Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»
Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №1 по дисциплине «Вычислительная математика»

Вариант 4

Выполнил:
Студент группы P3212
Данько Савелий Максимович
Преподаватель:

г. Санкт-Петербург
2025

Цель работы:

Реализация итерационного метода: “Метод простых итераций” для решения СЛАУ на языке программирования Python.

Описание метода:

УНИВЕРСИТЕТ ИТМО Лекция №1. Методы решения систем линейных уравнений

Итерационные методы. Метод простой итерации

Рассмотрим систему линейных уравнений с невырожденной матрицей ($\det A \neq 0$):

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \qquad\qquad \ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases} \qquad (5)$$

Приведем систему уравнений к виду (6), выразив неизвестные $x_1, x_2, \ldots, x_n$ соответственно из первого, второго и т.д. уравнений системы (5):

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

Итерационные методы. Метод простой итерации

Обозначим:

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

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

Тогда получим:

$$\begin{cases} x_1 = c_{11}x_1 + c_{12}x_2 + \dots + c_{1n}\,x_n + d_1 \\ x_2 = c_{21}x_1 + c_{22}x_2 + \dots + c_{2n}\,x_n + d_2 \\ \dots\dots \\ x_n = c_{n1}x_1 + c_{n2}x_2 + \dots + c_{nn}\,x_n + d_n \end{cases}$$

Или в векторно-матричном виде: $\boldsymbol{x = Cx + D}$, где x – вектор неизвестных, C – матрица коэффициентов преобразованной системы размерности $n*n$, D – вектор правых частей преобразованной системы.

Итерационные методы. Метод простой итерации

Систему (6) представим в сокращенном виде:

$$\boldsymbol{x_i = \sum_{j=1}^{n} c_{ij}x_j + d_i}\ , \qquad i = 1, 2, \dots, n$$

$$c_{ij} = \begin{cases} 0, \text{ при } i = j \\ -\dfrac{a_{ij}}{a_{ii}}, \text{ при } i \neq j \end{cases} \qquad d_i = \frac{b_i}{a_{ii}} \quad i = 1, 2, \dots, n$$

Рабочая формула метода простой итерации:

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

где k – номер итерации.

За начальное (нулевое) приближение выбирают вектор свободных членов: $x^{(0)} = D$ или нулевой вектор: $x^{(0)} = 0$

Следующее приближение: $\vec{x}^{(1)} = c\vec{x}^{(0)} + \vec{d},\ \ \vec{x}^{(2)} = c\vec{x}^{(1)} + \vec{d}$ ...

$$\vec{x}^{(k)} = c\vec{x}^{(k-1)} + \vec{d}$$

Листинг программы:
Полный код программы:
[ссылка на git]()

Реализация метода простых итераций:
```
def simple_iteration_method(A, b, eps, max_iterations=1000):
   n = len(A)
   x_old = [0.0] * n
   iterations = 0

   while iterations < max_iterations:
       x_new = [0.0] * n
       for i in range(n):
           s = 0.0
           for j in range(n):
               if j != i:
                   s += A[i][j] * x_old[j]
           x_new[i] = (b[i] - s) / A[i][i]
       error_vector = [abs(x_new[i] - x_old[i]) for i in range(n)]
       iterations += 1
       if max(error_vector) < eps:
           return x_new, iterations, error_vector
       x_old = x_new
   print("Максимальное число итераций достигнуто.")
   return x_new, iterations, error_vector
```

Пример работы программы:

Ввод с клавиатуры:

```
Решение СЛАУ методом простых итераций.
Выберите способ ввода данных:
1 - с клавиатуры
2 - из файла
Ваш выбор (1 или 2): 1
Введите размерность матрицы n (<=20): 3
Введите коэффициенты матрицы A построчно, разделяя пробелами:
Строка 1: 2 2 10
Строка 2: 10 1 1
Строка 3: 2 10 1
Введите вектор правых частей b (числа через пробел): 14 12 13
Введите требуемую точность eps: 0.01

Проверка диагонального преобладания матрицы A...
Исходная матрица не обладает диагональным преобладанием.
Пытаемся переставить строки для обеспечения диагонального преобладания...
Перестановка строк выполнена. Новая матрица имеет диагональное преобладание.

Норма матрицы A (∞-норма): 14.0

Выполняется метод простых итераций...

Найденное решение:
  x[1] = 0.999568
  x[2] = 0.99946
  x[3] = 0.999316

Количество итераций: 6

Вектор погрешностей |x_i^(k) - x_i^(k-1)|:
  Компонента 1: 0.0019320000000000448
  Компонента 2: 0.0024599999999999067
  Компонента 3: 0.0030840000000001977
```

Ввод из файла:
Пример файла input.txt:

```
3
2 2 10
10 1 1
2 10 1
14 12 13
0.01
```

Пример работы:

```
Решение СЛАУ методом простых итераций.
Выберите способ ввода данных:
1 – с клавиатуры
2 – из файла
Ваш выбор (1 или 2): 2
Введите имя файла: input.txt

Проверка диагонального преобладания матрицы A...
Исходная матрица не обладает диагональным преобладанием.
Пытаемся переставить строки для обеспечения диагонального преобладания...
Перестановка строк выполнена. Новая матрица имеет диагональное преобладание.

Норма матрицы A (∞-норма): 14.0

Выполняется метод простых итераций...

Найденное решение:
  x[1] = 0.999568
  x[2] = 0.99946
  x[3] = 0.999316

Количество итераций: 6

Вектор погрешностей |x_i^(k) - x_i^(k-1)|:
  Компонента 1: 0.0019320000000000448
  Компонента 2: 0.0024599999999999067
  Компонента 3: 0.0030840000000001977
```

Вывод:

В процессе выполнения данной лабораторной работы я смог реализовать итерационный метод “Метод простых итераций” для решения СЛАУ на языке программирования Python.