Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования
Московский государственный технический университет имени Н.Э. Баумана

Факультет: Информатика и системы управления
Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа №7
по курсу «Численные методы»
«Решение системы линейных алгебраических уравнений методом простой итерации и методом Зейделя»

Выполнила:
студентка группы ИУ9-62Б
Самохвалова П. С.

Проверила:
Домрачева А. Б.

Москва, 2023

Цель:

Анализ метода простой итерации и метода Зейделя для решения системы линейных алгебраических уравнений.

Постановка задачи:

Дано: $Ax = b$, $x = (x_i)$, $b = (b_i)$, $A = (a_{ij})$, $i, j = \overline{1, n}$.

Найти: $x = (x_i)$.

Тестовый пример:

Вариант 21

$$A = \begin{pmatrix} 10.0 + \alpha & -1.0 & 0.2 & 2.0 \\ 1.0 & 12.0 - \alpha & -2.0 & 0.1 \\ 0.3 & -4.0 & 12.0 - \alpha & 1.0 \\ 0.2 & -0.3 & -0.5 & 8.0 - \alpha \end{pmatrix}$$

$$b = \begin{pmatrix} 1.0 + \beta \\ 2.0 - \beta \\ 3.0 \\ 1.0 \end{pmatrix}$$

$$\alpha = 0.1 * 21, \quad \beta = 0.1 * 21$$

Задание:

1. Решить систему методом простой итерации с относительной погрешностью 0.01:
   а) преобразовать систему к виду x = Fx + c, распечатать матрицу F и столбец c;
   б) найти норму ||F||;
   в) Для каждой итерации распечатать абсолютную и относительную ошибки.
2. Решить систему методом Зейделя, обеспечить $||x^k - x^{k-1}|| \leq 10^{-4}$

Описание методов:

Преобразуем исходную СЛАУ к виду $x = Fx + c$, $x = (x_i)$, $c = (c_i)$, $F = (f_{ij})$, $i, j = \overline{1, n}$.

Зададимся некоторым начальным приближением к решению $x^0 = (x_1^0, ..., x_n^0)^T$. Используя метод простой итерации каждое последующее приближение (итерацию) находим по предыдущему:

$$x^k = Fx^{k-1} + c, \quad k = 1, 2, ...$$

Последовательность приближений сходится к точному решению

$$\lim_{k \to \inf} x^k = x,$$

если какая-либо норма матрицы F меньше единицы, например,

$$||F|| = \max_{1 \le i \le n} \sum_{j=1}^{n} |f_{ij}| < 1.$$

При этом абсолютная погрешность очередной итерации

$$\Delta_k \le \frac{||F||}{1 - ||F||} ||x^k - x^{k-1}||,$$

а относительная погрешность -

$$\delta_k = \frac{\Delta_k}{||x^k||},$$

где $||x^k|| = \max_{1 \le i \le n} |x_i^k|$.

Всякую невырожденную систему Ax = b можно привести к виду x = Fx + c, для которого ||F|| < 1 и метод простой итерации сходится.

Если для всех строк $|a_{ii}| > \sum\limits_{j \ne i} |a_{ij}|, \quad i = \overline{1, n}$ (диагональные элементы превалируют) , то для приведения к требуемому виду достаточно разрешить каждое уравнение системы относительно ведущей неизвестной. Тогда

$$f_{ij} = \begin{cases} 0 & \text{, i = j} \\ -\frac{a_{ij}}{a_{ii}} & , i \ne j \end{cases}$$

и $c_i = \frac{b_i}{a_{ii}}, \quad i, j = \overline{1, n}; \quad ||F|| < 1$.

Метод Зейделя является модификацией метода простой итерации и сходится, как правило, быстрее. Для того, чтобы решить СЛАУ методом Зейделя необходимо представить матрицу F как сумму нижнетреугольной матрицы F_d и верхнетреугольной матрицы F_u.

Тогда k-ая итерация метода будет удовлетворять рекуррентному соотношению

$$x^k = F_d x^k + F_u x^{k-1} + c, \quad k = 1, 2, ...$$

Листинг 1. Решение СЛАУ методом простой итерации и методом Зейделя

```
def norm_v(x):
    norm = 0
    n = len(x)
    for i in range(n):
        if abs(x[i]) > norm:
            norm = abs(x[i])
    return norm


def sub_v(x, y):
```

```python
    n = len(x)
    sub = [0] * n
    for i in range(n):
        sub[i] = x[i] - y[i]
    return sub


def sum_v(x, y):
    n = len(x)
    sum = [0] * n
    for i in range(n):
        sum[i] = x[i] + y[i]
    return sum


def mul_mv(a, x):
    n = len(x)
    mul = [0] * n
    for i in range(n):
        for j in range(n):
            mul[i] += a[i][j] * x[j]
    return mul


k = 21
a = 0.1 * k
b = 0.1 * k
am = [[10.0 + a, -1.0, 0.2, 2.0],
      [1.0, 12.0 - a, -2.0, 0.1],
      [0.3, -4.0, 12.0 - a, 1.0],
      [0.2, -0.3, -0.5, 8.0 - a]]
bm = [1.0 + b, 2.0 - b, 3.0, 1.0]

n = 4

fm = [[0] * n for i in range(n)]
cm = [0] * n

for i in range(n):
```

```
    for j in range(n):
        if i != j:
            fm[i][j] = -am[i][j] / am[i][i]
for i in range(n):
    cm[i] = bm[i] / am[i][i]

print("F =")
for i in range(n):
    for j in range(n):
        print("{:24} ".format(str(fm[i][j])), end = "")
    print()
print()

print("c =")
for i in range(n):
    print("{:24}".format(str(cm[i])))
print()

fm_norm = 0
for i in range(n):
    s = 0
    for j in range(n):
        s += abs(fm[i][j])
    if s > fm_norm:
        fm_norm = s

print("||F|| =", fm_norm)
print()

k = 0
xk = cm[:]
otn_delta = 1

print("A simple iteration method with a relative error of 0.01")
print()

while otn_delta > 0.01:
    k += 1
    xk_last = xk[:]
```

```
    xk = sum_v(mul_mv(fm, xk_last), cm)
    abs_delta = norm_v(sub_v(xk, xk_last))
    otn_delta = abs_delta / norm_v(xk)
    print("{:4} {:24} {:24}".format(str(k), str(abs_delta), str(otn_delta)))

print()

fd = [[0] * n for i in range(n)]
fu = [[0] * n for i in range(n)]

for i in range(n):
    for j in range(n):
        if j < i:
            fd[i][j] = fm[i][j]

for i in range(n):
    for j in range(n):
        if i <= j:
            fu[i][j] = fm[i][j]

print("Seidel's method with an absolute error of 0.0001")
print()

k = 0
xk = cm[:]
abs_delta = 1

while abs_delta > 0.0001:
    k += 1
    xk_last = xk[:]
    xk = mul_mv(fu, xk_last)
    xk = mul_mv(fd, xk)
    xk = sum_v(xk, cm)
    abs_delta = norm_v(sub_v(xk, xk_last))
    otn_delta = abs_delta / norm_v(xk)
    print("{:4} {:24} {:24}".format(str(k), str(abs_delta), str(otn_delta)))
```

Результаты работы:

$$F = \begin{pmatrix} 0 & 0.08264462809917356 & -0.01652892561983471 & -0.1652892561983471 \\ -0.10101010101010101 & 0 & 0.20202020202020202 & -0.010101010101010102 \\ -0.0303030303030303 & 0.40404040404040403 & 0 & -0.10101010101010101 \\ -0.03389830508474576 & 0.05084745762711864 & 0.0847457627118644 & 0 \end{pmatrix},$$

$$c = \begin{pmatrix} 0.25619834710743805 \\ -0.010101010101010109 \\ 0.30303030303030304 \\ 0.1694915254237288 \end{pmatrix},$$

$||F|| = 0.5353535353535354.$

Таблица погрешностей, получаемых на каждой итерации при решении СЛАУ методом простой итерации с относительной погрешностью 0.01:

Номер итерации	Абсолютная погрешность	Относительная погрешность
1	0.03385868773177919	0.12354248038332181
2	0.01294805238429747	0.04511309071213607
3	0.0025578023329712873	0.008946285079489141

Таблица погрешностей, получаемых на каждой итерации при решении СЛАУ методом Зейделя с абсолютной погрешностью 0.0001:

Номер итерации	Абсолютная погрешность	Относительная погрешность
1	0.02506893293170409	0.07640655687051644
2	0.0020527513408289955	0.006217594985869593
3	0.00016866968910478342	0.0005106241027756813
4	1.3853778181194265e-05	4.193863411030351e-05

Выводы:

В результате выполнения лабораторной работы был изучен метод простой итерации и метод Зейделя решения системы линейных алгебраических уравнений, была написана реализация на языке программирования Python. При применении метода простой итерации к тестовому примеру потребовались 3 итерации для достижения относительной погрешности 0.01. На 3 итерации абсолютная погрешность составила 0.0025578023329712873, относительная погрешность составила 0.008946285079489141. При применении метода Зейделя к тестовому примеру потребовались 4 итерации для достижения абсолютной погрешности 0.0001. На 4 итерации абсолютная погрешность составила 1.3853778181194265e-05, относительная погрешность составила 4.193863411030351e-05.