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 № 3

по курсу «Численные методы линейной алгебры»

«Оценки источников погрешностей решения СЛАУ методом Гаусса»

Студент группы ИУ9-72Б Виленский С. Д.

Преподаватель Посевин Д. П.

Москва 2024

1 Задание

Реализовать формулы оценок ошибок округления и исходных данных.

2 Результаты

Исходный код программы представлен в листингах 1- 3.

Листинг 1 — Реализация алгоритмов оценки источников погрешностей

```
using LinearAlgebra
using Random

## Generating input data

function generate_matrix(n::Int64)::Matrix{Float64}
    return rand(-100:0.01:100, n, n)
end
function generate_matrix_delta(n::Int64)::Matrix{Float64}
    return rand(-10:0.01:10, n, n)
end

function generate_vector(n::Int64)::Vector{Float64}
    return rand(-100:0.01:100, n)
end
function generate_vector_delta(n::Int64)::Vector{Float64}
    return rand(-10:0.01:10, n)
end

## Norms

function euclidean_norm(A)::Float64
    return sqrt(sum(abs2, A))
end

function uniform_norm(x::Vector{Float64})::Float64
    return maximum(abs.(x))
end

function uniform_norm(x::Matrix{Float64})::Float64
    return maximum(sum(abs, x, dims=2))
end

## Condition number of a matrix

function get_condition_number(A::Matrix{Float64}, _norm::Function)::
    Float64
    return _norm(inv(A)) * _norm(A)
end

## Growth rate of matrix elements

function gauss_growth_factor(A::Matrix{Float64})::Float64
    n = size(A, 1)
```

Листинг 2 — Реализация алгоритмов оценки источников погрешностей

```
    max_initial = maximum(abs.(A))
    max_during = max_initial

    A_work = copy(A)

    for k in 1:n-1
        for i in k+1:n
            if A_work[k, k] != 0
                factor = A_work[i, k] / A_work[k, k]
                for j in k:n
                    A_work[i, j] -= factor * A_work[k, j]
                end
            end
        end
        max_during = max(max_during, maximum(abs.(A_work)))
    end

    return max_during / max_initial
end

## Error estimates

function error_rounding(
    A::Matrix{Float64},
    p::Int64,
    t::Int64,
    _norm::Function
)::Float64
    nu_A = get_condition_number(A, _norm)
    n = size(A, 1)
    g_A = gauss_growth_factor(A)
    return nu_A * n * g_A / p ^ t
end

function error_input_data(
    A::Matrix{Float64}, delta_A::Matrix{Float64},
    f::Vector{Float64}, delta_f::Vector{Float64},
    _norm::Function
)::Float64
    nu_A = get_condition_number(A, _norm)
    error_A = _norm(delta_A) / _norm(A)
    error_f = _norm(delta_f) / _norm(f)
    return nu_A * (error_A + error_f)
end

## Solution error

function error_result(
    A::Matrix{Float64}, delta_A::Matrix{Float64},
    f::Vector{Float64}, delta_f::Vector{Float64},
    _norm::Function
)::Float64
    x = A \ f
    delta_x = (A + delta_A) \ (f + delta_f) - x
    return _norm(delta_x) / _norm(x)
end
```

Листинг 3 — Реализация алгоритмов оценки источников погрешностей

```
## Testing

A = [
    100. 99.;
    99. 98.
]
delta_A = [
    0. 0.;
    0. 0.
]
f = [
    199.,
    197.
]
delta_f = [
    -.01,
    .01
]

print("Relative error of input data: ", error_input_data(A, delta_A, f,
    delta_f, uniform_norm))
print("\nRelative error of result: ", error_result(A, delta_A, f,
    delta_f, uniform_norm))

A = generate_matrix(5)
delta_A = generate_matrix_delta(5)
f = generate_vector(5)
delta_f = generate_vector_delta(5)

_norm = euclidean_norm

print("Relative error of rounding: ", error_rounding(A, 2, 11, _norm))
print("\nRelative error of input data: ", error_input_data(A, delta_A, f
    , delta_f, _norm))
print("\nRelative error of result: ", error_result(A, delta_A, f,
    delta_f, _norm))
```

Результат работы программы представлен в листинге 4.

Листинг 4 — Результат работы программы

```
Relative error of input data: 1.9899999999989821
Relative error of result: 1.9899999999940312

Relative error of rounding: 0.11516827877965007
Relative error of input data: 1.32573555646218
Relative error of result: 0.1490879688109449
```

3 Выводы

Проанализировав графики зависимостей ошибок разных методов от диагонального доминирования матриц можно сделать вывод о том, что самым оптимальным является модификация метода Гаусса перестановкой по строкам и столбцам, модификации метода Гаусса перестановкой по строкам или по столбцам относительно имеет близкую погрешность и классический метод Гаусса среди прочих имеет наибольшую относительную ошибку.