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:	Новокшанов Евгений Андреевич
Группа:	РК6-56Б
Тип задания:	лабораторная работа
Тема:	LU-разложение

Студент	___________ подпись, дата	Новокшанов Е.А. Фамилия, И.О.
Преподаватель	___________ подпись, дата	___________ Фамилия, И.О.

Москва, 2022

Содержание

LU-разложение

Задание

Дана СЛАУ $A_1 x = b_2$:

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

Дана СЛАУ $A_2 x = b_2$:

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

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 $Ax = b$, где матрица A представлена в виде LU-разложения с матрицей перестановок P, т.е. $PA = LU$.

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 a_{11} и b_1 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 $\tilde{x}$.

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 log-log 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).

Цель выполнения лабораторной работы

Исследование вычислительной устойчивости LU-разложения.

1 Базовая часть

Лабораторная работа была выполнена на языке Python3.9 с использованием библиотек matplotlib версии 3.6.0 и numpy версии 1.23.3.

Пункт №1: Разработка функции для LU-разложения

LU-разложение представляет собой разложение известной матрицы A на две треугольные матрицы L и U. Данное разложение возможно для любой невырожденной матрицы A. Программная реализация алгоритма LU-разложения представляет собой преобразование матрицы C равной матрице A. Накаждой i-й итерации:

1. Каждый C_{ji} элемент делится на элемент главной диагонали C_{ii}, при $j > i$;

2. Из каждого C_{jk} элемента, находящегося ниже i-й строки и правее i-го столбца вычитается произведение $C_{ji}C_{ik}$.

В результате выполнения алгоритма получается матрица $C = L + U - E$, т.е. объединение 2-х треугольных матриц в одной. Функция lu(A) является программной реализацией алгоритма приведенного выше и приводится в Листинге 1.

Листинг 1. Программная реализация разложения матрицы A на матрицы L и U

```
def lu(A):
  n = len(A)
  C = np.array(A.copy())
  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]
```

```
    U = np.triu(C)
    L = np.tril(C, -1)
    return L, U
```

Пункт №2: Разработка функции для вычисления решения СЛАУ

Полученные матрицы L и U из первого пункта мы используем для решения СЛАУ $LUx = b$. Для этого разработана функция solve(L, U, b). Функция расчитывает итоговый столбец x по следующему алгоритму:

1. Пусть $y = Ux$. Тогда получаем 2 СЛАУ: $Ly = b$ и $y = Ux$.
2. Решаем первую СЛАУ: $Ly = b$.
3. Найденное решение y подставляем во вторую СЛАУ и находим решение: Ux = y.
4. Выводим x.

Листинг 2. Программная реализация решения СЛАУ с помощью LU-разложения

```
def solve(L, U, b):
    n = len(b)
    X = np.array([0. for i in range(n)])
    Y = np.array([0. for i in range(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):
        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
```

Пункт №3: Решение СЛАУ с помощью разработанных функций

Было получено решение СЛАУ (1) с помощью разработанных функций lu(A)(Листинг 1) и solve(L, U, b)(Листинг 2) :
$x = [-1, 2, 0, 1]^T$.
Полученное решение СЛАУ сошлось с точным ее решением $x = [-1, 2, 0, 1]^T$.

2 Продвинутая часть

Пункт №1. Доказательство необходимости перестановки при решении СЛАУ методом Гаусса.

Для доказательства была рассмотрена следующая расширенная матрица:

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

На первой итерации метода Гаусса обнуляются элементы столбца под первым элементом главной диагонали. Для этого из все строк с номером больше 1 вычитается первая строка, домноженная на $\frac{\tilde{a_{i1}}}{\tilde{a_{11}}}$:

$$a_{ij}^{(1)} = a_{ij} - a_{1j}\frac{a_{i1}}{a_{11}}, i = 2, \dots, n;$$

где n - размер матрицы.
Матрица после этого принимает следующий вид:

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

На следующих итерациях необходимо обнулить все элементы под главной диагональю, однако для этого требуется из третьей строки вычесть вторую, домноженную на $\frac{\tilde{a_{32}}}{a_{22}}$. Диагональный элемент a_{22} был обнулен, следовательно дальнешие действия невозможны. Для продолжения решения СЛАУ (2) методом Гаусса необходима перестановка строк матриц таким образом, чтобы вторым диагональным элементом стало чило не равное нулю(меняем местами 2 и 3 строку). Вторая и третья строки расширенной матрицы (3) были переставлены между собой:

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

Тогда решение СЛАУ:

$$x = [1, -7, 4]^T \tag{4}$$

Пункт №2,3,4: Программная реализация решения СЛАУ с возможностью частичного выбора главного элемента

Для решения некоторых СЛАУ требуется делать перестановку строк, что не учитывало наше решение при реализации функции lu(A). Подобные перестановки можно учесть, используя матрицу перестановок P, которая образуется с помощью перестановки строк единичной матрицы E. При домножении матрицы P на матрицу A справа получается преобразованная матрица, которая учитывает все необходимые перестановки. Следовательно LU-разложение матрицы A находится следующим образом:

$$PA = LU.$$

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

$$|a_{kk}^{(k)}| = \max_{2 \leqslant i \leqslant n} |a_{ik}^{(k)}|.$$

Перестановки будут фиксироваться в соответствующей матрице перестановок P. Алгоритм программной реализации LU-разложения с возможнотью частичного выбора главного элемента является модифицированным алгоритмом из 1-го пункта базовой части. На каждой i-той итерации перед преобразованием элементов с помощью раннее указанного алгоритма, следует провести поиск опорного(максимального по модулю) элемента среди элементов i-того столбца, находящихся не выше i-той строки. После этого строка с максимальным по модулю элементом меняется местами с i-той строкой. Аналогичные перестановки выполняются и со строками матрицы P. В Листинге 3 представлена программная реализация модифицированной функции lu(A, permute), которая возвращает треугольные матрицы L, U и матрицу перестановок P.

Листинг 3. Программная реализация LU-разложения с возможностью частичного выбора главного элемента

```python
def lu_perm(A: list, permute: bool):
  n = len(A)
  C = np.array(A.copy())
  P = np.array([np.array([0**(abs(i-j)) for j in range(n)]) for i in range(n)],
      dtype=np.float64)
  for i in range(n):
    max_abs = 0
    max_row = -1
    for j in range(i, n):
      if(abs(C[j][i]) > max_abs):
        max_abs = abs(C[j][i])
        max_row = j
    if(max_abs!=0):
      if(permute):
        P[[max_row, i]] = P[[i, max_row]]
        C[[max_row, i]] = C[[i, max_row]]
      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]
  U = np.triu(C)
  L = np.tril(C, -1)
  return L, U, P
```

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 P. В таком случае решение СЛАУ будет находиться по следующей формуле:

$$PAx = Pb.$$

Программная реализация модифицированной функции solve(L, U, P, b) представлена в Листинге 4. Функция принимает на вход матрицу b, нижнюю и верхнюю треугольные матрицы L и U, полученные в результате работы функции lu(A, permute), а так же матрицу перестановок P. В результате своей работы функция возвращает матрицу решений x.

Листинг 4. Программная реализация решения СЛАУ с использованим матрицы перестановок P

```
def solve(L, U, P, b):
  n = len(b)
  X = np.zeros(n, dtype=np.float64)
  Y = np.zeros(n, dtype=np.float64)
  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):
    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
```

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

$$\hat{x} = [1, -7, 4]^T. \tag{5}$$

Данное решение сошлось с решением (4).

Пункт №5: Доказательство, что модифицированная и не модифицированная СЛАУ имеют одинаковое решение

Модифицированная СЛАУ (2) имеет следущий вид:

$$\begin{bmatrix} 3+\lambda & 1 & -3 \\ 6 & 2 & 5 \\ 1 & 4 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -16+\lambda \\ 12 \\ -39 \end{bmatrix}, \lambda = 10^{-p}, p \in N. \tag{6}$$

Данную СЛАУ можно переписать в виде системы уравнений:

$$\begin{cases} (3+\lambda)x_1 + 1x_2 - 3x_3 = -16 + \lambda \\ 6x_1 + 2x_2 + 5x_3 = 12 \\ 1x_1 + 4x_2 - 3x_3 = -39 \end{cases} \tag{7}$$

Выражение x_1 через x_2 и x_3:

$$\begin{cases} x_1 = \frac{-16+\lambda-x_2+3x_3}{3+\lambda} \\ 6x_1 + 2x_2 + 5x_3 = 12 \\ x_1 + 4x_2 - 3x_3 = -39 \end{cases} \tag{8}$$

Подстановка x_1 выраженного в системе(8) в оставшиеся уравнения этой системы:

$$\begin{cases} x_1 = \frac{-16+\lambda-x_2+3x_3}{3+\lambda} \\ \frac{6(-16+\lambda-x_2+3x_3)}{3+\lambda} + 2x_2 + 5x_3 = 12 \\ \frac{-16+\lambda-x_2+3x_3}{3+\lambda} + 4x_2 - 3x_3 = -39 \end{cases} \tag{9}$$

$$\begin{cases} x_1 = \frac{-16+\lambda-x_2+3x_3}{3+\lambda} \\ -96 + 6\lambda + x_2(-6 + 6 + 2\lambda) + x_3(18 + 15 + 5\lambda) = 12(3 + \lambda) \\ -16 + \lambda + x_2(-1 + 12 + 4\lambda) + x_3(3 - 9 - 3\lambda) = -39(3 + \lambda) \end{cases} \tag{10}$$

$$\begin{cases} x_1 = \frac{-16+\lambda-x_2+3x_3}{3+\lambda} \\ x_2 \cdot 2\lambda + x_3(33 + 5\lambda) = 132 + 6\lambda \\ x_2(11 + 4\lambda) + x_3(-6 - 3\lambda) = 23 + 40\lambda \end{cases} \tag{11}$$

Выражение x_2 через x_3 из второго уравнения системы(11):

$$\begin{cases} x_1 = \frac{-16+\lambda-x_2+3x_3}{3+\lambda} \\ x_2 = \frac{-x_3(33+5\lambda)+132+6\lambda}{2\lambda} \\ x_2(11 + 4\lambda) + x_3(-6 - 3\lambda) = 23 + 40\lambda \end{cases} \tag{12}$$

Подстановка x_2 выраженного в системе(13) в оставшееся уравнение:

$$\begin{cases} x_1 = \frac{-16+\lambda-x_2+3x_3}{3+\lambda} \\ x_2 = \frac{-x_3(33+5\lambda)+132+6\lambda}{2\lambda} \\ \frac{(-x_3(33+5\lambda)+132+6\lambda)(11+4\lambda)}{2\lambda} + x_3(-6 - 3\lambda) = 23 + 40\lambda \end{cases} \tag{13}$$

Вид системы(13) после упрощения:

$$\begin{cases} x_1 = \frac{-16+\lambda-x_2+3x_3}{3+\lambda} \\ x_2 = \frac{-x_3(33+5\lambda)+132+6\lambda}{2\lambda} \\ x_3(-26\lambda^2 - 199\lambda - 363) = -104\lambda^2 - 796\lambda - 1452 \end{cases} \tag{14}$$

Выразим x_3:

$$\begin{cases} x_1 = \frac{-16+\lambda-x_2+3x_3}{3+\lambda} \\ x_2 = \frac{-x_3(33+5\lambda)+132+6\lambda}{2\lambda} \\ x_3 = \frac{-104\lambda^2-796\lambda-1452}{-26\lambda^2-199\lambda-363} = 4 \end{cases} \tag{15}$$

Подставка полученного значения x_3 в системе(15) в другие уравнения системы:

$$\begin{cases} x_1 = 1 \\ x_2 = -7 \\ x_3 = 4 \end{cases}$$

Следовательно, решение модифицированной СЛАУ имеет следующий вид:

$$\tilde{x} = [1, -7, 4]^T. \tag{16}$$

Таким образом, модифицированная СЛАУ имеет то же решение, что и СЛАУ без изменений a_{11} и b_1.

Пункт №6: Решение модифицированной СЛАУ с частичным выбором главного элемента и без него

С помощью разработанных функций lu(A, permute) и solve(L, U, P, b), представленных в Листингах 3 и 4, были получены следующие решения модифицированной СЛАУ (6) в зависимости от $p \in 0, \ldots, 12$ 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 (6) для $p \in [0; 12]$ без частичного выбора главного элемента

p	$\tilde{x_1}$	$\tilde{x_2}$	$\tilde{x_3}$
0	1.0000000000000000	-7.0000000000000000	4.0000000000000000
1	1.0000000000000022	-7.0000000000000036	4.0000000000000009
2	0.9999999999995326	-6.9999999999985967	3.9999999999999987
3	0.9999999999999999	-7.0000000000000000	4.0000000000000000
4	0.9999999999966699	-6.9999999999900080	4.0000000000000000
5	0.9999999999555911	-6.9999999998667723	4.0000000000000000
6	0.9999999972244419	-6.9999999916733247	4.0000000000000000
7	1.0000000233146848	-7.0000000699440532	4.0000000000000009
8	1.0000002331468394	-7.0000006994405179	4.0000000000000009
9	1.0000000000000000	-7.0000000000000000	4.0000000000000000
10	0.9999822364330750	-6.9999467092992269	3.9999999999999991
11	1.0000888178346223	-7.0002664535038637	4.0000000000000009
12	0.9967809967809971	-6.9903429903429908	3.9999999999999996

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

Таблица 2. Решения модифицированной СЛАУ (7) для $p \in [0; 12]$ с частичным выбором главного элемента

p	$\tilde{x_1}$	$\tilde{x_2}$	$\tilde{x_3}$
0	1.0000000000000002	-7.0000000000000009	4.0000000000000000
1	1.0000000000000002	-7.0000000000000009	4.0000000000000000
2	0.9999999999999994	-7.0000000000000000	4.0000000000000009
3	1.0000000000000002	-7.0000000000000009	4.0000000000000000
4	1.0000000000000011	-7.0000000000000018	3.9999999999999996
5	1.0000000000000002	-7.0000000000000009	4.0000000000000000
6	0.9999999999999994	-7.0000000000000000	4.0000000000000009
7	1.0000000000000002	-7.0000000000000009	4.0000000000000000
8	1.0000000000000011	-7.0000000000000018	3.9999999999999996
9	1.0000000000000002	-7.0000000000000009	4.0000000000000000
10	1.0000000000000002	-7.0000000000000009	4.0000000000000000
11	1.0000000000000002	-7.0000000000000009	4.0000000000000000
12	1.0000000000000002	-7.0000000000000009	4.0000000000000000

Сравнивая результаты решений с частичным выбором главного элемента и без для разных p, которые представлены в таблицах 1 и 2 можно сделать вывод:
решения, полученные с применением частичного выбора главного элемента, являются точными, их погрешности сопоставимы с машинным эпсилон. Однако в решение без частичного выбора наблюдаются видимые неточности.

Пункт №7: График зависимости относительной погрешности вычисления E от p

По полученным данным таблицы 1 и 2 получен график зависимости E от p, где E равен:

$$E = \frac{||\hat{x} - \tilde{x}||_\infty}{||\hat{x}||_\infty}.$$

Зависимость отнсительной погрешности вычислений от p показана на рисунке 1.

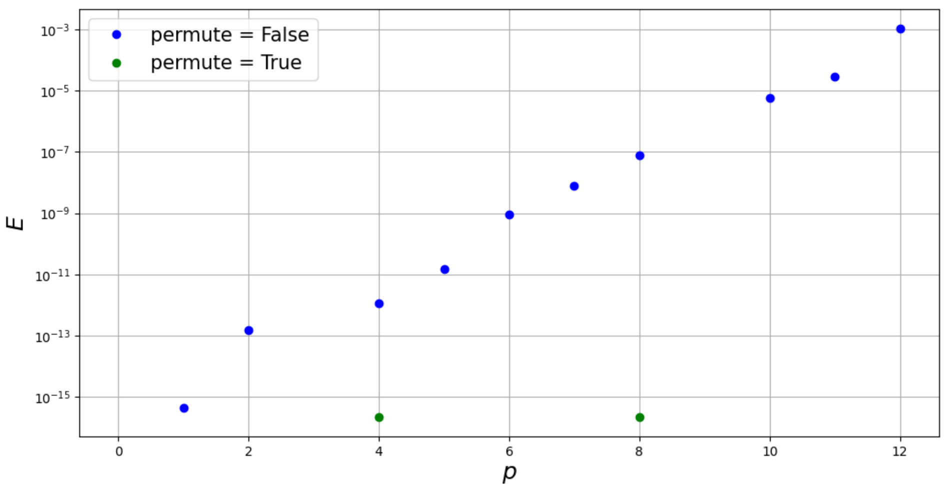

Рис. 1. График зависимости отнсительной погрешности вычислений от p

Для более наглядного представления взяты 100 линейно распределенных точек из отрезка $p \in [0; 12]$. Графики приведены на рисунке 2.

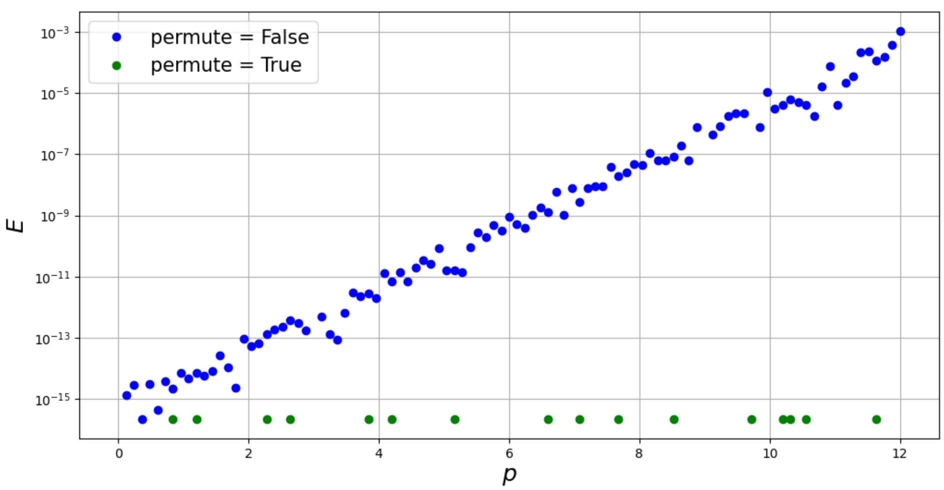

Рис. 2. График зависимости относительной погрешности вычисления E от p

В соответствии с графиками представленными на рисунке 1 и 2 можно сделать следующие выводы:

1. Относительная погрешность при решении с частичным выбором главного элемента равна нулю либо сопоставима с машинным эпсилон. Следовательно данный метод является вычислительно устойчивым;

2. Относительная погрешность при нахождении решения без частичного выбора главного элемента является вычислительно неустойчивой, так как наблюдается сремительное увеличение погрешности при увеличении p.

При решении СЛАУ на k-й итерации возможно деление одного элемента на элемент, который много меньше первого. Из этого следует появление существенной погрешности, и дальнейшее умножение на результат их деления, который принимает большое значение.

Решение СЛАУ с частичным выбором главного элемента показывает большую точность, однако меньшую скорость работы программы. Следовательно в некоторых задачах следует использовать решение более быстрое СЛАУ без частичного выбора главного элемента, который однако может привести к неправильному решению вследствии высокой погрешности.

Заключение

1. Были разработаны функции нахождения LU-разложения с матрицей перестановок P и без нее;

2. Были разработаны функции решения с помощью функций для LU-разложения;

3. Дана оценка вычислительной устойчивости разработанных функций;

4. Доказано, что модифицированная СЛАУ имеет то же решение, что и СЛАУ без добавления 10^{-p} к элементам матрицы A.

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

Новокшанов Е.А.Отчет о выполнении лабораторной работы по дисциплине «Вычислительная математика». [Электронный ресурс] — Москва: 2022. — 14 с. URL: https://sa2systems.ru:88 (система контроля версий кафедры РК6)

2022, осенний семестр