

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО»**

Отчет

по лабораторной работе №4 «Методы решения СЛАУ»

по дисциплине «**Прикладная математика**»

Автор: Константинова О.А., Векинцева В.А.

Факультет: Информационные технологии и программирование

Группа: М32111



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2023

Цель работы: изучить методы решения систем линейных алгебраических уравнений – точные методы Гаусса и итерационные методы (Зейделя или Якоби)

Задачи:

- 1) Реализовать метод Гаусса с выбором ведущего элемента для решения СЛАУ
- 2) Реализовать алгоритм LU-разложения с использованием разреженно-строчного формата хранения матрицы, а также метод решения СЛАУ с использованием LU-разложения
- 3) Реализовать итерационный метод решения СЛАУ (метод Зейделя, Якоби или верхней релаксации на выбор)
- 4) Провести исследование реализованных методов на системах с матрицами A^k , число обусловленности которых регулируется за счёт изменения диагонального преобладания
- 5) Оценить зависимость числа обусловленности и точности полученного решения в зависимости от параметра k
- 6) Провести аналогичные исследования на матрице Гильберта
- 7) Сравнить между собой прямые и итерационные методы по эффективности методов в зависимости от размеров матрицы

Ход работы:

Для выполнения данной лабораторной работы сгенерируем набор матриц СЛАУ размерностью 10, 50, 100, 1000, 10000 и 100000.

Для представления матриц в памяти используем разреженно-строчный формат хранения, который реализован библиотекой *scipy.sparse*, классом *csc_matrix*.

Разреженно-строчный формат – формат представления матрицы, удобной для случаев, когда матрица имеет большое количество нулей.

Число обусловленности задаётся за счёт изменения диагонального преобладания, т. е. внедиагональные элементы матрицы задаются случайным образом из набора чисел $[0, -1, -2, -3, -4]$, а диагональные получаются суммированием всех элементов из строки с противоположным знаком.

При этом диагональный элемент на первой строке всегда дополнительно суммируется с 10^{-k} (в нашей реализации мы задаём k равными 1, 2, 3, 4 и повторяем расчёт для каждого значения).

Константы (правая часть уравнения, свободные члены без неизвестных) для каждого уравнения СЛАУ рассчитываются подставлением значений 1, 2, ..., n в каждое уравнение. Тем самым, ожидаемое решение СЛАУ должно быть близко к значениям 1, 2, ..., n.

Полученное вычисленное решение мы оцениваем, сравнивая значения с 1, 2, ..., n и вычисляя абсолютную погрешность по формуле: $\varepsilon_{abs} = \max_{i=1, \dots, n} |x_i - \bar{x}_i|$, где x_i – искомая неизвестная, $\bar{x}_i = i$ – заранее известное решение для i-ой неизвестной.

Метод Гаусса с выбором ведущего элемента

В основе этого метода лежит идея последовательного исключения неизвестных, приводящая исходную систему с квадратной матрицей к легко разрешимой системе с верхней треугольной матрицей.

Метод состоит из двух этапов:

- 1) **прямой ход** – приведение исходной матрицы к верхней треугольной матрице
- 2) **обратный ход** – вычисление неизвестных в обратном порядке (снизу вверх)

Метод даёт точное решение, если все исходные данные точны и вычисления производятся точно, но при вычислении программным образом неизбежны округления и внесение погрешности в решение.

Основное накопление погрешности происходит на этапе прямого хода.

Модификация с выбором ведущего элемента на каждом шаге исключения i-ого неизвестного в качестве ведущего уравнения используется то, которое содержит максимальный по модулю коэффициент. Выбрав ведущее уравнение, мы меняем его позицию на i-ую.

Такой подход позволяет уменьшить накапливаемую погрешность, поскольку мы исключаем самый большой по модулю коэффициент из операции умножения с участием числа с погрешностью.

Сделаем расчёт для СЛАУ размерностью 10 с разными значениями параметра k, задающего матрицу:

Параметр k	Размерность матрицы	Время работы алгоритма	Абсолютная погрешность

1	10×10	0.046 сек	0.0000000000000002
2	10×10	0.046 сек	0.0000000000000004
3	10×10	0.047 сек	0.0000000000000004
4	10×10	0.049 сек	0.0000000000000005
5	10×10	0.045 сек	0.0000000000000004

Можно сделать вывод, что число k не влияет на результат видимым образом.

Метод LU-разложения

Метод является модификацией метода Гаусса, основанной на разложении матрицы СЛАУ на вспомогательные матрицы L и U , такие что: $A = L \times U$.

Элементы матриц рассчитываются по следующим формулам:

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}, i \leq j$$

$$l_{ij} = \frac{1}{u_{ij}} (a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}), i > j$$

$$l_{ij} = 1, i = j$$

Матрица L , вычисляемая в рамках алгоритма, является нижней треугольной с единицами на главной диагонали. Матрица U в результате вычислений получается верхней треугольной.

Поскольку обе матрицы треугольные, то решение СЛАУ с такими матрицами элементарно.

Получив LU-разложение, необходимо решить две вспомогательных СЛАУ.

1) $L \times y = b$, где b – константы (свободные члены) из исходной системы уравнений.

Найденное решение y нужно подставить во вторую вспомогательную систему:

2) $U \times x = y$ - решение этой системы и будет решением исходной задачи.

Сделаем расчёт для СЛАУ размерностью 10 с разными значениями параметра k , задающего матрицу:

Параметр k	Размерность матрицы	Время работы алгоритма	Абсолютная погрешность
1	10×10	0.034 сек	0.0000000000000002
2	10×10	0.034 сек	0.0000000000000004
3	10×10	0.034 сек	0.0000000000000003
4	10×10	0.034 сек	0.0000000000000003
5	10×10	0.034 сек	0.0000000000000002

Аналогично методу Гаусса, не видно влияния параметра k на погрешность и время работы.

Сильной стороной LU-разложения является то, что получив разложение для матрицы A, можно применять его для решения семейства систем уравнения, отличающихся только константами правой части.

Метод Зейделя

Метод относится к итерационным методам, которые обладают высокой эффективностью. Сам метод является улучшением *метода простых итераций*, который работает следующим образом: пусть имеется исходная система уравнений $A \cdot x = b$, приведём её к форме $x = D \cdot x + p$. Например, по формуле для i-го уравнения:

$$x_i = \frac{(-a_{i1} \cdot x_1 - a_{i2} \cdot x_2 - \dots - 0 \cdot x_i - \dots - a_{in} \cdot x_n)}{a_{ii}} + \frac{b_i}{a_{ii}}$$

Данная формула позволяет рассчитать элементы матрицы D и вектора p. Таким образом, на главной диагонали матрицы D будут нули.

Процесс поиска решения строится по итерационной схеме:

$$x^1 = D \cdot x^0 + p$$

$$x^2 = D \cdot x^1 + p$$

...

$$x^k = D \cdot x^{k-1} + p$$

Где x^0 – начальное приближение вектора решения системы (в нашей реализации взят нулевой вектор).

Работа алгоритма продолжается до тех пор, пока абсолютная погрешность $\varepsilon_{abs} = \max_{i=1,..,n} |x_i^k - x_i^{k-1}|$, полученного приближения на k-ом шаге не станет отличаться от приближения на предыдущем шаге меньше чем на заданную точность (в нашей реализации $\varepsilon = 0,000001$)

Отличие метода Зейделя заключается в том, что процесс итераций реализуется следующим образом: как только на k-ой итерации вычислена i-ая компонента x_i^k вектора x^k , её значение используют для вычисления последующих компонент $x_{i+1}^k, x_{i+2}^k, \dots, x_n^k$

$$x_1^k = (d_{11}x_1^{k-1} + d_{12}x_2^{k-1} + d_{13}x_3^{k-1} + \dots + d_{1n}x_n^{k-1}) + p_1$$

$$x_2^k = (d_{21}x_1^k + d_{22}x_2^{k-1} + d_{23}x_3^{k-1} + \dots + d_{2n}x_n^{k-1}) + p_2$$

$$x_3^k = (d_{31}x_1^k + d_{32}x_2^k + d_{33}x_3^{k-1} + \dots + d_{3n}x_n^{k-1}) + p_3$$

...

$$x_n^k = (d_{n1}x_1^k + d_{n2}x_2^k + d_{n3}x_3^k + \dots + d_{nn}x_n^{k-1}) + p_n$$

Метод Зейделя позволяет за то же количество шагов, что и метод простых итераций, получить более точный результат.

Сделаем расчёт для СЛАУ размерностью 10 с разными значениями параметра k, задающего матрицу:

Параметр k	Размерность матрицы	Время работы алгоритма	Абсолютная погрешность	Количество итераций
1	10×10	0.017 сек	0.0000001	11
2	10×10	0.019 сек	0.0000002	12
3	10×10	0.020 сек	0.0000001	13
4	10×10	0.021 сек	0.0000001	14
5	10×10	0.021 сек	0.00000009	14

Запуск алгоритма с различными параметрами k также не выявил закономерности абсолютной погрешности от значений параметра.

Сравнение работы методов

Запустим алгоритмы на матрицах разных размерностей:

Название метода	Параметр k	Размерность матрицы	Время работы алгоритма	Абсолютная погрешность	Количество итераций
Гаусс	1	10×10	0.046 сек	0.0000000000000003	-
LU	1	10×10	0.034 сек	0.0000000000000002	-
Зейдель	1	10×10	0.017 сек	0.000009	11
Гаусс	1	50×50	4.395 сек	0.0000000000000004	-
LU	1	50×50	1.820 сек	0.0000000000000004	-
Зейдель	1	50×50	0.327 сек	0.000006	11
Гаусс	1	100×100	Нет данных		
LU	1	100×100	Нет данных		
Зейдель	1	100×100	1.261 сек	0.00002	11
Гаусс	1	1000×1000	Нет данных		
LU	1	1000×1000	Нет данных		
Зейдель	1	1000×1000	143.387 сек	0.000007	13

Из результатов видно, что методы Гаусса и LU более точны, но в то же время работают значительно дольше, чем итерационный метод Зейделя. Также можно сделать вывод, что LU-разложение более хорош с точки зрения производительности.

Применимость для матрицы Гильберта

Матрица Гильберта задаётся по формуле для вычисления элементов:

$$a_{ij} = \frac{1}{i + j - 1}$$

Вид матрицы таков, что все элементы, кроме первого верхнего a_{11} , равного единице, являются дробями, меньшими единицы. Чем больше индексы i и j , тем меньше значение элементов.

Все методы проявили себя не лучшим образом на данной матрице:

- 1) сильно упала точность решения
- 2) метод Зейделя с точностью 0,000001 работает очень долго для любых размерностей матрицы (для размерности > 100 результаты получить вообще не удалось)
- 3) метод Зейделя оказался точнее на матрицах большой размерности

Название метода	Параметр k	Размерность матрицы	Время работы алгоритма	Абсолютная погрешность	Количество итераций
Гаусс	1	10×10	0.045 сек	0.002	-
LU	1	10×10	0.031 сек	0.0006	-
Зейдель	1	10×10	0.027 сек	5.2034	25
Гаусс	1	50×50	3.128 сек	8254.01723	-
LU	1	50×50	1.406 сек	2627.5258	-
Зейдель	1	50×50	6.155 сек	10.8245	317
Гаусс	1	100×100	24.617 сек	70216.5852	-

LU	1	100×100	12.334 сек	36470.3829	-
Зейдель	1	100×100	58.665 сек	23.3446	702