Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 2

по курсу «Численные методы линейной алгебры»

«Реализация метода Гаусса с перестановками»

Студент группы ИУ9-71Б Окутин Д.А.

Преподаватель Посевин Д. П.

Москва 2024

1 Цель работы

Реализовать три варианта метода Гаусса с перестановками и научиться оценивать погрешность решения системы линейных уравнений для матриц произвольной размерности.

2 Задание

1. Реализовать метод Гаусса с перестановками по столбцам, по строкам, по столбцам и строкам одновременно для действительных квадратных матриц произвольной размерности n.

2. Для проверки работоспособности алгоритмов необходимо использовать алгоритм тестирования задачи написанный в лабораторной работе №2 «Реализация метода Гаусса», который заключался в том, что мы заведомо определяем значения координат вектора x, данный вектор заведомо является решением уравнения $A* = b$, вычисляем b путем прямого перемножения матрицы A на вектор x и далее производим поиск решения уравнения $A* = b$ тем или иным методом Гаусса, получая x_{chisl}, после чего производим сравнение полученного x_{chisl} с заданным x, а также решением x_{bibl} , полученным с использованием сторонней библиотеки выбранной студеном. При этом сравнение производится по Евклидовой норме разности вектора $x - x_{chisl}$ и $x - x_{bibl}$.

3. На защите лабораторной работы студент должен показать умение оценивать погрешность вычислений в зависимости от выполнения условия диагонального преобладания матрицы, умение сравнивать погрешности вычислений полученных методом Гаусса с перестановками по столбцам, по строкам, по столбцам и строкам одновременно. Понимать связь теории с практикой.

4. Результат работы должен быть представлен в виде графиков зависимости абсолютной погрешности вычислений классическим методом Гаусса, методом Гаусса с перестановками по строкам, методом Гаусса с перестановками по столбцам, методом Гаусса с перестановками по столбцам и строкам, библиотечным методом от степени диагонального преобладания. Все графики должны быть построены на одной координатной плоскости. Напомним, что погрешность вычисления вектора x системы линейных алгебраических уравнений A·x=b тем или

иным способом рассчитывается по Евклидовой норме разности точного решения и решения полученного соответствующим методом. Степень диагонального преобладания вычисляется, как максимальная разность по i между модулем диагонального элемента и суммы модулей вне диагональных элементов. Очевидно, что если значение степени диагонального преобладания положительна, то условие диагонального преобладания выполняется, в противном случае — не выполняется. Поэтому график должен быть построен как для отрицательных значений степени диагонального преобладания, так и для положительных.

3 Реализация

Исходный код программы представлен в листингах 1– 6.

Листинг 1 — Вспомогательные функции

```
using Random
using LinearAlgebra

function euclidean_norm(vec::Vector)
    return sqrt(sum(vec.^2))
end

function generate_matrix(l::Int, r::Int, n::Int)
    return rand(n, n) .* (r - l) .+ l
end

function increase_diag_elems(a::Matrix, diagcoef::Float64)
    n = size(a, 1)
    for i in 1:n
        for j in 1:n
            if j != i
                a[i,i] += diagcoef * abs(a[i, j])
            end
        end
    end

    return a
end

function diag_dominance(matrix::Matrix)
    a=copy(matrix)
    return maximum(abs(a[i, i]) - sum(abs(a[i, j]) for j in 1:size(a, 2)
     if j != i) for i in 1:size(a, 1))
end

function test(method::Function, A::Matrix, x::Vector)
    x_calc = method(A, A*x)
    return euclidean_norm(x - x_calc)
end
```

Листинг 2 — Метод Гаусса

```

function gaussian_elimination(matrix::Matrix, vector::Vector)
    n = length(vector)

    A = copy(matrix)
    b = copy(vector)

    for k in 1:n-1
        for i in k+1:n
            factor = A[i, k] / A[k, k]
            A[i, k+1:end] -= factor * A[k, k+1:end]
            b[i] -= factor * b[k]
        end
    end

    x = zeros(n)
    x[n] = b[n] / A[n, n]
    for i in n-1:-1:1
        x[i] = (b[i] - dot(A[i, i+1:end], x[i+1:end])) / A[i, i]
    end

    return x
end
```

Листинг 3 — Метод Гаусса с перестановкой по столбцам

```

function gaussian_elimination_pivot(matrix::Matrix, vector::Vector)
    n = length(vector)

    A = copy(matrix)
    b = copy(vector)

    for k in 1:n-1
\begin{figure}
            \centering
            \includegraphics[width=0.5\linewidth]{image.png}
            \caption{Enter Caption}
            \label{fig:enter-label}
        \end{figure}
                maxindex = argmax(abs.(A[k:n, k])) + k - 1
        if maxindex != k
            A[[k, maxindex], k:end] = A[[maxindex, k], k:end]
            b[k], b[maxindex] = b[maxindex], b[k]
        end

        for i in k+1:n
            factor = A[i, k] / A[k, k]
            A[i, k+1:end] -= factor * A[k, k+1:end]
            b[i] -= factor * b[k]
        end
    end

    x = zeros(n)
    x[n] = b[n] / A[n, n]
    for i in n-1:-1:1
        x[i] = (b[i] - dot(A[i, i+1:end], x[i+1:end])) / A[i, i]
    end

    return x
end
```

Листинг 4 — Метод Гаусса с перестановкой по строкам

```

function gaussian_elimination_pivot_row(matrix::Matrix, vector::Vector)
    n = length(vector)

    A = copy(matrix)
    b = copy(vector)

    for k in 1:n-1
        maxindex = argmax(abs.(A[k, k:end])) + k - 1
        if maxindex != k
            A[[k, maxindex], :] = A[[maxindex, k], :]
            b[k], b[maxindex] = b[maxindex], b[k]
        end

        for i in k+1:n
            factor = A[i, k] / A[k, k]
            A[i, k+1:end] -= factor * A[k, k+1:end]
            b[i] -= factor * b[k]
        end
    end

    x = zeros(n)
    x[n] = b[n] / A[n, n]
    for i in n-1:-1:1
        x[i] = (b[i] - dot(A[i, i+1:end], x[i+1:end])) / A[i, i]
    end

    return x
end
```

Листинг 5 — Метод Гаусса с перестановкой по строкам и столбцам

```

function gauss_with_rows_and_columns_permutation(matrix::Matrix, vector::Vector)
    n = length(vector)

    A = copy(matrix)
    b = copy(vector)
    x_i = collect(1:n)
    x = zeros(Float64, n)

    for i in 1:n-1
        max_index_col = argmax(abs.(A[i:end, i])) + i - 1
        max_index_row = argmax(abs.(A[i, i:end])) + i - 1
        if abs(A[max_index_col, i]) > abs(A[i, max_index_row])
            A[[i, max_index_col], i:end] = A[[max_index_col, i], i:end]
            b[i], b[max_index_col] = b[max_index_col], b[i]
        else
            A[[i, max_index_row], :] = A[[max_index_row, i], :]
            b[i], b[max_index_row] = b[max_index_row], b[i]
        end

        for j in i+1:n
            f = A[j, i] / A[i, i]
            A[j, :] -= f * A[i, :]
            b[j] -= f * b[i]
        end
    end

    for i in n:-1:1
        x[i] = b[i] / A[i, i]
        for j in i-1:-1:1
            b[j] -= A[j, i] * x[i]
        end
    end

    x_copy = copy(x)
    for i in 1:n
        x[x_i[i]] = x_copy[i]
    end

    return x
end
```

Листинг 6 — Запус программы и построение графиков зависимости

```
using PyPlot
coefs = [i*0.2 for i in 1:3:21]

for d in [20, 100, 300]
    diag =Float64[]
    y_gauss = Float64[]
    y_gauss_row = Float64[]
    y_gauss_col = Float64[]
    y_gauss_row_col = Float64[]

    P = generate_matrix(-10, 10, d)
    for c in coefs
        A = copy(P)
        increase_diag_elems(P, c)
        x = ones(d)

        push!(diag, diag_dominance(A))
        push!(y_gauss, test(gaussian_elimination, A, x))
        push!(y_gauss_row, test(gaussian_elimination_pivot_row, A, x))
        push!(y_gauss_col, test(gaussian_elimination_pivot, A, x))
        push!(y_gauss_row_col, test(
   gauss_with_rows_and_columns_permutation, A, x))
    end

    PyPlot.figure(figsize=(6, 5))
    PyPlot.plot(diag, y_gauss, label="Classic Gauss")
    PyPlot.plot(diag, y_gauss_row, label="Gauss Rows")
    PyPlot.plot(diag, y_gauss_col, label="Gauss Columns")
    PyPlot.plot(diag, y_gauss_row_col, label="Gauss Rows & Columns")
    PyPlot.title("matrix $(d)x$(d)")
    PyPlot.xlabel("Diagonal Dominance")
    PyPlot.ylabel("Error")
    PyPlot.legend()
    PyPlot.grid()
    PyPlot.show()
end
```

4 Результаты

Результат запуска методов представлены на рисунках 1 - 3.

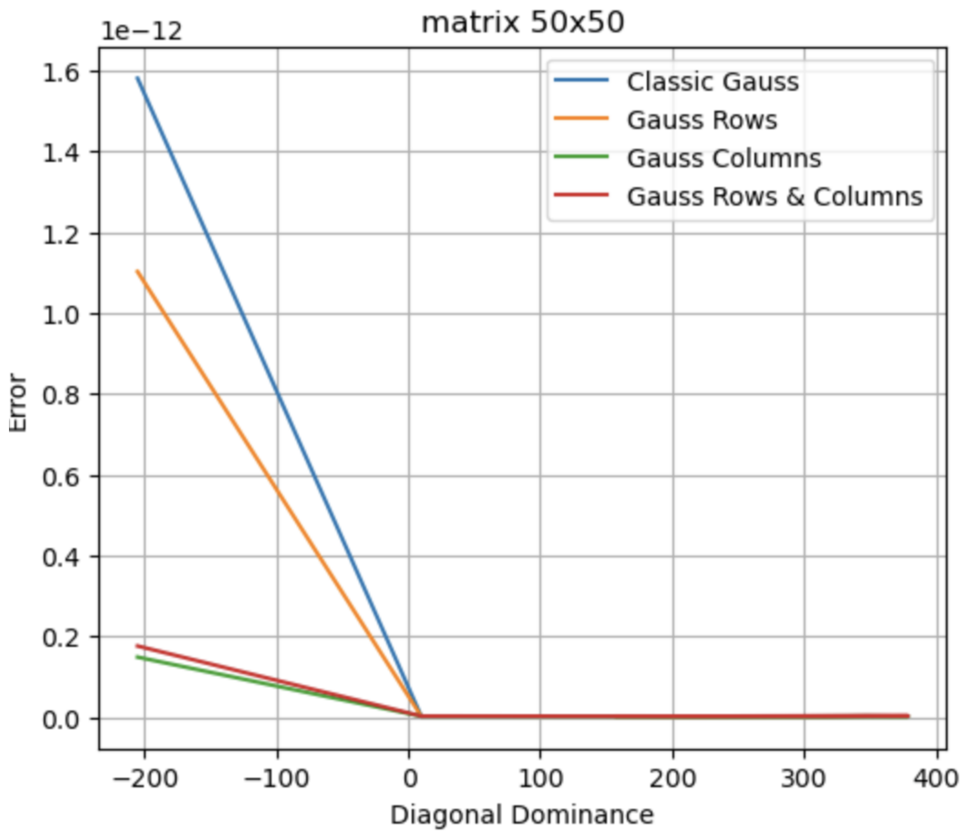

Рис. 1 — График зависимости для матрицы размерности 50x50

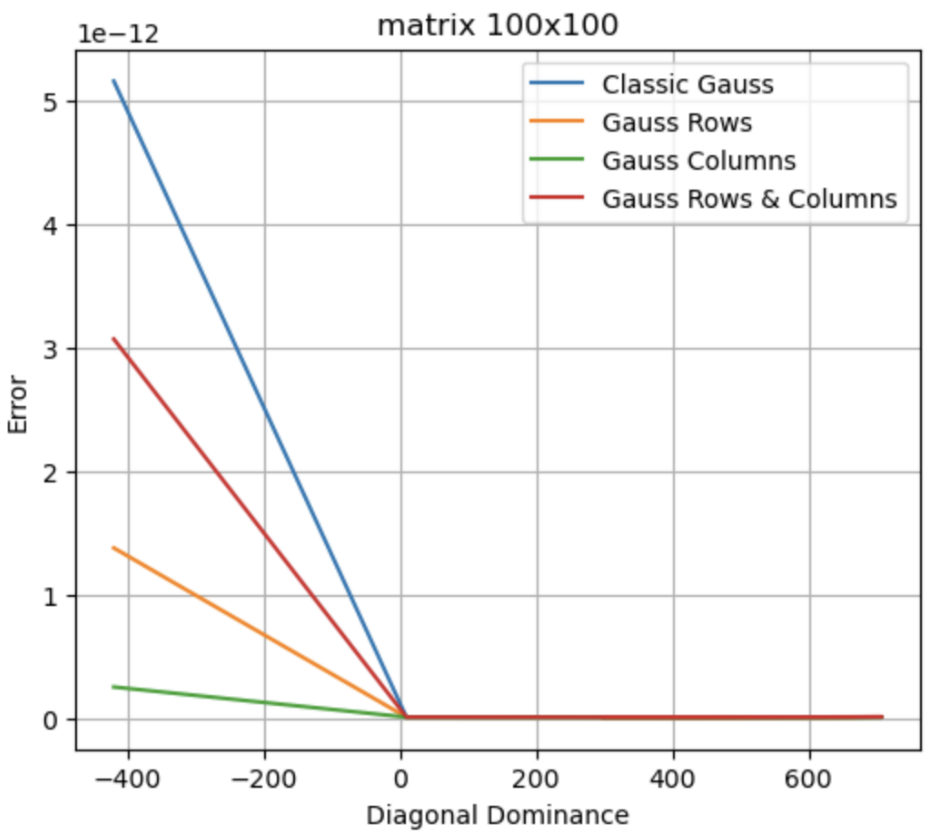

Рис. 2 — График зависимости для матрицы размерности 100x100

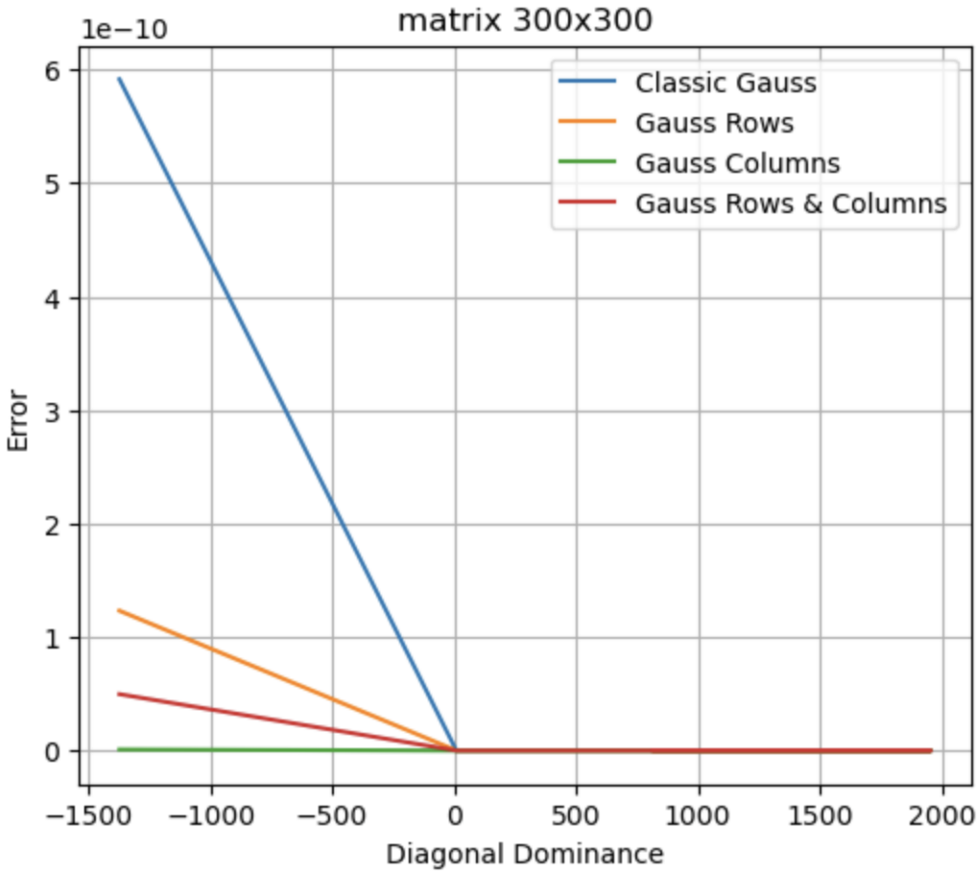

Рис. 3 — График зависимости для матрицы размерности 300x300

5 Выводы

В результте выполнения данной лабораторной работы были реализованы стадартный метод Гаусса, метод Гаусса с перестановками по столбцам, по строкам, по столбцам и строкам одновременно для действительных квадратных матриц произвольной размерности n на языке программирования Python.

Результат работы был представлен в виде графиков зависимости абсолютной погрешности вычислений классическим методом Гаусса, методом Гаусса с перестановками по строкам, методом Гаусса с перестановками по столбцам, методом Гаусса с перестановками по столбцам и строкам от степени диагонального преобладания.

Из графиков видно, что погрешность с размерностью матрицы растёт, что довольно ожидаемый результат, также видно, что классический метод Гаусса обладает более высокой погрешностю по отношению с другими вариациями, которые как раз таки и были направлены на уменьшение погрешности. При этом видно, что при положительной степени диагонального преобладания погрешности методов сходится к 0.