|  |  |
| --- | --- |
|  | **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 Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

**Лабораторная работа 5.2**

по курсу «Численные методы линейной алгебры».

«Вычисление собственных значений и собственных векторов матрицы методом Крылова.»

Выполнил:

студент группы ИУ9-71Б

Митрошкин Алексей

Проверил:

Посевин Д.П.

Москва, 2024

1. **Цель**

Реализовать метод вычисления собственных значений и собственных векторов симметричной матрицы методом А.Н. Крылова.

1. **Задание**

* Реализовать метод поиска собственных значений действительной симметричной матрицы A размером.
* Реализовать метод поиска собственных значений действительной симметричной матрицы A размером.

1. **Реализация**

Исходный код программы приведён в Листинге 1.

Листинг 1

|  |
| --- |
| using Polynomials  using LinearAlgebra  function krylov\_algo(matrix)  n = size(matrix, 1)  D = copy(matrix)  y = []  push!(y, ones(n)) # первый вектор берём произвольно    for i in 1:n  push!(y, D \* y[i]) # yk = D^k y0  end    # cобираем yn в матрицу в обратном порядке, чтобы составить СЛАУ  A = hcat(reverse(y[1:n])...)    yn = y[n + 1] # Последний вектор y\_n  p = [1.0]  append!(p, -(A \ yn)) # Решаем систему A \* p = yn      return y, p  end  function find\_eigen\_values(eqCoeffs)  reversed\_eqCoeffs = reverse(eqCoeffs)    p = Polynomial(reversed\_eqCoeffs)    roots\_p = roots(p)    return roots\_p  end  function find\_eigen\_vectors(y, lambdas, p)  n = length(p) - 1  xs = []  for i in 1:length(lambdas)  x = copy(y[n]) # Начинаем с последнего вектора y  q\_i = [1.0] # q\_0i = 1  for j in 1:(n-1)  push!(q\_i, lambdas[i] \* q\_i[j] + p[j+1])  x += q\_i[j+1] \* y[n - j]  end    push!(xs, x)  end  return xs  end  A = [2.2 1.0 0.5 2.0;  1.0 1.3 2.0 1.0;  0.5 2.0 0.5 1.6;  2.0 1.0 1.6 2.0]  y, p = krylov\_algo(A)  eigen\_result = eigen(A)  println("Коэффициенты характеристического многочлена p:\n", p)  println("Y:")  for i in 1:length(y)  println(" y\_$i: ", y[i])  end  eigens = find\_eigen\_values(p)  println("\nСЗ: ", eigens)  println("СЗ (библиотечно): ", eigen\_result.values)  e\_vects = find\_eigen\_vectors(y, eigens, p)  println("СВ:")  for i in 1:length(e\_vects)  println(" $i: ", e\_vects[i])  end |

В листиге 2 приведён результат работы программы.

Листинг 2

|  |
| --- |
| Коэффициенты характеристического многочлена p:  [1.0, -5.99999999999997, -0.20000000000011334, 12.734999999999701, -2.761600000000019]  Y:  y\_1: [1.0, 1.0, 1.0, 1.0]  y\_2: [5.7, 5.3, 4.6, 6.6]  y\_3: [33.34, 28.389999999999997, 26.310000000000002, 37.260000000000005]  y\_4: [189.413, 160.127, 146.221, 211.68600000000004]  y\_5: [1073.3181, 901.7061000000001, 826.7686000000001, 1196.2786]  СЗ: [-1.4200865939506169, 0.2226359271326228, 1.5454183350533741, 5.652032331764599]  СЗ (библиотечно): [-1.4200865939506189, 0.22263592713261104, 1.5454183350534132, 5.6520323317645795]  СВ:  1: [-0.995513846778179, 2.313048594068521, -3.3951871391295594, 1.4941956531551757]  2: [0.7281712311126203, 0.6346229145042859, -0.21408564920973028, -0.9837198605984483]  3: [2.304311380657996, -2.097832505855852, -1.779368116766154, 0.7395782920663088]  4: [165.9500312350099, 139.25316099728505, 127.58764090510725, 184.9089459153796] |