ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

ЧИСЛЕННОЕ РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ
(Вариант 9)

Выполнил студент 3 курса ПМиИ
Кондратьев Виталий

***Цель занятия*:** изучение численных методов решения систем линейных алгебраических уравнений, практическое решение систем на ЭВМ.

Задания к работе.

Написать, отладить и выполнить программы решения систем линейных алгебраических уравнений, записанных в векторно-матричной форме $A\ x = b$ и приведенных в таблице. В колонке x* приведено точное решение. Решить систему методом Гаусса с выбором главного элемента и методом Зейделя.

Оценить погрешности методов.

№	A				b	x*
9	$1{,}85$	$0{,}70$	$-0{,}12$	$-0{,}18$	$8{,}41$	3
	$0{,}16$	$0{,}19$	$0{,}79$	$0{,}11$	$-0{,}23$	4
	$1{,}13$	$2{,}77$	$0{,}18$	$-0{,}20$	$13{,}91$	-2
	$1{,}14$	$1{,}01$	$0{,}55$	$3{,}22$	$9{,}58$	1

Перенесем значения из таблиц в программу и убедимся в точности решения:

```
Матрица коэффициентов A:
[[ 1.85  0.7  -0.12 -0.18]
 [ 0.16  0.19  0.79  0.11]
 [ 1.13  2.77  0.18 -0.2 ]
 [ 1.14  1.01  0.55  3.22]]

Вектор-столбец свободных членов b:
[ 8.41 -0.23 13.91  9.58]

Точное решение:  [ 3.  4. -2.  1.]
```

Метод Гаусса с выбором главного элемента

Алгоритм:

1. **Прямой ход:**
   1.1. Для каждого столбца матрицы (кроме последнего):
   1.1.1. Найти строку с максимальным по модулю элементом в данном столбце.
   1.1.2. Если максимальный элемент равен 0, то система несовместна.
   1.1.3. Переставить строки местами так, чтобы максимальный элемент оказался на диагонали.
   1.1.4. Для всех строк, следующих за текущей, вычесть из них текущую строку, умноженную на такой коэффициент, чтобы элемент в текущем столбце стал нулем.
2. **Обратный ход:**
   2.1. Начиная с последней строки и двигаясь к первой:
   2.1.1. Разделить все элементы строки на диагональный элемент.
   2.1.2. Из каждой строки, стоящей выше, вычесть текущую строку, умноженную на такой коэффициент, чтобы элемент в текущем столбце стал нулем.
3. **Вычисление значений переменных:**
   3.1. Подставить значения всех неизвестных, кроме одного, в последнее уравнение системы.
   3.2. Вычислить значение оставшейся переменной.
   3.3. Подставить полученное значение в предпоследнее уравнение и вычислить значение следующей переменной.
   3.4. Продолжать этот процесс до тех пор, пока не будут найдены значения всех переменных.
4. **Проверка**
   4.1. Подставить найденные значения всех переменных в исходную систему уравнений.
   4.2. Если все уравнения системы выполняются, то решение найдено верно.

Решение:

```
Решение системы методом Гаусса с выбором главного элемента:
Переменная  Значение                 Погрешность
x1          3.000000000000000        0.000000000000002
x2          4.000000000000000        0.000000000000000
x3          -2.000000000000000       0.000000000000002
x4          1.000000000000000        0.000000000000002
```

Метод Зейделя

Алгоритм метода Зейделя (простой вариант):

1. **Начало:**
   1.1. Запиши систему уравнений.
   1.2. Угадай значения x1, x2, ..., xn.
2. **Повторяй:**
   2.1. Для каждого уравнения:
   2.1.1. Подставь уже найденные значения x1, x2, ..., xn в правую часть.
   2.1.2. Реши уравнение относительно x, стоящего слева.
   2.1.3. Запомни новое значение x.
   2.2. Проверь:
   2.2.1. Если все x изменились мало (или не изменились), то готово!
   2.2.2. Если нет, то повтори все сначала.
3. **Ответ:**
   3.1. Значения x1, x2, ..., xn - это решение системы уравнений.

Решение:

```
Решение системы методом Зейделя:
Переменная  Значение             Погрешность
x1          3.000000019300396    0.000000026164233
x2          3.999999987673903    0.000000012696349
x3          -1.999999984103966   0.000000008367785
x4          0.999999994476155    0.000000000509134
Количество итераций =  45
```

Вывод

Сравнение метода Гаусса с выбором главного элемента и метода Зейделя

Метод Гаусса с выбором главного элемента и метод Зейделя – два популярных метода решения систем линейных уравнений. Каждый из них обладает своими преимуществами и недостатками.

Точность:

• *Метод Гаусса с выбором главного элемента обеспечивает более высокую точность* решения, чем метод Зейделя. Это связано с тем, что метод Гаусса не использует деление на малые числа, что может привести к **ошибкам округления**.

Скорость:

• *Метод Зейделя*, как правило, работает <u>*быстрее*</u>, чем метод Гаусса, особенно для *хорошо обусловленных систем* с диагональным преобладанием.

Сходимость:

• *Метод Зейделя* может <u>*не сходиться*</u> для некоторых систем линейных уравнений.

• *Метод Гаусса с выбором главного элемента* всегда сходится, но может требовать <u>*больше итераций*</u>, чем метод Зейделя.

Выбор метода:

• *Метод Гаусса с выбором главного элемента* рекомендуется использовать, когда <u>*требуется высокая точность*</u> решения.

• *Метод Зейделя* может быть более подходящим, когда <u>*важна скорость*</u>, а требования к точности не так высоки.

Важно:

• Для достижения <u>*оптимальной сходимости*</u> метода Зейделя может потребоваться <u>*настройка параметров*</u>.

В целом:

• Выбор метода решения системы линейных уравнений зависит от *конкретных требований* к решению, таких как <u>*точность, скорость и устойчивость*</u> метода.

Дополнительные сведения:

• Для *более подробного сравнения* методов Гаусса и Зейделя рекомендуется ознакомиться с соответствующей литературой.

• *Выбор оптимального метода* для конкретной задачи может быть выполнен с помощью <u>*экспериментального сравнения*</u> различных методов.

ПРИЛОЖЕНИЕ

```
"""
Лабораторная работа №2
Студент ОНК «ИВТ» ВШ КНиИИ направления ПМиИ 3 курса
Кондратьев Виталий
Вариант 9
"""

import numpy as np


def gauss_method(A, b):
    """
    Решение системы линейных уравнений Методом Гаусса с выбором главного
элемента.
    """
    n = len(A)
    Ab = np.concatenate((A, b.reshape(n, 1)), axis=1)

    # Прямой ход метода Гаусса с выбором главного элемента
    for i in range(n):
        # Находим максимальный элемент в столбце и меняем строки, если нужно
        max_row = i + np.argmax(np.abs(Ab[i:, i]))
        Ab[[i, max_row]] = Ab[[max_row, i]]
        # Итерируемся по строкам и вычитаем кратное первой строке число
        for j in range(i + 1, n):
            Ab[j] -= Ab[i] * Ab[j, i] / Ab[i, i]

    # Обратный ход метода Гаусса
    x = np.zeros(n)
    for i in range(n - 1, -1, -1):
        x[i] = (Ab[i, n] - Ab[i, i + 1:n].dot(x[i + 1:n])) / Ab[i, i]

    # Вычисление погрешности
    error = np.zeros(n)
    for i in range(n):
        error[i] = np.abs(np.sum(A[i] * x) - b[i])

    # Вывод результатов
    print("{:<12}{:<25}{:<25}".format("Переменная", "Значение",
"Погрешность"))
    for i in range(n):
        print("{:<12}{:<25.15f}{:<25.15f}".format(f"x{i + 1}", x[i],
error[i]))


def seidel_method(A, b, x0, max_iter=100000, tol=1e-16):
    """
    Решение системы линейных уравнений методом Зейделя.
    """
    count = 0
    # Подсчет транспонированной матрицы A и b
    A_transpose = A.T @ A
    b_transpose = A.T @ b
    # Количество неизвестных переменных
    n = len(x0)
    # Начальное приближение
    x = np.copy(x0)
    while True:
        # Создание нового приближения решения
        count += 1
        x_new = np.zeros_like(x)
```

```python
        for i in range(n):
            # Сумма произведений коэффициентов на значения переменных
            s1 = A_transpose[i, :i] @ x_new[:i]
            s2 = A_transpose[i, i + 1:] @ x[i + 1:]
            # Новое значение i-ой переменной
            x_new[i] = (b_transpose[i] - s1 - s2) / A_transpose[i, i]
        # Проверка на достижение требуемой точности
        if np.allclose(x, x_new, rtol=tol):
            break
        # Обновление значения решения
        x = np.copy(x_new)
        # Проверка на превышение максимального количества итераций
        if max_iter <= 0:
            raise ValueError("Number of iterations exceeded")
        max_iter -= 1

    # Вывод результатов
    print("{:<12}{:<25}{:<25}".format("Переменная", "Значение",
"Погрешность"))
    error = abs(A @ x - b)
    for i in range(n):
        print("{:<12}{:<25.15f}{:<25.15f}".format(f"x{i + 1}", x[i],
error[i]))
    print("Количество итераций = ", count)


A = np.array([[1.85, 0.70, -0.12, -0.18],
              [0.16, 0.19, 0.79, 0.11],
              [1.13, 2.77, 0.18, -0.20],
              [1.14, 1.01, 0.55, 3.22]])

b = np.array([8.41, -0.23, 13.91, 9.58])

print("Матрица коэффициентов A:")
print(str(A) + '\n')

print("Вектор-столбец свободных членов b:")
print(str(b) + '\n')

# Вывод точного решения
print("Точное решение: ", np.linalg.solve(A, b))

# Метод Гаусса
print("\nРешение системы методом Гаусса с выбором главного элемента:")
gauss_method(A, b)

# Метод Зейделя
print("\nРешение системы методом Зейделя:")
x0 = np.zeros_like(b)
x = seidel_method(A, b, x0)
```