УДК 517.9

Кулбашева Зарина Аалиевна, магистрант, Кыргызский национальный университет им. Ж.Баласагына, г. Бишкек
e-mail: aalievna@yandex.ru

Уралиева Гулзина Алымбековна, магистрант, Кыргызский национальный университет им. Ж.Баласагына, г. Бишкек
e-mail: uralievagulzina0@gmail.com

ЧИСЛЕННОЕ РЕШЕНИЕ ЛИНЕЙНОГО РАЗНОСТНОГО УРАВНЕНИЯ С ПОСТОЯННЫМИ КОЭФФИЦИЕНТАМИ

Аннотация. В статье рассматривается численное решение линейного разностного уравнения с постоянными коэффициентами. Реализован алгоритм на языке Python, позволяющий находить общее решение уравнения на основе корней характеристического многочлена. В работе использованы библиотеки NumPy и Matplotlib для вычислений и визуализации. Представлены теоретические основы метода, подробное описание постановки задачи, а также реализация алгоритма с графическим отображением результатов.

Ключевые слова: линейное разностное уравнение, корни многочлена, постоянные коэффициенты.

The article discusses the numerical solution of a linear difference equation with constant coefficients. An algorithm implemented in Python is presented, which allows for finding the general solution of the equation based on the roots of the characteristic polynomial. The work utilizes the NumPy and Matplotlib libraries for computation and visualization. Theoretical foundations of the method are provided, along with a detailed problem statement and an implementation of the algorithm featuring graphical representation of the results.

Keywords: linear difference equation, roots of a polynomial, constant coefficients.

Постановка задачи

Рассматривается задача нахождения числовой последовательности , удовлетворяющей линейному однородному разностному уравнению с постоянными коэффициентами, которые рассматриваются в работах [1]-[3]:

где p — порядок уравнения, a — заданные параметры системы. Начальные условия задаются пользователем и определяют конкретное решение. Такие уравнения применяются при аппроксимации дифференциальных задач методами конечных разностей и в моделировании дискретных процессов.

Теоретические основы

Рассматриваемое линейное однородное разностное уравнение с постоянными коэффициентами имеет вид:

$$\sum_{k=0}^{p} a_k u_{n+k} = 0,$$

где a_k — действительные постоянные коэффициенты, а p — порядок уравнения. Это уравнение представляет собой дискретный аналог линейных дифференциальных уравнений с постоянными коэффициентами и широко применяется при численном моделировании физических и технических процессов, описываемых во временной или пространственной дискретности [1].

Для получения решения применяют метод характеристического уравнения. Характеристическое уравнение имеет вид:

$$r^p + a_{p-1} r^{p-1} + \cdots + a_1 r + a_0 = 0.$$

Анализ корней данного алгебраического уравнения позволяет определить структуру общего решения разностного уравнения [2]:

- Если все корни r_i вещественные и различны, то общее решение имеет вид:

$$u_n = C_1 r_1^n + C_2 r_2^n + \cdots + C_p r_p^n,$$

где C_i — произвольные постоянные, определяемые начальными условиями.

- При наличии кратных корней, решение дополняется множителями n^k в соответствии с кратностью:

$$u_n = (C_1 + C_2 n + \cdots + C_k n^{k-1}) r^n.$$

- В случае комплексных сопряжённых корней возникает колебательное поведение:

$$u_n = \rho^n(A\cos(n\theta) + B\sin(n\theta)),$$

где ρ — модуль корня, θ — его аргумент[3].

Таким образом, теоретическая модель основана на классическом подходе к линейным разностным уравнениям и позволяет не только численно вычислить решения, но и качественно оценить их поведение (устойчивость, осцилляции, экспоненциальный рост или затухание).

Реализация решения

Шаг 1: Ввод данных

Пользователь вручную вводит:

- порядок уравнения ppp;
- коэффициенты характеристического уравнения $a_0, a_1, \dots, a_p$;
- начальные условия $y_0, y_1, \dots, y_{p-1}$.

Это задаёт структуру уравнения и конкретизирует задачу.

Шаг 2: Построение характеристического уравнения

На основе введённых коэффициентов формируется характеристический многочлен:

$$r^p + a_{p-1}r^{p-1} + \cdots + a_1 r + a_0 = 0$$

Это уравнение описывает общую структуру решения.

Шаг 3: Нахождение корней

С помощью np.roots() находятся корни характеристического многочлена $r_1, r_2, \dots, r_p$.

Тип и кратность корней определяют форму общего решения:

- вещественные — экспоненциальный рост/спад;
- комплексные — колебания;
- кратные — добавляются множители n, n^2 и т. д.

Шаг 4: Построение системы уравнений для констант

Для определения констант $C_1, C_2, \dots, C_p$ формируется система:

$$\begin{bmatrix} r_1^0 & r_2^0 & \dots & r_p^0 \\ r_1^1 & r_2^1 & \dots & r_p^1 \\ \vdots & \vdots & \ddots & \vdots \\ r_1^{p-1} & r_2^{p-1} & \dots & r_p^{p-1} \end{bmatrix} \cdot \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_p \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{p-1} \end{bmatrix}$$

Решается с помощью np.linalg.solve.

Это позволяет точно учитывать начальные условия.

Шаг 5: Вычисление последовательности

Для каждого $n = 0, 1, \dots, 10$ считается значение по формуле общего решения:

$$y_n = \sum_{k=1}^{p} C_k \cdot r_k^n$$

(возможно, с учётом комплексных и кратных корней — как в коде).

Шаг 6: Построение графика

С помощью matplotlib.pyplot строится график:

- ось X — шаг n,
- ось Y — значение y_n,
- добавлены подписи и сетка.

Это позволяет наглядно видеть поведение решения (рост, спад, колебания и т.п.).

При вводе данных указаных на следующем рисунке

```
Введите порядок разностного уравнения (например, 4 для уравнения 4-го порядка): 5
Введите коэффициенты для разностного уравнения порядка 5 (от a_4 до a_0):
a_5: 1
a_4: 2
a_3: 3
a_2: 1
a_1: 2
a_0: 1
Введите начальные условия (y_0, y_1, ..., y_4):
y_0: 2
y_1: 1
y_2: 3
y_3: 1
y_4: 2
```

Рис 1. Скриншот ввода данных.

Получаем нижеследующий ответ

```
Корни характеристического уравнения: [-1.05409157+1.29244561j -1.05409157-1.29244561j  0.29794291+0.80522687j
  0.29794291-0.80522687j -0.48770269+0.j        ]
Константы C: [ 1.98879626-0.36322225j  0.00248856-1.02276661j  0.02374695-0.76520371j
 -0.03609981-0.44817308j  0.0456294 -0.09838444j]
```

Рис 2. Скриншоты результатов численного решения

А также график

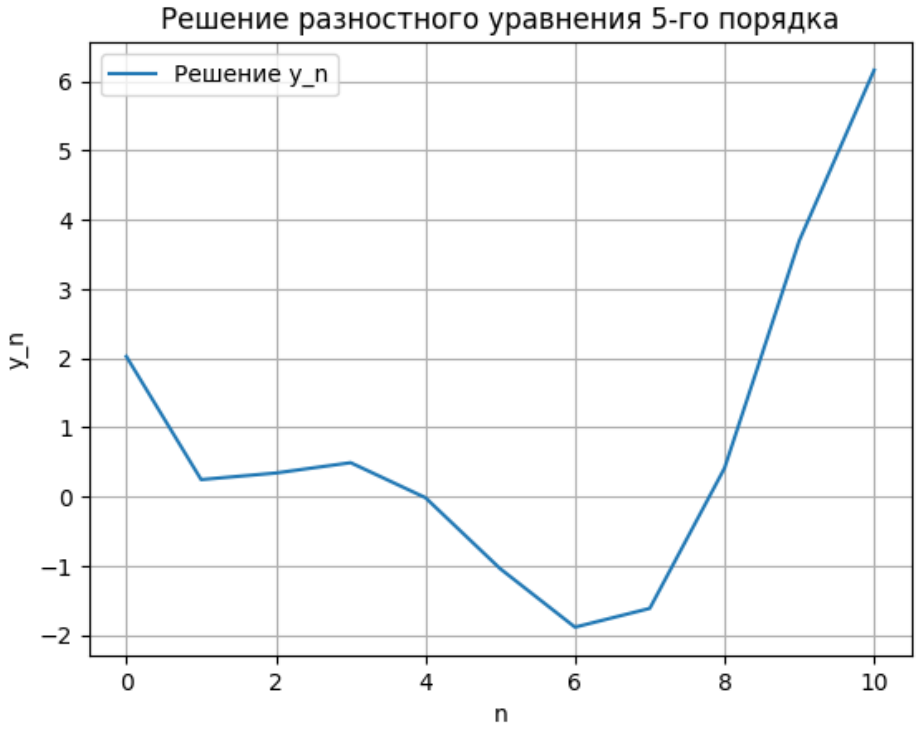

Рис 3. График численного решения.

Программная реализация задачи выполнена на языке Python с использованием библиотек NumPy и Matplotlib. Ниже приведён полный текст программы:

```
import numpy as np
import matplotlib.pyplot as plt

# Функция для решения характеристического уравнения
def solve_characteristic(coefficients):
    roots = np.roots(coefficients)
    return roots

# Функция для общего аналитического решения
def general_solution(roots, n, constants):
```

```
        solution = 0
        for i, r in enumerate(roots):
            if np.isreal(r):
                solution += constants[i] * r**n
            else:
                solution += constants[i] * np.exp(r.real * n) *
(np.cos(r.imag * n) + 1j * np.sin(r.imag * n))
        return solution

    # Функция нахождения констант C_i по начальным условиям
    def find_constants(roots, initial_conditions, p):
        A = np.array([[r**i for i in range(p)] for r in roots])
        constants = np.linalg.solve(A, initial_conditions)
        return constants

    # Ввод порядка уравнения
    p = int(input("Введите порядок разностного уравнения: "))

    # Ввод коэффициентов
    coefficients = []
    print(f"Введите коэффициенты от a_{p} до a_0:")
    for i in range(p + 1):
        coeff = float(input(f"a_{p - i}: "))
        coefficients.append(coeff)

    # Ввод начальных условий
    initial_conditions = []
    print(f"Введите начальные условия (y_0, ..., y_{p-1}):")
    for i in range(p):
        y_n = float(input(f"y_{i}: "))
        initial_conditions.append(y_n)

    # Нахождение корней
    roots = solve_characteristic(coefficients)
    print("Корни характеристического уравнения:", roots)

    # Нахождение констант C_i
    C = find_constants(roots, initial_conditions, p)
    print("Константы C:", C)
```

```
# Вычисление решения для n от 0 до 10
n_values = np.arange(0, 11)
solution_values = [general_solution(roots, n, C) for n in n_values]
solution_values_real = [np.real(sol) for sol in solution_values]

# Визуализация результатов
plt.plot(n_values, solution_values_real, label="Решение y_n")
plt.xlabel('n')
plt.ylabel('y_n')
plt.title(f'Решение разностного уравнения {p}-го порядка')
plt.grid(True)
plt.legend()
plt.show()
```

Программа реализует последовательные этапы: ввод данных, вычисление корней, нахождение констант, построение численного решения и визуализация. Она может служить базовым шаблоном для анализа поведения линейных дискретных систем и дальнейшего расширения в задачах моделирования и исследований.

Список литературы

1. Самойленко, А. М., & Перестюк, А. И. (2002). *Дифференциальные и разностные уравнения с дискретным временем*. Киев: Наукова думка.
2. Элсгольц, Л. Э. (1979). *Дифференциальные уравнения и вариационное исчисление*. Москва: Наука.
3. Гольдштейн, Х. (2003). *Классическая механика*. Москва: Наука. (Глава 2: Математические методы)