ШАМАНАЕВ П. А., КАТИН Д. А., ДЕСЯЕВ Е. В.

РЕАЛИЗАЦИЯ АЛГОРИТМА НАХОЖДЕНИЯ РЕЗОЛЬВЕНТЫ МАТРИЦЫ С ИСПОЛЬЗОВАНИЕМ ПРИСОЕДИНЕННОЙ МАТРИЦЫ И ХАРАКТЕРИСТИЧЕСКОГО МНОГОЧЛЕНА

Аннотация. В настоящей работе излагается реализация алгоритма вычисления резольвенты матрицы с использованием присоединенной матрицы и характеристического многочлена матрицы на языке Python. На графиках приведены зависимости скорости работы алгоритма при различных размерностях матрицы.

Ключевые слова: резольвента матрицы, присоединенная матрица, характеристический многочлен, наибольший общий делитель, Python.

SHAMANAEV P. A., KATIN D. A., DESYAEV E. V.

IMPLEMENTATION OF AN ALGORITHM FOR FINDING THE RESOLVENT OF A MATRIX USING THE ADJECT MATRIX AND CHARACTERISTIC POLYNOMIAL

Abstract. The article describes the implementation of an algorithm for calculating the resolvent of a matrix using the adjoint matrix and the characteristic polynomial of the matrix in Python. The graphs show the speed of the algorithm for various matrix dimensions.

Keywords: matrix resolvent, adjoint matrix, characteristic polynomial, greatest common divisor, Python.

Введение. При решении многих задач требуется вычислять резольвенту матрицы [1], не зная ее собственных значений. В частности, такая задача возникает при решении систем линейных алгебраических уравнений с малым параметром методом Ляпунова-Шмидта [2; 3].

В работе [1] приведен подход вычисления резольвенты матрицы с использованием присоединенной матрицы и характеристического многочлена матрицы. Для вычисления последних Д. К. Фаддеевым предложен метод одновременного вычисления коэффициентов характеристического многочлена и присоединенной матрицы [1].

Изложим алгоритм вычисления резольвенты матрицы на основе подхода, изложенного в [1] и метода Д. К. Фаддеева одновременного вычисления коэффициентов характеристического многочлена и присоединенной матрицы.

Алгоритм вычисления резольвенты матрицы. Для вычисления резольвенты постоянной $(m \times m)$ −матрицы A

$$R\,(\lambda) = [\lambda E - A]^{-1}$$

воспользуемся формулой [1, с. 112]

$$R\,(\lambda) = \frac{1}{\chi_{min}(\lambda)} C(\lambda),$$

где $C(\lambda)$ – приведенная присоединенная матрица для матрицы $[\lambda E - A]$ (см. [1, с. 99]); $\chi_{min}(\lambda)$ – минимальный характеристический многочлен матрицы A.

Приведенную присоединенную матрицу $C(\lambda)$ и минимальный характеристический многочлен матрицы A будем вычислять по формулам [1, с. 99-100]

$$C(\lambda) = \frac{1}{d(\lambda)}\, B_A(\lambda)$$

$$\chi_{min}(\lambda) = \frac{\Delta(\lambda)}{d(\lambda)}$$

где $B_A(\lambda)$ – присоединенная матрица для матрицы A [1, с. 92], $d(\lambda)$ – наибольший общий делитель всех элементов матрицы $B_A(\lambda)$,

$$\Delta(\lambda) = \lambda^n - p_1\lambda^{n-1} - p_2\lambda^{n-2} - \cdots - p_{n-1}\lambda - p_n,$$

– характеристический многочлен матрицы A.

Для нахождения присоединенной матрицы $B_A(\lambda)$ воспользуемся формулой [1, с. 94]

$$B_A(\lambda) = B_0\lambda^{n-1} + B_1\lambda^{n-2} + B_2\lambda^{n-3} + \cdots + B_{n-1},$$

где B_0 – единичная $(m \times m)$ −матрица, а $(m \times m)$ −матрицы $B_k\,, k = 1, \dots, n-1$, находятся методом Д. К. Фаддеева по формулам [1, с. 97]

$$A_k = A\, B_{k-1},$$

$$p_k = \frac{1}{k}\, Sp(A_k),$$

$$B_k = A_k - p_k E,$$

$$k = 1, \dots, m.$$

Здесь E_0 – единичная $(m \times m)$ −матрица.

Программная реализация алгоритм вычисления резольвенты матрицы. Приведем фрагменты алгоритма вычисления резольвенты матрицы A на языке Python с использованием пакета SymPy.

Фрагмент кода инициализации матрицы A с целыми случайными элементами $a_{ij}, i, j = 1, \dots, m$ из отрезка $[-10, 10]$:

```
max_aij = 10
A = np.random.randint(-max_aij, max_aij + 1, (m, m))
```

Фрагмент кода реализации метода Д. К. Фаддеева одновременного вычисления характеристического многочлена $\Delta(\lambda)$ и присоединенной матрицы $B_A(\lambda)$:

```
def fadeev_method(A):
    m = len(A)
    E = sympy.eye(m)
    B = [0 for i in range(m + 1)]
    B[0] = sympy.eye(m)
    tmpA = [0 for i in range(m + 1)]
    p = [0 for i in range(m + 1)]
    for k in range(1, m + 1):
        tmpA[k] = A * B[k - 1]
        p[k] = 1 / k * matrix_trace(tmpA[k])
        B[k] = tmpA[k] - p[k] * E
    deltaLambda = lyamda ** m
    for i in range(1, m + 1):
        deltaLambda -= p[i] * lyamda ** (m - i)
    B_A = sympy.zeros(m)
    for k in range(m):
        B_A += B[k] * lyamda ** (m - 1 - k)
    return B_A, deltaLambda
```

Фрагмент кода вычисления наибольшего общего делителя всех элементов присоединенной матрицы $B_A(\lambda)$:

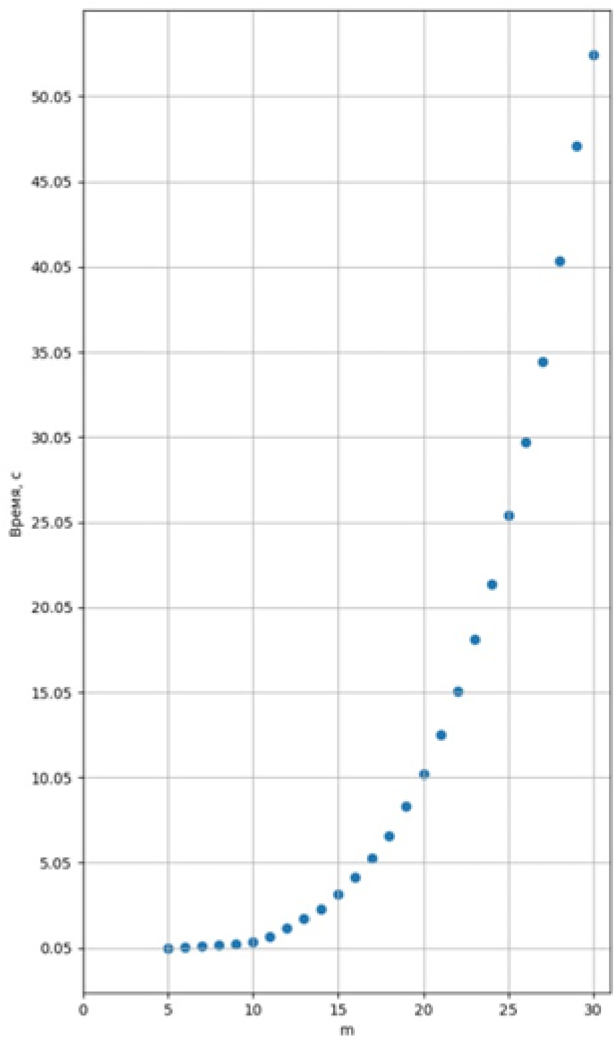
```
d = B_A[0, 0]
for i in range(m):
    for j in range(m):
        d = sympy.gcd(d, B_A[i, j])
```

Фрагмент кода вычисления приведенной присоединенной матрицы $C(\lambda)$, минимального характеристического многочлена $\chi_{min}(\lambda)$ и резольвенты $R\,(\lambda)$ матрицы A:

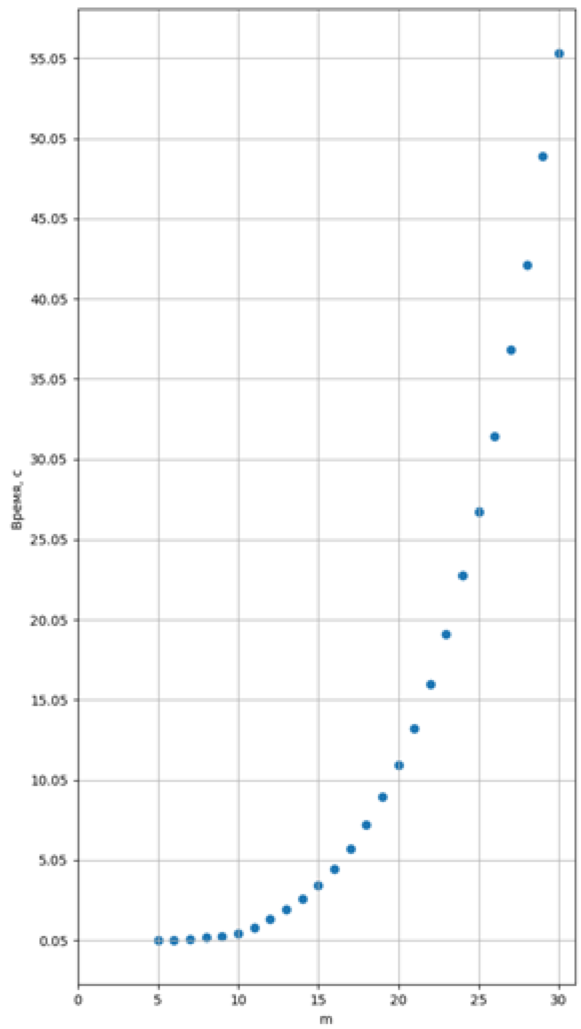
```
C = 1 / d * B_A
x_min = deltaLambda.as_expr() / d
R = 1 / x_min * C
```

Вычислительный эксперимент. Вычисления проводились на ноутбуке с процессором 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz и операционной системой Windows 11 Pro.

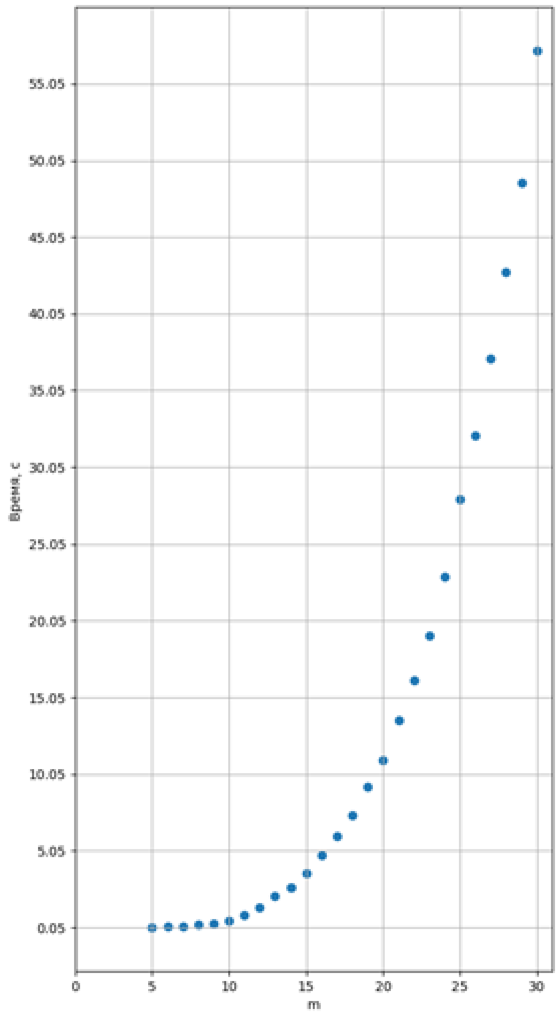
На графиках представлены зависимости скорости работы алгоритма вычисления резольвенты матрицы A с целыми случайными элементами при различных размерностях матрицы.



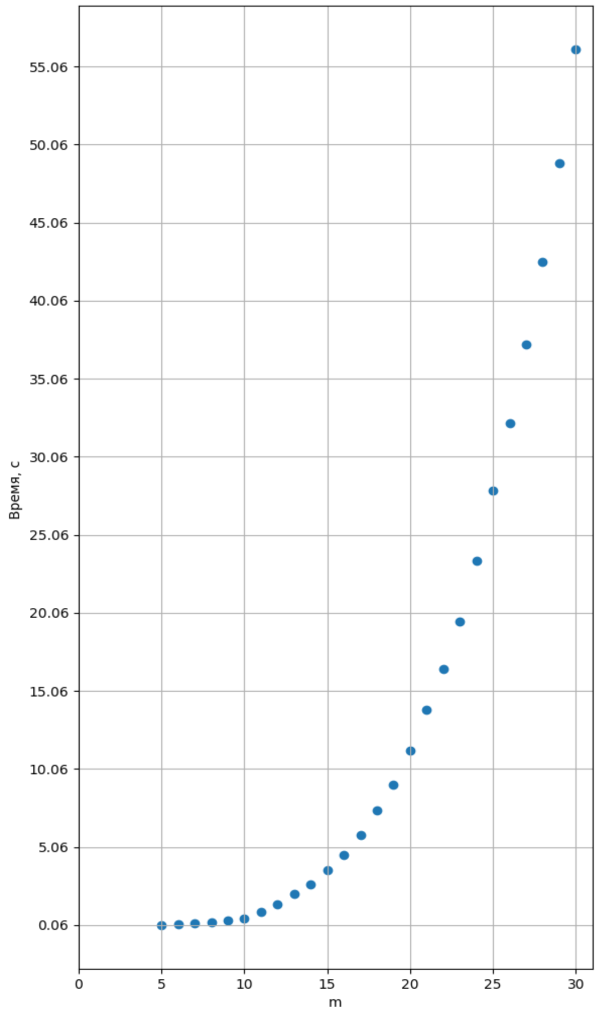
a)



b)



c)



d)

Рисунок. Графики зависимостей скорости работы алгоритма от размерности матрицы A при условиях: $a)$ $|a_{ij}| \leq 10$, $b)$ $|a_{ij}| \leq 10^5$, $c)$ $|a_{ij}| \leq 10^6$, $d)$ $|a_{ij}| \leq 10^7$, $i, j = 1, \dots, m$

Из графиков видно, что при увеличении значений коэффициентов матрицы A по абсолютной величине скорость работы алгоритма увеличивается незначительно.

Для сравнительного анализа использовалась функция пакета Sympy нахождения обратной матрицы, зависящей от параметра. Вычисления показали, что при $m = 30$ и $|a_{ij}| \leq 10$ эта функция уже затрачивает порядка 10 мин., что значительно превышает скорость работы разработанного алгоритма вычисления резольвенты матрицы с использованием приведенной присоединенной матрицы и минимального характеристического многочлена.

СПИСОК ЛИТЕРАТУРЫ

1. Гантмахер Ф. Р. Теория матриц. – М.: Наука, 1966. – 576 с.
2. Вайнберг М. М., Треногин В. А. Теория ветвления решений нелинейных уравнений. – М.: Наука, 1964. – 524 с.
3. Шаманаев П. А., Прохоров С. А. Алгоритм решения систем линейных алгебраических уравнений с малым параметром методом Ляпунова-Шмидта в регулярном случае [Электронный ресурс] // Математическое моделирование, численные методы и комплексы программ имени Е. В. Воскресенского: IX Международная научная молодежная школа-семинар (Саранск, 8-11 октября 2020 г.). – С. 129–131. – Режим доступа: https://conf.svmo.ru/files/2020/papers/paper40.pdf (дата обращения: 23.11.2023).