Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Вычислительная математика

Лабораторная работа №2.

Метод Гаусса — Зейделя

Выполнил:

Маликов Глеб Игоревич

Группа № P3224

Преподаватели:

Перл Ольга Вячеславовна

Хохлов Александр Алексеевич

г. Санкт-Петербург

2024

Оглавление

Задание

Решите систему линейных алгебраических уравнений, реализуя метод Гаусса-Зейделя.

Формат входных данных:

```
n
a11 a12 ... a1n b1
a21 a22 ... a2n b2
...
an1 an2 ... ann bn
```

Формат вывода:

```
x1
x2
...
xn
```

, где x1..xn - значения неизвестных.

Если для текущей матрицы нет диагонального преобладания, вам следует попытаться найти его путем перестановки столбцов или / и строк. Если после такой операции преобладание диагонали по-прежнему отсутствует, должно быть напечатано следующее сообщение:

"The system has no diagonal dominance for this method. Method of the Gauss-Seidel is not applicable.". Для этого задайте значение переменной isMethodApplicable и сообщение об ошибке.

Описание численного метода

Метод Гаусса — Зейделя решения системы линейных уравнений

Метод Гаусса-Зейделя — это итерационный метод, используемый для решения системы линейных уравнений. Хотя его можно применить к любой матрице с ненулевыми элементами на диагонали, сходимость гарантирована только в том случае, если матрица строго диагонально доминирующая или симметричная и положительно определенная.

В данной работе метод Гаусса — Зейделя будет применяться только для таких матриц, которые являются слабо диагонально доминирующими, но имеют хотя бы одно строку, где $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$

Пусть имеется система линейных уравнений с матрицей A и векторами X и B, где A и B известные значения, X вектор неизвестных переменных. Метод Гаусса – Зейделя в данном случае используется для нахождения приближённого значения вектора X. Начальное приближение для X часто приравнивают к нулевому вектору. Последующие приближения обозначаются как X^k.

Решение получается итеративно через формулу $Lx^{(k+1)} = b - Ux^{(k)}$, где матрица A разложена на нижнюю треугольную компоненту L и строго верхнюю треугольную компоненту U так, что A = L + U.

$$A = \underbrace{\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}}_{L} + \underbrace{\begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}}_{U}.$$

Рисунок 1 - Декомпозиция матрицы A

Так, система линейных уравнений может быть описана следующим образом:

$$Ax = b$$

$$(L + U)x = b$$

$$Lx = b - Ux$$

$$x^{(k+1)} = L^{-1}\left(b - Ux^{(k)}\right)$$

Предыдущую формулу можно описать для каждого значения X в k+1 итерации:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right), i = \overline{1,n}$$

Блок-схемы

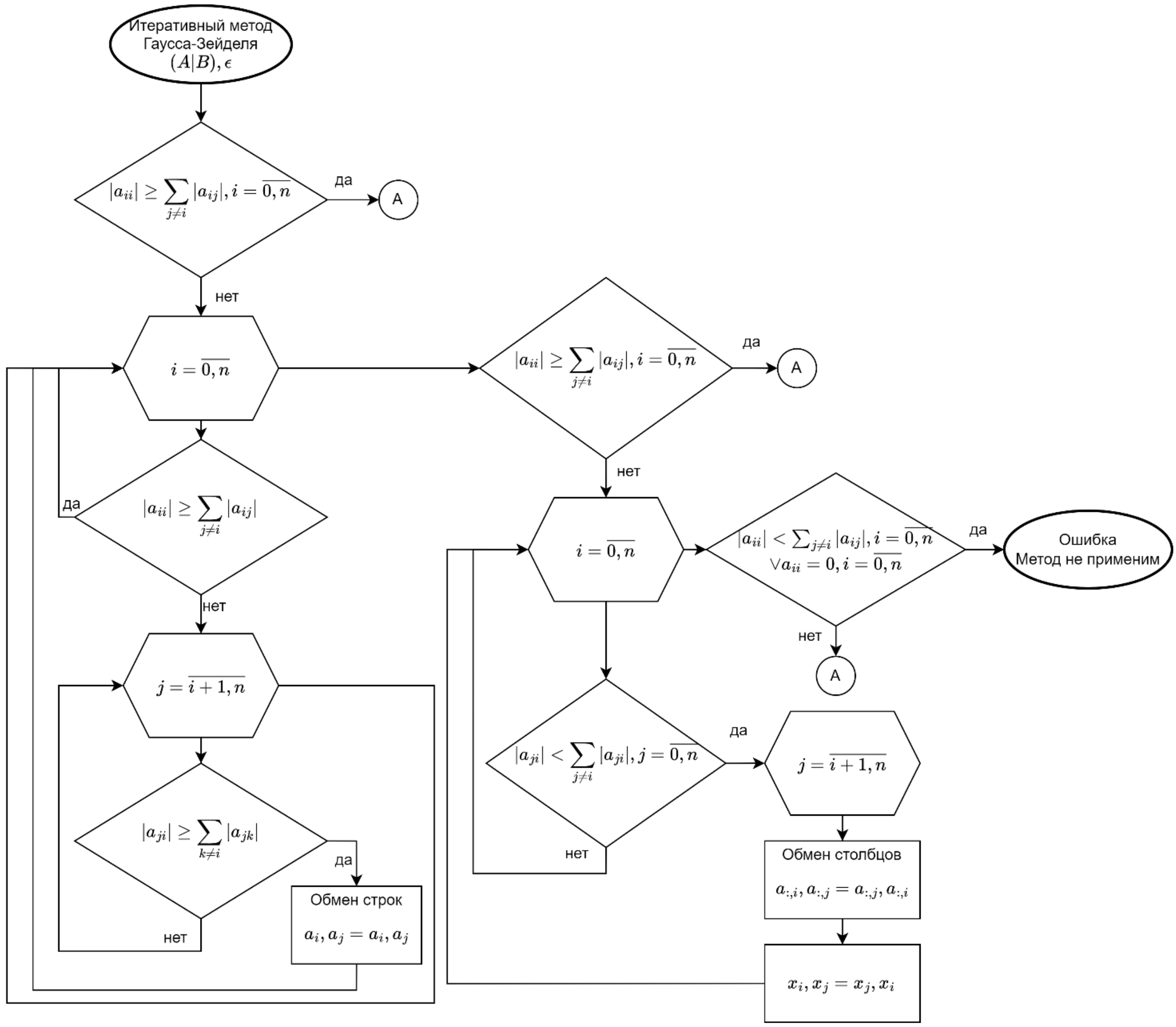

Схема 1 - Преобразования для диагонального доминирования

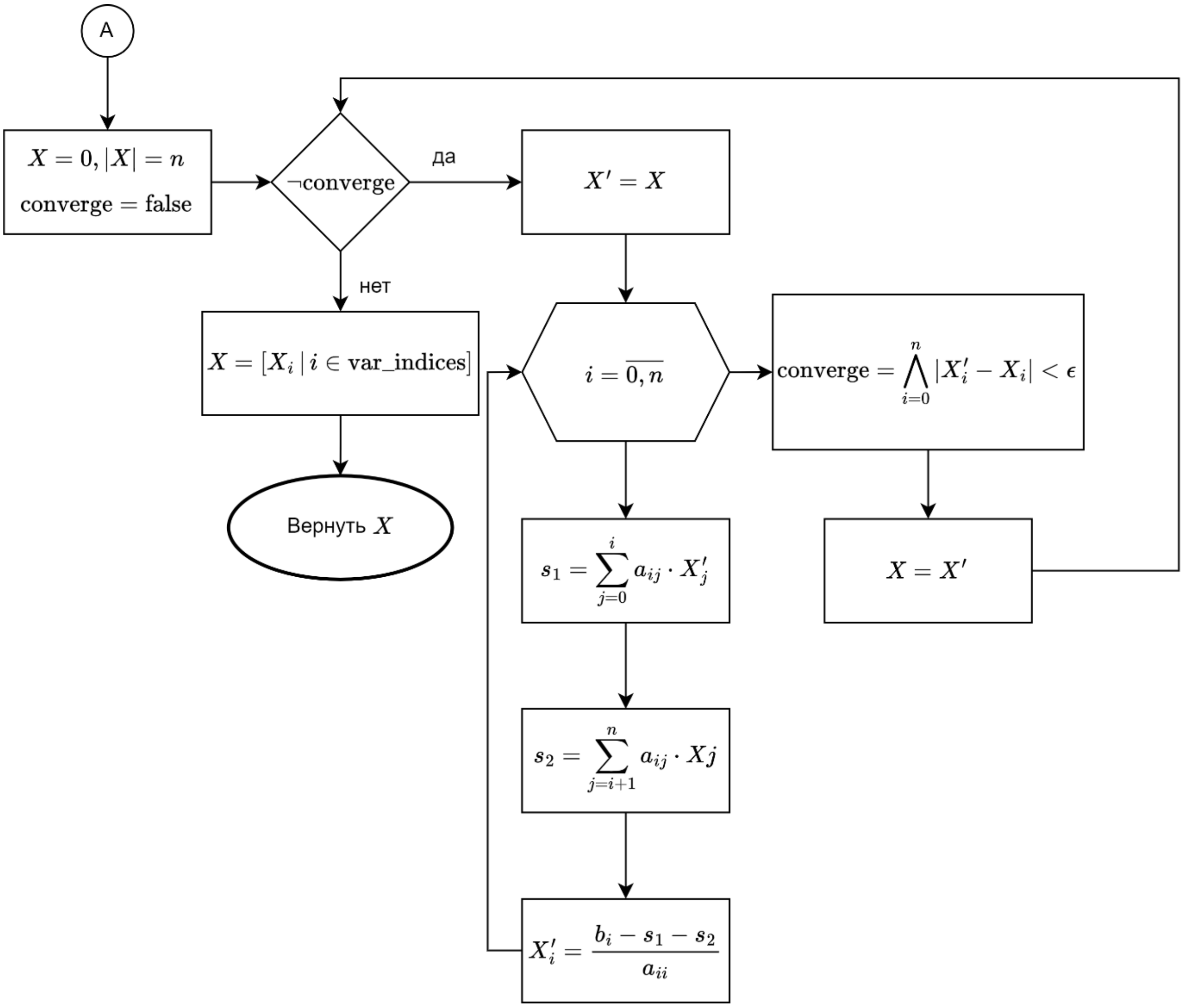

Схема 2 - Вычисление вектора X методом Гаусса-Зейделя

Код

solveByGaussSeidel

```
@staticmethod
    def solveByGaussSeidel(n, matrix, epsilon):
        """
        Решает систему линейных уравнений с использованием метода Гаусса-Зейделя.
        :param n: Количество уравнений.
        :param matrix: 2D список, представляющий коэффициентную матрицу, дополненную вектором правой части.
        :param epsilon: Желаемая точность.
        :return: Вектор решения, если успешно, пустой список в противном случае.
        """

        # Индексы переменных
        var_indices = [i for i in range(n)]

        if not is_diagonally_dominant(matrix):
            # Произвести перестановку строк и столбцов, чтобы достичь диагонального преобладания
            for i in range(n):
                if get_number_dominance_in_list(matrix[i], i) == "Not dominant":
                    # Искать строку, в которой число доминирующее
                    for j in range(i + 1, n):
                        if get_number_dominance_in_list(matrix[j], i) != "Not dominant":
                            matrix = swap_rows(matrix, i, j)
                            break
            if not is_diagonally_dominant(matrix):
                # Изменение столбцов, если изменение строк не помогло
                for i in range(n):
                    if get_number_dominance_in_list([matrix[j][i] for j in range(n)], i) == "Not dominant":
                        for j in range(i + 1, n):
                            if get_number_dominance_in_list([matrix[k][j] for k in range(n)], i) != "Not dominant":
                                matrix = swap_columns(matrix, i, j)
                                # Сохранить новый порядок переменных
                                var_indices[i], var_indices[j] = var_indices[j], var_indices[i]
                                break
        if not is_diagonally_dominant(matrix) or any(matrix[i][i] == 0 for i in range(n)):
            # Система не может иметь диагонального преобладания, метод Гаусса-Зейделя не применим
            Result.isMethodApplicable = False
```

```
            Result.errorMessage = ("The system has no diagonal dominance 
for this method. Method of the "
                                   "Gauss-Seidel is not applicable.")
            return []

        x = gauss_seidel(matrix, epsilon)
        # Восстановление исходного порядка переменных, если он был изменен
        x = [x[i] for i in var_indices]
        return x
```

gauss_seidel

```
def gauss_seidel(matrix, epsilon):
    """
    Выполняет итерацию Гаусса-Зейделя для решения системы линейных 
уравнений.
    :param matrix: 2D список, представляющий коэффициентную матрицу, 
дополненную вектором правой части.
    :param epsilon: Желаемая точность.
    :return: Вектор решения.
    """
    x = [0] * n  # Вектор начального приближения
    converge = False

    while not converge:
        x_copy = x[:]

        for i in range(n):
            s1 = sum(matrix[i][j] * x_copy[j] for j in range(i))
            s2 = sum(matrix[i][j] * x[j] for j in range(i + 1, n))
            x_copy[i] = (matrix[i][-1] - s1 - s2) / matrix[i][i]

        # Проверка сходимости
        converge = all(abs(x_copy[i] - x[i]) < epsilon for i in range(n))
        x = x_copy

    return x
```

is_diagonally_dominant

```
def is_diagonally_dominant(matrix):
    """
    Проверяет, имеет ли матрица диагональное преобладание и хотя бы одна 
строка имеет строго диагональное преобладание.
    :param matrix: 2D список, представляющий коэффициентную матрицу, 
дополненную вектором правой части.
    :return: True, если матрица имеет диагональное преобладание, False в 
противном случае.
    """
```

```
    found_strictly_dominant = False
    for i in range(n):
        if is_number_dominant_in_row(matrix[i], i) == "Not dominant":
            return False
        elif is_number_dominant_in_row(matrix[i], i) == "Strictly
dominant":
            found_strictly_dominant = True
    return found_strictly_dominant
```

get_number_dominance_in_list

```
def get_number_dominance_in_list(row, index):
    """
    Проверяет, является ли число в данном индексе доминирующим в данной
строке. Игнорируется последнее число.
    :param row: 1D список, представляющий строку в матрице.
    :param index: Индекс числа для проверки.
    :return: "Strictly dominant", если число строго доминирует, "Weakly
dominant", если число слабо доминирует,
    "Not dominant" в противном случае.
    """
    value = abs(row[index])
    sum_of_other_values = sum(abs(row[i]) for i in range(n) if i != index)
    if value > sum_of_other_values:
        return "Strictly dominant"
    elif value == sum_of_other_values:
        return "Weakly dominant"
    else:
        return "Not dominant"
```

Пример работы программы

Пример 1

Тривиальная система

Ввод:

```
2
1 0 2
0 1 3
1e-10
```

Вывод:

```
2.0
3.0
```

Информация:

```
swap_columns: 0 times
swap_rows: 0 times
get_number_dominance_in_list: 4 times
is_diagonally_dominant: 1 times
Iterations: 2
```

Пример 2

Диагонально доминирующая система

Ввод:

```
3
4 -1 2 3
-1 4 1 2
2 -1 4 2
1e-10
```

Вывод:

```
0.7708333333774092
0.6250000000237136
0.2708333333172238
```

Информация:

```
swap_columns: 0 times
swap_rows: 0 times
get_number_dominance_in_list: 6 times
is_diagonally_dominant: 1 times
```

```
Iterations: 16
```

Пример 3

Данная система решается классическим методом Гаусса за одно преобразование

Ввод:

```
2
3 2 3
4 5 6
1e-10
```

Вывод:

```
0.42857142865633024
0.8571428570749358
```

Информация:

```
swap_columns: 0 times
swap_rows: 0 times
get_number_dominance_in_list: 4 times
is_diagonally_dominant: 1 times
Iterations: 37
```

Пример 4

Данная система не является изначально диагонально доминирующей системой

Ввод:

```
5
-2 45 0 4 0 0
1 0 0 0 0 0
5 6 4 7 78 4
0 0 5 0 0 -4
1 1 1 10 1 1
1e-10
```

Вывод:

```
0.0
-0.015444821289071133
-0.8
0.17375423949909857
0.07790242629795817
```

Информация:

```
swap_columns: 0 times
swap_rows: 3 times
get_number_dominance_in_list: 29 times
is_diagonally_dominant: 3 times
Iterations: 8
```

Пример 5

Данная система не может быть диагонально доминирующей системой

Ввод:

```
2
2 3 11
5 7 13
1e-10
```

Вывод:

```
The system has no diagonal dominance for this method. Method of the Gauss-
Seidel is not applicable.
```

Информация:

```
swap_columns: 0 times
swap_rows: 0 times
get_number_dominance_in_list: 9 times
is_diagonally_dominant: 3 times
```

Пример 6

Решение системы приближено к нулевому X вектору

Ввод:

```
4
0 0 -2 0 3
-3 0 0 0 0
0 2 0 0.4 0
0 0 0 -10 0
1e-10
```

Вывод:

```
-0.0
0.0
-1.5
-0.0
```

Информация:

```
swap_columns: 0 times
swap_rows: 2 times
get_number_dominance_in_list: 23 times
is_diagonally_dominant: 3 times
Iterations: 2
```

Пример 7

Все значения X должны получиться одинаковыми

Ввод:

```
7
7 1 1 1 1 1 1 10
1 7 1 1 1 1 1 10
1 1 7 1 1 1 1 10
1 1 1 7 1 1 1 10
1 1 1 1 7 1 1 10
1 1 1 1 1 7 1 10
1 1 1 1 1 1 7 10
1e-10
```

Вывод:

```
0.7692307692231085
0.7692307692331118
0.7692307692384025
0.7692307692384842
0.769230769235258
0.7692307692313237
0.7692307692286159
```

Информация:

```
swap_columns: 0 times
swap_rows: 0 times
get_number_dominance_in_list: 14 times
is_diagonally_dominant: 1 times
Iterations: 16
```

Пример 8

Большая система изначально не диагональная

Ввод:

```
20
60 -1 2 0 6 0 0 0 6 0 0 0 0 0 0 0 0 0 0 0 6
-1 11 -1 3 25 0 0 0 25 0 0 0 0 0 -77 0 7 0 0 0 25
2 -1 120 -1 -11 0 0 0 -11 0 0 0 0 0 0 0 0 0 0 0 -11
0 3 -1 80 15 0 0 0 5 0 663 0 0 0 0 6 0 0 0 0 15
0 0 0 0 30 1 -1 2 6 0 0 0 0 0 0 0 0 0 0 0 60
0 0 0 0 0 -1 11 -1 25 0 0 0 0 -52 0 0 0 0 0 747 25
0 0 0 0 0 2 -1 100 -11 0 0 0 0 0 0 0 -5 0 0 0 -112
0 0 0 0 0 0 3 -1 55 0 0 0 0 0 0 0 0 0 0 0 15
0 0 0 0 0 0 0 0 0 19 -1 2 0 6 -2 0 0 6 0 0 6
0 0 0 0 0 0 0 0 0 -1 11 -1 3 25 0 -525 0 25 0 0 25
0 0 0 0 0 0 0 0 0 2 -1 100 -1 -11 0 0 0 -11 0 0 -11
0 0 0 0 0 0 0 0 0 3 -1 8 18 0 0 0 5 0 0 0 15
0 0 0 0 0 0 40 0 0 0 0 0 0 9 10 -1 2 6 0 0 6
2 60 0 0 0 0 0 0 0 0 0 0 0 -1 11 -1 25 0 0 0 25
0 0 0 -567 0 0 0 0 0 9 0 87 0 0 2 -29 10 -11 0 0 -11
0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 -1 15 0 0 0 15
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 10 -1 2 6
0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 -1 11 -1 25
0 0 0 0 0 0 0 0 0 0 0 0 5 110 0 0 0 2 -1 10 -11
0 0 0 0 0 51 2 0 0 0 0 0 0 0 0 0 0 0 3 -1 15
1e-10
```

Вывод:

```
-0.13159420856600695
-0.04632728540955989
0.1182740936964875
0.019272203068346955
2.008908873209871
0.15994188992166605
-0.11482698437088468
-1.0506794362172076
```

```
0.25988730030718993
0.14467940054728942
-0.026620822508690876
-0.02936312621099864
0.570346619246039
-0.12108194167537485
0.4882162402240019
-0.011330761995608916
0.901601367822159
0.8310201583314367
2.3631522758280057
0.01683924472153273
```

Информация:

```
swap_columns: 0 times
swap_rows: 15 times
get_number_dominance_in_list: 182 times
is_diagonally_dominant: 3 times
Iterations: 12
```

Пример 9

Данная система не является изначально диагонально доминирующей системой

Ввод:

```
8
10 -1 2 0 6 0 0 0 6
-1 11 -1 3 25 0 2 0 25
2 -1 0 -1 -11 0 0 0 -11
0 3 -1 8 15 0 0 0 15
1 7 0 0 0 10 -1 2 6
0 0 20 0 0 -1 -6 -1 25
0 6 0 0 45 2 -1 10 -11
0 -63 0 2 0 0 3 -1 15
1e-10
```

Вывод:

```
The system has no diagonal dominance for this method. Method of the Gauss-
Seidel is not applicable.
```

Информация:

```
swap_columns: 1 times

swap_rows: 3 times

get_number_dominance_in_list: 54 times

is_diagonally_dominant: 3 times
```

Пример 10

Система диагонально доминирующая но имеет значение равное нулю в диагонали

Ввод:

```
4

10 -1 2 0 6

-1 11 -1 3 25

0 0 0 0 2

0 3 -1 8 15

1e-10
```

Вывод:

```
The system has no diagonal dominance for this method. Method of the Gauss-Seidel is not applicable.
```

Информация:

```
swap_columns: 0 times

swap_rows: 0 times

get_number_dominance_in_list: 8 times

is_diagonally_dominant: 1 times
```

Вывод

Метод Гаусса-Зейделя обладает преимуществом перед некоторыми другими методами, включая метод Гаусса, поскольку он может дать приближенное решение за меньшее количество шагов и не требует хранения всех предыдущих шагов. Однако его эффективность зависит от свойств матрицы системы. Метод Гаусса-Зейделя сходится только для матриц, которые являются диагонально доминирующими или симметричными и положительно определенными. В то же время, он может быть менее эффективным, чем некоторые другие итерационные методы, такие как метод сопряженных градиентов, для больших систем или систем с определенными свойствами. Кроме того, как показано в примере 3, иногда данный метод требует на много больше итерации для решения небольших систем, в то время как другие методы могут не иметь таких ограничений.

Алгоритмическая сложность метода Гаусса-Зейделя составляет $O(n^2)$, рассматривая с точки зрения каждой итерации, где n — это количество уравнений в системе. Количество итераций, необходимых для достижения сходимости, может варьироваться в зависимости от свойств матрицы системы и заданной точности. Если опишем число k как число итерации, то алгоритмическая сложность будет составлять $O(kn^2)$

Код корректно реализует метод Гаусса-Зейделя. Он проверяет, является ли матрица диагонально доминирующей, и, если это не так, пытается сделать ее таковой путем перестановки строк и столбцов. Если матрица не может быть приведена к диагонально доминирующему виду, код корректно выводит сообщение об ошибке. Код также корректно реализует итерационную процедуру Гаусса-Зейделя и останавливается, когда достигнута заданная точность.

Численная ошибка метода Гаусса-Зейделя, как и любого итерационного метода, зависит от нескольких факторов. Во-первых, она зависит от заданной точности: чем меньше значение эпсилон, тем точнее будет решение, но это также может потребовать большего числа итераций. Во-вторых, численная ошибка может возрастать с увеличением размера системы из-за накопления ошибок округления при выполнении большего числа операций.