Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени
Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Робототехники и комплексной автоматизации»
КАФЕДРА «Системы автоматизированного проектирования (РК-6)»

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

по дисциплине «Вычислительная математика»

Студент:	Горохова Алексадра Сергеевна
Группа:	РК6-56Б
Тип задания:	лабораторная работа
Тема:	LU-разложение

Студент	________ подпись, дата	Горохова А.С. Фамилия, И.О.
Преподаватель	________ подпись, дата	________ Фамилия, И.О.

Москва, 2022

Содержание

LU-разложение

1 Задание

Дана СЛАУ $\boldsymbol{A_1 x} = \boldsymbol{b_1}$:

$$\begin{bmatrix} 1 & 1 & 0 & 3 \\ 2 & 1 & -1 & 1 \\ 3 & -1 & -1 & 2 \\ -1 & 2 & 3 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ -3 \\ 4 \end{bmatrix}. \tag{1}$$

Дана СЛАУ $\boldsymbol{A_1 x} = \boldsymbol{b_1}$:

$$\begin{bmatrix} 3 & 1 & -3 \\ 6 & 2 & 5 \\ 1 & 4 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -16 \\ 12 \\ -39 \end{bmatrix}. \tag{2}$$

Требуется (базовая часть):

1. Написать функцию lu(A), которая производит LU-разложение матрицы A и возвращает матрицы L и U.

2. Написать функцию solve(L, U, b), которая возвращает решение СЛАУ $Ax = b$, где матрица A представлена в виде LU-разложения.

3. Найти решение СЛАУ 1 с помощью разработанной функции lu(A) и сравнить с точным решением этой СЛАУ: $x = [-1, 2, 0, 1]^T$.

Требуется (продвинутая часть):

1. Доказать, что для решения СЛАУ 2 необходимо на определенной итерации метода Гаусса произвести перестановку строк.

2. Модифицировать функцию lu(A, permute) так, чтобы она принимала аргумент permute и возвращает матрицы L, U и P. Если permute=True, то LU-разложение должно происходить с частичным выбором главного элемента, а возвращаемая матрица P при этом должна быть соответствующей матрицей перестановок. Если permute=False, то частичного выбора происходить не должно, а возращаемая матрица P должна быть единичной.

3. Модифицировать функцию solve(L, U, P, b) так, чтобы она принимала на вход аргумент P и возвращала решение СЛАУ $\boldsymbol{Ax} = \boldsymbol{b}$, где матрица $\boldsymbol{A}$ представлена в виде LU-разложения с матрицей перестановок $\boldsymbol{P}$, т.е. $\boldsymbol{PA} = \boldsymbol{LU}$.

4. Найти решение СЛАУ 2 с помощью разработанной функции solve(L, U, P, b), обозначаемое здесь и далее $\hat{x}$.

5. Доказать, что модифицированная СЛАУ, полученная с помощью добавления к элементам $a11$ и $b1$ СЛАУ 2 малого числа 10^{-p}, где p – произвольное целое число, имеет то же решение, что и исходная СЛАУ. Здесь и далее решение модифицированной СЛАУ будет обозначаться как .

6. С помощью разработанных функций lu(A, permute) и solve(L, U, P, b) найти решения модифицированной СЛАУ для $p \in [0, 12]$ для двух случаев:
   - без частичного выбора главного элемента,
   - с частичным выбором главного элемента.

7. Для обоих случаев построить semilogy график зависимости относительной погрешности вычисления $E = \frac{\|\hat{x}-\tilde{x}\|_\infty}{\|\hat{x}\|_\infty}$ от p. Проанализировав полученные результаты, сделайте подробный вывод о вычислительной устойчивости/неустойчивости решения СЛАУ с помощью LU-разложения для конкретного рассматриваемого случая и опишите, что нужно в этом контексте иметь в виду потенциальному пользователю ваших функций lu(A, permute) и solve(L, U, P, b).

2 Цель выполнения лабораторной работы

Реализация LU-разложения и исследование его на вычислительную устойчивость.

3 Базовая часть

3.1 Разработка функции, производящей LU - разложение

$\boldsymbol{LU}$-разложение - представление известной матрицы $\boldsymbol{A}$ в виде произведения двух матриц $\boldsymbol{A} = \boldsymbol{LU}, \boldsymbol{L}$ - нижняя треугольная матрица, а $\boldsymbol{U}$ - верхняя. $\boldsymbol{LU}$-разложение возможно только для невырожденных матриц.
Нижняя и верхняя матрицы $\boldsymbol{L}$ и $\boldsymbol{U}$ программно вычисляются по следующему алгоритму, который выполняется последовательно для $i = 1, \ldots, n$, где n - размерность матрицы.
Пусть матрица $\boldsymbol{C} = \boldsymbol{A}$. Тогда на каждой i-ой итерации:

1. все элементы, находящиеся ниже i-ой строки i-го столбца, делятся на опорный элемент главной диагонали;

2. из всех элементов c_{jk} таких, что $j > i$ и $k > i$, вычитается произведение элементов $c_{ji}c_{ik}$.

В результате работы алгоритма получается матрица $\boldsymbol{C}$:

$$C = L + U - E, \text{ где } E \text{ – единичная матрица.} \tag{3}$$

На основании данного алгоритма была разработана функция lu(A), производящая разложение матрицы $\boldsymbol{A}$ на нижнюю и верхнюю треугольные матрицы $\boldsymbol{L}$ и $\boldsymbol{U}$ соответственно. Её программная реализация представлена в листинге 1.

Листинг 1. Функция, производящая LU-разложение

```python
def lu(A):
    n = len(A)
    C = np.array(A.copy())
    U = np.zeros((n, n), dtype=np.float64)
    L = np.zeros((n, n), dtype=np.float64)
    for i in range(n):
        for j in range(i + 1, n):
            C[j][i] /= C[i][i]
            for k in range(i + 1, n):
                C[j][k] -= C[j][i] * C[i][k]
    for i in range(n):
        for j in range(n):
            if i > j:
                L[i][j] = C[i][j]
            if i == j:
                L[i][j] = 1
                U[i][j] = C[i][j]
            if i < j:
                U[i][j] = C[i][j]
    return L, U
```

3.2 Разработка функции, вычисляющей решение СЛАУ

После того, как матрица $\boldsymbol{A}$ была разложена на матрицы $\boldsymbol{L}$ и $\boldsymbol{U}$, решение СЛАУ находится следующим образом.
Вектор $\boldsymbol{y}$ обозначается как $\boldsymbol{y} = \boldsymbol{U}\boldsymbol{x}$. Тогда он находится как решение уравнения:

$$Ly = b.$$

Вектор $\boldsymbol{x}$ является решением уравнения:

$$Ux = y.$$

Векторы x и y программно вычисляются следующим образом.

1. Значения вектора $\boldsymbol{y}$:
$$y_i = b_i - \sum_{k=1}^{i-1} l_{ik} y_k,$$
где $i = 1, \ldots, n$, где n - размерность матрицы.

2. Значения вектора $\boldsymbol{x}$:
$$x_i = \frac{1}{u_{ii}} \left(y_i - \sum_{k=i+1}^{i-1} u_{ik} x_k \right),$$
где $i = 1, \ldots, n$, где n - размерность матрицы

Для нахождения решения СЛАУ $\boldsymbol{Ax} = \boldsymbol{b}$, где матрица $\boldsymbol{A}$ представлена в виде $\boldsymbol{LU}$-разложения, с помощью алгоритма, приведенного выше, была разработана функция solve(L, U, b), реализация которой представлена в листинге 2. Она принимает на вход нижнюю и верхнюю матрицы $\boldsymbol{L}$ и $\boldsymbol{U}$ соответственно, а также матрицу $\boldsymbol{b}$. Возвращает матрицу решений $\boldsymbol{x}$.

Листинг 2. Функция вычисления решения СЛАУ $\boldsymbol{Ax} = \boldsymbol{b}$

```
def solve(L, U, b):
  n = len(b)
  X = np.zeros(n)
  Y = np.zeros(n)
  for i in range(n):
      Y[i] = b[i] - sum([L[i][k] * Y[k] for k in range(i)])
  for i in range(n):
      summa = sum([U[n - i - 1][n - k - 1] * X[n - k - 1] for k in range(i)])
      y = Y[n - i - 1]
      X[n - i - 1] = (y - summa) / U[n - i - 1][n - i - 1]
  return X
```

3.3 Решение СЛАУ 1

С помощью разработанных функций lu(A) и solve(L, U, b), представленных в листингах 1 и 2 соответственно, было получено решение СЛАУ 1:

$$x^* = [-1, 2, 0, 1]^T.$$

Решение данной СЛАУ сошлось с точным ее решением, представленным в задании.

4 Продвинутая часть

4.1 Доказательство того, что для решения СЛАУ 2 необходима перестановка строк

Для доказательства была рассмотрена следующая расширенная матрица:

$$\tilde{A}_2 = \left[\begin{array}{ccc|c} 3 & 1 & -3 & -16 \\ 6 & 2 & 5 & 12 \\ 1 & 4 & -3 & -39 \end{array}\right]. \tag{4}$$

На первой итерации метода Гаусса были обнулены элементы столбца под первым элементом главной диагонали. Для этого из всех строк была вычтена первая, домноженная на $\frac{\tilde{a}_{i1}}{\tilde{a}_{11}}$:

$$\tilde{a}_{ij}^{(1)} = \tilde{a}_{ij} - \frac{\tilde{a}_{i1}}{\tilde{a}_{11}}\tilde{a}_{1j}, \quad i = 2, \ldots, n,$$

где n - размер матрицы. Тогда в результате матрица после первой итерации примет вид:

$$\tilde{A}_2 = \left[\begin{array}{ccc|c} 3 & 1 & -3 & -16 \\ 0 & 0 & 11 & 44 \\ 0 & 3\frac{2}{3} & -2 & -33\frac{2}{3} \end{array}\right].$$

При выполнении второй итерации, необходимо обнулить значения столбца под вторым элементом главной диагонали. Для этого требуется из третьей строки вычесть вторую, домноженную на $\frac{\tilde{a}_{32}}{\tilde{a}_{22}}$. Однако, при выполнении первой итерации, диагональный элемент $\tilde{a}_{22}$ был обнулен, следовательно, данное действие невозможно. В таком случае, для продолжения решения СЛАУ 2 методом Гаусса необходима перестановка строк матриц так, чтобы вторым диагональным элементом стал наибольший по модулю элемент второго столбца:

$$\left|\tilde{a}_{22}^{(2)}\right| = \max_{2 \le i \le n} \left|\tilde{a}_{i2}^{(2)}\right|. \tag{5}$$

Тогда согласно 5, вторая и третья строки расширенной матрицы 4 были переставлены между собой:

$$\tilde{A}_2 = \left[\begin{array}{ccc|c} 3 & 1 & -3 & -16 \\ 0 & 3\frac{2}{3} & -2 & -33\frac{2}{3} \\ 0 & 0 & 11 & 44 \end{array}\right]. \tag{6}$$

Тогда решение СЛАУ:

$$\tilde{x} = [1, -7, 4]. \tag{7}$$

4.2 Программная реализация решения СЛАУ 2 с частичным выбором главного элемента

Как было доказано в предыдущем пункте, для решения СЛАУ 2 необходимо произвести перестановку строк. Подобные перестановки можно учесть, используя матрицу перестановок $\boldsymbol{P}$, которая образуется с помощью перестановки строк единичной матрицы $\boldsymbol{E}$. При умножении такой матрицы перестановок справа на матрицу $\boldsymbol{A}$ происходит перестановка столбцов. В таком случае $\boldsymbol{LU}$-разложение матрицы $\boldsymbol{A}$ находится следующим образом:

$$PA = LU.$$

Тогда $\boldsymbol{LU}$-разложение с частичным выбором главного элемента будет производиться за счет перестановки строк матрицы $\boldsymbol{A}$ так, что диагональным элементом становится наибольший по модулю элемент k-го столбца:

$$\left|\tilde{a}_{kk}^{(k)}\right| = \max_{k \le i \le n} \left|\tilde{a}_{ik}^{(k)}\right|.$$

Перестановки будут фиксироваться в соответствующей матрице перестановок $\boldsymbol{P}$. Программная реализация функции lu(A, permute), производящей $\boldsymbol{LU}$-разложение с частичным выбором главного элемента, представлена в листинге 3. Функция принимает на вход матрицу $\boldsymbol{A}$ и аргумент permute, который может принимать значения True или

False соответственно. Если permute=True, то $\boldsymbol{LU}$-разложение происходит с частичным выбором главного элемента, а возвращаемая матрица $\boldsymbol{P}$ - соответствующая матрица перестановок. Если permute=False, производится $\boldsymbol{LU}$-разложение без частичного выбора, а возращаемая матрица $\boldsymbol{P}$ - единичная. Также функция возвращает нижнюю и верхнюю треугольные матрицы $\boldsymbol{L}$ и $\boldsymbol{U}$ соответственно.

Листинг 3. Функция, производящая LU-разложение с частичным выбором главного элемента или без него

```
def lu(A, permute):
    n = len(A)
    C = np.array(A.copy())
    P = np.zeros((n, n))
    U = np.zeros((n, n), dtype=float)
    L = np.zeros((n, n), dtype=float)
    for i in range(n):
        for j in range(n):
            P[i][i] = 1
    for i in range(n):
        #поиск опорного элемента
        sup_el_value = 0
        sup_el = -1
        for row in range(i, n):
            if np.abs(C[row][i]) > sup_el_value:
                sup_el_value = np.abs(C[row][i])
                sup_el = row
        if sup_el_value != 0:
            if permute:
            #меняем местами i-ю строку и строку с опорным элементом
                P[[sup_el, i]] = P[[i, sup_el]]
                C[[sup_el, i]] = C[[i, sup_el]]
            for j in range(i + 1, n):
                C[j][i] /= C[i][i]
                for k in range(i + 1, n):
                    C[j][k] -= C[j][i] * C[i][k]
    for i in range(n):
        for j in range(n):
            if i > j:
                L[i][j] = C[i][j]
            if i == j:
                L[i][j] = 1
                U[i][j] = C[i][j]
            if i < j:
                U[i][j] = C[i][j]
    return L, U, P
```

$\boldsymbol{LU}$-разложение может производиться как с частичным выбором главного элемента, так и без него. Поэтому необходимо в функцию solve(L, U, b), представленную в листинге 2, передавать еще и матрицу перестановок $\boldsymbol{P}$. В таком случае решение СЛАУ будет находиться по следующей формуле:

$$PAx = Pb.$$

Программная реализация функции solve(L, U, P, b) представлена в листинге 4. Функция принимает на вход матрицу $\boldsymbol{b}$, нижнюю и верхнюю треугольные матрицы $\boldsymbol{L}$ и $\boldsymbol{U}$, полученные в результате LU-разложения матрицы $\boldsymbol{A}$, домноженной на матрицу перестановок $\boldsymbol{P}$. Возвращает матрицу решений $\boldsymbol{x}$.

Листинг 4. Функция вычисления решения СЛАУ $\boldsymbol{PAx} = \boldsymbol{Pb}$

```python
def solve(L, U, P, b):
    n = len(b)
    X = np.zeros(n)
    Y = np.zeros(n)
    b = P.dot(b)
    for i in range(n):
        Y[i] = b[i] - sum([L[i][k] * Y[k] for k in range(i)])
    for i in range(n):
        X[n - i - 1] = (Y[n - i - 1] - sum([U[n - i - 1][n - k - 1] * X[n - k - 1] for
            k in range(i)])) / U[n - i - 1][n - i - 1]
    return X
```

В результате разработки функции solve(L, U, P, b), представленной в листинге 4, было получено решение СЛАУ 2:

$$\hat{x} = [1, -7, 4]. \tag{8}$$

Данное решение сошлось с решением 7 из доказательства первого пункта продвинутой части.

4.3 Решение модифицированной СЛАУ 2

Согласно условию, модифицированная СЛАУ 2 имеет следущий вид:

$$\begin{bmatrix} 3+\delta & 1 & -3 \\ 6 & 2 & 5 \\ 1 & 4 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -16+\delta \\ 12 \\ -39 \end{bmatrix}, \quad \text{где } \delta = 10^{-p}. \tag{9}$$

Модифицированная СЛАУ 9 была представлена в виде следующей системы уравнений:

$$\begin{cases} (3+\delta)x_1 + x_2 - 3x_3 = -16 + \delta, \\ 6x_1 + 2x_2 + 5x_3 = 12, \\ x_1 + 4x_2 - 3x_3 = -39. \end{cases} \tag{10}$$

Из первого уравнения системы 10 был выражен x_1:

$$x_1 = \frac{-16 + \delta - x_2 + 3x_3}{3 + \delta}. \tag{11}$$

Полученное выражение 11 было подставлено в исходую систему 10:

$$\begin{cases} \frac{6(-16+\delta-x_2+3x_3)}{3+\delta} + 2x_2 + 5x_3 = 12, \\ \\ \frac{-16+\delta-x_2+3x_3}{3+\delta} + 4x_2 - 3x_3 = -39. \end{cases} \tag{12}$$

Система 12 была упрощена:

$$\begin{cases} 2x_2\delta + 6\delta + 33x_3 + 5x_3\delta - 96 = 12(3 + \delta), \\ 4x_2\delta + 11x_2 + \delta - 6x_3 - 3x_3\delta - 16 = -39(3 + \delta). \end{cases} \tag{13}$$

Из первого уравнения системы 13 был выражен x_2:

$$x_2 = \frac{-33x_3 - 5x_3\delta + 6\delta + 132}{2\delta}. \tag{14}$$

Выражение 14 было подставлено во второе уравнение системы 13:

$$\frac{4\delta\frac{-33x_3-5x_3\delta+6\delta+132}{2\delta} + 11\frac{-33x_3-5x_3\delta+6\delta+132}{2\delta} + \delta - 6x_3 - 3x_3\delta - 16}{3 + \delta} = -39. \tag{15}$$

Полученное выражение 15 было упрощено, в результате чего приняло следующий вид:

$$\frac{-26\delta^2x_3 - 199\delta x_3 - 363x_3 + 26\delta^2 + 562\delta + 1452}{2\delta(3 + \delta)} = -39. \tag{16}$$

Уравнение 16 было домножено на $2\delta(3 + \delta)$:

$$-26\delta^2x_3 - 199\delta x_3 - 363x_3 + 26\delta^2 + 562\delta + 1452 = -234\delta - 78\delta^2. \tag{17}$$

В уравнении 17 был вынесен x_3 за скобки и приведены однородные слагаемые:

$$x_3(-26\delta^2 - 199\delta - 363) = -104\delta^2 - 796\delta - 1452. \tag{18}$$

Тогда из 18 выражается x_3:

$$x_3 = \frac{-104\delta^2 - 796\delta - 1452}{-26\delta^2 - 199\delta - 363} = 4. \tag{19}$$

Чтобы найти x_2, в выражение 14 было подставлено значение x_3, полученное в 19:

$$x_2 = \frac{-33 \cdot 4 - 20\delta + 6\delta + 132}{2\delta} = -7. \tag{20}$$

Для того, чтобы найти x_1, в выражение 11 были подставлены полученные в 19 и 20 значения для x_3 и x_2 соответственно:

$$x_1 = \frac{-16 + \delta + 7 + 12}{3 + \delta} = 1. \tag{21}$$

Тогда, согласно 19, 20 и 21, решение модифицированной СЛАУ 9:

$$\tilde{x} = [4, -7, 1]. \tag{22}$$

Таким образом, видно, что модифицированная СЛАУ 9, полученная с помощью добавления к элементам $a11$ и $b1$ малого числа δ, имеет то же решение 22, что и СЛАУ 2 - 8.

4.4 Решения модифицированной СЛАУ при $p \in [0; 12]$

С помощью разработанных функций lu(A, permute) и solve(L, U, P, b), представленных в листингах 3 и 4, были получены решения модифицированной СЛАУ 9 в зависимости от $p \in [0; 12]$ и значения переменной permute для двух случаев:

- без частичного выбора главного элемента (permute=False),
- с частичным выбором главного элемента (permute=True).

В таблице 1 представлены решения, полученные без частичного выбора главного элемента.

Таблица 1. Решения модифицированной СЛАУ 2 для $p \in [0; 12]$ без частичного выбора главного элемента

p	$\tilde{x}_1$	$\tilde{x}_2$	$\tilde{x}_3$
0	1.000000000000	-7.000000000000	4.000000000000
1	1.000000000000	-7.000000000000	4.000000000000
2	1.000000000000	-6.999999999999	4.000000000000
3	1.000000000000	-7.000000000000	4.000000000000
4	0.999999999997	-6.999999999990	4.000000000000
5	0.999999999956	-6.999999999867	4.000000000000
6	0.999999997224	-6.999999991673	4.000000000000
7	1.000000023315	-7.000000069944	4.000000000000
8	1.000000233147	-7.000000699441	4.000000000000
9	1.000000000000	-7.000000000000	4.000000000000
10	0.999982236433	-6.999946709299	4.000000000000
11	1.000088817835	-7.000266453504	4.000000000000
12	0.996780996781	-6.990342990343	4.000000000000

В таблице 2 представлены решения, полученные с частичным выбором главного элемента.

Таблица 2. Решения модифицированной СЛАУ 2 для $p \in [0; 12]$ с частичным выбором главного элемента

p	$\tilde{x}_1$	$\tilde{x}_2$	$\tilde{x}_3$
0	1.000000000000	-7.000000000000	4.000000000000
1	1.000000000000	-7.000000000000	4.000000000000
2	1.000000000000	-7.000000000000	4.000000000000
3	1.000000000000	-7.000000000000	4.000000000000
4	1.000000000000	-7.000000000000	4.000000000000
5	1.000000000000	-7.000000000000	4.000000000000
6	1.000000000000	-7.000000000000	4.000000000000
7	1.000000000000	-7.000000000000	4.000000000000
8	1.000000000000	-7.000000000000	4.000000000000
9	1.000000000000	-7.000000000000	4.000000000000
10	1.000000000000	-7.000000000000	4.000000000000
11	1.000000000000	-7.000000000000	4.000000000000
12	1.000000000000	-7.000000000000	4.000000000000

Из результатов, приведенных в таблицах 1 и 2, следует, что решения, полученные с помощью частичного выбора главного элемента, являются точными, их погрешности чрезвычайно малы и сопоставимы с машинным эпсилон. Решения, полученные без частичного выбора главного элемента имеют значения, отклоняющиеся от точного.

4.5 Относительная погрешность вычислений

Относительная погрешность вычислений вычисляется по следующей формуле:

$$E = \frac{\|\hat{x} - \tilde{x}\|_\infty}{\|\tilde{x}\|_\infty}, \tag{23}$$

где $\|\hat{x}\|$ - норма вектора 18, а $\|\tilde{x}\|$ - норма вектора решения модифицированной СЛАУ 9. Норма вектора определяется следующим выражением:

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}, \text{ где } n \text{ - размерность вектора.} \tag{24}$$

В соответствии с 23 и 24 были получены графики зависимости относительной погрешности E для решений модифицированной СЛАУ 9 с частичным выбором главного элемента и без него от p.
Для более наглядного представления были взяты 100 линейно распределенных точек из отрезка $[0; 12]$. Графики приведены на рисунке 1.

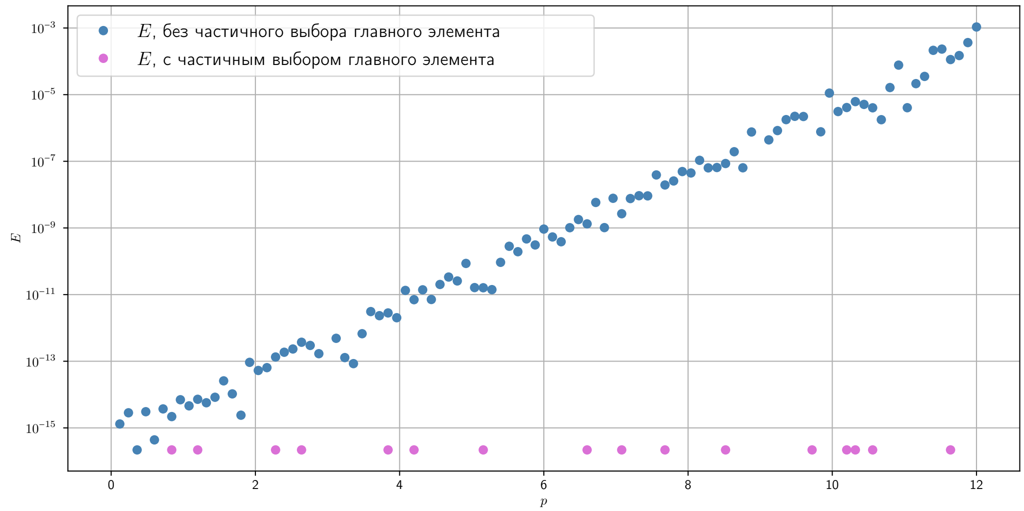

Рис. 1. График зависимости относительной погрешности вычисления E от p

Как можно заметить на приведенных на рисунке 1 графикам, при решении модифицированной СЛАУ 9 без частичного выбора главного элемента, относительная погрешность возрастает с увеличением p. Это связано с тем, что на k-ой итерации прямого хода метода Гаусса элемент $c_{kk}^{(k)}$ мал по сравнению с $c_{ik}^{(k)}$. В таком случае результат их деления m_{ik} будет иметь большое значение, что приводит к усилению погрешности округления. При этом, с увеличением p, опорный элемент главной диагонали $c_{kk}^{(k)}$ уменьшается, что приводит к усилению накопления погрешности. Однако решение СЛАУ 9 методом $\boldsymbol{LU}$-разложения с частичным выбором главного элемента устраняет эту проблему. В таком случае относительная погрешность вычислений равна нулю, либо принимает значения, сравнимые с машинным эпсилон. Это связано с тем, что, при использовании данного метода, выбирается наибольший во всем столбце элемент $c_{kk}^{(k)}$, на который далее будет производиться деление.

Тогда можно сделать вывод, что решение СЛАУ методом $\boldsymbol{LU}$-разложения без частичного выбора главного элемента является вычислительно неустойчивой операцией. Метод $\boldsymbol{LU}$-разложения с частичным выбором главного элемента, наоборот, является вычислительно устойчивой операцией.

Потенциальному пользователю функций lu(A, permute) и solve(L, U, P, b) стоит выбирать метод решения с частичным выбором главного элемента, поскольку он подходит для задач, чувствительных к погрешности округления, или имеющих достаточно малые в сравнении с остальными элементы главной диагонали матрицы $\boldsymbol{A}$.

5 Заключение

1. В ходе выполнения лабораторной работы были разработаны функции $\boldsymbol{LU}$ и $\boldsymbol{LUP}$-разложения матрицы. Также были разработаны функции решения СЛАУ

с помощью $\boldsymbol{LU}$ и $\boldsymbol{LUP}$-разложения.

2. Было приведено доказательство необходимости перестановки строк для решения СЛАУ 2.

3. Было доказано, что модифицированная СЛАУ 9, полученная с помощью добавления к первым элементам матриц $\boldsymbol{A}$ и $\boldsymbol{b}$ малого числа 10^{-p}, где p - произвольное целое число, имеет то же решение, что и исходная СЛАУ 2.

4. На основании построенных графиков зависимости относительной погрешности E от p, был сделан вывод о вычислительной устойчивости $\boldsymbol{LU}$-разложения в случае с использованием частичного выбора главного элемента и о его вычислительной неустойчивости в случае без использования частичного выбора главного элемента.

5. Были даны рекомендации потенциальному пользователю разработанных в ходе лабораторной работы функций lu(A, permute) и solve(L, U, P, b).

Список использованных источников

1. Першин А.Ю. Лекции по курсу «Вычислительная математика». Москва, 2018-2021. С. 140. URL: `https://archrk6.bmstu.ru/index.php/f/810046`.

2. Соколов, А.П. Инструкция по выполнению лабораторных работ (общая). Москва: Соколов, А.П., 2018-2021. С. 9. URL: `https://archrk6.bmstu.ru`. (облачный сервис кафедры РК6).

3. Соколов, А.П. Инструкция по выполнению заданий к семинарским занятиям (общая). Москва: Соколов, А.П., 2018-2022. С. 7. URL: `https://archrk6.bmstu.ru`. (облачный сервис кафедры РК6).

4. Першин А.Ю. Сборник задач семинарских занятий по курсу «Вычислительная математика»: Учебное пособие. / Под редакцией Соколова А.П. [Электронный ресурс]. Москва, 2018-2021. С. 20. URL: `https://archrk6.bmstu.ru`. (облачный сервис кафедры РК6).

5. Першин А.Ю., Соколов А.П. Сборник постановок задач на лабораторные работы по курсу «Вычислительная математика»: Учебное пособие. [Электронный ресурс]. Москва, 2021. С. 54. URL: `https://archrk6.bmstu.ru`. (облачный сервис кафедры РК6).

Выходные данные

Горохова А.С.Отчет о выполнении лабораторной работы по дисциплине «Вычислительная математика». [Электронный ресурс] — Москва: 2022. — 15 с. URL: `https://sa2systems.ru:88` (система контроля версий кафедры РК6)

Постановка: *доцент кафедры РК-6, PhD А.Ю. Першин*
Решение и вёрстка: *студент группы РК6-56Б, Горохова А.С.*

2022, осенний семестр