

Федеральное государственное автономное образовательное учреждение
высшего образования

«Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники



УНИВЕРСИТЕТ ИТМО

Лабораторная работа №1 по дисциплине
«Вычислительная математика»

"Решение системы линейных алгебраических уравнений
СЛАУ"

Выполнил: Ермаков Никита Сергеевич

Группа: Р32082

Преподаватель: Машина Е.А

г. Санкт-Петербург

2023

Цель работы:

Научится писать программы для прямых и итерационных методов, понять их отличия, преимущества и недостатки

Задание лабораторной работы:

Вариант 4 => метод простых итераций

1. № варианта определяется как номер в списке группы согласно ИСУ.
2. В программе численный метод должен быть реализован в виде отдельной подпрограммы или класса, в который входные/выходные данные передаются в качестве параметров.
3. Размерность матрицы $n \leq 20$ (задается из файла или с клавиатуры - по выбору конечного пользователя).
4. Должна быть реализована возможность ввода коэффициентов матрицы, как с клавиатуры, так и из файла (по выбору конечного пользователя).

Для итерационных методов должно быть реализовано:

- Точность задается с клавиатуры/файла
- Проверка диагонального преобладания (в случае, если диагональное преобладание в исходной матрице отсутствует, сделать перестановку строк/столбцов до тех пор, пока преобладание не будет достигнуто). В случае невозможности достижения диагонального преобладания - выводить соответствующее сообщение.
- Вывод вектора неизвестных: x_1, x_2, \dots, x_n
- Вывод количества итераций, за которое было найдено решение.
- Вывод вектора погрешностей: $|x_i(k) - x_{i(k-1)}|$

Описание метода, расчетные формулы:

Метод простой итерации - это численный метод, используемый для решения систем линейных уравнений. Он основан на преобразовании исходной системы уравнений в эквивалентную систему, которая может быть решена итерационным процессом.

Итерационный процесс начинается с некоторого начального приближения решения. Затем каждое последующее приближение вычисляется на основе предыдущего, путем применения некоторой функции перехода (итерационной формулы), которая зависит от матрицы системы и ее правой части.

Преимуществами метода простой итерации являются его простота и универсальность: он может быть применен к любой системе линейных уравнений. Однако он может быть неэффективным для систем с большим числом уравнений, особенно если требуется достичь высокой точности решения.

Формула для итераций:

$$x_i^{(k+1)} = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{a_{ij}}{a_{ii}} x_j^k, \quad i = 1, 2, \dots, n$$

где k - номер итерации.

- Начальное приближение выбирается как вектор правых частей уравнения или нулевой вектор
- Сам итерационный процесс продолжается до тех пор, пока значения разниц вычисленных переменных от их значений на прошлом шаге не будет меньше, чем заданное заранее значение погрешности ε

$$\max_{1 \leq i \leq n} |x_i^{(k)} - x_i^{(k-1)}| \leq \varepsilon$$

Функция, реализующая вычисления:

```
def simple_iterations(matrix, rhs, precision):
    n = len(matrix)
    x = [0] * n
    error = [0] * n
    iteration_count = 0

    while True:
        new_x = [0] * n
        for i in range(n):
            s = sum(matrix[i][j] * x[j] for j in range(n) if j != i)
            new_x[i] = (rhs[i] - s) / matrix[i][i]

        error = [abs(new_x[i] - x[i]) for i in range(n)]
        if max(error) < precision:
            break

        x = new_x
        iteration_count += 1

    return x, error, iteration_count
```

Функция принимает матрицу коэффициентов, вектор правых частей уравнения и необходимую точность

Примеры и результат работы программы:

```
Привет пользователь, эта программа предназначена для решения систем линейных уравнений методом простых итераций.
Она не найдет точное решение, но найдет решение с заданной тобою точностью
Выберите способ ввода данных:
1 - Ввод с клавиатуры
2 - Чтение из файла
Введите номер выбранного варианта: 1
Введите размерность матрицы n: 3
Введите матрицу A:
Введите строку матрицы: 2 2 10
Введите строку матрицы: 10 1 1
Введите строку матрицы: 2 10 1
Введите вектор правой части b:
Введите элементы вектора правой части: 14 12 13
Введите точность (например: 0.01): 0.01
Матрица после преобразования (если было выполнено):
[10.0, 2.0, 2.0]
[1.0, 10.0, 1.0]
[1.0, 2.0, 10.0]
Вектор решения: [1.0024000000000002, 1.0015, 1.00192]
Количество итераций: 5
Вектор погрешностей: [0.0030840000000001977, 0.0019320000000000448, 0.0024599999999999067]
```

```
Привет пользователь, эта программа предназначена для решения систем линейных уравнений методом простых итераций.
Она не найдет точное решение, но найдет решение с заданной тобою точностью
Выберите способ ввода данных:
1 - Ввод с клавиатуры
2 - Чтение из файла
Введите номер выбранного варианта: 1
Введите размерность матрицы n: 3
Введите матрицу A:
Введите строку матрицы: 10 2 -1
Введите строку матрицы: -2 -6 -1
Введите строку матрицы: 1 -3 12
Введите вектор правой части b:
Введите элементы вектора правой части: 5 24.42 36
Введите точность (например: 0.01): 0.01
Матрица после преобразования (если было выполнено):
[10.0, 2.0, -1.0]
[-2.0, -6.0, -1.0]
[1.0, -3.0, 12.0]
Вектор решения: [1.6425069444444447, -4.893200462962963, 1.6377907407407406]
Количество итераций: 5
Вектор погрешностей: [8.777777777813078e-05, 0.0027330246913574996, 0.002033564814814648]
```

Выводы по работе:

Метод простых итераций удобен для вычисления приближенных значений корней системы уравнений и нетрудно реализуется в виде программы. Но также необходимо перед началом итерационного процесса убедиться в том, что будет обеспечена сходимость. Один из способов - преобразовать матрицу к виду с диагональным преобладанием, что может занять значительное время (даже при $n \leq 20$), если перебирать все перестановки строк и столбцов.