Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский Нижегородский
государственный университет им. Н.И. Лобачевского»

Институт Информационных технологий, математики и механики

Отчёт по лабораторной работе

Методы решения систем линейных уравнений

Выполнил:
студент гр. 381806-01

Алибеков М.Р.

Проверил:
Доцент каф. ДУМЧА, ИИТММ

Эгамов А.И.

Нижний Новгород
2020 г.

Содержание

1. Введение

Лабораторная работа направлена на изучение вопроса решения систем линейных алгебраических уравнений различными методами и сравнение их производительности.

Система линейных алгебраических уравнений (линейная система, СЛУ, СЛАУ) – система уравнений, каждое из которых является линейным – алгебраическим уравнением первой степени.

Решение систем линейных алгебраических уравнений – одно из классических задач линейной алгебры, во многом определившая её объекты и методы. Помимо этого, линейные алгебраические уравнения и методы их решения играют важную роль во многих прикладных направлениях.

Методы решения СЛАУ разделяют на прямые и итерационные.

Прямые методы дают алгоритм, по которому можно найти точное решение систем линейных алгебраических уравнений. К наиболее известным прямым методам относят: метод Гаусса, метод Крамера, метод LU-разложения.

Итерационные методы основаны на использовании повторяющегося процесса и позволяют получить решение в результате последовательных приближений. В число итерационных методов входят: метод простых итераций, метод Зейделя, метод релаксации, метод Якоби.

2. Постановка задачи

В рамках лабораторной работы ставится задача реализации программного комплекса с графическим пользовательским интерфейсом (на одном из языков программирования высокого уровня), который позволит решать системы линейных уравнений от n.переменных ($n \in [2, 16]$) следующими методами: методом Гаусса, методом Крамера, методом Зейделя, методом простых итераций, методом верхней релаксации, методом LU-разложения.

В задаче присутствуют явные ограничения на количество переменных ($n \in [2, 16]$).

Программный комплекс должен обладать следующими возможностями:

- Получение входных данных в двух режимах:
  - ручной ввод входных данных
  - автоматическая генерация случайных коэффицентов
- Нахождение решения каждым из представленных выше методами.
- Замеры времени (производительности) для каждого из методов.

Для работы с каждым из методов решения СЛАУ должен быть представлен отдельный класс. Также целесообразно иметь некий общий абстрактный класс, содержащий общие для всех методов действия и признаки.

Программное решение должно выглядеть приблизительно следующим образом:

1. Модуль Form1 – поддержка GUI (графического пользовательского интерфейса) для работы с программой.
2. Модуль SLAESolvingMethod – общий абстрактный класс, содержащий общие для всех методов действия и признаки.
3. Модуль GaussMethod – модуль для решения СЛАУ методом Гаусса.
4. Модуль CramerMethod – модуль для решения СЛАУ методом Крамера.
5. Модуль SeidelMethod – модуль для решения СЛАУ методом Зейделя.
6. Модуль SimpleIterationMethod – модуль для решения СЛАУ методом простых итераций.
7. Модуль UpperRelaxationMethod – модуль для решения СЛАУ методом верхней релаксации.
8. Модуль LUDecompositionMethod – модуль для решения СЛАУ методом LU-разложения.

3. Теоретическая основа

3.1 Метод Гаусса

Метод Гаусса — классический метод решения системы линейных алгебраических уравнений (СЛАУ). Назван в честь немецкого математика Карла Фридриха Гаусса. Это метод последовательного исключения переменных, когда с помощью элементарных преобразований система уравнений приводится к равносильной системе треугольного вида, из которой последовательно, начиная с последних (по номеру), находятся все переменные системы.

Данный метод чрезвычайно подробно рассматривается в рамках курса Линейной алгебры, поэтому подробно останавливаться на нём не стану.

3.2 Метод Крамера

Пусть дана СЛУ $Ax = b$, заданная матрицами A, b:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

Данная система называется крамеровской, если $\Delta(A) \neq 0$. Такая система имеет единственное решение, и оно находится по формулам Крамера.

$$x_j = \frac{\Delta_j}{\Delta}, j = 1, 2, \dots, n,$$

где определитель Δ_j получается из определителя Δ заменой j-ого столбца столбцом свободных членов системы.

3.3 Метод LU-разложения

Пусть дана СЛУ $Ax = b$, заданная матрицами A, b:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

LU- разложение – представление матрицы A в виде произведения двух матриц, $A = LU$, где L – нижняя треугольная матрица, а U – верхняя треугольная матрица.

$$L = \begin{pmatrix} l_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ l_{n1} & \cdots & l_{nn} \end{pmatrix}, \quad U = \begin{pmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & u_{nn} \end{pmatrix}$$

Для решения $Ax = b$ сначала находим LU –разложение $A = LU$, затем решаем систему $Ly = b$, где $y = Ux$. В конце решаем систему $y = Ux$ и получаем ответ.

Применение матрицы L равносильно проведению прямого хода метода Гаусса, а применение матрицы U равносильно обратному ходу метода Гаусса.

3.4 Метод простых итераций

Пусть дана СЛУ $Ax = b$, заданная матрицами A, b:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

Данный метод является итерационным, поэтому для запуска алгоритма помимо СЛАУ необходимо задать требуемую точность решения.

Сам алгоритм выглядит следующим образом:

- Задать вектор начального приближения (например, вектор свободных членов).
- Рассчитать следующее приближение по формуле:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{i \neq j} a_{ij} x_j^{(k)})$$

- Если норма вектора невязки выше заданной, перейти на предыдущий шаг, а иначе – решение найдено.

3.5 Метод Зейделя

Метод Зейделя представляет собой некоторое изменения метода простых итераций. В нем при вычислении $(k+1)$ – ого приближения неизвестной x_i используются уже вычисленные значения $(k+1)$ – ого приближения для неизвестных $x_1, x_2, \dots, x_{i-1}$. Если для приведенной системы

$$\begin{cases} x_1 = b_{11}x_1 + b_{12}x_2 + \dots + b_{1n}x_n + h_1, \\ x_2 = b_{21}x_1 + b_{22}x_2 + \dots + b_{2n}x_n + h_2, \\ \dots \\ x_n = b_{n1}x_1 + b_{n2}x_2 + \dots + b_{nn}x_n + h_n. \end{cases}$$

уже найдено k – е приближение $x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}$, то $(k+1)$ – е приближение находится по формулам

$$\begin{cases} x_1^{(k+1)} = b_{11}x_1^{(k)} + b_{12}x_2^{(k)} + \dots + b_{1n}x_n^{(k)} + h_1, \\ x_2^{(k+1)} = b_{21}x_1^{(k+1)} + b_{22}x_2^{(k)} + \dots + b_{2n}x_n^{(k)} + h_2, \\ x_3^{(k+1)} = b_{31}x_1^{(k+1)} + b_{32}x_2^{(k+1)} + \dots + b_{3n}x_n^{(k)} + h_3, \\ \dots \\ x_n^{(k+1)} = b_{n1}x_1^{(k+1)} + \dots b_{n,n-1}x_{n-1}^{(k+1)} + b_{nn}x_n^{(k)} + h_n. \end{cases}$$

$$k = 0,1,2, \dots .$$

Если матрицу B итерационного процесса Зейделя представить в виде $B = B_1 + B_2$, где

$$B_1 = \begin{pmatrix} 0 & 0 & \dots & 0 \\ b_{21} & 0 & \cdots & 0 \\ \dots & \dots & \dots & \dots \\ b_{n1} & b_{n2} & \cdots & 0 \end{pmatrix}, \qquad B_2 = \begin{pmatrix} b_1 & b_{12} & \dots & b_{1n} \\ 0 & b_{22} & \cdots & b_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \cdots & b_{nn} \end{pmatrix},$$

то систему можно записать в матричной форме следующим образом:

$$x^{(k+1)} = B_1 x^{(k+1)} + B_2 x^{(k)} + h ,$$

или, что тоже самое, в виде

$$(E - B_1)x^{(k+1)} = B_2 x^{(k)} + h .$$

Матрица $E - B_1$ является левой нижней треугольной матрицей с единицами по главной диагонали. Поэтому она имеет обратную матрицу $(E - B_1)^{-1}$. Умножив слева обе части равенства на матрицу $(E - B_1)^{-1}$, приведем к равенству

$$x^{(k+1)} = (E - B_1)^{-1} B_2 x^{(k)} + (E - B_1)^{-1} h .$$

Таким образом, итерационный процесс Зейделя эквивалентен процессу простой итерации. Для сходимости обоих этих матриц достаточно, что какая-либо норма матрицы $(E - B_1)^{-1} B_2$ была меньше единицы.

3.6 Метод верхней релаксации

Среди явных одношаговых итерационных методов наибольшее распространение получил метод верхних релаксаций. Это связано с тем, что метод верхних релаксаций содержит свободный параметр ω, изменяя который можно получать различную скорость сходимости итерационного процесса.

Наиболее эффективно этот метод применяется при решении множества близких алгебраических систем линейных уравнений. На первом этапе проводится решение одной из систем с различными значениями итерационного параметра ω и из анализа скорости сходимости итерационного процесса выбирается оптимальное значение этого параметра. Затем все остальные системы решаются с выбранным значением ω.

Основная вычислительная формула имеет вид:

$$x_j^{i+1} + \omega \sum_{k=1}^{j-1} \frac{a_{jk}}{a_{jj}} x_k^{i+1} = (1-\omega) x_j^i - \omega \sum_{k=j+1}^{m} \frac{a_{jk}}{a_{jj}} x_k^i + \omega \frac{f_i}{a_{jj}}$$

В остальном данный метод полностью идентичен методу простых итераций.

4. Руководство пользователя

При запуске программы перед пользователем появляется интерфейс управления:

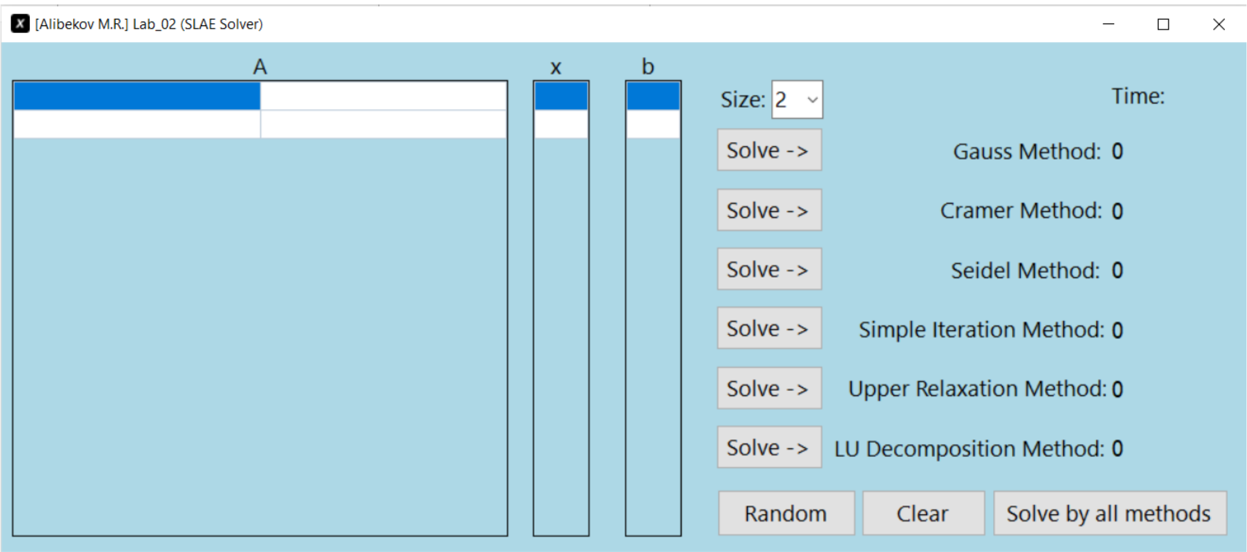

Рисунок 1. Окно приложения

В случае использования ручного режима пользователь может самостоятельно ввести необходимые коэффициенты, следуя подсказке, отображаемой рядом с полем ввода. После заполнения можно выбрать решение каким-либо определённым методом (для этого нажмите на кнопку “Solve ->”), либо всеми сразу (“Solve by all methods”). При решении также производятся замеры времени.

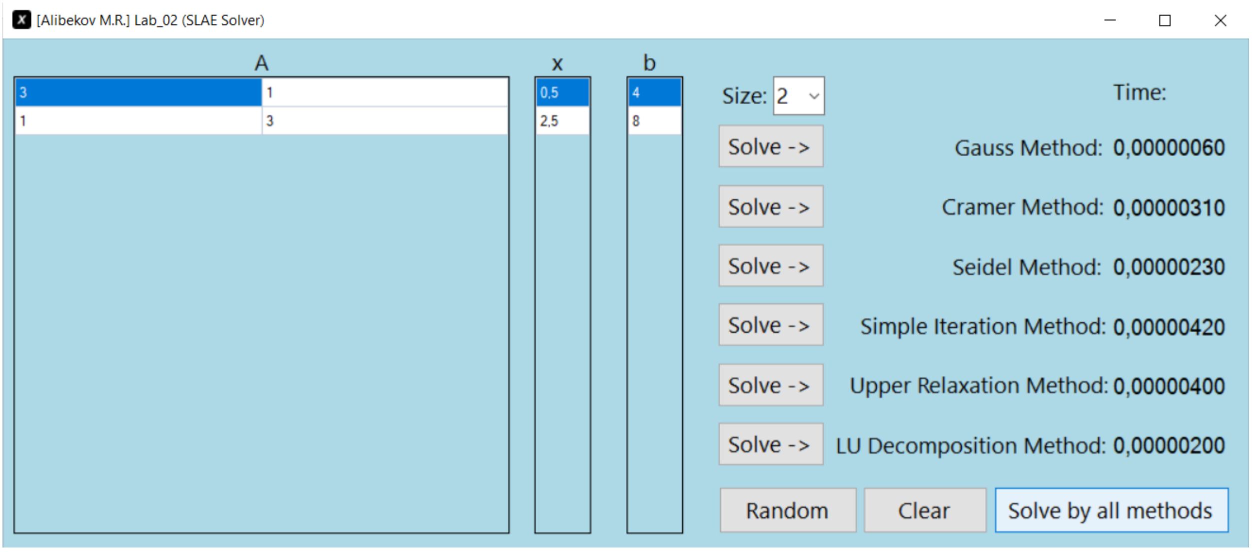

Рисунок 2. Решение всеми методами

Помимо прочего можно также менять число переменных (“Size”), использовать рандомные (случайные) значения коэффициентов (“Random”), а также полностью очистить все значения (“Clear”).

5. Заключение

В результате лабораторной работы разработан программный комплекс на языке C#, позволяющий решать системы линейных алгебраических уравнений различными методами и сравнивать их производительность. При этом сами уравнения можно как задать вручную, так и сгенерировать автоматически.

В результате работы программы на экране отображаются вектор-решение x и время выполнения подсчётов каждым из представленных методов (Гаусса, Крамера, Зейделя, простых итераций, LU-разложения, верхней релаксации).

В ходе экспериментов сравнили разные численные методы по времени выполнения. И метод Зейделя в некоторых случаях приводит к более быстрой сходимости, чем метод простых итераций. Однако ввиду относительно малых размеров матриц (от 2 до 16 переменных), данные результаты могут сильно отличаться от тех, что будут для СЛАУ с большим количеством переменных.

Таким образом, цели, поставленные в лабораторной работе, были успешно достигнуты.

6. Литература

1. Г.С. Шевцов, О.Г. Крюкова, Б.И. Мызникова. Численные методы линейной алгебры: Лань, 2011 – 496с.
2. Герберт Шилдт. C# 4.0: Полное руководство: ООО “И.Д. Вильямс”, 2011 – 1056 с.
3. Курс лекций по Вычислительным Методам 5-ого семестра в 2020-2021 учебных Годах направления ФИИТ в Институте информационных технологий, математики и механики в ННГУ им. Лобачевского.
4. Амосов А.А., Дубинский Ю.А., Копченова Н.В. Вычислительные методы для инженеров: Учеб. пособие. — М.: Высш. шк., 1994. — 544 с.
5. Волков Е.А. Численные методы. [Электронный ресурс]: учеб. — Электрон. дан. — СПБ.: Лань, 2008 — 256 с. — Режим доступа: http://e.lanbook.com/book/54 — Загл. с экрана.
6. Д.П. Кострамов, А.П. Фаворский Вводные лекции по численным методам: Учеб. Пособие. – М.: Логос, 2004. – 184 с.: ил.