



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»
«Вычисление ошибки округления и ошибки исходных данных для
СЛАУ»

Студент группы ИУ9-71Б Окутин Д. А.

Преподаватель Посевин Д. П.

Москва 2024

1 Цель

Цель данной лабораторной работы: оценить погрешность решения системы уравнений. Посчитать ошибку округления при решении методом Гаусса, посчитать ошибку исходных данных при создании искусственного возмущения системы.

2 Задание

1. Решить задачу на поиск ошибки исходных данных на листочке.
2. Проверить полученное решение, реализовав вычисление ошибки программно.
3. Вычислить ошибку округления в результате решения СЛАУ методом Гаусса.

3 Реализация

Исходный код представлен в листинге 1 - 4.

Листинг 1: Вспомогательные функции

```
1
2  using Random
3  using LinearAlgebra
4
5  function euclidean_norm(vec::Vector)
6      return sqrt(sum(vec.^2))
7  end
8
9  function generate_matrix(l::Float64, r::Float64, n::Int)
10     return rand(n, n) .* (r - l) .+ l
11 end
12
13 function generate_vector(l::Float64, r::Float64, n::Int)
14     return rand(n) .* (r - l) .+ l
15 end
16
17 function vector_norm(vec::Vector)
18     max_elem = argmax(abs.(vec))
19     return abs(vec[max_elem])
```

```

20 end
21
22 function matrix_norm(matrix::Matrix)
23     max_norm = -100000000
24     for i in 1:size(matrix,1)
25         max_norm = max(sum(abs.(matrix[i,1:end])),max_norm)
26     end
27
28
29     return max_norm
30 end
31
32 t = 15.0 #
33 p = 2.0 #
34
35 function findCondNum(A::Matrix)
36     mu_a = matrix_norm(A)*matrix_norm(inv(A))
37
38     if mu_a < 100
39         println("                :
40                 = $mu_a")
41     else
42         println("                :
43                 = $mu_a")
44     end
45     return mu_a
46 end

```

Листинг 2: Метод Гаусса

```

1
2 function gaussian_elimination(matrix::Matrix, vector::Vector)
3     n = length(vector)
4
5     A = copy(matrix)
6     b = copy(vector)
7
8     max_first_step = maximum(abs.(A))
9     max_growth_coef = -100000000
10
11     #
12     for k in 1:n-1
13         for i in k+1:n
14             factor = A[i, k] / A[k, k]
15             A[i, 1:end] -= factor * A[k, 1:end]
16             b[i] -= factor * b[k]

```

```

17         end
18
19         max_growth_coef = max(maximum(abs.(A))/max_first_step ,
20         max_growth_coef)
21     end
22
23     #
24     x = zeros(n)
25     x[n] = b[n] / A[n, n]
26     for i in n-1:-1:1
27         x[i] = (b[i] - dot(A[i, i+1:end], x[i+1:end])) / A[i, i]
28     end
29
30     return x, max_growth_coef
31 end

```

Листинг 3: Проверка СЛАУ из задания с листочка

```

1
2  A = [503.0  3.0; 251.0  651.0]
3  f = [556.0; 862.0]
4  res = A \ f
5  println("                : $res")
6
7  A_shifted = [503.0  3.0; 251.0  651.0]
8  f_shifted = [557.3; 863.3]
9
10 delta_A = A_shifted - A
11 delta_f = f_shifted - f
12
13
14 res_shifted = A_shifted \ f_shifted
15 println("\ n                :
16     $res_shifted\n")
17
18 mu_a = findCondNum(A)
19
20 println(vector_norm(delta_f)/vector_norm(f))
21
22 final = mu_a * (vector_norm(delta_f)/vector_norm(f)+ matrix_norm(
23     delta_A)/matrix_norm(A))
24 println("\ n                -
25                                : $final")
26
27 norm_diff = abs(vector_norm(res_shifted) - vector_norm(res))

```

```

25     println("\ n                                     :
        $norm_diff")
26
27     res_gauss, max_growth_coef = gaussian_elimination(A, f)
28     println("\ ng(A): $max_growth_coef")
29
30     final_1 = mu_a * max_growth_coef * 2 * (p ^ (-t))
31     println("\ n                                     -
                                                : $final_1")

```

Листинг 4: Проверка случайной СЛАУ

```

1
2     n=5
3     A = generate_matrix(-10.0,10.0,n)
4     f = generate_vector(-10.0,10.0,n)
5     res = A \ f
6     println("                                     : $res")
7
8     delta_A = generate_matrix(-0.01,0.01,n)
9     delta_f = generate_vector(-0.01,0.01,n)
10
11     A_shifted = A + delta_A
12     f_shifted = f + delta_f
13     res_shifted = A_shifted \ f_shifted
14     println("\ n                                     :
        $res_shifted\n")
15
16     mu_a = findCondNum(A)
17     final = mu_a * (vector_norm(delta_f)/vector_norm(f)+ matrix_norm(
        delta_A)/matrix_norm(A))
18     println("\ n                                     -
                                                : $final")
19
20     norm_diff = abs(vector_norm(res_shifted) - vector_norm(res))
21     println("\ n                                     :
        $norm_diff")
22
23     res_gauss, max_growth_coef = gaussian_elimination(A, f)
24     println("\ ng(A): $max_growth_coef")
25
26     final_1 = mu_a * max_growth_coef * n * (p ^ (-t))
27     println("\ n                                     -
                                                : $final_1")

```

4 Результаты

Результат представлен на рисунке 1.

```
Ответ для исходной СЛАУ: [1.0999999999999999, 0.9000000000000001]
Ответ для возмущенной СЛАУ: [1.102578512396694, 0.9010027548209367]
Система хорошо обусловлена: коэффициент = 2.081750841750842
0.0015081206496519194
Погрешность из-за ошибки входных данных: 0.0031395314318747097
Разность норм результатов: 0.0025785123966941548
g(A): 0.9977004333446325
Погрешность из-за ошибки вычислений: 0.0001267678049884258
```

Рис. 1 — Полученные результаты

5 Выводы

В результате данной лабораторной работе были изучены 2 оценки: 1) оценка ошибки округления 2) оценка ошибки исходных данных. Была релизована программа на языке julia, которая помогает вычислять обе оценки по СЛАУ для дальнейшего анализа.