МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени М.В.Ломоносова

Факультет вычислительной математики и кибернетики

Компьютерный практикум по учебному курсу

«ВВЕДЕНИЕ В ЧИСЛЕННЫЕ МЕТОДЫ»

ЗАДАНИЕ № 1_1

ОТЧЕТ

о выполненном задании

студента 203 учебной группы факультета ВМК МГУ

Кудисова Артёма Аркадьевича

гор. Москва

2020 год

Цель работы

Изучить классический метод Гаусса (а также модифицированный метод Гаусса), применяемый для решения системы линейных алгебраических уравнений.

Постановка задачи

Дана система уравнений Ax=f порядка n×n с невырожденной матрицей A. Необходимо написать программу, решающую данную систему линейных алгебраических уравнений заданного пользователем размера (n – параметр программы) методом Гаусса и методом Гаусса с выбором главного элемента.

Необходимо предусмотреть возможность задания элементов матрицы системы и ее правой части как во входном файле данных, так и путем задания специальных формул.

Основные цели

1. Решить заданную СЛАУ методом Гаусса и методом Гаусса с выбором главного элемента;
2. Вычислить определитель матрицы det(A);
3. Вычислить обратную матрицу A^{-1};
4. Определить число обусловленности M_A=$||A||\times||A^{-1}||$;
5. Исследовать вопрос вычислительной устойчивости метода Гаусса (при больших значениях параметра n);
6. Правильность решения СЛАУ подтвердить системой тестов (например, можно использовать ресурсы on-line системы http://www.wolframalpha.com, пакета Maple и т.п.).

Метод Гаусса

Пусть дана система $a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n = f_i$, $i = \overline{1, n}$

В матричном виде ее можно представить как $Ax = f$, $A \in \mathbb{R}^{n\times n}$, $|A| \neq 0$ (по условию), $x \in \mathbb{R}^{n\times 1}$, $f \in \mathbb{R}^{n\times 1}$

Основная суть метода Гаусса заключена в приведении заданной системы к верхнему треугольному виду и дальнейшем вычислении неизвестных $x_1, x_2, ..., x_n$ из этой треугольной системы.

Исходя из описания метод Гаусса удобно разделить на 2 этапа:

1. Прямой ход: на данном шаге мы приводим систему к верхнему треугольному виду

2. Обратный ход: последовательно отыскиваем неизвестные $x_1, x_2, ..., x_n$

Прямой ход

Не ограничивая общности, будем считать, что элемент a_{11} отличен от нуля (в противном случае найдем элемент $a_{1i} \neq 0$ и поменяем местами неизвестные x_1 и x_i, изменив соотвествующим образом нумерацию неизвестных, заметим, что данный элемент непременно существует, так как по условию предполагается, что матрица A невырождена)

Разделим первое уравнение на a_{11} и введем в первом уравнение в качестве новых коэффициентов следующие отношения

$$c_{12} = \frac{a_{12}}{a_{11}},\ c_{13} = \frac{a_{13}}{a_{11}},\ ...,\ c_{1n} = \frac{a_{1n}}{a_{11}},\ \hat{f}_1 = \frac{f_1}{a_{11}}$$

Вычтем из каждого i-ого уравнения системы, где $i = \overline{2, n}$, первое уравнение, домноженное на a_{i1}. Таким образом, мы исключим из всех уравнений, кроме первого, неизвестное x_1. После этого наша система примет вид

$$\begin{cases} x_1 + c_{12}x_2 + ... + c_{1n}x_n & = & \hat{f}_1 \\ \qquad \tilde{a}_{i2}x_2 + ... + \tilde{a}_{in}x_n & = & \tilde{f}_i,\ i = \overline{2, n} \end{cases}$$

$$\text{где} \begin{array}{rcl} \tilde{a}_{ij} & = & a_{ij} - c_{1j}a_{i1} \\ \tilde{f}_i & = & f_i - \hat{f}_1 a_{i1},\ j = \overline{2, n} \end{array}$$

Теперь выделим из этой системы ее "укороченную"версию, содержащую n-1 уравнение

$$\tilde{a}_{i2}x_2 + ... + \tilde{a}_{in}x_n = \tilde{f}_i,\ i = \overline{2, n}$$

Аналогичным образом продолжим процесс исключения и через n-1 шагов наша система примет треугольный вид

$$
\begin{aligned}
x_1 + c_{12}x_2 + c_{13}x_3 + ... + c_{1n}x_n &= \hat{f}_1 \\
x_2 + c_{23}x_3 + ... + c_{2n}x_n &= \hat{f}_2 \\
\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots & \\
x_{n-1} + c_{(n-1)n}x_n &= \hat{f}_{n-1} \\
x_n &= \hat{f}_n
\end{aligned}
$$

Обратный ход

Суть обратного хода состоит в последовательном отыскании неизвестных $x_1, x_2, ..., x_n$ из полученной системы в обратном порядке

$$
\begin{aligned}
x_n &= \hat{f}_n \\
x_{n-1} &= \hat{f}_{n-1} - c_{(n-1)n}x_n \\
x_{n-2} &= \hat{f}_{n-2} - c_{(n-2)n}x_n - c_{(n-2)(n-1)}x_{n-1} \\
& \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\
x_1 &= \hat{f}_1 - c_{1n}x_n - c_{1(n-1)}x_{n-1} - ... - c_{12}x_2
\end{aligned}
$$

Модификация метод Гаусса

Оригинальный метод Гаусса подвержен вычислительным ошибкам, неизбежным при компьютерных расчетах из-за невозможности точно представлять числа с плавающей точкой.

Так в ходе решения в матрице C у нас могут образоваться большие по модулю элементы c_{ij}, $i, j = \overline{1, n}$. Тогда во время обратного прохода найденные с ошибками неизвестные x_i, $i = \overline{2, n}$, будут умножаться на большие по модулю элементы матрицы C, тем самым дальше увеличивая эти ошибки.

Этого можно избежать, если все элементы матрицы C наоборот удовлетворяют условию $|c_{ij}| \leq 1$, $i, j = \overline{1, n}$

Выполнения этого условия нетрудно добиться, если на каждом шаге метода Гаусса ведущим элементом будет становиться наибольший по модулю. Пусть на шаге i наибольшим по модулю является элемент $\tilde{a}_{ij}$. Тогда поменяем в системе i-ый и j-ый столбцы местами, изменив соответствующим образом нумерацию неизвестных, и продолжим выполнять Гаусса.

Определитель

Вычислить определитель матрицы системы в ходе метода Гаусса нетрудно, так как матрица системы приводится к диагональному виду с единицами на главной диагонали с помощью линейных преобразований строк (без их перестановок), перестановок столбцов и делений строк на ведующий элемент.

Из курса линейной алгебры хорошо известно, что:

- Определитель диагональной матрицы равен произведению ее диагональных элементов.
- Линейные преобразования строк (без перестановок) не меняют определитель матрицы.
- Перестановки столбцов изменяют только знак определителя.
- Деление строки матрицы на ненулевое число a ведет к тому, что и определитель матрицы также уменьшается в a раз.

Таким образом, определитель матрицы системы равен произведению ведущих элементов, умноженному на $(-1)^p$, где p - число перестановок столбцов.

Обратная матрица

В ходе прямого и обратного проходов метода Гаусса матрица A системы приводится к единичному виду I. Это означает, что построить обратную матрицу чрезвычайно просто.

Из курса линейной алгебры известно, что $[A|I] -> [I|A^{-1}]$, причем все линейные преобразования происходят над строками.

Однако, в ходе нашего метода Гаусса возможны также перестановки столбцов, из-за чего в действительности мы ищем обратную матрицу не исходной матрицы A, а матрицы B, полученной из A некоторой перестановой столбцов. Перейти от полученной обратной матрицы B^{-1} к A^{-1} можно путем соответствующей перестановки строк B^{-1}.

Описание программы

Программа написана на языке python с использованием библиотеки numpy. Данная библиотека предоставляет широкие возможности для работы с данными в матричном виде. Несмотря на то, что в numpy реализовано огромное количество функций, легко вычисляющих в том числе определители и обратные матрицы, были использованы лишь базовые возможности библиотки, такие как перемножение матриц и их транспонирование. (Спектральная норма матрицы, необходимая для отыскания числа обусловленности, рассчитывалась при помощи встроенной функции numpy.linalg.norm[...., 2])

Программа состоит из 5 функций:

1. def forward_pass(matrix: np.ndarray, f: np.ndarray, select: bool = False) -> dict

   Функция forward_pass реализует прямой проход метода Гаусса.

   Аргументы:

   - matrix - матрица системы
   - f - значения уравнений системы
   - select - логическая переменная, отвечающая за использование модифицированного метода Гаусса

   Функция возвращает словарь, содержащий треугольную матрицу системы, измененные значения уравнений (в ходе прямого прохода), перестановку столбцов матрицы, посчитанный определитель и заготовку для обратной матрицы.

2. def backward_pass(matrix: np.ndarray, f: np.ndarray, inv_blank: np.ndarray, perm: np.ndarray) -> dict

   Данная функция реализует обратный проход метода Гаусса.

   Аргументы:

   - matrix - треугольная матрица системы
   - f - значения уравнений
   - inv_blank - заготовка обратной матрицы
   - perm - перестановка столбцов матрицы

Функция возвращает словарь, содержащий значения неизвестных и обратную матрицу.

3. def dif(first: np.ndarray, second: np.ndarray) -> float

   Принимает 2 вектора одинаковых размеров и возвращает расстояние между ними.

4. def get_equation() -> dict

   Эта функция отвечает за получение входных данных и их преобразование к нужному формату.

   Возвращается словарь, содержащий матрицу системы, значения уравнений системы и логическую переменную, отвечающую за выбор между обычным методом Гаусса и его модифицированной версией.

5. def main()

   Главная функция - запускает все остальные функции в необходимом порядке, осуществляет формирование входных аргументов, распоковку возвращаемых значений и вывод ответа.

Код программы

```
import numpy as np


def forward_pass(matrix: np.ndarray,
                 f: np.ndarray,
                 select: bool = False) -> dict:
    correct = True
    count_perm = 0
    diag_elements = list()

    n = matrix.shape[0]
    perm = np.arange(n)

    inv_blank = np.diag(np.ones(n, dtype=np.float64))
    matrix_t = matrix.T

    for ind in range(n):
        tmp = matrix[ind][ind]

        if select or tmp == 0:
            i = np.argmax(np.abs(matrix[ind]))
```

```python
            if i != ind:
                matrix_t[[i, ind]] = matrix_t[[ind, i]]

                count_perm += 1
                perm[[ind, i]] = i, ind

        diag_element = matrix[ind][ind]
        diag_elements.append(diag_element)

        if diag_element == 0:
            correct = False
            break

        matrix[ind] = matrix[ind] / diag_element
        inv_blank[ind] = inv_blank[ind] / diag_element
        f[ind] /= diag_element

        f[ind + 1:] -= matrix_t[ind][ind + 1:] * f[ind]
        inv_blank[ind + 1:] -= np.array([matrix_t[ind][ind + 1:]]).T \
                               @ np.array([inv_blank[ind]])
        matrix[ind + 1:] -= np.array([matrix_t[ind][ind + 1:]]).T \
                            @ np.array([matrix[ind]])

    det = (-1) ** count_perm
    for element in diag_elements:
        det *= element

    return {"correct": correct,
            "matrix": matrix,
            "f": f,
            "perm": perm,
            "det": det,
            "inv_blank": inv_blank}


def backward_pass(matrix: np.ndarray,
                  f: np.ndarray,
                  inv_blank: np.ndarray,
                  perm: np.ndarray) -> dict:
    n = matrix.shape[0]
    for ind in range(1, n):
        f[:n - ind] -= matrix.T[n - ind][:n - ind] * f[n - ind]
        inv_blank[:n - ind] -= np.array([matrix.T[n - ind][:n - ind]]).T \
                               @ np.array([inv_blank[n - ind]])

    return {"ans": f[perm],
            "inv_matrix": inv_blank}
```

```
def dif(first: np.ndarray, second: np.ndarray) -> float:
    tmp = first - second
    return np.sqrt(np.sum(tmp * tmp))


def get_equation() -> dict:
    ans = {"matrix": None,
           "f": None,
           "select": None}

    ch = input("Select the leading elements? y/n ")
    if ch == 'y':
        ans["select"] = True
    elif ch == 'n':
        ans["select"] = False
    else:
        print("Wrong input")
        return ans

    ch = input("\nWould you like to enter formulas for matrices "
               "or input file?\n"
               "Print 1 in the first case and 2 in the second: ")
    if ch == '1':
        try:
            n = int(input("\nFirst specify the count of "
                          "unknown variables: "))
            m = int(input("Now enter the value of m: "))

            ans["f"] = np.array([m * n - x ** 3 for x in range(1, n + 1)],
                                dtype=np.float64)
            ans["matrix"] = np.array([[(i + j) / (m + n) if i == j
                                       else n + m * m + j / m + i / n
                                       for j in range(1, n + 1)]
                                      for i in range(1, n + 1)],
                                     dtype=np.float64)

        except (ValueError, ZeroDivisionError):
            print("Wrong input")
            return ans

    elif ch == '2':
        try:

            n = int(input("\nFirst specify the count "
                          "of unknown variables: "))
            file = input("Enter file name: ")
```

```
            matrix = []
            f = []

            with open(file, "r") as file:
                for line in file:
                    nums = [float(x) for x in line.strip().split()]
                    if len(nums) != n + 1:
                        print("Wrong input")
                        return ans

                    matrix.append(nums[:n])
                    f.append(nums[n])

            ans["matrix"] = np.array(matrix, dtype=np.float64)
            ans["f"] = np.array([float(x) for x in f], dtype=np.float64)

        except ValueError:
            print("Wrong input")
            return ans

    else:
        print("Wrong input")

    return ans


def main():
    res = get_equation()

    matrix = res["matrix"]
    f = res["f"]
    select = res["select"]

    if matrix is None:
        return

    f_copy = f.copy()
    matrix_copy = matrix.copy()
    res = forward_pass(matrix, f, select)

    if res["correct"] is False:
        print("\nDeterminant is 0!")
        print("Violated the conditions of the problem")
        return

    perm = res["perm"]
    det = res["det"]
```

```
    matrix = res["matrix"]
    inv_blank = res["inv_blank"]
    f = res["f"]

    print("\nDeterminant is " + str(det))

    ans = backward_pass(matrix, f, inv_blank, perm)
    print("\nAnswer is ", end="")
    print(ans["ans"])
    print(f"\nResidual is {dif(matrix_copy @ ans['ans'].T, f_copy)}")

    cond_num = np.linalg.norm(matrix_copy, 2) * \
               np.linalg.norm(ans['inv_matrix'], 2)

    print(f"\nCondition number is {cond_num}")
    print("\nInverse matrix is")
    print(ans["inv_matrix"])


if __name__ == '__main__':
    main()
```

Тестирование

Приложение 1 - 13, приложение 2 (1 - 4)
Для проверки решений использовался ресурс www.wolframalpha.com

Используемые обозначения:

1. Determinant - определитель
2. Answer - найденные значения неизвестных
3. Residual - невязка решения
4. Condition number - число обусловленности
5. Inverse matrix - обратная матрица

1. $$\begin{cases} 3x_1 - 2x_2 + 2x_3 - 2x_4 &= 8 \\ 2x_1 - x_2 + 2x_3 &= 4 \\ 2x_1 + x_2 + 4x_3 + 8x_4 &= -1 \\ x_1 + 3x_2 - 6x_3 + 2x_4 &= 3 \end{cases}$$

   - **Обычный метод Гауса**

```
Determinant is 24.000000000000014

Answer is [ 2.  -3.  -1.5  0.5]

Residual is 2.8780270159990897e-15

Condition number is 108.07594171827499

Inverse matrix is
[[-0.5         1.33333333 -0.16666667  0.16666667]
 [-5.          8.66666667 -1.33333333  0.33333333]
 [-2.          3.5        -0.5        -0.        ]
 [ 1.75       -3.16666667  0.58333333 -0.08333333]]
```

- **Модифицированный метод Гаусса**

```
Determinant is 24.00000000000001

Answer is [ 2.  -3.  -1.5  0.5]

Residual is 4.440892098500626e-16

Condition number is 108.07594171827493

Inverse matrix is
[[-0.5         1.33333333 -0.16666667  0.16666667]
 [-5.          8.66666667 -1.33333333  0.33333333]
 [-2.          3.5        -0.5        -0.        ]
 [ 1.75       -3.16666667  0.58333333 -0.08333333]]
```

- **www.wolframalpha.com**

```
Determinant is 24

Answer is [ 2  -3  -1.5  0.5]

Inverse matrix is
             [[  -6    16    -2    2]
1/12    x     [ -60   104   -16    4]
              [ -24    42    -6    0]
              [  21   -38     7   -1]]
```

2. $$\begin{cases} 2x_1 + 3x_2 + x_3 + 2x_4 &= 4 \\ 4x_1 + 3x_2 + x_3 + x_4 &= 5 \\ x_1 - 7x_2 - x_3 - 2x_4 &= 7 \\ 2x_1 + 5x_2 + x_3 + x_4 &= 1 \end{cases}$$

- **Обычный метод Гаусса**

```
Determinant is 1.999999999999996

Answer is [-3. -5. 39. -7.]

Residual is 1.0658141036401503e-14

Condition number is 272.86505886926767

Inverse matrix is
[[ -1    2    -1   -2  ]
 [ -1    1.5  -1   -1.5]
 [  8  -14.5   9   16.5]
 [ -1    3    -2   -4  ]]
```

- **Модифицированный метод Гаусса**

```
Determinant is 1.9999999999999978

Answer is [-3. -5. 39. -7.]

Residual is 7.105427357601002e-15

Condition number is 272.8650588692672

Inverse matrix is
[[ -1.    2.   -1.   -2. ]
 [ -1.    1.5  -1.   -1.5]
 [  8.  -14.5   9.   16.5]
 [ -1.    3.   -2.   -4. ]]
```

- **www.wolframalpha.com**

```
Determinant is 2

Answer is [-3  -5  39  -7]

Inverse matrix is
            [[ -2    4   -2   -4 ]
1/2    x     [ -2    3   -2   -3 ]
             [ 16  -29   18   33 ]
             [ -2    6   -4   -8 ]]
```

3. $$\begin{cases} x_1 - x_2 + x_3 - x_4 &= 0 \\ 4x_1 + x_2 - x_4 &= 0 \\ 2x_1 + x_2 - 2x_3 + x_4 &= 0 \\ 5x_1 + x_2 - 4x_4 &= 0 \end{cases}$$

- **Обычный метод Гаусса**

```
Determinant is 21.99999999999999

Answer is [ 0.  0. -0. -0.]

Residual is 0.0

Condition number is 11.394761747443686

Inverse matrix is
[[ 2.72727273e-01  2.27272727e-01  1.36363636e-01 -9.09090909e-02]
 [-1.00000000e+00  5.00000000e-01 -5.00000000e-01 -1.11022302e-16]
 [-1.81818182e-01  6.81818182e-01 -5.90909091e-01 -2.72727273e-01]
 [ 9.09090909e-02  4.09090909e-01  4.54545455e-02 -3.63636364e-01]]
```

- **Модифицированный метод Гаусса**

```
Determinant is 21.99999999999999

Answer is [ 0.  0. -0. -0.]

Residual is 0.0

Condition number is 11.394761747443686

Inverse matrix is
[[ 2.72727273e-01  2.27272727e-01  1.36363636e-01 -9.09090909e-02]
 [-1.00000000e+00  5.00000000e-01 -5.00000000e-01 -1.11022302e-16]
 [-1.81818182e-01  6.81818182e-01 -5.90909091e-01 -2.72727273e-01]
 [ 9.09090909e-02  4.09090909e-01  4.54545455e-02 -3.63636364e-01]]
```

- **www.wolframalpha.com**

```
Determinant is 22

Answer is [ 0  0  0  0]

Inverse matrix is
             [[   6    5     3   -2]
1/22    x     [ -22   11   -11    0]
              [  -4   15   -13    0]
              [   2    9     1   -8]]
```

4. $n = 50,\ m = 15$

$f_i = mn - i^3$

$$A_{ij} = \begin{cases} (i+j)/(m+n), \ i \neq j \\ n + m^2 + \frac{j}{m} + \frac{i}{n}, \ i = j \end{cases} \text{ где } i, j = \overline{1, n}$$

- **Обычный метод Гаусса**

```
Determinant is -5.89296721859522e+123

Answer is [-120.37223965 -120.33237023 -120.24890725 -120.1000768
-119.86412267 -119.51930632 -119.04390685 -118.41622101
-117.61456315 -116.61726521 -115.40267672 -113.94916476
-112.23511397 -110.23892649 -107.939022   -105.31383762
-102.34182799  -99.00146519  -95.27123871  -91.12965549
 -86.55523985  -81.52653352  -76.02209557  -70.02050241
 -63.50034782  -56.44024285  -48.81881587  -40.61471254
 -31.80659573  -22.37314562  -12.29305956   -1.54505213
   9.89214489   22.03978256   34.91909477   48.5512983
  62.95759277   78.15916072   94.17716761  111.0327618
 128.74707463  147.34122038  166.83629632  187.25338273
 208.61354288  230.93782311  254.24725277  278.56284431
 303.90559325  330.2964782 ]

Residual is 1.6974873992478887e-05

Condition number is 49.384787435591484

Inverse matrix is
[[-3.56159525e-03  7.40306944e-05 ...
   7.40910753e-05  7.40923476e-05]
 [ 7.40181472e-05 -3.56086645e-03 ...
   7.40828587e-05  7.40841935e-05]
 [ 7.40069018e-05  7.40083266e-05 ...
   7.40746454e-05  7.40760428e-05]
 ...
 [ 7.35055455e-05  7.35098135e-05 ...
   7.37084694e-05  7.37126553e-05]
 [ 7.34945074e-05  7.34988380e-05 ...
  -3.52680256e-03  7.37046548e-05]
 [ 7.34834738e-05  7.34878669e-05 ...
   7.36923488e-05 -3.52608182e-03]]
```

Невязка решения равна 1.6974873992478887e-05

- **Модифицированный метод Гаусса**

```
Determinant is -5.892967218597558e+123

Answer is [-120.37223964 -120.33237023 -120.24890725 -120.1000768
 -119.86412267 -119.51930632 -119.04390685 -118.41622101
 -117.61456315 -116.61726521 -115.40267672 -113.94916476
 -112.23511397 -110.23892649 -107.93902199 -105.31383762
 -102.34182799  -99.00146519  -95.27123871  -91.12965549
  -86.55523986  -81.52653352  -76.02209557  -70.02050241
  -63.50034782  -56.44024285  -48.81881587  -40.61471254
```

```
 -31.80659573  -22.37314562  -12.29305956   -1.54505213
   9.89214489   22.03978255   34.91909477   48.5512983
  62.95759277   78.15916072   94.17716761  111.0327618
 128.74707463  147.34122038  166.83629632  187.25338273
 208.61354288  230.93782311  254.24725277  278.56284431
 303.90559325  330.2964782 ]

Residual is 5.813485580565809e-10

Condition number is 49.38478743571305

Inverse matrix is
[[ 7.34834738e-05  7.34878669e-05 ...
   7.36923488e-05 -3.52608182e-03]
 [ 7.40181472e-05 -3.56086645e-03 ...
   7.40828587e-05  7.40841935e-05]
 [ 7.40069018e-05  7.40083266e-05 ...
   7.40746454e-05  7.40760428e-05]
 ...
 [ 7.35055455e-05  7.35098135e-05 ...
   7.37084694e-05  7.37126553e-05]
 [ 7.34945074e-05  7.34988380e-05 ...
  -3.52680256e-03  7.37046548e-05]
 [-3.56159525e-03  7.40306944e-05 ...
   7.40910753e-05  7.40923476e-05]]
```

Невязка решения равна 5.813485580565809e-10

5. $n = 200,\ m = 10$

   $f_i = mn - i^3$

   $A_{ij} = \begin{cases} (i+j)/(m+n),\ i \neq j \\ n + m^2 + \frac{j}{m} + \frac{i}{n},\ i = j \end{cases}$ где $i, j = \overline{1, n}$

   - **Обычный метод Гаусса**

```
Determinant is -inf

Answer is [-6718.37705678 -6716.74065529 -6715.06532758 -6713.33114064
 -6711.51818455 -6709.60656839 -6707.57643386 -6705.40794401
 -6703.08128755 -6700.57667773 -6697.87435548 -6694.95458322
 -6691.79765021 -6688.3838709  -6684.69358568 -6680.70715597
 -6676.40497284 -6671.76744865 -6666.77502265 -6661.40815796
 -6655.64734145 -6649.47308641 -6642.86593076 -6635.80643514
 -6628.27518708 -6620.25279594 -6611.7198977  -6602.65715291
 -6593.04524465 -6582.86488139 -6572.09679792 -6560.72174894
 -6548.72051656 -6536.07390665 -6522.76274934 -6508.76789647
```

```
 -6494.07022808 -6478.65064574 -6462.49007457 -6445.56946656
 -6427.86979194 -6409.37205205 -6390.05726714 -6369.90648297
 -6348.90076896 -6327.02121773 -6304.24894615 -6280.56509487
 -6255.95082784 -6230.38733247 -6203.85582107 -6176.33752707
 -6147.81370947 -6118.26564963 -6087.67465421 -6056.02204844
 -6023.28918672 -5989.45744431 -5954.50821867 -5918.42293151
 -5881.18302877 -5842.76997748 -5803.16526886 -5762.35041771
 -5720.30696182 -5677.01645948 -5632.46049614 -5586.6206769
 -5539.4786316  -5491.01601158 -5441.21449311 -5390.05577212
 -5337.52157063 -5283.59363074 -5228.25371878 -5171.48362397
 -5113.26515656 -5053.58015071 -4992.41046262 -4929.73797163
 -4865.54457941 -4799.81220856 -4732.52280612 -4663.65834075
 -4593.20080264 -4521.13220665 -4447.43458755 -4372.09000218
 -4295.08053163 -4216.38827849 -4135.99536574 -4053.88394147
 -3970.03617248 -3884.43425134 -3797.06038851 -3707.89682065
 -3616.92580334 -3524.12961394 -3429.49055436 -3332.99094635
 -3234.61313155 -3134.33947928 -3032.15237449 -2928.0342273
 -2821.96746731 -2713.93454968 -2603.91794621 -2491.90015217
 -2377.86368598 -2261.79108585 -2143.66491118 -2023.4677445
 -1901.18218832 -1776.79086706 -1650.27642698 -1521.62153329
 -1390.80887594 -1257.82116355 -1122.64112768  -985.25151883
  -845.63511137  -703.77469825  -559.65309594  -413.25313964
  -264.55768576  -113.5496158     39.78817407   195.47276299
   353.52120954   513.95055119   676.7778087    842.01997467
  1009.69402661  1179.8169194   1352.40558752  1527.47694338
  1705.04788213  1885.1352751   2067.75597466  2252.92681096
  2440.66459721  2630.98612174  2823.90815495  3019.44744649
  3217.62072543  3418.44470113  3621.9360616   3828.1114747
  4036.98758918  4248.58103165  4462.9084104   4679.98631308
  4899.83130638  5122.45993677  5347.88873339  5576.13419939
  5807.21282585  6041.14107724  6277.93540131  6517.61222538
  6760.18795676  7005.67898224  7254.10166981  7505.47236702
  7759.80740012  8017.12308178  8277.43569621  8540.76151384
  8807.1167834   9076.51773485  9348.98057826  9624.52150329
  9903.15668131 10184.90226325 10469.77438183 10757.78914802
 11048.96265582 11343.31097898 11640.85017126 11941.5962663
 12245.56528357 12552.77321593 12863.23604128 13176.96971756
 13493.99018403 13814.31335932 14137.95514449 14464.9314207
 14795.25805044 15128.95087556 15466.02572264 15806.49839476
 16150.38467971 16497.70034351 16848.46113562 17202.68278411
 17560.38100138 17921.57147873 18286.26988913 18654.49188699]

Residual is 0.021392098006514395

Condition number is 205.9617419516225

Inverse matrix is
[[-3.31540318e-03  1.68680563e-05 ...
```

```
  1.65739652e-05  1.65725607e-05]
 [ 1.68627918e-05 -3.31435177e-03 ...
  1.65706310e-05  1.65692433e-05]
 [ 1.68559446e-05  1.68543954e-05 ...
  1.65672989e-05  1.65659279e-05]
 ...
 [ 1.55990969e-05  1.56010106e-05 ...
  1.59556689e-05  1.59573625e-05]
 [ 1.55930297e-05  1.55949601e-05 ...
 -3.11884586e-03  1.59544248e-05]
 [ 1.55869660e-05  1.55889131e-05 ...
  1.59497655e-05 -3.11790913e-03]]
```

Невязка решения равна 0.021392098006514395
Из-за большой размерности матрицы ($n = 200$) произошло переполнение при вычислении определителя, из-за он посчитан неверно.

- **Модифицированный метод Гаусса**

```
Determinant is -inf

Answer is [-6718.37706181 -6716.74065518 -6715.0653273  -6713.33114071
 -6711.51818329 -6709.60656822 -6707.57643394 -6705.40794411
 -6703.08128758 -6700.57667833 -6697.87435543 -6694.95458304
 -6691.7976503  -6688.38387138 -6684.69358534 -6680.70715618
 -6676.40497274 -6671.7674487  -6666.77502251 -6661.40815737
 -6655.6473412  -6649.47308655 -6642.86593063 -6635.80643524
 -6628.2751867  -6620.25279587 -6611.71989806 -6602.65715304
 -6593.04524496 -6582.86488232 -6572.09679795 -6560.72174896
 -6548.72051669 -6536.07390669 -6522.76274869 -6508.76789653
 -6494.07022815 -6478.65064552 -6462.49007466 -6445.56946553
 -6427.86979207 -6409.37205207 -6390.05726723 -6369.90648304
 -6348.90076881 -6327.02121758 -6304.24894611 -6280.56509483
 -6255.95082784 -6230.38733279 -6203.85582096 -6176.33752709
 -6147.81370948 -6118.26564983 -6087.67465328 -6056.02204837
 -6023.28918697 -5989.45744424 -5954.50821864 -5918.42293186
 -5881.18302879 -5842.76997747 -5803.16526909 -5762.35041792
 -5720.30696128 -5677.01645952 -5632.46049597 -5586.62067691
 -5539.47863152 -5491.01601189 -5441.2144929  -5390.05577229
 -5337.52157052 -5283.59363083 -5228.25371913 -5171.48362399
 -5113.26515664 -5053.58015088 -4992.41046306 -4929.73797209
 -4865.54457932 -4799.8122086  -4732.52280617 -4663.65834067
 -4593.20080307 -4521.13220668 -4447.43458707 -4372.09000207
 -4295.08053172 -4216.38827822 -4135.99536593 -4053.88394132
 -3970.03617292 -3884.43425132 -3797.0603891  -3707.89682082
 -3616.92580297 -3524.12961395 -3429.49055405 -3332.99094536
```

```
 -3234.6131318  -3134.33947904 -3032.15237452 -2928.03422735
 -2821.96746832 -2713.93454985 -2603.91794598 -2491.9001523
 -2377.86368595 -2261.79108557 -2143.66491126 -2023.46774458
 -1901.18218846 -1776.79086723 -1650.27642656 -1521.6215334
 -1390.80887599 -1257.82116382 -1122.64112756  -985.25151908
  -845.6351114   -703.77469861  -559.65309592  -413.25313956
  -264.5576868   -113.54961586    39.78817407   195.47276289
   353.52120961   513.95055239   676.77780857   842.01997469
  1009.69402653  1179.81691917  1352.40558698  1527.47694364
  1705.04788224  1885.13527525  2067.75597457  2252.92681156
  2440.66459709  2630.98612153  2823.90815483  3019.44744651
  3217.62072571  3418.44470124  3621.93606156  3828.11147486
  4036.98758906  4248.58103187  4462.90841079  4679.98631314
  4899.83130614  5122.45993688  5347.88873238  5576.13419961
  5807.21282555  6041.14107716  6277.93540149  6517.61222562
  6760.18795679  7005.67898232  7254.10166975  7505.47236678
  7759.80740135  8017.12308167  8277.43569622  8540.7615138
  8807.11678356  9076.51773503  9348.98057814  9624.52150326
  9903.15668121 10184.90226333 10469.77438147 10757.78914802
 11048.96265597 11343.31097893 11640.85017113 11941.59626747
 12245.56528357 12552.77321576 12863.23604113 13176.96971757
 13493.99018376 13814.31335925 14137.95514444 14464.93142065
 14795.25805012 15128.95087604 15466.0257226  15806.49839501
 16150.38467951 16497.70034343 16848.46113517 17202.6827843
 17560.38100151 17921.57147869 18286.26988896 18654.49188667]

Residual is 2.709132565429942e-06

Condition number is 205.96174195826018

Inverse matrix is
[[ 1.55869660e-05  1.55889131e-05 ...
   1.59497655e-05 -3.11790913e-03]
 [ 1.68627918e-05 -3.31435177e-03 ...
   1.65706310e-05  1.65692433e-05]
 [ 1.68559446e-05  1.68543954e-05 ...
   1.65672989e-05  1.65659279e-05]
 ...
 [ 1.55990969e-05  1.56010106e-05 ...
   1.59556689e-05  1.59573625e-05]
 [ 1.55930297e-05  1.55949601e-05 ...
  -3.11884586e-03  1.59544248e-05]
 [-3.31540318e-03  1.68680564e-05 ...
   1.65739652e-05  1.65725608e-05]]
```

Невязка решения равна 2.709132565429942e-06

Из-за большой размерности матрицы ($n = 200$) произошло переполнение при вычислении определителя, из-за он посчитан неверно.

Выводы

В ходе работы был рассмотрен классический метод Гаусса для решения СЛАУ, а также его модифицированная версия. На конкретных примерах была показана корректность реализации данных методов.

Стоит отметить, что модифицированный метод Гаусса оказался эффективнее и устойчивее. Так на примерах СЛАУ небольшого размера[1] видно, что невязка решения при использовании модифицированного метода Гаусса на порядок меньше, чем если бы применялась его классическая версия. Вычисленные определители также немного точнее (но это разница крайне мала).

Если же сравнивать на СЛАУ больших размеров, то и здесь модифицированный метод Гаусса показывает себя значительно лучше. Для хорошо обусловленной матрицы[2] он позволил уменьшить невязку решения на целых 5 порядков ($1.6975 \cdot 10^{-5} \ \rightarrow \ 5.8135 \cdot 10^{-10}$), для плохо обусловленной[3] - на 4 ($0.0214 \ \rightarrow \ 2.7091 \cdot 10^{-6}$).

[1] - примеры 1, 2
[2] - пример 4
[3] - пример 5