

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

Факультет программной инженерии и компьютерной техники

Отчет по лабораторной работе №1

По дисциплине «Вычислительная математика»

Вариант 9

Выполнил: Камянецкий Н.В.

Группа: Р3208

Преподаватель: Машина Е.А.

Санкт-Петербург

~ 2024 ~

Цель работы:

Разработать программу, которая будет реализовать метод простых итераций:

- 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 - вывести 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 работы:

Итерационные методы – это методы последовательных приближений. Задается некоторое начальное приближение. Далее с помощью определенного алгоритма проводится один цикл вычислений - итерация. В результате итерации находят новое приближение. Итерации проводятся до получения решения с требуемой точностью.

Дана матрица вида чё

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases}$$

Для которой должно выполняться достаточное условие сходимости

Теорема. Достаточным условием сходимости итерационного процесса к решению системы при любом начальном векторе $x_i^{(0)}$ является выполнение условия преобладания диагональных элементов или доминирование диагонали:

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|, \quad i = 1, 2, \dots, n$$

Если достаточное условие выполняется, то мы выражаем неизвестные:

$$\begin{cases} x_1 = \frac{a_{12}}{a_{11}}x_2 + \frac{a_{13}}{a_{11}}x_3 + \dots + \frac{a_{1n}}{a_{11}}x_n - \frac{b_1}{a_{11}} \\ x_2 = \frac{a_{21}}{a_{22}}x_1 + \frac{a_{23}}{a_{22}}x_3 + \dots + \frac{a_{2n}}{a_{22}}x_n - \frac{b_2}{a_{22}} \\ \dots \dots \\ x_n = \frac{a_{n1}}{a_{nn}}x_1 + \frac{a_{n2}}{a_{nn}}x_2 + \dots + \frac{a_{n-1n-1}}{a_{nn}}x_{n-1} - \frac{b_n}{a_{nn}} \end{cases} \quad (6)$$

После чего выделяется матрица коэффициентов С и вектор свободных членов d

$$c_{ij} = \begin{cases} 0, & \text{при } i = j \\ -\frac{a_{ij}}{a_{ii}}, & \text{при } i \neq j \end{cases}$$

$$d_i = \frac{b_i}{a_{ii}} \quad i = 1, 2, \dots, n$$

Для последующего приближения для нормы матрицы (матрицы C) должно выполняться условие сходимости, то есть:

$$\|C\| < 1$$

Далее вектор свободных членов принимается за начальное (нулевое) приближение, а все последующие приближения вычисляются следующим образом:

$$\begin{array}{l} x_1^1 \rightarrow x_2^0 \quad x_3^0 \quad x_4^0 \\ x_2^1 \rightarrow x_1^1 \quad x_3^0 \quad x_4^0 \\ x_3^1 \rightarrow x_1^1 \quad x_2^1 \quad x_4^0 \\ x_4^1 \rightarrow x_1^1 \quad x_2^1 \quad x_3^1 \end{array}$$

Также высчитывается критерий по абсолютным отклонениям:

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

Листинг программы:

```
# Функция для перестановки и проверки сходимости
def relocate(A, B, n):
    new_matrix_b = [0 for i in range(n)]
    new_matrix = [[0 for i in range(n)] for i in range(n)]
    flag = 0;
    for i in range(n):
        for j in range(n):
            if abs(A[j][i]) > abs(module_sum(A[j])) - abs(A[j][i]):
                new_matrix[i]=A[j]
                new_matrix_b[i]=B[j]
                flag += 1;
                break

    if flag != n:
        print("Условие не может быть выполнено")
        exit()

    for i in range(n):
        divider = new_matrix[i][i]
        for j in range(n):
            new_matrix[i][j]=new_matrix[i][j]/divider*-1
            new_matrix_b[i] = new_matrix_b[i]/divider
            new_matrix[i][i] = 0
    if check_convergence_condition(new_matrix):
        print("Условие сходимости не выполнено")
        exit()
    return new_matrix, new_matrix_b

# Функция для подсчета суммы строки
def module_sum(row):
    sum = 0
    for i in range(len(row)):
        sum += abs(row[i])
    return sum

# Функция для нахождения ответа
def answer(array, B, k, n):
    indexes = B[:]
    end = B[:]
    it = 0
    while True:
        it+=1
```

```

s=0
for i in range(n):
    for j in range(n):
        s += float(indexes[j]) * float(array[i][j])
    s += B[i]
    indexes[i] = s
    s = 0
print(it, indexes)
if(max(max_value(indexes, end, n)) <= k):
    break
if(it > 5):
    break
end = indexes[:]

```

Условие сходимости

```

def check_convergence_condition(matrix_c):
    max_element = max([sum(map(abs, row)) for row in matrix_c])
    if max_element >= 1:
        return True
    return False

```

Функция по поиску разницы значений массивов

```

def max_value(A, B, n):
    array_result = [0 for i in range(n)]
    for i in range(n):
        array_result[i] = abs(abs(A[i]) - abs(B[i]))
    return array_result

```

```

def key_word():

```

Ввод размерности матрицы

```

n = int(input("Введите размерность матрицы: "))

```

Ввод матрицы A

```

A = []

```

```

print("Введите матрицу A:")

```

```

for _ in range(n):

```

```

    row = list(map(float, input().split()))

```

```

    A.append(row)

```

Ввод вектора b

```

print("Введите вектор b:")

```

```

b = list(map(float, input().split()))

```

```

# Ввод точности
print("Введите точность")
k = float(input())

A, b = relocate(A, b, n)
answer(A, b, k, n)

def file_mode():
    with open("lab1/task.txt", "r") as file:
        lines = [line.strip() for line in file]
        n=int(lines[0])
        lines.pop(0)
        A = [[0 for i in range(n)] for i in range(n)]
        for i in range(n):
            A[i] = list(map(float, lines[0].split()))
            lines.pop(0)
        b=list(map(float, lines[0].split()))
        lines.pop(0)
        k=float(lines[0])
        A, b = relocate(A, b, n)
        answer(A, b, k, n)

def choose_mode():

    if (input("Вы желаете взять данные из файла task.txt? [y]")=="y"):
        file_mode()
    else:
        key_word()

choose_mode()

```

Пример программы при ввода с файла:

```

Вы желаете взять данные из файла task.txt? [y]y
1 [0.9299999999999999, 0.9740000000000001, 1.0191999999999999]
2 [1.00068, 0.997944, 1.0002752]
3 [1.00017808, 0.999936864, 0.9999770111999999]

```

Task.txt:

```
task.txt
3
2 2 10
10 1 1
2 10 1
14 12 13
0.01
```

Пример программы при вводе руками:

```
Введите матрицу A:
2 2 10
10 1 1
2 10 1
Введите вектор b:
14 12 13
Введите точность
0.01
1 [0.9299999999999999, 0.9740000000000001, 1.0191999999999999]
2 [1.00068, 0.997944, 1.0002752]
3 [1.00017808, 0.999936864, 0.9999770111999999]
PS D:\ITMO\Course 2\Выч мат>
```

Вывод:

В результате данной лабораторной работы я написал программу, которая позволяет решать СЛАУ размера $n \times n$ при помощи метода Гаусса-Зейделя.