



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»

Отчет по лабораторной работе № 2
«Реализация метода Гаусса с перестановками»
по курсу
«Численные методы линейной алгебры»

Выполнил:

Студент группы ИУ9-72Б

Караник А.А.

Проверил:

Посевин Д.П.

2024 г.

Цель работы

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

Постановка задачи

- Реализовать метод Гаусса с перестановками по столбцам, по строкам, по столбцам и строкам одновременно для действительных квадратных матриц произвольной размерности n .
- Для проверки работоспособности алгоритмов необходимо использовать алгоритм тестирования, который заключался в том, что мы заведомо определяем значения координат вектора x , данный вектор является решением уравнения $A \cdot x = b$; вычисляем b путем прямого перемножения матрицы A на вектор x и далее производим поиск решения уравнения $A \cdot x = b$ тем или иным методом Гаусса, получая $x_{\text{числ}}$, после чего производим сравнение полученного $x_{\text{числ}}$ с заданным x , а также решением $x_{\text{библ}}$, полученным с использованием сторонней библиотеки выбранной студеном. При этом сравнение производится по Евклидовой норме разности вектора $x_{\text{числ}}$ и $x_{\text{библ}}$.
- 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.
- 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 $A \cdot x = b$ 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.

Реализация

Исходный код программы:

```
using Random
using LinearAlgebra
using Plots
using Distributions

function gauss_classic(A, f)
    A_aug = hcat(A, f)
    n = size(A, 1)

    for i in 1:n
        pivot = A_aug[i, i]
        A_aug[i, :] /= pivot

        for j in i+1:n
            factor = A_aug[j, i]
            A_aug[j, :] -= factor * A_aug[i, :]
        end
    end

    x = zeros(n)
    for i in n:-1:1
        x[i] = A_aug[i, end] - A_aug[i, i+1:end-1]' * x[i+1:end]
    end

    return x
end

function gauss_by_rows(A, f)
    n = size(A, 1)
    A_aug = [A f]
    row_perm = collect(1:n)

    for k in 1:n-1
        max_col = argmax(abs.(A_aug[k, k:n])) + k - 1

        if max_col != k
            A_aug[:, [k, max_col]] = A_aug[:, [max_col, k]]
            row_perm[[k, max_col]] = row_perm[[max_col, k]]
        end

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

    x = zeros(n)
    for j in n:-1:1
        x[j] = (A_aug[j, end] - A_aug[j, j+1:end-1]' * x[j+1:end]) / A_aug[j, j]
    end
end
```

```

        return x[row_perm]
    end

function gauss_by_columns(A, f)
    n = size(A, 1)
    A_aug = [A f]

    for k in 1:n-1
        max_row = argmax(abs.(A_aug[k:n, k])) + k - 1

        if max_row != k
            A_aug[[k, max_row], :] = A_aug[[max_row, k], :]
        end

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

    x = zeros(n)
    for j in n:-1:1
        x[j] = (A_aug[j, end] - A_aug[j, j+1:end-1]' * x[j+1:end]) / A_aug[j, j]
    end

    return x
end

function gauss_combined(A, f)
    n = size(A, 1)
    A_aug = [A f]
    row_perm = collect(1:n)
    col_perm = collect(1:n)

    for k in 1:n-1

        max_val, max_index = findmax(abs.(A_aug[k:n, k:n]))
        max_row, max_col = max_index[1] + k - 1, max_index[2] + k - 1

        if max_row != k
            A_aug[[k, max_row], :] = A_aug[[max_row, k], :]
            row_perm[[k, max_row]] = row_perm[[max_row, k]]
        end

        if max_col != k
            A_aug[:, [k, max_col]] = A_aug[:, [max_col, k]]
            col_perm[[k, max_col]] = col_perm[[max_col, k]]
        end

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

    x = zeros(n)
    for j in n:-1:1
        x[j] = (A_aug[j, end] - A_aug[j, j+1:end-1]' * x[j+1:end]) / A_aug[j, j]
    end

    result = zeros(n)
    for i in 1:n
        result[col_perm[i]] = x[i]
    end

    return result
end

function library(A, b)
    return A \ b
end

function euclidean_norm(vec)
    return sqrt(sum(element^2 for element in vec))
end

function mulvec(A, vector)
    result = Float64[]
    for i in 1:size(A, 1)
        element = 0.0
        for j in 1:length(vector)
            element += A[i, j] * vector[j]
        end
        push!(result, element)
    end
    return result
end

function generate_matrix(l, r, n)
    return rand(Uniform(l,r), n, n)
end

```

```

end

function increase_diag_elements(A, diag)
    n = size(A, 1)
    for i in 1:n
        A[i, i] += diag * sum(abs(A[i, j]) for j in 1:n if j != i)
    end
    return A
end

function diag_dominance(A)
    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 calculate(method, A::Array{Float64}, x::Array{Float64})
    b = mulvec(A, x)
    x_calc = method(A, b)
    return euclidean_norm(x - x_calc)
end

n = 100
diag = Float64[]
y_gauss = Float64[]
y_gauss_row = Float64[]
y_gauss_columns = Float64[]
y_gauss_combined = Float64[]
y_library = Float64[]
coefs = [i * 0.2 for i in 1:2:10]

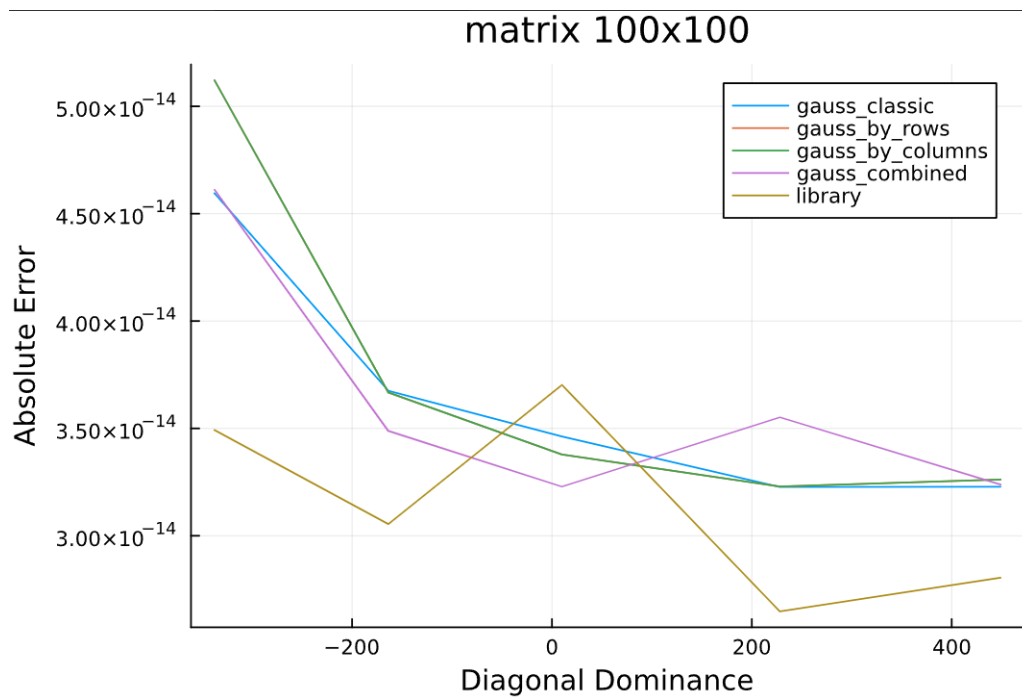
for c in coefs
    A = generate_matrix(-10.0, 10.0, n)
    A = increase_diag_elements(A, c)
    x = rand(Uniform(-10.0, 10.0), n)
    push!(diag, diag_dominance(A))
    push!(y_gauss, calculate(gauss_classic, A, x))
    push!(y_gauss_row, calculate(gauss_by_rows, A, x))
    push!(y_gauss_columns, calculate(gauss_by_columns, A, x))
    push!(y_gauss_combined, calculate(gauss_combined, A, x))
    push!(y_library, calculate(library, A, x))
end

p = plot(diag,
    [y_gauss, y_gauss_row, y_gauss_columns, y_gauss_combined, y_library],
    label=["gauss_classic" "gauss_by_rows" "gauss_by_columns" "gauss_combined" "library"],
    title=("$n \times n$ matrix"),
    xlabel="Diagonal Dominance",
    ylabel="Absolute Error")

display(p)

```

Результат работы:



Вывод

В ходе выполнения лабораторной работы были реализованы различные варианты метода Гаусса для решения систем линейных уравнений. Кроме того, был разработан и применен алгоритм тестирования для проверки корректности реализации. Тестирование состояло в сравнении численного решения системы с заведомо известным точным решением. Также результат работы был представлен в виде графиков зависимости абсолютной погрешности от степени диагонального преобладания.