

3821Б1ПМ on 2 Kucenev

N2

$$P(z) = a_0 z^4 + a_1 z^3 + a_2 z^2 + a_3 z + a_4$$

$$3 \leq a_0 \leq 4, 6 \leq a_1 \leq 10, 9 \leq a_2 \leq 20, 3 \leq a_3 \leq 4, 2 \leq a_4 \leq 4$$

$$Q_1 = \underline{a_4} + \underline{a_3}z + \overline{a_2}z^2 + \overline{a_1}z^3 + \underline{a_0}z^4 = 0$$

$$Q_2 = \underline{a_4} + \overline{a_3}z + \overline{a_2}z^2 + \underline{a_1}z^3 + \underline{a_0}z^4 = 0$$

$$Q_3 = \overline{a_4} + \overline{a_3}z + \underline{a_2}z^2 + \underline{a_1}z^3 + \overline{a_0}z^4 = 0$$

$$Q_4 = \overline{a_4} + \underline{a_3}z + \underline{a_2}z^2 + \overline{a_1}z^3 + \overline{a_0}z^4 = 0$$

$$Q_1(z) = 2 + 3z + 20z^2 + 10z^3 + 3z^4 = 0 \text{ уст.}$$

$$Q_2(z) = 2 + 4z + 20z^2 + 6z^3 + 3z^4 = 0 \text{ уст.}$$

$$Q_3(z) = 4 + 4z + 9z^2 + 6z^3 + 4z^4 = 0 \text{ уст.}$$

$$Q_4(z) = 4 + 3z + 9z^2 + 10z^3 + 4z^4 = 0 \text{ не уст.}$$

С помощью python проверили уст. полиномов

Перепроверим Q_4

$$\begin{bmatrix} 4 & 0 & 9 & 0 & 4 \\ 0 & 10 & 0 & 3 & 0 \end{bmatrix} \xrightarrow{\lambda_1 = \frac{4}{10}} \begin{bmatrix} 0 & \frac{39}{5} & 0 & 4 \\ 10 & 0 & 3 & 0 \end{bmatrix} \xrightarrow{\lambda_2 = \frac{10.5}{39}} \begin{bmatrix} \frac{39}{5} & 0 & 4 \\ 0 & -\frac{83}{39} & 0 \end{bmatrix} \xrightarrow{\lambda_3 = -\frac{39^2}{5 \cdot 83}}$$

Видим, что $\lambda_3 < 0 \Rightarrow$ хотя бы один корень будет ≥ 0

$\Rightarrow Q_4$ - не уст. $\Rightarrow P(z)$ - не облад. робастной уст.

С помощью программы ищем корни, проверяем их действительную часть.

```
import numpy as np

def is_polynomial_stable():
    # Получаем коэффициенты полинома от пользователя
    coefficients_input = input("Введите коэффициенты полинома от старшей
    степени к младшей, разделенные пробелами: ")
    coefficients = list(map(float, coefficients_input.split()))

    # Удаляем ведущие нули
    while coefficients and coefficients[0] == 0:
        coefficients.pop(0)

    # Проверяем, что полином не нулевой
    if not coefficients:
        print("Полином тождественно равен нулю. Нет корней.")
        return False
```

```
# Проверяем случай константы
if len(coefficients) == 1:
    print("Полином является ненулевой константой. Условно
устойчив.")
    return True

# Вычисляем корни полинома
roots = np.roots(coefficients)

# Проверяем действительные части корней
for root in roots:
    if root.real >= 0:
        print(f"Найден корень с неотрицательной действительной
частью: {root}")
        return False

print("Все корни имеют отрицательные действительные части.")
return True

if __name__ == "__main__":
    if is_polynomial_stable():
        print("Полином устойчив.")
    else:
        print("Полином неустойчив.")
```