scratch.py

```python
import numpy as np

if __name__ == "__main__":
    A = np.matrix([[ 0,  1,  0, -1,  0,  0],
        [ 0,  0,  1,  1, -1,  0],
        [-1,  0,  0,  0, -1, -1],
        [ 1,  0,  0,  0,  1,  0],
        [ 0,  1,  0,  0,  0,  1],
        [ 0,  0,  1,  1,  0,  0]])

    B = (A * A + np.identity(6))
    for i in range(1, 4):
        # print(B, end="\n")
        if not np.count_nonzero(B):
            print(i)
        B = B * B

    B = A - 1j * np.identity(6)
    for i in range(1, 4):
        C = A + 1j * np.identity(6)
        for k in range(1, 4):
            # print(B * C, end="\n\n")
            if not np.count_nonzero(B * C):
                print((i, k))
            C = C * C
        B = B * B

    print("-" * 50)

    print("\n+i")

    B = A - 1j * np.identity(6)
    for i in range(1, 4):
        print(np.linalg.matrix_rank(B))
        B = B * B

    print("\n-i")

    C = A + 1j * np.identity(6)
    for i in range(1, 4):
        print(np.linalg.matrix_rank(C))
        C = C * C

"""
output:

3
(3, 3)
--------------------------------------------------

+i
5
4
3

-i
5
4
3

"""
```

LINAG Ü9

8.9.5. α) $V \ldots VR/\mathbb{R}$ $\quad f \in L(V,V)$ $\quad B \ldots$ Basis von V

$$A := \langle B^*, f(B) \rangle = \begin{pmatrix} 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & -1 & 0 \\ -1 & 0 & 0 & 0 & -1 & -1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

a) ges: $\chi_f(X)$, $\mu_f(X)$, $\chi_{f_\mathbb{C}}(X)$, $\mu_{f_\mathbb{C}}(X)$

$$\det \begin{pmatrix} 0-X & 1 & 0 & -1 & 0 & 0 \\ 0 & 0-X & 1 & 1 & -1 & 0 \\ -1 & 0 & 0-X & 0 & -1 & -1 \\ 1 & 0 & 0 & 0-X & 1 & 0 \\ 0 & 1 & 0 & 0 & 0-X & 1 \\ 0 & 0 & 1 & 1 & 0 & 0-X \end{pmatrix} = (X^2+1)^3 = \chi_f(X)$$

Da $\mu_f(X)$ ein Teiler von $\chi_f(X)$ ist und $\chi_f(X)$ ein Annulatorpolynom ist, kann $\mu_f(X)$ nur (X^2+1), $(X^2+1)^2$ oder $(X^2+1)^3$ sein.

$$A^2+1 = \begin{pmatrix} 0 & 0 & 1 & 1 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 \\ 0 & -2 & 0 & 0 & 0 & -1 \\ 0 & 2 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \neq 0 \qquad (A^2+1)^2 = \begin{pmatrix} 0 & 0 & -4 & -4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & -4 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \neq 0 \Rightarrow \underbrace{(A^2+1)^3}_{\mu_f(X)} = 0$$

Aus Satz 8.9.2 folgt $\chi_f(X) = \chi_{f_\mathbb{C}}(X) = (X^2+1)^3 = (X-i)^3(X+i)^3$

$\mu_{f_\mathbb{C}}(X)$ kann nur $(X-i)^l(X+i)^m$ mit $l,m \in \{1,2,3\}$ sein. Durch nachrechnen liefert nur $(X-i)^3(X+i)^3 = 0$ $\Rightarrow \mu_{f_\mathbb{C}}(X) = (X-i)^3(X+i)^3$

b) ges: A in J-NF und A in reeller J-NF

$\mathrm{rg}(A-(-i)\cdot E) = 5 \qquad \mathrm{rg}((A-(-i)\cdot E)^2) = 4 \qquad \mathrm{rg}((A-(-i)\cdot E)^3) = 3$

$\mathrm{rg}(A-i\cdot E) = 5 \qquad \mathrm{rg}((A-i\cdot E)^2) = 4 \qquad \mathrm{rg}((A-i\cdot E)^3) = 3$

$\pm i$:	0	1	2	3	4	
r	6	5	4	3	3	…
u		1	1	1	0	…
k		0	0	1	0	…

$$\Rightarrow \text{J-NF von } A = \begin{pmatrix} -i & 1 & 0 & 0 & 0 & 0 \\ 0 & -i & 1 & 0 & 0 & 0 \\ 0 & 0 & -i & 0 & 0 & 0 \\ 0 & 0 & 0 & i & 1 & 0 \\ 0 & 0 & 0 & 0 & i & 1 \\ 0 & 0 & 0 & 0 & 0 & i \end{pmatrix}$$

$$\text{reelle J-NF von } A = \begin{pmatrix} J_3(0) & -i\cdot E_3 \\ i\cdot E_3 & J_3(0) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & -i & 0 & 0 \\ 0 & 0 & 1 & 0 & -i & 0 \\ 0 & 0 & 0 & 0 & 0 & -i \\ i & 0 & 0 & 0 & 1 & 0 \\ 0 & i & 0 & 0 & 0 & 1 \\ 0 & 0 & i & 0 & 0 & 0 \end{pmatrix}$$ laut 8.9.4 im Buch

(siehe auch Python Code)