МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А.И. ГЕРЦЕНА»



Направление подготовки

09.03.01 – Информатика и вычислительная техника

Профиль «Технологии разработки программного обеспечения»

**Лабораторная работа №2**

**«Решение систем линейных алгебраических уравнений»**

|  | Работу выполнил:  Балаев Жамал,  Васильева Марина,  Иванов Никита,  Шардт Максим  Рожков Максим  очная форма обучения  курс: 2; группа:ИВТ-1.1 |
| --- | --- |
|  | Научный руководитель:  Профессор Власова Елена Зотиковна |

Санкт-Петербург

2023

# Содержание

Отчет [Балаева Ж. Б. ИВТ 1.1 4](#_30j0zll)

Отчет [Васильевой М. А. 1.1 11](#_dljygzvolopu)

Отчет [Иванова Н. Р. ИВТ 1.1 18](#_eidz17wbp01w)

Отчет [Рожкова М. В. ИВТ 1.1 25](#_cirllp2egikw)

Отчет [Шардта М. А. ИВТ 1.1 32](#_joor4e7o17tt)

### Выполнил Балаев Ж. Б. ИВТ 1.1

# Задача 1. Модифицировать программу по решению СЛУ методом Гаусса для реализации метода Гаусса-Жордана

## Математическая модель

(1)

;

(2)

(3)

## Моделирование

Алгоритм реализует приведение матрицы к треугольному виду метод Гаусса-Жордана. На каждом шаге цикла программы выполняет деление текущей строки на диагональный элемент, затем вычитает эту строку умноженную на коэффициент из всех остальных строк матрицы.

| def gauss\_jordan(a):  n = len(a)  for i in range(n):  for k in range(n):  if k == i:  continue  a[k][i] = round(a[k][i] / a[i][i], precision)  for j in range(i + 1, n + 1):  a[k][j] -= round(a[i][j] \* a[k][i], precision)  a[k][i] = 0  return a |
| --- |

Решением системы уравнений является вектор значений на главной диагонали. Данная функция собирает необходимые значения в один массив.

| def gauss\_jordan\_result(a):  x = [0] \* len(a)  for i in range(len(a)):  x[i] = round(a[i][-1] / a[i][i], 2)  return x |
| --- |

## Результат выполнения контрольного примера

# 

Рис. 1. Результат выполнения кода

| Результаты выполнения программы при разных значениях столбца свободных членов | | | |
| --- | --- | --- | --- |
| При начальных значениях из контрольного примера | Измененные значения на 0,1 | Измененные значения на 0,01 | Измененные значения на 0,001 |
| 1.0 1.0 1.0 1.0 | 3.0 -0.2 0.5 1.3 | 1.2 0.88 0.95 1.03 | 1.02 0.988 0.995 1.003 |

Табл. 1. Сводная таблица результатов

## Вывод

Нами был реализован программный модуль для решения СЛУ методом Гаусса-Жордана.

# Задача 2. Модифицировать программу по решению СЛУ методом Гаусса для реализации алгоритма с выбором главного элемента (по строкам, по столбцам, по всей матрице).

## Математическая модель

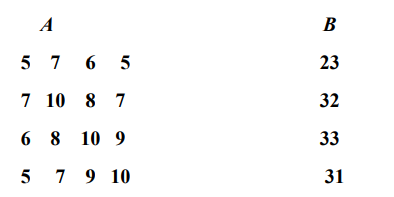
(1)

(2)

(3)

(4)

## Контрольный пример



## Моделирование

В ходе итерации в матрицы с главным элементом, Затем главный элемент преобразуется в 1, а остальные элементы на этой строке делятся на значение главного элемента. Для нижних строк из каждого значения вычитается кратное главному элементу, чтобы обнулить эти значения.

Для выбора главного элемента были написаны вспомогательные функции, которые содержатся в отдельном модуле.

| def gauss\_jordan(a):  n = len(a)  for i in range(n):  p = a[i][i]  a[i][i] = 1  for j in range(i + 1, n + 1):  a[i][j] = round(a[i][j] / p, precision)   for k in range(i + 1, n):  if a[k][i] != 0:  for j in range(i + 1, n + 1):  a[k][j] -= a[k][i] \* a[i][j]  a[k][i] = 0   return a |
| --- |

В каждой итерации обратного хода из переменной aa вычитается произведение значения элемента a[i][j] и соответствующего решения x[j]. В конце итерации значение aa делится на значение главного элемента a[i][i].

| def gauss\_reverse(a):  n = len(a)  x = [0] \* len(a)  for i in range(n-1, -1, -1):  x[i] = 0  aa = a[i][n]  for j in range(n, i, -1):  aa -= a[i][j] \* x[j]  x[i] = round(aa, precision)   return x |
| --- |

После завершения итераций по матрице столбец свободных членов содержит решение системы уравнений.

# 

## Результат выполнения контрольного примера

# 

Рис. 2. Результаты вычислений

| Результаты выполнения программы при разных значениях столбца свободных членов | | | |
| --- | --- | --- | --- |
| При начальных значениях из контрольного примера | Измененные значения на 0,1 | Измененные значения на 0,01 | Измененные значения на 0,001 |
| 1.0 1.0 1.0 1.0 | 3.0 -0.2 0.5 1.3 | 1.2 0.88 0.95 1.03 | 1.02 0.988 0.995 1.003 |

Табл. 2. Сводная таблица результатов

## Вывод

Нами был реализован программный модуль для решения системы линейных уравнений методом Гаусса для реализации алгоритма с выбором главного элемента (по строкам, по столбцам, по всей матрице).

# Задача 3. Модифицировать программу по решению СЛУ методом Гаусса для реализации алгоритма обратного хода с распараллеливанием вычислений

## Математическая модель

(1)

(2)

(3)

(4)

## Моделирование

В ходе алгоритма вычисляется множитель по формуле (1) для каждой строки и затем с учетом множителя вычитаются из элементов нижестоящей строки по формуле (2). Таким образом матрица приводится к треугольному виду.

| def gauss\_direct(a):  n = len(a)  for i in range(n - 1):  for k in range(i + 1, n):  a[k][i] = round(a[k][i] / a[i][i], precision)  for j in range(i + 1, n + 1):  a[k][j] = round(a[k][j] - a[i][j] \* a[k][i], precision)  a[k][i] = 0  return a |
| --- |

Данный алгоритм реализует обратный ход, используя цикл для обновления коэффициентов в столбце свободных членов, что позволяет ускорить вычисления.

| def gauss\_reverse(a):  n = len(a)  x = [0] \* len(a)  for i in range(n - 1, -1, -1):  x[i] = round(a[i][n] / a[i][i], precision)  for k in range(i):  a[k][n] -= a[k][i] \* x[i]   return x |
| --- |

Также как и предыдущем модуле присутствует возможность выбора точности вычислений.

## Результат выполнения контрольного примера

# 

Рис. 3. Результаты вычислений

| Результаты выполнения программы при разных значениях столбца свободных членов | | | |
| --- | --- | --- | --- |
| При начальных значениях из контрольного примера | Измененные значения на 0,1 | Измененные значения на 0,01 | Измененные значения на 0,001 |
| 1.0 1.0 1.0 1.0 | 3.0 -0.2 0.5 1.3 | 1.2 0.88 0.95 1.03 | 1.02 0.988 0.995 1.003 |

Табл. 3. Сводная таблица результатов

## Вывод

Нами был реализован программный модуль для решения СЛУ методом Гаусса с распараллеливанием вычислений.

### Выполнила [Васильева М. А.](#_wl4bpvrj617c) 1.1

# Задача 1. Модифицировать программу по решению СЛУ методом Гаусса для реализации метода Гаусса-Жордана

## Математическая модель

(1)

;

(2)

(3)

## Моделирование

Алгоритм реализует приведение матрицы к треугольному виду метод Гаусса-Жордана. На каждом шаге цикла программы выполняет деление текущей строки на диагональный элемент, затем вычитает эту строку умноженную на коэффициент из всех остальных строк матрицы.

| def gauss\_jordan(a):  n = len(a)  for i in range(n):  for k in range(n):  if k == i:  continue  a[k][i] = round(a[k][i] / a[i][i], precision)  for j in range(i + 1, n + 1):  a[k][j] -= round(a[i][j] \* a[k][i], precision)  a[k][i] = 0  return a |
| --- |

Решением системы уравнений является вектор значений на главной диагонали. Данная функция собирает необходимые значения в один массив.

| def gauss\_jordan\_result(a):  x = [0] \* len(a)  for i in range(len(a)):  x[i] = round(a[i][-1] / a[i][i], 2)  return x |
| --- |

## Результат выполнения контрольного примера

# 

Рис. 1. Результат выполнения кода

| Результаты выполнения программы при разных значениях столбца свободных членов | | | |
| --- | --- | --- | --- |
| При начальных значениях из контрольного примера | Измененные значения на 0,1 | Измененные значения на 0,01 | Измененные значения на 0,001 |
| 1.0 1.0 1.0 1.0 | 3.0 -0.2 0.5 1.3 | 1.2 0.88 0.95 1.03 | 1.02 0.988 0.995 1.003 |

Табл. 1. Сводная таблица результатов

## Вывод

Нами был реализован программный модуль для решения СЛУ методом Гаусса-Жордана.

# Задача 2. Модифицировать программу по решению СЛУ методом Гаусса для реализации алгоритма с выбором главного элемента (по строкам, по столбцам, по всей матрице).

## Математическая модель

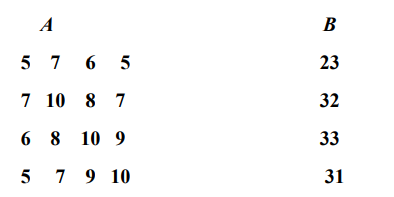
(1)

(2)

(3)

(4)

## Контрольный пример



## Моделирование

В ходе итерации в матрицы с главным элементом, Затем главный элемент преобразуется в 1, а остальные элементы на этой строке делятся на значение главного элемента. Для нижних строк из каждого значения вычитается кратное главному элементу, чтобы обнулить эти значения.

Для выбора главного элемента были написаны вспомогательные функции, которые содержатся в отдельном модуле.

| def gauss\_jordan(a):  n = len(a)  for i in range(n):  p = a[i][i]  a[i][i] = 1  for j in range(i + 1, n + 1):  a[i][j] = round(a[i][j] / p, precision)   for k in range(i + 1, n):  if a[k][i] != 0:  for j in range(i + 1, n + 1):  a[k][j] -= a[k][i] \* a[i][j]  a[k][i] = 0   return a |
| --- |

В каждой итерации обратного хода из переменной aa вычитается произведение значения элемента a[i][j] и соответствующего решения x[j]. В конце итерации значение aa делится на значение главного элемента a[i][i].

| def gauss\_reverse(a):  n = len(a)  x = [0] \* len(a)  for i in range(n-1, -1, -1):  x[i] = 0  aa = a[i][n]  for j in range(n, i, -1):  aa -= a[i][j] \* x[j]  x[i] = round(aa, precision)   return x |
| --- |

После завершения итераций по матрице столбец свободных членов содержит решение системы уравнений.

# 

## Результат выполнения контрольного примера

# 

Рис. 2. Результаты вычислений

| Результаты выполнения программы при разных значениях столбца свободных членов | | | |
| --- | --- | --- | --- |
| При начальных значениях из контрольного примера | Измененные значения на 0,1 | Измененные значения на 0,01 | Измененные значения на 0,001 |
| 1.0 1.0 1.0 1.0 | 3.0 -0.2 0.5 1.3 | 1.2 0.88 0.95 1.03 | 1.02 0.988 0.995 1.003 |

Табл. 2. Сводная таблица результатов

## Вывод

Нами был реализован программный модуль для решения системы линейных уравнений методом Гаусса для реализации алгоритма с выбором главного элемента (по строкам, по столбцам, по всей матрице).

# Задача 3. Модифицировать программу по решению СЛУ методом Гаусса для реализации алгоритма обратного хода с распараллеливанием вычислений

## Математическая модель

(1)

(2)

(3)

(4)

## Моделирование

В ходе алгоритма вычисляется множитель по формуле (1) для каждой строки и затем с учетом множителя вычитаются из элементов нижестоящей строки по формуле (2). Таким образом матрица приводится к треугольному виду.

| def gauss\_direct(a):  n = len(a)  for i in range(n - 1):  for k in range(i + 1, n):  a[k][i] = round(a[k][i] / a[i][i], precision)  for j in range(i + 1, n + 1):  a[k][j] = round(a[k][j] - a[i][j] \* a[k][i], precision)  a[k][i] = 0  return a |
| --- |

Данный алгоритм реализует обратный ход, используя цикл для обновления коэффициентов в столбце свободных членов, что позволяет ускорить вычисления.

| def gauss\_reverse(a):  n = len(a)  x = [0] \* len(a)  for i in range(n - 1, -1, -1):  x[i] = round(a[i][n] / a[i][i], precision)  for k in range(i):  a[k][n] -= a[k][i] \* x[i]   return x |
| --- |

Также как и предыдущем модуле присутствует возможность выбора точности вычислений.

## Результат выполнения контрольного примера

# 

Рис. 3. Результаты вычислений

| Результаты выполнения программы при разных значениях столбца свободных членов | | | |
| --- | --- | --- | --- |
| При начальных значениях из контрольного примера | Измененные значения на 0,1 | Измененные значения на 0,01 | Измененные значения на 0,001 |
| 1.0 1.0 1.0 1.0 | 3.0 -0.2 0.5 1.3 | 1.2 0.88 0.95 1.03 | 1.02 0.988 0.995 1.003 |

Табл. 3. Сводная таблица результатов

## Вывод

Нами был реализован программный модуль для решения СЛУ методом Гаусса с распараллеливанием вычислений.

### Выполнил [Иванов Н. Р.](https://docs.google.com/document/d/1uJ-PphvyBSg3V-l16ZUAOtt2b-tTnhSR82t9XZbdA9c/edit#heading=h.vr2u23c4g3jp) ИВТ 1.1

# Задача 1. Модифицировать программу по решению СЛУ методом Гаусса для реализации метода Гаусса-Жордана

## Математическая модель

(1)

;

(2)

(3)

## Моделирование

Алгоритм реализует приведение матрицы к треугольному виду метод Гаусса-Жордана. На каждом шаге цикла программы выполняет деление текущей строки на диагональный элемент, затем вычитает эту строку умноженную на коэффициент из всех остальных строк матрицы.

| def gauss\_jordan(a):  n = len(a)  for i in range(n):  for k in range(n):  if k == i:  continue  a[k][i] = round(a[k][i] / a[i][i], precision)  for j in range(i + 1, n + 1):  a[k][j] -= round(a[i][j] \* a[k][i], precision)  a[k][i] = 0  return a |
| --- |

Решением системы уравнений является вектор значений на главной диагонали. Данная функция собирает необходимые значения в один массив.

| def gauss\_jordan\_result(a):  x = [0] \* len(a)  for i in range(len(a)):  x[i] = round(a[i][-1] / a[i][i], 2)  return x |
| --- |

## Результат выполнения контрольного примера

# 

Рис. 1. Результат выполнения кода

| Результаты выполнения программы при разных значениях столбца свободных членов | | | |
| --- | --- | --- | --- |
| При начальных значениях из контрольного примера | Измененные значения на 0,1 | Измененные значения на 0,01 | Измененные значения на 0,001 |
| 1.0 1.0 1.0 1.0 | 3.0 -0.2 0.5 1.3 | 1.2 0.88 0.95 1.03 | 1.02 0.988 0.995 1.003 |

Табл. 1. Сводная таблица результатов

## Вывод

Нами был реализован программный модуль для решения СЛУ методом Гаусса-Жордана.

# Задача 2. Модифицировать программу по решению СЛУ методом Гаусса для реализации алгоритма с выбором главного элемента (по строкам, по столбцам, по всей матрице).

## Математическая модель

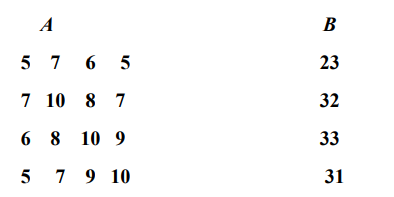
(1)

(2)

(3)

(4)

## Контрольный пример



## Моделирование

В ходе итерации в матрицы с главным элементом, Затем главный элемент преобразуется в 1, а остальные элементы на этой строке делятся на значение главного элемента. Для нижних строк из каждого значения вычитается кратное главному элементу, чтобы обнулить эти значения.

Для выбора главного элемента были написаны вспомогательные функции, которые содержатся в отдельном модуле.

| def gauss\_jordan(a):  n = len(a)  for i in range(n):  p = a[i][i]  a[i][i] = 1  for j in range(i + 1, n + 1):  a[i][j] = round(a[i][j] / p, precision)   for k in range(i + 1, n):  if a[k][i] != 0:  for j in range(i + 1, n + 1):  a[k][j] -= a[k][i] \* a[i][j]  a[k][i] = 0   return a |
| --- |

В каждой итерации обратного хода из переменной aa вычитается произведение значения элемента a[i][j] и соответствующего решения x[j]. В конце итерации значение aa делится на значение главного элемента a[i][i].

| def gauss\_reverse(a):  n = len(a)  x = [0] \* len(a)  for i in range(n-1, -1, -1):  x[i] = 0  aa = a[i][n]  for j in range(n, i, -1):  aa -= a[i][j] \* x[j]  x[i] = round(aa, precision)   return x |
| --- |

После завершения итераций по матрице столбец свободных членов содержит решение системы уравнений.

# 

## Результат выполнения контрольного примера

# 

Рис. 2. Результаты вычислений

| Результаты выполнения программы при разных значениях столбца свободных членов | | | |
| --- | --- | --- | --- |
| При начальных значениях из контрольного примера | Измененные значения на 0,1 | Измененные значения на 0,01 | Измененные значения на 0,001 |
| 1.0 1.0 1.0 1.0 | 3.0 -0.2 0.5 1.3 | 1.2 0.88 0.95 1.03 | 1.02 0.988 0.995 1.003 |

Табл. 2. Сводная таблица результатов

## Вывод

Нами был реализован программный модуль для решения системы линейных уравнений методом Гаусса для реализации алгоритма с выбором главного элемента (по строкам, по столбцам, по всей матрице).

# Задача 3. Модифицировать программу по решению СЛУ методом Гаусса для реализации алгоритма обратного хода с распараллеливанием вычислений

## Математическая модель

(1)

(2)

(3)

(4)

## Моделирование

В ходе алгоритма вычисляется множитель по формуле (1) для каждой строки и затем с учетом множителя вычитаются из элементов нижестоящей строки по формуле (2). Таким образом матрица приводится к треугольному виду.

| def gauss\_direct(a):  n = len(a)  for i in range(n - 1):  for k in range(i + 1, n):  a[k][i] = round(a[k][i] / a[i][i], precision)  for j in range(i + 1, n + 1):  a[k][j] = round(a[k][j] - a[i][j] \* a[k][i], precision)  a[k][i] = 0  return a |
| --- |

Данный алгоритм реализует обратный ход, используя цикл для обновления коэффициентов в столбце свободных членов, что позволяет ускорить вычисления.

| def gauss\_reverse(a):  n = len(a)  x = [0] \* len(a)  for i in range(n - 1, -1, -1):  x[i] = round(a[i][n] / a[i][i], precision)  for k in range(i):  a[k][n] -= a[k][i] \* x[i]   return x |
| --- |

Также как и предыдущем модуле присутствует возможность выбора точности вычислений.

## Результат выполнения контрольного примера

# 

Рис. 3. Результаты вычислений

| Результаты выполнения программы при разных значениях столбца свободных членов | | | |
| --- | --- | --- | --- |
| При начальных значениях из контрольного примера | Измененные значения на 0,1 | Измененные значения на 0,01 | Измененные значения на 0,001 |
| 1.0 1.0 1.0 1.0 | 3.0 -0.2 0.5 1.3 | 1.2 0.88 0.95 1.03 | 1.02 0.988 0.995 1.003 |

Табл. 3. Сводная таблица результатов

## Вывод

Нами был реализован программный модуль для решения СЛУ методом Гаусса с распараллеливанием вычислений.

### Выполнил [Рожков М. В.](https://docs.google.com/document/d/1uJ-PphvyBSg3V-l16ZUAOtt2b-tTnhSR82t9XZbdA9c/edit#heading=h.8eb1w5y4s5hj) ИВТ 1.1

# Задача 1. Модифицировать программу по решению СЛУ методом Гаусса для реализации метода Гаусса-Жордана

## Математическая модель

(1)

;

(2)

(3)

## Моделирование

Алгоритм реализует приведение матрицы к треугольному виду метод Гаусса-Жордана. На каждом шаге цикла программы выполняет деление текущей строки на диагональный элемент, затем вычитает эту строку умноженную на коэффициент из всех остальных строк матрицы.

| def gauss\_jordan(a):  n = len(a)  for i in range(n):  for k in range(n):  if k == i:  continue  a[k][i] = round(a[k][i] / a[i][i], precision)  for j in range(i + 1, n + 1):  a[k][j] -= round(a[i][j] \* a[k][i], precision)  a[k][i] = 0  return a |
| --- |

Решением системы уравнений является вектор значений на главной диагонали. Данная функция собирает необходимые значения в один массив.

| def gauss\_jordan\_result(a):  x = [0] \* len(a)  for i in range(len(a)):  x[i] = round(a[i][-1] / a[i][i], 2)  return x |
| --- |

## Результат выполнения контрольного примера

# 

Рис. 1. Результат выполнения кода

| Результаты выполнения программы при разных значениях столбца свободных членов | | | |
| --- | --- | --- | --- |
| При начальных значениях из контрольного примера | Измененные значения на 0,1 | Измененные значения на 0,01 | Измененные значения на 0,001 |
| 1.0 1.0 1.0 1.0 | 3.0 -0.2 0.5 1.3 | 1.2 0.88 0.95 1.03 | 1.02 0.988 0.995 1.003 |

Табл. 1. Сводная таблица результатов

## Вывод

Нами был реализован программный модуль для решения СЛУ методом Гаусса-Жордана.

# Задача 2. Модифицировать программу по решению СЛУ методом Гаусса для реализации алгоритма с выбором главного элемента (по строкам, по столбцам, по всей матрице).

## Математическая модель

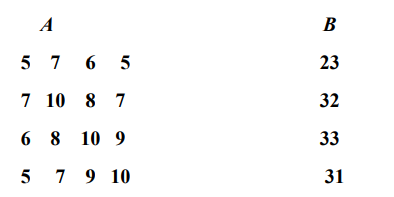
(1)

(2)

(3)

(4)

## Контрольный пример



## Моделирование

В ходе итерации в матрицы с главным элементом, Затем главный элемент преобразуется в 1, а остальные элементы на этой строке делятся на значение главного элемента. Для нижних строк из каждого значения вычитается кратное главному элементу, чтобы обнулить эти значения.

Для выбора главного элемента были написаны вспомогательные функции, которые содержатся в отдельном модуле.

| def gauss\_jordan(a):  n = len(a)  for i in range(n):  p = a[i][i]  a[i][i] = 1  for j in range(i + 1, n + 1):  a[i][j] = round(a[i][j] / p, precision)   for k in range(i + 1, n):  if a[k][i] != 0:  for j in range(i + 1, n + 1):  a[k][j] -= a[k][i] \* a[i][j]  a[k][i] = 0   return a |
| --- |

В каждой итерации обратного хода из переменной aa вычитается произведение значения элемента a[i][j] и соответствующего решения x[j]. В конце итерации значение aa делится на значение главного элемента a[i][i].

| def gauss\_reverse(a):  n = len(a)  x = [0] \* len(a)  for i in range(n-1, -1, -1):  x[i] = 0  aa = a[i][n]  for j in range(n, i, -1):  aa -= a[i][j] \* x[j]  x[i] = round(aa, precision)   return x |
| --- |

После завершения итераций по матрице столбец свободных членов содержит решение системы уравнений.

# 

## Результат выполнения контрольного примера

# 

Рис. 2. Результаты вычислений

| Результаты выполнения программы при разных значениях столбца свободных членов | | | |
| --- | --- | --- | --- |
| При начальных значениях из контрольного примера | Измененные значения на 0,1 | Измененные значения на 0,01 | Измененные значения на 0,001 |
| 1.0 1.0 1.0 1.0 | 3.0 -0.2 0.5 1.3 | 1.2 0.88 0.95 1.03 | 1.02 0.988 0.995 1.003 |

Табл. 2. Сводная таблица результатов

## Вывод

Нами был реализован программный модуль для решения системы линейных уравнений методом Гаусса для реализации алгоритма с выбором главного элемента (по строкам, по столбцам, по всей матрице).

# Задача 3. Модифицировать программу по решению СЛУ методом Гаусса для реализации алгоритма обратного хода с распараллеливанием вычислений

## Математическая модель

(1)

(2)

(3)

(4)

## Моделирование

В ходе алгоритма вычисляется множитель по формуле (1) для каждой строки и затем с учетом множителя вычитаются из элементов нижестоящей строки по формуле (2). Таким образом матрица приводится к треугольному виду.

| def gauss\_direct(a):  n = len(a)  for i in range(n - 1):  for k in range(i + 1, n):  a[k][i] = round(a[k][i] / a[i][i], precision)  for j in range(i + 1, n + 1):  a[k][j] = round(a[k][j] - a[i][j] \* a[k][i], precision)  a[k][i] = 0  return a |
| --- |

Данный алгоритм реализует обратный ход, используя цикл для обновления коэффициентов в столбце свободных членов, что позволяет ускорить вычисления.

| def gauss\_reverse(a):  n = len(a)  x = [0] \* len(a)  for i in range(n - 1, -1, -1):  x[i] = round(a[i][n] / a[i][i], precision)  for k in range(i):  a[k][n] -= a[k][i] \* x[i]   return x |
| --- |

Также как и предыдущем модуле присутствует возможность выбора точности вычислений.

## Результат выполнения контрольного примера

# 

Рис. 3. Результаты вычислений

| Результаты выполнения программы при разных значениях столбца свободных членов | | | |
| --- | --- | --- | --- |
| При начальных значениях из контрольного примера | Измененные значения на 0,1 | Измененные значения на 0,01 | Измененные значения на 0,001 |
| 1.0 1.0 1.0 1.0 | 3.0 -0.2 0.5 1.3 | 1.2 0.88 0.95 1.03 | 1.02 0.988 0.995 1.003 |

Табл. 3. Сводная таблица результатов

## Вывод

Нами был реализован программный модуль для решения СЛУ методом Гаусса с распараллеливанием вычислений.

### Выполнил Шардт М. А. ИВТ 1.1

# Задача 1. Модифицировать программу по решению СЛУ методом Гаусса для реализации метода Гаусса-Жордана

## Математическая модель

(1)

;

(2)

(3)

## Моделирование

Алгоритм реализует приведение матрицы к треугольному виду метод Гаусса-Жордана. На каждом шаге цикла программы выполняет деление текущей строки на диагональный элемент, затем вычитает эту строку умноженную на коэффициент из всех остальных строк матрицы.

| def gauss\_jordan(a):  n = len(a)  for i in range(n):  for k in range(n):  if k == i:  continue  a[k][i] = round(a[k][i] / a[i][i], precision)  for j in range(i + 1, n + 1):  a[k][j] -= round(a[i][j] \* a[k][i], precision)  a[k][i] = 0  return a |
| --- |

Решением системы уравнений является вектор значений на главной диагонали. Данная функция собирает необходимые значения в один массив.

| def gauss\_jordan\_result(a):  x = [0] \* len(a)  for i in range(len(a)):  x[i] = round(a[i][-1] / a[i][i], 2)  return x |
| --- |

## Результат выполнения контрольного примера

# 

Рис. 1. Результат выполнения кода

| Результаты выполнения программы при разных значениях столбца свободных членов | | | |
| --- | --- | --- | --- |
| При начальных значениях из контрольного примера | Измененные значения на 0,1 | Измененные значения на 0,01 | Измененные значения на 0,001 |
| 1.0 1.0 1.0 1.0 | 3.0 -0.2 0.5 1.3 | 1.2 0.88 0.95 1.03 | 1.02 0.988 0.995 1.003 |

Табл. 1. Сводная таблица результатов

## Вывод

Нами был реализован программный модуль для решения СЛУ методом Гаусса-Жордана.

# Задача 2. Модифицировать программу по решению СЛУ методом Гаусса для реализации алгоритма с выбором главного элемента (по строкам, по столбцам, по всей матрице).

## Математическая модель

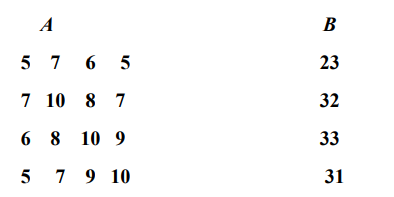
(1)

(2)

(3)

(4)

## Контрольный пример



## Моделирование

В ходе итерации в матрицы с главным элементом, Затем главный элемент преобразуется в 1, а остальные элементы на этой строке делятся на значение главного элемента. Для нижних строк из каждого значения вычитается кратное главному элементу, чтобы обнулить эти значения.

Для выбора главного элемента были написаны вспомогательные функции, которые содержатся в отдельном модуле.

| def gauss\_jordan(a):  n = len(a)  for i in range(n):  p = a[i][i]  a[i][i] = 1  for j in range(i + 1, n + 1):  a[i][j] = round(a[i][j] / p, precision)   for k in range(i + 1, n):  if a[k][i] != 0:  for j in range(i + 1, n + 1):  a[k][j] -= a[k][i] \* a[i][j]  a[k][i] = 0   return a |
| --- |

В каждой итерации обратного хода из переменной aa вычитается произведение значения элемента a[i][j] и соответствующего решения x[j]. В конце итерации значение aa делится на значение главного элемента a[i][i].

| def gauss\_reverse(a):  n = len(a)  x = [0] \* len(a)  for i in range(n-1, -1, -1):  x[i] = 0  aa = a[i][n]  for j in range(n, i, -1):  aa -= a[i][j] \* x[j]  x[i] = round(aa, precision)   return x |
| --- |

После завершения итераций по матрице столбец свободных членов содержит решение системы уравнений.

# 

## Результат выполнения контрольного примера

# 

Рис. 2. Результаты вычислений

| Результаты выполнения программы при разных значениях столбца свободных членов | | | |
| --- | --- | --- | --- |
| При начальных значениях из контрольного примера | Измененные значения на 0,1 | Измененные значения на 0,01 | Измененные значения на 0,001 |
| 1.0 1.0 1.0 1.0 | 3.0 -0.2 0.5 1.3 | 1.2 0.88 0.95 1.03 | 1.02 0.988 0.995 1.003 |

Табл. 2. Сводная таблица результатов

## Вывод

Нами был реализован программный модуль для решения системы линейных уравнений методом Гаусса для реализации алгоритма с выбором главного элемента (по строкам, по столбцам, по всей матрице).

# Задача 3. Модифицировать программу по решению СЛУ методом Гаусса для реализации алгоритма обратного хода с распараллеливанием вычислений

## Математическая модель

(1)

(2)

(3)

(4)

## Моделирование

В ходе алгоритма вычисляется множитель по формуле (1) для каждой строки и затем с учетом множителя вычитаются из элементов нижестоящей строки по формуле (2). Таким образом матрица приводится к треугольному виду.

| def gauss\_direct(a):  n = len(a)  for i in range(n - 1):  for k in range(i + 1, n):  a[k][i] = round(a[k][i] / a[i][i], precision)  for j in range(i + 1, n + 1):  a[k][j] = round(a[k][j] - a[i][j] \* a[k][i], precision)  a[k][i] = 0  return a |
| --- |

Данный алгоритм реализует обратный ход, используя цикл для обновления коэффициентов в столбце свободных членов, что позволяет ускорить вычисления.

| def gauss\_reverse(a):  n = len(a)  x = [0] \* len(a)  for i in range(n - 1, -1, -1):  x[i] = round(a[i][n] / a[i][i], precision)  for k in range(i):  a[k][n] -= a[k][i] \* x[i]   return x |
| --- |

Также как и предыдущем модуле присутствует возможность выбора точности вычислений.

## Результат выполнения контрольного примера

# 

Рис. 3. Результаты вычислений

| Результаты выполнения программы при разных значениях столбца свободных членов | | | |
| --- | --- | --- | --- |
| При начальных значениях из контрольного примера | Измененные значения на 0,1 | Измененные значения на 0,01 | Измененные значения на 0,001 |
| 1.0 1.0 1.0 1.0 | 3.0 -0.2 0.5 1.3 | 1.2 0.88 0.95 1.03 | 1.02 0.988 0.995 1.003 |

Табл. 3. Сводная таблица результатов

## Вывод

Нами был реализован программный модуль для решения СЛУ методом Гаусса с распараллеливанием вычислений.