$$A^T Ax = A^T b ; \quad (3.10)$$

показано [1], что эта система всегда совместна и множество псевдорешений системы (3.7) совпадает с множеством решений СЛАУ (3.10).

На практике особенно важен случай, когда $m \geq n$ и ранг матрицы A совпадает с числом n её столбцов. В этом случае матрица $A^T A$ невырождена, являясь симметричной и положительно определённой. Поэтому нормальное псевдорешение СЛАУ (3.7) получается как результат численного решения системы (3.10) – например, методом Холецкого.

В данном частном случае для псевдообратной матрицы справедлива [15] следующая формула:

$$A^+ = (A^T A)^{-1} A^T ; \quad (3.11)$$

данная формула находит многочисленные применения в механике – например, в задачах кинематического анализа манипуляционных роботов [23] и при идентификации параметров механических систем [4].

В то же время метод нормальных уравнений не является универсальным. Если матрица A не слишком хорошо обусловлена, то при переходе к матрице $A^T A$ число обусловленности последней нередко оказывается примерно равным квадрату числа обусловленности исходной матрицы. Это ведёт к тому, что в решении будет потеряно вдвое больше значащих цифр, чем в методе *SVD*-разложения [3].

Ещё одним практически значимым классом СЛАУ (3.7) являются системы, содержащие меньше уравнений, чем неизвестных. Матрица A в этом случае имеет больше столбцов, чем строк ($m < n$). Покажем, что, если матрица A имеет полный ранг, то её псевдообращение может быть осуществлено по формуле

$$A^+ = A^T (A A^T)^{-1}. \quad (3.12)$$

Подставим в левую часть формулы (3.12) сингулярное разложение $A = USV^T$:

$$\begin{aligned} A^T (A A^T)^{-1} &= VS^T U^T (USV^T VS^T U^T)^{-1} = VS^T U^T (U S S^T U^T)^{-1} = \\ &= VS^T U^T (U^T)^{-1} (S S^T)^{-1} U^{-1} = VS^T (S S^T)^{-1} U^T. \end{aligned}$$

Учитывая, что здесь все сингулярные числа $\sigma_j > 0$, имеем:

$$S^T(SS^T)^{-1} = \begin{pmatrix} \sigma_0 & & 0 \\ & \ddots & \\ 0 & & \sigma_{m-1} \\ \vdots & & \vdots \end{pmatrix} \begin{pmatrix} \frac{1}{\sigma_0^2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sigma_{m-1}^2} \end{pmatrix} = \begin{pmatrix} 1/\sigma_0 & & 0 \\ & \ddots & \\ 0 & & 1/\sigma_{m-1} \\ \vdots & & \vdots \end{pmatrix} \equiv S^+.$$

Таким образом, $A^T(AA^T)^{-1} = VS^+U^T \equiv A^+$.

Упражнение. Действуя аналогично, докажите справедливость формулы (3.11).

Представление псевдообратной матрицы в виде (3.12) сводит построение нормального **решения** к расчёту по формуле $x = A^T z$, где $AA^T z = b$. Особенности решения последней СЛАУ аналогичны изложенным выше особенностям решения нормальных уравнений (3.10).

Теперь перейдём к рассмотрению прикладной задачи, в которой возникает потребность в нахождении нормального решения СЛАУ.

3.4. Применение матричных разложений для решения обратной задачи о скоростях избыточного манипулятора

Рассмотрим обратную задачу о скоростях плоского четырёхзвенного манипулятора, схема которого изображена на рис. 3.3. Звенья, образующие простую кинематическую цепь, соединены вращательными шарнирами. На четвёртом звене манипулятора расположен схват.

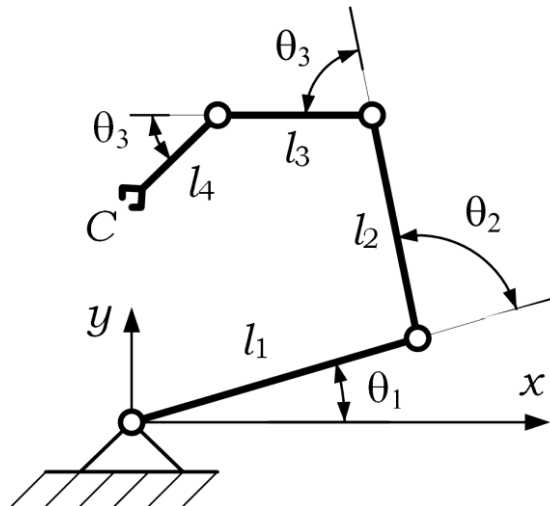


Рис. 3.3. Кинематическая схема манипулятора

Требуется решить обратную задачу о скоростях: по известным значениям проекций V_{C_x} и V_{C_y} скорости центра C схвата и угловой скорости Ω последнего определить обобщённые скорости манипулятора $\dot{\theta}_i$, $i=1, \dots, 4$. Текущие значения обобщённых координат θ_i , $i=1, \dots, 4$, также даны.

Поставленная задача сводится к нахождению столбца обобщённых скоростей \dot{q} путём решения СЛАУ

$$J(q) \cdot \dot{q} = u, \quad (3.13)$$

где $u = (V_{C_x} \quad V_{C_y} \quad \Omega)^T$, $q = (\theta_1 \dots \theta_4)^T$,

$$J = \left(\begin{array}{c|c|c|c} -l_1 s_1 - l_2 s_2 - l_3 s_3 - l_4 s_4 & -l_2 s_2 - l_3 s_3 - l_4 s_4 & -l_3 s_3 - l_4 s_4 & -l_4 s_4 \\ \hline l_1 c_1 + l_2 c_2 + l_3 c_3 + l_4 c_4 & l_2 c_2 + l_3 c_3 + l_4 c_4 & l_3 c_3 + l_4 c_4 & l_4 c_4 \\ \hline 1 & 1 & 1 & 1 \end{array} \right).$$

Здесь l_i , $i=1, \dots, 4$, – длина i -го звена манипулятора,

$$c_1 = \cos \theta_1, \quad c_2 = \cos(\theta_1 + \theta_2), \quad c_3 = \cos(\theta_1 + \theta_2 + \theta_3), \quad c_4 = \cos(\theta_1 + \dots + \theta_4),$$

$$s_1 = \sin \theta_1, \quad s_2 = \sin(\theta_1 + \theta_2), \quad s_3 = \sin(\theta_1 + \theta_2 + \theta_3), \quad s_4 = \sin(\theta_1 + \dots + \theta_4).$$

Далее полагаем, что конфигурация манипулятора – невырожденная [11], так что матрица J имеет полный ранг.

Решение обратной задачи о скоростях является неотъемлемой частью алгоритма управления манипулятором по вектору скорости [23]. В случае равных размерностей столбцов q и u решение СЛАУ (3.13) не вызывает трудностей: можно воспользоваться методами LU - или QR -разложений.

В рассматриваемом случае манипулятор обладает *кинематической избыточностью*: количество обобщённых координат больше числа параметров, определяющих программное движение исполнительного органа (схвата). Система кинематических уравнений (3.13) имеет бесконечно много решений, из которых мы выберем нормальное решение

$$\dot{q} = J^+ u,$$

для построения которого воспользуемся несколькими способами.

Исходные данные для решения задачи таковы:

$$\theta_1 = \pi/2, \quad \theta_2 = -\pi/2, \quad \theta_3 = \pi/2, \quad \theta_4 = -\pi/2; \quad l_i = 1 \text{ м}, \quad i=1, \dots, 4;$$

$$V_{C_x} = -1 \text{ м/с}, \quad V_{C_y} = 1 \text{ м/с}, \quad \Omega = 0.$$

Матрица СЛАУ (3.13) и столбец свободных членов равны:

$$J = \begin{pmatrix} -2 & -1 & -1 & 0 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \quad u = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}.$$

Необходимые расчёты проведём в среде **Scilab**. Программный код, приведённый ниже, следует дополнить определением функций **lmdl** и **lmdu** (см. раздел 1.1) для выполнения левого матричного деления на треугольные матрицы.

Введём исходные данные:

```
--> J = [-2 -1 -1 0; 2 2 1 1; 1 1 1 1];
--> u = [-1; 1; 0];
```

1. Построим нормальное решение с помощью сингулярного разложения $J = USV^T$. В рассматриваемой задаче удобнее пользоваться компактной формой *SVD*-разложения. В конкретном случае она имеет вид:

$$J = \tilde{U} \tilde{S} V^T,$$

где \tilde{S} – диагональная квадратная матрица порядка $m=3$, на диагонали которой расположены сингулярные числа; \tilde{U} состоит из первых $m=3$ столбцов матрицы U (остальные столбцы умножаются на нулевой блок матрицы S и на результат не влияют).

Можно доказать, что расчёт псевдообратной матрицы можно осуществить по формуле $J^+ = V \tilde{S}^+ \tilde{U}^T$, а нахождение нормального решения выполняется по формулам:

$$y = \tilde{U}^T u, \quad z = \tilde{S}^+ y, \quad \dot{q} = Vz. \quad (3.14)$$

Реализуем этот алгоритм в **Scilab**. Для получения сингулярного разложения в компактной форме воспользуемся функцией **svd** и зададим у неё второй аргумент "e":

```
--> [U_, S_, V] = svd(J, "e");
```

Нормальное решение \dot{q} (столбец **dq_svd**) в соответствии с (3.14):

```
--> y = U_'*u; z = y ./diag(S_);
```

```
--> dq_svd = V*z; dq_svd'
```

```
ans =
```

```
0.5    0.5   - 0.5   - 0.5
```

Таким образом, $\dot{\theta}_1 = \dot{\theta}_2 = -\dot{\theta}_3 = -\dot{\theta}_4 = 0,5$ рад/с.

Отметим, что в **Scilab** реализована функция **pinv**, вычисляющая псевдообратную матрицу с помощью сингулярного разложения:

```
--> ( pinv(J)*u ) '
ans =
0.5    0.5  - 0.5  - 0.5
```

2. Построим нормальное решение на основе представления псевдообратной матрицы в форме $J^+ = J^T(JJ^T)^{-1}$ с использованием разложения Холецкого.

Решим вспомогательную СЛАУ $JJ^T z = u$ методом Холецкого:

```
--> L = chol(J*J') ' ;
--> y = lmdl(L, u) ; // y = u \ L.
--> z = lmdu(L', y) ; // z = y \ L^T.
```

Нормальное решение $\dot{q} = J^T z$ (столбец **dq_ch**):

```
--> dq_ch = J' * z ; dq_ch '
ans =
0.5    0.5  - 0.5  - 0.5
```

3. Построим нормальное решение на основе представления псевдообратной матрицы в форме $J^+ = J^T(JJ^T)^{-1}$ с использованием **QR-разложения** $J^T = QR$.

Несмотря на то, что здесь матрицы J^T и R прямоугольные (строк больше, чем столбцов), принципиальных сложностей при обобщении рассмотренных ранее методов построения QR -разложения нет. В данной конкретной задаче удобнее пользоваться компактной формой QR -разложения:

$$J^T = \tilde{Q}\tilde{R}, \quad (3.15)$$

где \tilde{R} – квадратная верхняя треугольная матрица порядка $m=3$, состоящая из первых $m=3$ строк R ; \tilde{Q} состоит из первых $m=3$ столбцов матрицы Q (остальные столбцы умножаются на нулевой блок матрицы R и на результат не влияют).

Подстановка (3.15) в (3.14) даёт:

$$J^+ = \tilde{Q}(\tilde{R}^T)^{-1},$$

что позволяет найти нормальное решение СЛАУ (3.13) по формулам:

$$y = u \setminus \tilde{R}^T, \quad \dot{q} = \tilde{Q} y.$$

Реализуем этот алгоритм в **Scilab**. Для получения QR -разложения в компактной форме воспользуемся функцией **qr** и зададим у неё второй аргумент "e":

```
--> [Q_, R_] = qr(J', "e");
      Найдём нормальное решение  $\dot{q} = \tilde{Q} u$  (столбец dq_qr):
--> y = lmdl(R_', u); //  $y = u \backslash \tilde{R}^T$ .
--> dq_qr = Q_*y; dq_qr'
ans =
  0.5    0.5   -0.5   -0.5
```

Итак, результаты решения поставленной задачи тремя способами совпадают с точностью до вычислительных погрешностей.

В табл. 3.1 приведены нормы невязок и погрешности нормальных решений (последние сравнивались с точным решением задачи $\dot{q}^* = (1/2 \ 1/2 \ -1/2 \ -1/2)^T$).

Таблица 3.1

Сравнительный анализ решений, полученных на компьютере

Нормальные решения	$\dot{q} = \mathbf{dq_svd}$	$\dot{q} = \mathbf{dq_ch}$	$\dot{q} = \mathbf{dq_qr}$
Норма невязки $\ J\dot{q} - u\ _\infty$	$8,882 \cdot 10^{-16}$	$4,441 \cdot 10^{-16}$	$2,220 \cdot 10^{-16}$
Погрешность $\ \dot{q} - \dot{q}^*\ _\infty$	$2,220 \cdot 10^{-16}$	$4,441 \cdot 10^{-16}$	$1,665 \cdot 10^{-16}$

В рассматриваемом примере для всех трёх способов точности решений и невязки сопоставимы друг с другом (незначительное преимущество – у решения, полученного с помощью QR -разложения). Это объясняется тем, что матрица J не является плохо обусловленной. Действительно, её сингулярные числа (это – диагональные элементы матрицы \mathbf{S}_-) равны

$$\sigma_0 = 4,3344, \quad \sigma_1 = 1,0000, \quad \sigma_2 = 0,464,$$

а нормальное число обусловленности таково:

$$\text{cond}^+ A = \sigma_{\max} / \sigma_{\min} = \sigma_0 / \sigma_2 = 9,4.$$

3.5. Метод наименьших квадратов

Пусть две величины, x и y , связаны функциональной зависимостью: $y = f(x)$. Предполагается, что эта зависимость существует, но нам доступны лишь значения функции f при отдельных значениях аргумента, причём не точные, а найденные в серии экспериментов с определёнными погрешностями.

Обозначим через X_i , $i = 0, \dots, m-1$, упомянутые отдельные значения аргумента, а через Y_i – найденные экспериментально значения функции f в точках X_i (узлах).

Будем искать для функции f приближение в виде непрерывной функции F (аппроксиманта). Значения этой функции в узлах будут отличаться от значений Y_i ; обозначим $\varepsilon_i = F(X_i) - Y_i$.

Отклонения ε_i могут принимать как положительные, так и отрицательные значения. Чтобы не учитывать знаки, возведём каждое отклонение в квадрат и просуммируем по всем узлам:

$$\sum_{i=0}^{i < m} \varepsilon_i^2 \equiv \sum_{i=0}^{i < m} [F(X_i) - Y_i]^2 = Q. \quad (3.16)$$

Величину Q можно рассматривать как суммарную меру отклонения приближающей функции от исходной. Метод построения аппроксиманта F из условия минимизации величины Q называется **методом наименьших квадратов** (МНК) [21,22].

Ограничимся случаем *линейной задачи* МНК, когда аппроксимант выбирают в виде линейной комбинации

$$F(x) = c_0 \varphi_0(x) + c_1 \varphi_1(x) + \dots + c_{n-1} \varphi_{n-1}(x), \quad (3.17)$$

где $\varphi_0(x), \dots, \varphi_{n-1}(x)$ – *базисные функции*, которые предполагаются линейно независимыми в узлах. Считаем, что $n \leq m$.

Необходимым условием экстремума величины Q являются равенства

$$\frac{\partial Q}{\partial c_k} = 0, \quad k = 0, \dots, n-1;$$

вычисляя частные производные и сокращая на 2, получаем СЛАУ относительно неизвестных c_i

$$\sum_{i=0}^{i < m} [c_0 \varphi_0(X_i) + \dots + c_{n-1} \varphi_{n-1}(X_{n-1}) - Y_i] \varphi_0(X_i) = 0, \quad \dots \quad (3.18)$$

$$\sum_{i=0}^{i < m} [c_0 \varphi_0(X_i) + \dots + c_{n-1} \varphi_{n-1}(X_{n-1}) - Y_i] \varphi_{n-1}(X_i) = 0,$$

которая называется **нормальной системой МНК**. Матрицей этой СЛАУ служит матрица Грама системы базисных функций, которая симметрична и – при сделанных предположениях – положительно определена [22]. Её элемент с индексами kl представляет собой сумму

$$\sum_{i=0}^{i < m} \varphi_k(X_i) \varphi_l(X_i).$$

Введём в рассмотрение для рассматриваемой системы базисных функций **матрицу плана** – прямоугольную матрицу $A \in \mathbb{R}^{m \times n}$ с элементами $a_{ik} = \varphi_k(X_i)$. Тогда матрица системы (3.18) представима как произведение $A^T A$, а сама эта система принимает [22] вид

$$A^T A c = A^T Y, \quad (3.19)$$

где c и Y – столбцы из коэффициентов c_i и значений Y_i .

Таким образом, система (3.18) служит нормальной системой уравнений для СЛАУ $A c = Y$. Значит, решение линейной задачи МНК сводится либо к решению нормальной системы МНК (скажем, методом Холецкого), что разумно в случае хорошей обусловленности матрицы этой системы, либо (наиболее надёжный способ) к нахождению нормального псевдорешения СЛАУ $A c = Y$ методом *SVD*-разложения [3].

Заметим, что в литературе по вычислительной линейной алгебре линейная задача МНК нередко трактуется чисто алгебраически и отождествляется с задачей нахождения псевдорешения СЛАУ [1,3].

3.6. Определение характеристик колебаний с помощью метода наименьших квадратов

Рассмотрим задачу аппроксимации экспериментальных данных, полученных при наблюдении колебательного процесса, который с высокой точностью может быть описан законом

$$f(t) = c_0 + c_1 \cos t + c_2 \sin t + c_3 \cos 2t + c_4 \sin 2t, \quad (3.20)$$

где t – безразмерное время, играющее роль независимой переменной, $c_i, i = 0, \dots, 4$, – неизвестные постоянные коэффициенты.

Для оценки неизвестных параметров в выражении (3.20) воспользуемся методом наименьших квадратов. В качестве базисных функций выберем

$$\varphi_0(t) = 1, \quad \varphi_1(t) = \cos t, \quad \varphi_2(t) = \sin t, \quad \varphi_3(t) = \cos 2t, \quad \varphi_4(t) = \sin 2t.$$

Поставленную задачу решим в среде **Scilab** несколькими способами. Программный код, приведённый ниже, следует дополнить определением функций **lmdl** и **lmdu** (см. раздел 1.1) для выполнения левого матричного деления на треугольные матрицы.

Зададим исходные данные – массивы значений времени **time** и аппроксимируемой функции **data**=*Y*:

```
--> time = [0.47 1.20 1.93 2.66 3.39 4.12 4.85 ...
--> 5.58 6.31 7.04 7.77 8.50 9.23 9.96 10.69 ...
--> 11.42 12.15 12.88 13.61 14.34]';
--> data = [-0.29 -0.31 -0.29 -0.2 0.03 0.06 ...
--> 0.17 -0.02 -0.24 -0.39 -0.35 -0.21 -0.17 ...
--> 0.08 0.15 0.16 -0.08 -0.28 -0.35 -0.37]';
```

Вычислим столбцы значений базисных функций в узлах:

```
--> phi0 = ones(time);
--> phi1 = cos(time); phi2 = sin(time);
--> phi3 = cos(2*time); phi4 = sin(2*time);
```

Сформируем матрицу плана *A*:

```
--> A = [phi0, phi1, phi2, phi3, phi4];
```

1. Решим линейную задачу наименьших квадратов с помощью метода сингулярного разложения, а точнее – его компактной формы $A = \tilde{U} \tilde{S} V^T$ (см. раздел 3.4):

```
--> [U_, S_, V] = svd(A, "e"); // A = \tilde{U} \tilde{S} V^T
```

Анализ сингулярных чисел, расположенных на диагонали матрицы **S_**, позволяет сделать выводы об очень хорошей обусловленности задачи – максимальное и минимальное из них отличаются примерно в 1,5 раза:

```
--> diag(S_) '
ans =
    4.5609    3.3355    3.2696    2.9626    2.9335
```

Построим решение задачи МНК:

```
--> y = U_'*data; // y = \tilde{U}^T Y.
--> z = y ./diag(S_); // z = \tilde{S}^+ y.
```

Столбец искомых коэффициентов $c = Vz$ (массив **c_svd**):

```
--> c_svd = V*z; c_svd'
ans =
    - 0.1154    - 0.0643    - 0.2509    - 0.0307    - 0.0124
```

2. Найдём искомые коэффициенты, решив систему нормальных уравнений (3.19) методом Холецкого. Получим для матрицы $A^T A$ разложение Холецкого и найдём вспомогательный столбец $y = (A^T Y) \setminus L$:

```
--> L = chol(A'*A)'; y = lmdl(L, A'*data);
```

Столбец искомых коэффициентов $c = y \setminus L^T$ (массив **c_ch**):

```
--> c_ch = lmdu(L', y); c_ch'
```

```
ans =
```

```
- 0.1154 - 0.0643 - 0.2509 - 0.0307 - 0.0124
```

3. Решим задачу МНК с помощью QR -разложения, а точнее – его компактной формы $A = \tilde{Q}\tilde{R}$ (см. раздел 3.4). Подстановка последнего соотношения в формулу (3.11) позволяет найти псевдообратную матрицу в виде $A^+ = \tilde{R}^{-1}\tilde{Q}^T$. Таким образом, решение задачи МНК можно найти по формуле $c = A^+ Y = (\tilde{Q}^T Y) \setminus \tilde{R}$.

Построим компактную форму QR -разложения и найдём столбец c (массив **c_qr**):

```
--> [Q_, R_] = qr(A, "e"); // A = Q_ R_
```

```
--> c_qr = lmdu(R_, Q_'*data); c_qr'
```

```
ans =
```

```
- 0.1154 - 0.0643 - 0.2509 - 0.0307 - 0.0124
```

Итак, все три рассмотренных способа дают одинаковый (с точностью до вычислительных погрешностей) результат:

$c_0 = -0,1154$, $c_1 = -0,0643$, $c_2 = -0,2509$, $c_3 = -0,0307$, $c_4 = -0,0124$.

Проиллюстрируем результаты решения задачи, построив график полученного аппроксиманта. Зададим массив промежуточных точек **t** и вычислим в этих точках значения базисных функций:

```
--> t = linspace(0,15,150)';
```

```
--> AA = [ones(t) cos(t) sin(t) cos(2*t) sin(2*t)];
```

Построим график аппроксиманта $F(t)$, на который нанесём также исходные данные $Y_i = f(t_i)$ задачи (рис. 3.4):

```
--> plot(t, AA*c_svd, 'k'); plot(time, data, 'ko');
```

```
--> legend(["$F(t)$" "Y_i"], -1);
```

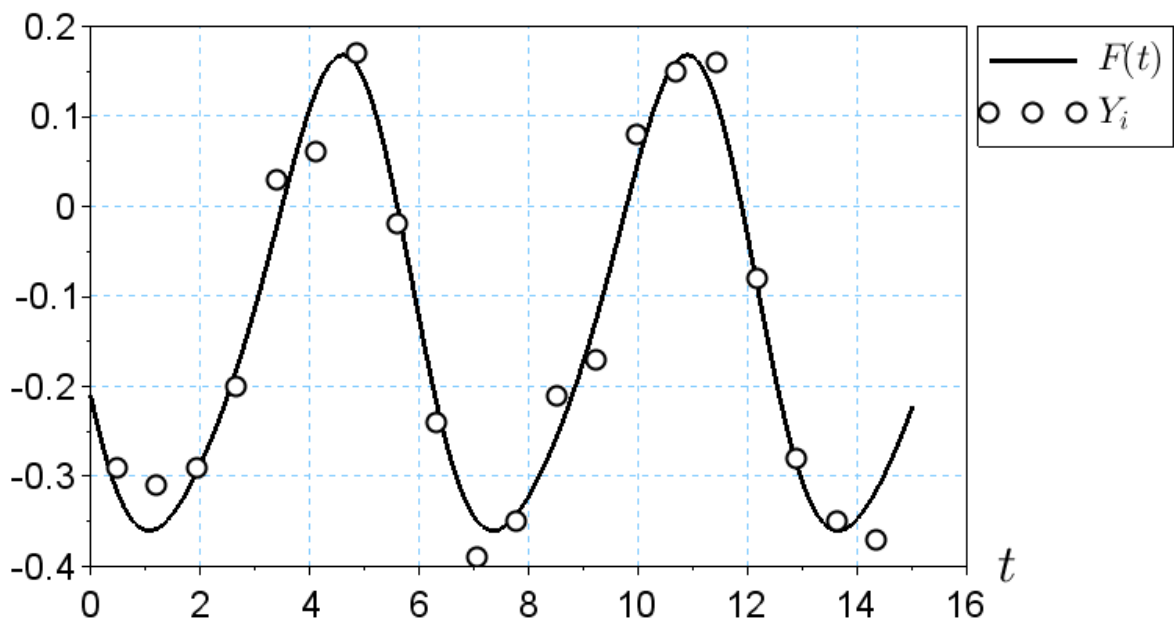



Рис. 3.4. График аппроксимирующей функции

Завершая данный раздел, отметим, что задачи определения характеристик периодической функции, подобные решённой здесь, возникают, например, при обработке экспериментальных данных о вынужденных колебаниях чувствительных элементов волновых твердотельных и микромеханических гироскопов. Построенные в результате экспериментов зависимости коэффициентов в выражениях вида (3.20) от частоты внешнего задающего воздействия используются для идентификации параметров математических моделей гироскопов (выполняемой, в том числе, и с применением МНК) [24].

Контрольные вопросы и задания

1. Что такое SVD -разложение матрицы? Для каких классов матриц оно существует? Каков порядок решения СЛАУ с помощью метода сингулярного разложения?
2. Выпишите все варианты SVD -разложения для матриц 1-го порядка: а) $A = (5)$; б) $A = (-5)$.
3. Что такое сингулярные числа матрицы A ? Какова их взаимосвязь с собственными значениями матриц $A^T A$ и AA^T ?
4. Дайте определения псевдорешения и нормального псевдорешения СЛАУ. В каком случае нормальное псевдорешение является нормальным решением СЛАУ?
5. Опишите алгоритм построения нормального псевдорешения методом SVD -разложения.

6. Что такое псевдообратная матрица? Запишите уравнения Пенроуза, которым удовлетворяет псевдообратная матрица. Однозначно ли определяется псевдообратная матрица?

7. Что такое нормальное число обусловленности? Каким образом оно используется для оценки точности решений и псевдорешений?

8. Каким образом можно улучшить вычислительную устойчивость нахождения псевдорешения плохо обусловленной СЛАУ?

9. Что такое нормальная система уравнений для СЛАУ?

10. Как, не прибегая к построению SVD -разложения, найти псевдорешение СЛАУ с матрицей полного ранга? Рассмотрите два случая: уравнений больше, чем неизвестных; неизвестных больше, чем уравнений. Всегда ли уместно применение этих способов?

11. В чём состоит линейная задача наименьших квадратов? Какие алгоритмы её решения вам известны?

12. Решите задачу МНК с исходными данными:

$$f(0) = 1, \quad f(1) = 3, \quad f(2) = 5, \quad f(3) = 7.$$

Постройте аппроксиманты

$$F(x) = c_0 + c_1x, \quad F(x) = c_0 + c_1x + c_2x^2.$$

Указание. Решить нормальную систему уравнений МНК (3.19).

4. СИСТЕМЫ УРАВНЕНИЙ С ТРЁХДИАГОНАЛЬНЫМИ МАТРИЦАМИ

4.1. Метод прогонки

В различных приложениях (таких, как решение краевых задач для дифференциальных уравнений [3] или задачи интерполяции кубическими сплайнами [25]) нередко возникает необходимость в решении СЛАУ высокого порядка, матрицы которых имеют специфическую структуру: ненулевые элементы лежат только на главной диагонали матрицы и на двух соседних с ней слева и справа диагоналях. В таких случаях можно резко сократить объём вычислений за счёт учёта специфики этих матриц.

Именно, рассмотрим СЛАУ $Ax = s$ порядка $n + 1$ с трёхдиагональной матрицей A :

$$\begin{aligned} a_0 x_0 + b_0 x_1 &= s_0, \\ c_1 x_0 + a_1 x_1 + b_1 x_2 &= s_1, \\ &\dots \\ c_i x_{i-1} + a_i x_i + b_i x_{i+1} &= s_i, \\ &\dots \\ c_n x_{n-1} + a_n x_n &= s_n. \end{aligned} \tag{4.1}$$

Элементы a_0, \dots, a_n образуют, таким образом, диагональ матрицы A ; элементы b_0, \dots, b_{n-1} – её наддиагональные, а элементы c_1, \dots, c_n – поддиагональные элементы. Остальные элементы матрицы A – заведомо нулевые, и в памяти компьютера их не хранят.

Рассматриваемый ниже метод решения СЛАУ (4.1) – **метод прогонки** – есть такой вариант метода Гаусса, в котором операции выполняются только над *ненулевыми* элементами трёхдиагональной матрицы, причём либо метод ведущего элемента не делается (*монотонная прогонка*), либо ведущий элемент выбирают в текущей строке (*немонотонная прогонка*).

Мы приводим здесь, следуя [3,25], лишь алгоритм монотонной прогонки (изложение алгоритма немонотонной прогонки можно найти в [25]). Достаточным условием применимости монотонной прогонки является наличие у матрицы A диагонального преобладания; условия диагонального преобладания (см. раздел 1.3) для трёхдиагональной матрицы принимают вид:

$$\begin{aligned} |a_0| &\geq |b_0| + \delta, \\ |a_i| &\geq |c_i| + |b_i| + \delta, \quad i = 1, \dots, n-1, \\ |a_n| &\geq |c_n| + \delta. \end{aligned} \tag{4.2}$$

Алгоритм состоит из двух этапов. На первом этапе (*прямой ход прогонки*) уравнения системы последовательно – по мере возрастания индекса i – преобразуют к виду

$$x_i = u_i + v_i x_{i+1}$$

(последнее уравнение примет вид $x_n = u_n$); на втором этапе (*обратный ход прогонки*) полученные уравнения последовательно решают, причём индекс i теперь последовательно убывает.

Приведём расчётные формулы алгоритма.

I. Прямой ход прогонки.

1) при $i = 0$ имеем: $q_0 = a_0$, $u_0 = s_0 / q_0$, $v_0 = -b_0 / q_0$.

2) при $i = 1, \dots, n-1$:

$$q_i = a_i + c_i v_{i-1}, \quad u_i = \frac{s_i - c_i u_{i-1}}{q_i}, \quad v_i = -\frac{b_i}{q_i}.$$

3) при $i = n$:

$$q_n = a_n + c_n v_{n-1}, \quad u_n = \frac{s_n - c_n u_{n-1}}{q_n}.$$

II. Обратный ход прогонки.

1) при $i = n$ имеем: $x_n = u_n$.

2) при $i = n-1, \dots, 0$: $x_i = u_i + v_i x_{i+1}$.

Число арифметических операций в методе прогонки составляет $8n$ (вместо $O(n^3)$, как это было при решении СЛАУ общего вида).

Заметим, что метод прогонки можно интерпретировать [3] как построение LU -разложения матрицы A . При этом матрицы U и L оказываются двухдиагональными. У матрицы U диагональные элементы равны 1, а наддиагональные – значениям $-v_i$; у матрицы L диагональные элементы равны q_i , а поддиагональные равны c_i (таким образом, здесь унитреугольной оказывается именно матрица U).

Следовательно, суть прямого хода состоит в вычислении элементов q_i и v_i матриц L и U и одновременном решении СЛАУ $Lu = s$; суть обратного хода – решение СЛАУ $Ux = u$.

4.2. Кусочные многочлены и сплайны

Рассмотрим на отрезке $[a, b]$ числовой прямой \mathbb{R} сетку Δ , т.е. [25] упорядоченный набор точек (узлов): $a = X_0 < X_1 < \dots < X_n = b$. Здесь n – число интервалов разбиения, $n+1$ – число узлов сетки, а $n-1$ – число её внутренних узлов.

Заданная на $[a, b]$ функция $F(x)$ – **кусочный многочлен** с узлами на сетке Δ , имеющий *степень* t и *дефект* r , если на каждом интервале разбиения функция $F(x)$ совпадает с некоторым алгебраическим много-

членом $F_i(x)$ – i -м **звеном** кусочного многочлена, причём степень каждого из звеньев не превосходит m , а гладкость самой функции $F(x)$ при этом не меньше $m-r$, т.е. $F \in C^{m-r}[a, b]$ [3,26].

Пусть $\mathbb{P}_{m,r}^\Delta$ – множество всех охарактеризованных в предыдущем абзаце кусочных многочленов. Это – векторное пространство, размерность которого равна $m+1+r(n-1)$ [26]. Для входящих в $\mathbb{P}_{m,r}^\Delta$ функций число m ограничивает сверху возможные значения степеней звеньев, а число $m-r$ ограничивает снизу возможную гладкость кусочного многочлена.

Каждое звено кусочного многочлена задаётся $m+1$ коэффициентом. Обычно для i -го звена используют тейлоровское представление:

$$F_i(x) = a_i + b_i(x - X_i) + c_i(x - X_i)^2 + \dots; \quad (4.3)$$

представление кусочного многочлена F при помощи набора тейлоровских коэффициентов звеньев называют **звенным представлением**.

Звенья кусочных многочленов, будучи обычными многочленами, имеют непрерывные производные сколь угодно высокого порядка; поэтому введённое выше требование к гладкости кусочного многочлена относится лишь к внутренним узлам сетки и означает, что в каждом из этих узлов он имеет непрерывные производные порядка $0, \dots, m-r$, совпадающие с соответствующими производными примыкающих к данному узлу звеньев. Каждое из условий совпадения значений производных соседних звеньев есть линейное условие, налагаемое на коэффициенты звеньев.

При $k > m-r$ k -я производная кусочного многочлена F имеет, вообще говоря, разрыв 1-го рода; производные порядка, большего m , можно в расчёт не брать (для них узлы – *устранимые* точки разрыва). Поэтому дефект r – это число высших производных, которым разрешено иметь разрывы 1-го рода во внутренних узлах сетки.

Таким образом, размерность пространства $\mathbb{P}_{m,r}^\Delta$ есть сумма размерности пространства \mathbb{P}_m многочленов степени не выше m и числа допустимых разрывов во внутренних узлах.

Между рассматриваемыми на одной и той же сетке пространствами кусочных многочленов при «соседних» значениях m и r имеют место следующие *условия включения*:

$$\mathbb{P}_{m,r}^\Delta \subset \mathbb{P}_{m,r+1}^\Delta, \quad \mathbb{P}_{m,r}^\Delta \subset \mathbb{P}_{m+1,r+1}^\Delta. \quad (4.4)$$

Заметим также, что производная кусочного многочлена из пространства $\mathbb{P}_{m,r}^\Delta$ есть кусочный многочлен из пространства $\mathbb{P}_{m-1,r}^\Delta$ (т.е. при дифференцировании степень и гладкость убывают на единицу, дефект же сохраняется).

Теперь – о сплайнах. Традиционно **сплайн** определяют [27] как кусочный многочлен дефекта 1 (т.е. это – кусочный многочлен наивысшей гладкости, не сводящийся к обычному многочлену). Сплайны степени m образуют векторное пространство $\mathbb{P}_{m,1}^\Delta$ размерности $n+m$, причём $\mathbb{P}_{m,1}^\Delta \subset C^{m-1}[a,b]$.

Здесь следует отметить, что распространена и другая терминология, когда «сплайнами» называют [3,25] произвольные кусочные многочлены (здесь мы следовать ей не будем). При этом – во избежание недоумений – кусочные многочлены дефекта 1 иногда именуют [28] *простыми* сплайнами.

Известно также немало типов *обобщённых* сплайнов, звенья которых уже не являются алгебраическими многочленами: рациональные, экспоненциальные, тригонометрические, гиперболические и другие [29,30]. Многие из таких неполиномиальных сплайнов обладают значительными достоинствами. Так, применение сплайнов на основе однопараметрических групп диффеоморфизмов [26,30] позволяет гарантировать строгую монотонность интерполянта при интерполировании строго монотонной функции; сплайны по рациональным интерполянтам [31,32] обладают свойством безусловной сходимости интерполянтов для произвольной непрерывной функции при стремлении к нулю максимального шага сетки; интерполяция сплайнами на основе атомарных функций (*U*-сплайнами) обеспечивает [33] получение интерполянтов бесконечной гладкости. Но такие обобщения кусочных многочленов здесь не рассматриваются.

Одним из простейших примеров сплайнов являются **линейные сплайны** – сплайны из пространства $\mathbb{P}_{1,1}^\Delta \subset C[a,b]$.

Графиком линейного сплайна является ломаная линия, а звенное представление имеет вид

$$F_i(x) = a_i + b_i(x - X_i). \quad (4.5)$$

Линейные сплайны хорошо приспособлены для решения **основной задачи интерполяции**. Последняя состоит в том, что для заданного на сетке Δ набора значений $F_i \equiv f(X_i)$, $i = 0, \dots, n$ требуется построить функцию из заранее очерченного класса, принимающую в узлах те же значения.

В классе $\mathbb{P}_{1,1}^\Delta$ интерполянт функции f всегда существует и единственен, а коэффициенты звеньев даются [25] формулами

$$a_i = F_i, \quad b_i = (F_{i+1} - F_i)/(X_{i+1} - X_i).$$

Недостатки такой интерполяции: интерполянт не является гладкой функцией, а точность невелика, поскольку при $f \in C^2[X_0, X_n]$ для нормы разности интерполируемой функции и интерполянта (здесь и далее в качестве нормы заданной на отрезке функции рассматривается её *уклонение от нуля*, т.е. [3] максимум её модуля на отрезке) справедлива [25] оценка

$$\|f - F\| \leq \frac{M_2}{8} h_{\max}^2, \quad M_2 \equiv \|f^{(2)}\|. \quad (4.6)$$

Здесь h_{\max} – максимальный шаг сетки, т.е. наибольшее из расстояний между её соседними узлами.

Перейдём к кусочным многочленам степени 3. Среди них наибольшее применение нашли два вида таких многочленов; в обоих случаях звенное представление таково:

$$F_i(x) = a_i + b_i(x - X_i) + c_i(x - X_i)^2 + d_i(x - X_i)^3. \quad (4.7)$$

Эрмитовыми кубическими многочленами называют кусочные многочлены из пространства $\mathbb{P}_{3,2}^\Delta \subset C^1[a, b]$. Размерность этого пространства равна $2n + 2$, и такие кусочные многочлены хорошо приспособлены для решения задачи интерполяции по Эрмиту. Последняя состоит в том, что для узлов сетки Δ заданы значения $f(X_i)$ и $f'(X_i)$, $i = 0, \dots, n$, и требуется построить такую функцию из заранее очерченного класса, что она сама и её первая производная принимают в узлах эти значения.

В классе $\mathbb{P}_{3,2}^\Delta$ эрмитов интерполянт функции f всегда существует и единственен; первые два коэффициента в представлении (4.7) равны соответственно $a_i = f(X_i)$ и $b_i = f'(X_i)$, а остальные два можно найти [26] по расчётным формулам

$$h = X_{i+1} - X_i, \quad D = \frac{a_{i+1} - a_i}{h}, \quad N = b_{i+1} - 2D + b_i, \quad (4.8)$$

$$c_i = \frac{D - b_i - N}{h}, \quad d_i = \frac{N}{h^2}. \quad (4.9)$$

При интерполяции эрмитовыми кубическими многочленами гладкость интерполянта не слишком велика, зато достаточно высока точность: в случае $f \in C^4 [X_0, X_n]$ справедлива [3,25] оценка

$$\|f - F\| \leq \frac{M_4}{384} h_{\max}^4, \quad M_4 \equiv \|f^{(4)}\|. \quad (4.10)$$

Как и при интерполяции линейными сплайнами, интерполяция здесь является локальной: нахождение коэффициентов звенного представления интерполянта осуществляется на каждом интервале разбиения независимо.

Кубическими сплайнами называют кусочные многочлены из пространства $\mathbb{P}_{3,1}^\Delta \subset C^2[a,b]$. Размерность этого пространства равна $n+3$, и это создаёт при интерполяции определённые проблемы.

Именно, для решения задачи интерполяции по Эрмиту параметров явно не хватает; а в случае основной задачи интерполяции (где изначально задают $n+1$ значение) мы имеем два лишних параметра. Значит, помимо условий интерполирования, надо наложить на коэффициенты звенного представления (4.7) ещё два дополнительных условия.

Обычно эти условия накладывают, стремясь поточнее задать поведение интерполянта на концах отрезка интерполяции.

Итак: **граничные условия** для кубического сплайна – это [3] два условия, накладываемые на сплайн соответственно на левом и правом концах отрезка $[X_0, X_n]$ с целью обеспечить единственность решения основной задачи интерполяции. Известно несколько стандартных типов граничных условий; здесь мы рассмотрим только один.

Условия типа I [25]: $F'(X_0) = f'(X_0), F'(X_n) = f'(X_n)$.

Кубический сплайн, являющийся для функции f решением основной задачи интерполяции при дополнительных граничных условиях типа I называется **фундаментальным кубическим сплайном** функции f .

4.3. Интерполяция кубическими сплайнами

Обсудим вопрос вычисления коэффициентов фундаментального кубического сплайна. Заметим, что в силу условий включения (4.4)

$$\mathbb{P}_{3,1}^\Delta \subset \mathbb{P}_{3,2}^\Delta.$$

Смысл здесь – такой: если у эрмитова кубического многочлена вторая производная, которая может иметь во внутренних узлах разрывы, фактически оказалась непрерывной, то мы получаем кубический сплайн.

Задача интерполяции фундаментальными кубическими сплайнами отличается от задачи интерполяции эрмитовыми кубическими многочленами тем, что условия интерполирования $f'(x)$ во внутренних узлах сетки заменены условиями непрерывности $F''(x)$ в этих узлах. Значит, если по заданным значениям X_i , $a_i \equiv f(X_i)$ ($i = 0, \dots, n$), $b_0 \equiv f'(X_0)$, $b_n \equiv f'(X_n)$, найти коэффициенты b_i ($i = 1, \dots, n-1$), то задача построения фундаментального кубического сплайна будет сведена к задаче построения эрмитова кубического многочлена, решаемой по формулам (4.8)–(4.9).

Нам осталось рассмотреть порядок нахождения коэффициентов b_i , называемых **наклонами** сплайна в узлах, во внутренних узлах.

Данные коэффициенты находят, решая **уравнения для наклонов** кубического сплайна – систему линейных алгебраических уравнений относительно неизвестных b_0, \dots, b_n . В распространённом варианте её записи (используют и другие варианты) она имеет [25] вид

$$\begin{aligned} 2b_0 + \mu_0^* b_1 &= 2f'(X_0), \\ \lambda_i b_{i-1} + 2b_i + \mu_i b_{i+1} &= 3(\lambda_i D_{i-1} + \mu_i D_i), \quad i = 1, \dots, n-1, \\ \lambda_n^* b_{n-1} + 2b_n &= 2f'(X_n); \end{aligned} \quad (4.11)$$

где

$$D_i = \frac{a_{i+1} - a_i}{h_i}, \quad \lambda_i = \frac{h_i}{h_{i-1} + h_i}, \quad \mu_i = \frac{h_{i-1}}{h_{i-1} + h_i}, \quad i = 0, \dots, n-1,$$

а $\mu_0^* = \lambda_n^* = 0$ (при граничных условиях других типов эти два коэффициента отличны от нуля, да и правые части соответствующих уравнений видоизменяются).

Матрица СЛАУ (4.11) является трёхдиагональной. Для того, чтобы при решении данной СЛАУ можно было использовать метод монотонной прогонки (см. раздел 4.1), проверим наличие диагонального преобладания, т.е. выполнение условий (4.2).

Так как $\lambda_i + \mu_i = 1$, то условия диагонального преобладания принимают вид

$$2 \geq |\lambda_i| + |\mu_i| + \delta = \lambda_i + \mu_i + \delta = 1 + \delta,$$

и можно взять $\delta = 1$.

Учёт граничных условий ничего не изменит: в них диагональный элемент равен 2, а внедиагональные члены вообще отсутствуют.

Итак, диагональное преобладание имеет место. Далее, по следствию из теоремы Леви – Деспланка

$$\text{cond}_\infty A \leq \frac{\|A\|_\infty}{\delta} = \frac{2+1}{1} = 3 ;$$

в выкладках учтено, что ∞ -норма матрицы равна максимуму 1-норм строк.

Таким образом, задача вычисления наклонов кубического сплайна в вычислительном плане очень хорошо обусловлена, а процесс её решения методом монотонной прогонки устойчив.

Приведём справедливую при $f \in C^4 [X_0, X_n]$ оценку погрешности интерполяции фундаментальными кубическими сплайнами [3,25]:

$$\|f - F\| \leq \frac{5M_4}{384} h_{\max}^4, \quad M_4 \equiv \|f^{(4)}\|. \quad (4.12)$$

По сравнению с оценкой (4.10) для интерполяции эрмитовыми кубическими многочленами здесь множитель $1/384$ заменён на $5/384$.

Ясно, что оценка погрешности сейчас должна была получиться хуже, поскольку о функции f информации имеется меньше.

Заметим, что кубические сплайны нашли широкое применение в различных задачах робототехники. Так, при кинематическом анализе манипуляционных роботов весьма часто в виде таких сплайнов задают [11,23] либо законы изменения координат в сочленениях робота, либо координаты, определяющие положение и ориентацию рабочего органа.

4.4. Применение кубических сплайнов в задаче планирования траектории автономного летательного аппарата

Интерполяция сплайнами дискретных измерений позволяет перейти от значений характеристик какого-либо процесса, заданных в отдельных точках, к функциям, определённым и на промежутках между точками измерений. Непрерывность функций сплайнов и их производных до нужного порядка позволяет получать в явном виде решения дифференциальных уравнений движения исследуемого объекта.

Рассмотрим в качестве примера задачу планирования траектории облёта квадрокоптером воздушной линии электропередачи (ЛЭП). Необходимо произвести старт летательного аппарата с земли, облететь ЛЭП с левой стороны и зависнуть над ней (заключительный этап полёта квадрокоптера, включающий [34] его спуск и посадку, сейчас не рассматривается).

Законы изменения абсциссы $X(t)$ и ординаты $Y(t)$ квадрокоптера найдём в виде кубических сплайнов. Зададим характерные точки траектории летательного аппарата: значения горизонтальной X и вертикальной Y координат, горизонтальной $V_x = \dot{X}$ и вертикальной $V_y = \dot{Y}$ компонент его скорости в моменты времени t_i , $i=0, \dots, 4$ (см. табл. 1). Роль независимой переменной сейчас играет время t . Требованию равенства нулю скорости аппарата в начале и в конце манёвра соответствуют граничные условия типа I (см. табл. 1).

Таблица 1

**Исходные данные
в задаче планирования траектории квадрокоптера**

Номер узла i	0	1	2	3	4
t_i , с	0	4	7	12	17
$X(t_i)$, м	3	2	2	6	10
$V_x(t_i)$, м/с	0	-	-	-	0
$Y(t_i)$, м	0	7	18	23	22
$V_y(t_i)$, м/с	0	-	-	-	0

По формулам (4.11), составим СЛАУ для наклонов b_i , $i=0, \dots, 4$, сплайнов в узлах:

$$\text{для } X_i(t): \begin{cases} 2b_0 = 0, \\ \frac{3b_0}{7} + 2b_1 + \frac{4b_2}{7} = -\frac{9}{28}, \\ \frac{5b_1}{8} + 2b_2 + \frac{3b_3}{8} = \frac{9}{10}, \\ \frac{b_2}{2} + 2b_3 + \frac{b_4}{2} = \frac{12}{5}, \\ 2b_4 = 0; \end{cases} \quad \text{для } Y_i(t): \begin{cases} 2b_0 = 0, \\ \frac{3b_0}{7} + 2b_1 + \frac{4b_2}{7} = \frac{239}{28}, \\ \frac{5b_1}{8} + 2b_2 + \frac{3b_3}{8} = 8, \\ \frac{b_2}{2} + 2b_3 + \frac{b_4}{2} = \frac{6}{5}, \\ 2b_4 = 0. \end{cases} \quad (4.13)$$

Решим полученные СЛАУ методом прогонки:

$$\begin{aligned} \text{для } X_i(t): \quad & b_0 = b_4 = 0, \quad b_1 = -\frac{433}{1720}, \quad b_2 = \frac{137}{430}, \quad b_3 = \frac{1927}{1720}; \\ \text{для } Y_i(t): \quad & b_0 = b_4 = 0, \quad b_1 = \frac{52991}{15480}, \quad b_2 = \frac{11441}{3870}, \quad b_3 = -\frac{2153}{15480}. \end{aligned} \quad (4.14)$$

Отметим, что для получения точного решения можно воспользоваться готовыми инструментами различных математических пакетов – например, **Mathematica** [35] или **Maple** [36], поддерживающих символьные преобразования.

С учётом (4.7), (4.8), (4.9) найдём кубические многочлены $X_i(t)$ и $Y_i(t)$ при $t \in [t_i, t_{i+1}]$ для $i=0, \dots, 3$ – звенья сплайнов $X(t)$ и $Y(t)$ (см. табл. 2).

Таблица 2

**Результаты решения задачи сплайн-интерполяции:
звенья кубических сплайнов**

i	$X_i(t)$
0	$3 - 857 / 6880 \cdot t^2 + 427 / 27520 \cdot t^3$
1	$2 - 433 / 1720 \cdot (t - 4) + 53 / 860 \cdot (t - 4)^2 + 23 / 3096 \cdot (t - 4)^3$
2	$2 + 137 / 430 \cdot (t - 7) + 221 / 1720 \cdot (t - 7)^2 - 277 / 43000 \cdot (t - 7)^3$
3	$6 + 1927 / 1720 \cdot (t - 12) + 137 / 4300 \cdot (t - 12)^2 - 33 / 1720 \cdot (t - 12)^3$
	$Y_i(t)$
0	$28279 / 61920 \cdot t^2 - 1189 / 247680 \cdot t^3$
1	$7 + 52991 / 15480 \cdot (t - 4) + 3089 / 7740 \cdot (t - 4)^2 - 2953 / 27864 \cdot (t - 4)^3$
2	$18 + 11441 / 3870 \cdot (t - 7) - 8587 / 15480 \cdot (t - 7)^2 + 4217 / 129000 \cdot (t - 7)^3$
3	$23 - 2153 / 15480 \cdot (t - 12) - 2491 / 38700 \cdot (t - 12)^2 +$ $+ 4039 / 387000 \cdot (t - 12)^3$

На рис. 4.1 представлена траектория квадрокоптера, построенная по звеньям кубических сплайнов (табл. 2).

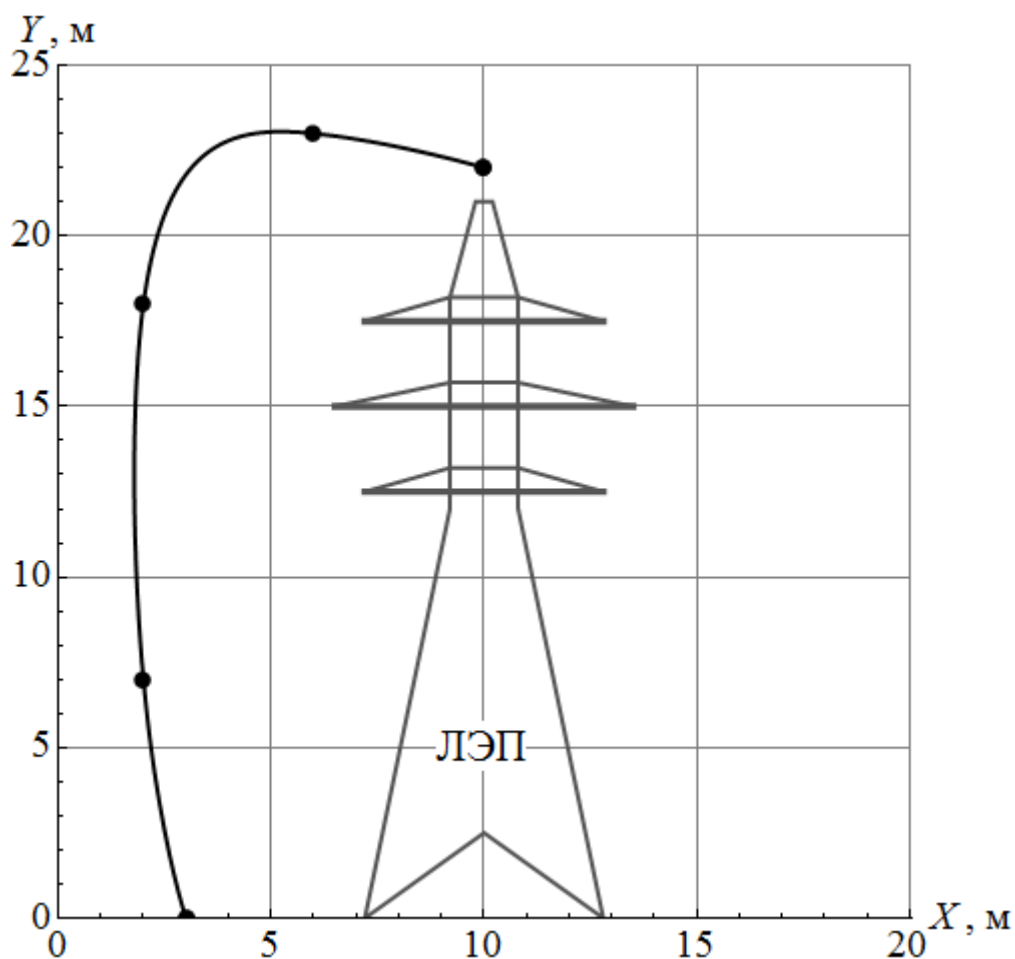


Рис. 4.1. Траектория квадрокоптера

На рис. 4.2 приведены графики модуля скорости $|V| = \sqrt{\dot{X}^2 + \dot{Y}^2}$ квадрокоптера и модуля его ускорения $|W| = \sqrt{\ddot{X}^2 + \ddot{Y}^2}$, демонстрирующие непрерывный характер изменения указанных величин.

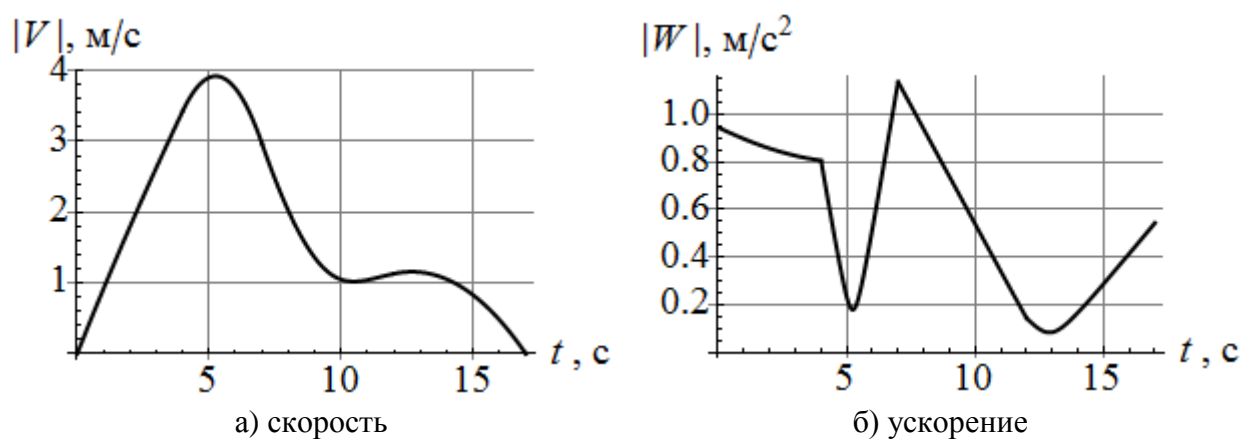


Рис. 4.2. График изменения модулей скорости и ускорения квадрокоптера

Проверим правильность нахождения наклонов сплайна $X(t)$, решив соответствующую СЛАУ (4.13) в среде **Scilab** методом монотонной прогонки. Зададим массивы диагональных, поддиагональных и наддиагональных элементов матрицы этой СЛАУ, а также массив правых частей уравнений:

```
--> a_cf = [2; 2; 2; 2; 2];
--> b_cf = [0; 4/7; 3/8; 1/2; 0];
--> c_cf = [0; 3/7; 5/8; 1/2; 0];
--> s_cf = [0; -9/28; 9/10; 12/5; 0];
```

В обозначениях раздела 4.1 $\mathbf{a_cf} = [a_0 \dots a_4]$, $\mathbf{b_cf} = [b_0 \dots b_3 \ 0]$, $\mathbf{c_sf} = [0 \ c_1 \dots c_4]$, $\mathbf{s_cf} = [s_0 \dots s_4]$.

Зададим вспомогательные переменные и массивы:

```
--> n = size(a_cf, 1);
--> q = zeros(a_cf); u = q; v = q;
```

Прямой ход прогонки:

```
--> q(1) = a_cf(1); u(1) = s_cf(1)/q(1);
--> v(1) = -b_cf(1)/q(1);
--> for i = 2:(n-1)
--> q(i) = a_cf(i)+c_cf(i)*v(i-1);
--> u(i) = (s_cf(i)-c_cf(i)*u(i-1))/q(i);
--> v(i) = -b_cf(i)/q(i);
--> end
--> q(n)=a_cf(n)+c_cf(n)*v(n-1);
--> u(n) = (s_cf(n)-c_cf(n)*u(n-1))/q(n);
```

Теперь выполним обратный ход прогонки и найдём решение СЛАУ (массив \mathbf{x}):

```
--> x = zeros(a_cf);
--> x(n) = u(n);
--> for i=(n-1):-1:1
--> x(i) = u(i)+v(i)*x(i+1);
--> end
```

Выведем на печать значения полученных наклонов сплайна $X(t)$:

```
--> x'
ans =
    0.   - 0.2517442    0.3186047    1.1203488    0.
```

Сравним этот результат с точным решением (4.14), записанным в массив $\mathbf{B} = [b_0 \dots b_4]$:

```
--> B = [0; -433/1720; 137/430; 1927/1720; 0];
--> norm(x - B, %inf)
ans =
    2.220D-16
```

Результаты совпадают с точностью до погрешностей выполнения арифметических операций на компьютере.

Отметим, что в среде **Scilab** есть встроенные средства для решения задачи интерполяции кубическими сплайнами. Продемонстрируем их работу. Зададим исходные данные для получения сплайна $X(t)$ – массивы времён **dataT** и координат **dataX** (табл. 1):

```
--> dataT = [0 4 7 12 17];
--> dataX = [3 2 2 6 10];
```

Горизонтальные скорости в начале и в конце манёвра:

```
--> Vx_s = 0; Vx_f = 0;
```

Применим функцию **splin** с опцией **"clamped"** для вычисления наклонов сплайна $X(t)$ при граничных условиях I типа, имеющих в рассматриваемом примере вид $\dot{X}(0) = Vx_s$, $\dot{X}(17) = Vx_f$:

```
--> bX = splin(dataT, dataX, "clamped", [Vx_s, Vx_f])
bX =
    0. - 0.2517442    0.3186047    1.1203488    0.
```

Зададим массив из 100 промежуточных точек, равномерно размещённых на отрезке $[0, 17]$:

```
--> t = linspace(0, 17, 100);
```

С помощью функции **interp** вычислим массив значений сплайна в промежуточных точках:

```
--> X = interp(t, dataT, dataX, bX);
```

Построим график интерполянта и наложим на него узловые точки траектории (см. рис. 4.3):

```
--> plot(t, X); plot(dataT, dataX, 'ko');
```

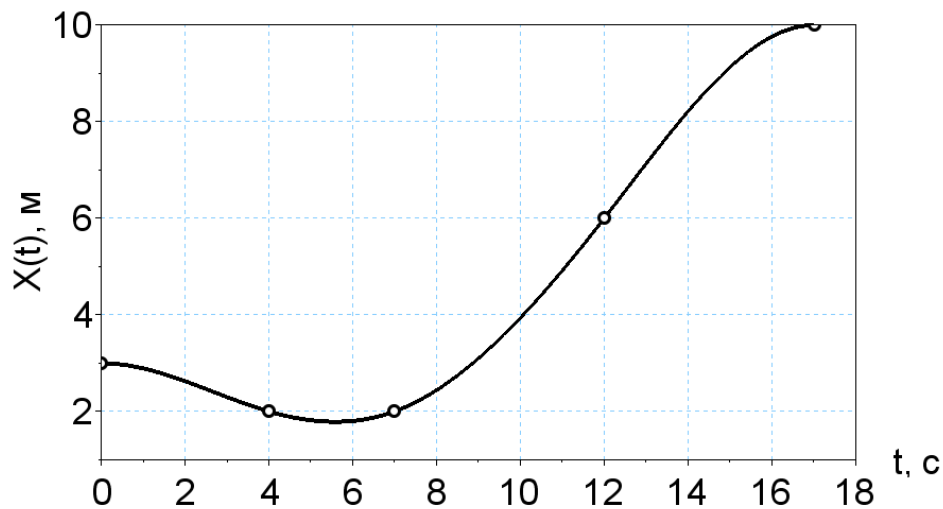


Рис. 4.3. График сплайна, определяющего закон изменения горизонтальной координаты квадрокоптера

Упражнение. Самостоятельно решите задачу построения сплайна $Y(t)$ в среде **Scilab**. Вычислите массив **Y** его значений в промежуточных точках и постройте траекторию квадрокоптера (**plot(x, Y)**).

Завершая данный раздел, отметим, что выполненные по схеме квадрокоптера автономные летательные аппараты, оснащённые интеллектуальными бортовыми системами автоматического управления, в последнее время всё шире применяются при дистанционном мониторинге технического состояния ЛЭП [37].

Контрольные вопросы и задания

1. Запишите основные расчётные соотношения для метода монотонной прогонки. Для решения какого класса СЛАУ подходит этот метод? Как связан метод монотонной прогонки с методом LU -разложения?
2. Дайте определение кусочного многочлена, дефекта кусочного многочлена и звена кусочного многочлена. Что такое звенное представление кусочного многочлена?
3. Какие кусочные многочлены называют сплайнами?
4. В чём состоит основная задача интерполяции? Каков порядок её решения с помощью линейных сплайнов?
5. В чём состоит задача интерполяции по Эрмиту? Какой порядок её решения с помощью эрмитовых кубических многочленов?
6. Что такое кубический сплайн? Какова степень его гладкости?
7. Что такое граничные условия для кубического сплайна? Какова их роль в решении основной задачи интерполяции с помощью кубических сплайнов? В чём состоят граничные условия I типа?

8. Опишите алгоритм решения основной задачи интерполяции кубическими сплайнами с граничными условиями типа I.

9. Какой вид имеют уравнения наклонов кубического сплайна? Почему процесс их решения методом монотонной прогонки устойчив?

10. С помощью метода монотонной прогонки решите СЛАУ $Ax = b$, где

$$A = \begin{pmatrix} 16 & 4 & 0 & 0 \\ 4 & 33 & 8 & 0 \\ 0 & -4 & 15 & -4 \\ 0 & 0 & 4 & -33 \end{pmatrix}, \quad b = \begin{pmatrix} -28 \\ 41 \\ 22 \\ -25 \end{pmatrix}.$$

Заключение

В данном учебном пособии рассмотрены методы решения СЛАУ, использующие основные матричные разложения: Холецкого, LU -разложения без выбора и с частичным выбором ведущего элемента, QR -разложения (методы вращений и отражений), SVD -разложения. Отдельно рассмотрены задачи нахождения нормальных решений и псевдорешений СЛАУ произвольного вида, решения СЛАУ с трёхдиагональной матрицей методом монотонной прогонки.

Также рассмотрены задачи приближения функций по методу наименьших квадратов и интерполяции кубическими сплайнами, в которых возникает необходимость решения СЛАУ специальных классов.

Изложенные методы применены для решения прикладных задач механики и робототехники: обратной задачи о скоростях избыточного манипулятора, планированию траектории автономного летательного аппарата методом сплайн-интерполяции, решению системы уравнений малых свободных колебаний линейной консервативной системы с конечным числом степеней свободы, а также идентификации параметров вынужденных колебаний механической системы (чувствительного элемента гироскопа).

Приведены примеры решения задач с использованием рассмотренных методов в среде **Scilab**.

Список рекомендованной литературы

Основной

1. **Канатников А.Н., Крищенко А.П.** Линейная алгебра. 5-е изд. М.: Издательство МГТУ им. Н.Э. Баумана, 2015. 336 с. (Математика в техническом университете. Вып. 4). ISBN 978-5-7038-4284-3.
2. **Зализняк В.Е.** Численные методы. Основы научных вычислений. 2-е изд. М.: Юрайт, 2017. 356 с. ISBN 978-5-534-02714-3.
3. **Амосов А.А., Дубинский Ю.А., Копчёнова Н.В.** Вычислительные методы. 4-е изд. СПб.: Лань, 2014. 672 с. ISBN 978-5-8114-1623-3.
4. **Адамов Б.И., Капустина О.М., Меркурьев И.В., Панкратьева Г.В.** Кинематика, динамика и управление движением мобильного робота-манипулятора KUKA youBot. М.: Издательство МЭИ, 2018. 71 с. ISBN 978-5-7046-1978-9.

Дополнительный

5. **Алексеев Е.Р., Чеснокова О.В., Рудченко Е.А.** Scilab: Решение инженерных и математических задач // М.: ALT Linux; БИНОМ. Лаборатория знаний, 2008. 269 с. ISBN 978-5-94774-890-1.
6. **Колкер А.Б., Ливенец Д.А., Кошелева А.И.** Обоснование выбора программного обеспечения для робототехники // Автоматика и программная инженерия. 2012. № 1 (1). С. 51–64.
7. **Дохтаева И.А., Рожина Е.Н.** Определение достоинств среды программирования Scilab путем сравнительного анализа языков программирования MatLab и Scilab // Прогрессивные технологии и процессы: Сборник научных статей Международной молодёжной научно-практической конференции (Курск, 25–26 сентября 2014 г.). Т. 1. Курск: Университетская книга, 2014. ISBN 978-5-9905749-5-3. С. 204–207.
8. **Анисимова Э.С., Ибатуллин Р.Р.** Использование программной системы Scilab в обучении решению задач вычислительной математики // Современные наукоёмкие технологии. 2016. № 8-1. С. 91–95.
9. **Гантмахер Ф.Р.** Теория матриц. 5-е изд. М.: Физматлит, 2010. 560 с. ISBN 978-5-9221-0524-8.
10. **Бровко Г.Л.** Элементы математического аппарата механики сплошной среды. М.: Физматлит, 2015. 424 с. ISBN 978-5-9221-1634-3.
11. **Корецкий А.В., Осадченко Н.В.** Компьютерное моделирование кинематики манипуляционных роботов. М.: Издательство МЭИ, 2000. 48 с.

12. **Хорн Р., Джонсон Ч.** Матричный анализ. М.: Мир, 1989. 655 с. ISBN 5-03-001042-4.
13. **Панкратьева Г.В., Захаров С.В., Теселько Н.В.** Исследование колебаний в консервативных системах. М.: Издательский дом МЭИ, 2007. 32 с.
14. **Зими́на О.В.** Линейная алгебра и аналитическая геометрия. Учебный комплекс. М.: Издательство МЭИ, 2000. 328 с. ISBN 5-7046-0632-6.
15. **Воеводин В.В., Кузнецов Ю.А.** Матрицы и вычисления. М.: Наука, 1984. 320 с.
16. **Икрамов Х.Д.** Несимметричная проблема собственных значений. Численные методы. М.: Наука, 1991. 240 с. ISBN 5-02-014462-2.
17. **Shivakumar P.N., Sivakumar K.C., Zhang Yang.** Infinite Matrices and Their Recent Applications. Switzerland: Springer, 2016. x + 118 p. ISBN 978-3-319-30179-2.
18. **Подольский М.Е., Черенкова С.В., Иванова М.А.** О новом подходе к анализу физического смысла и математического описания тензорных величин в прикладных задачах механики // Морские интеллектуальные технологии. 2017. Т. 3, № 4 (38). С. 166–176.
19. **Кильчевский Н.А.** Курс теоретической механики. Т. 1. Кинематика, статика, динамика точки. 2-е изд. М.: Наука, 1977. 480 с.
20. **Винберг Э.Б.** Курс алгебры. 3-е изд. М.: МЦНМО, 2017. 592 с. ISBN 978-5-4439-0209-8.
21. **Дэннис Дж., Шнабель Р.** Численные методы безусловной оптимизации и решения нелинейных уравнений. М.: Мир, 1988. 440 с. ISBN 5-03-001102-1.
22. **Форсайт Дж., Малькольм М., Моулер К.** Машинные методы математических вычислений. М.: Мир, 1980. 280 с.
23. **Зенкевич С.Л., Ющенко А.С.** Основы управления манипуляционными роботами. 2-е изд. М.: Издательство МГТУ им. Н. Э. Баумана, 2004. 480 с. (Робототехника). ISBN 5-7038-2567-9.
24. **Маслов А.А., Маслов Д.А., Меркурьев И.В.** Идентификация параметров волнового твердотельного гироскопа с учётом нелинейности колебаний резонатора // Приборы и системы. Управление, контроль, диагностика. 2014. № 5. С. 18–23.
25. **Завьялов Ю.С., Квасов Б.И., Мирошниченко В.Л.** Методы сплайн-функций. М.: Наука, 1980. 352 с.
26. **Осадченко Н.В.** Локальная монотонная интерполяция и однопараметрические группы // Пространство, время и фундаментальные взаимодействия. 2017. № 2. С. 60–73.
27. **Бахвалов Н.С., Жидков Н.П., Кобельков Г.М.** Численные методы. 3-е изд. М.: БИНОМ. Лаборатория знаний, 2003. 632 с. ISBN 5-94774-060-5.

28. **Волков Ю.С., Субботин Ю.Н.** 50 лет задаче Шёнберга о сходимости сплайн-интерполяции // Труды Института математики и механики УрО РАН. 2014. Т. 20, № 1. С. 52–67.
29. **Богданов В.В., Волков Ю.С.** Выбор параметров обобщённых кубических сплайнов при выпуклой интерполяции // Сибирский журнал вычислительной математики. 2006. Т. 9, № 1. С. 5–22.
30. **Осадченко Н.В.** Монотонная сплайн-интерполяция класса C^2 на основе однопараметрических групп диффеоморфизмов // Пространство, время и фундаментальные взаимодействия. 2017. № 3. С. 12–27.
31. **Рамазанов А.-Р.К., Магомедова В.Г.** Сплайны по рациональным интерполяциям // Дагестанские электронные математические известия. 2015. № 4. С. 21–30.
32. **Рамазанов А.-Р.К., Магомедова В.Г.** Оценка скорости сходимости сплайнов по рациональным интерполяциям через индуцированные функции // Вестник Дагестанского государственного университета. Серия 1. Естественные науки. 2016. Т. 31, вып. 2. С. 35–40.
33. **Комова О.И., Маслов А.Н., Осадченко Н.В.** Атомарные функции и построение программного движения сварочного робота // Вестник МГТУ им. Н.Э. Баумана. Серия: Естественные науки. 2018. № 5 (80). С. 15–36. doi: 10.18698/1812-3368-2018-5-15-36
34. **Павловский В.Е., Яцун С.Ф., Емельянова О.В., Савицкий А.В.** Моделирование и исследование процессов управления квадрокоптером // Робототехника и техническая кибернетика. 2014. № 4 (5). С. 49–57.
35. **Дьяконов В.П.** Mathematica 5.1/5.2/6 в математических и научно-технических расчётах. 2-е изд. М.: Солон-Пресс, 2010. 744 с. ISBN 978-5-91359-045-9.
36. **Кирсанов М.Н.** Практика программирования в системе Maple. М.: Издательский дом МЭИ, 2011. 208 с. ISBN 978-5-383-0061-3.
37. **Попов Н.И., Емельянова О.В.** Динамические особенности мониторинга воздушных линий электропередачи с помощью квадрокоптера // Современные проблемы науки и образования. 2014. № 2. Ст. 105 (8 с.).