Requirements specification:

Given the following open loop transfer function with four poles at S = 0, S= -25 S=-50 + j 10 and S = -50 –j 10 and no zeroes. It is required to write a program to draw the root locus following the rules.

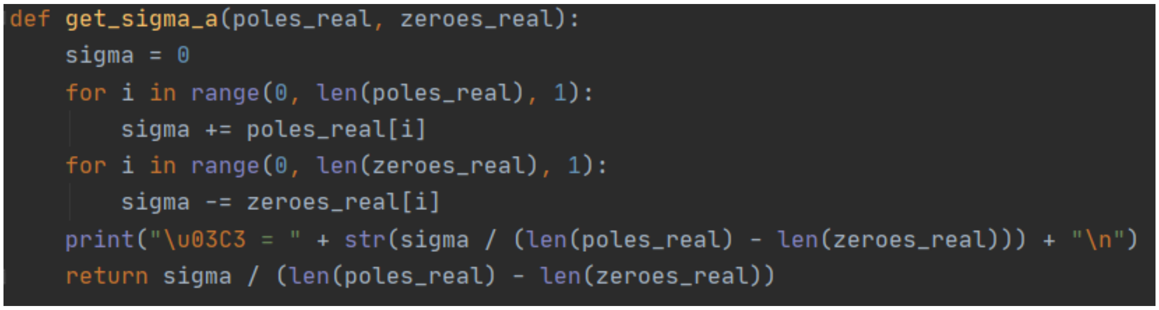
Main data structures:

1- Arrays
2- tubles

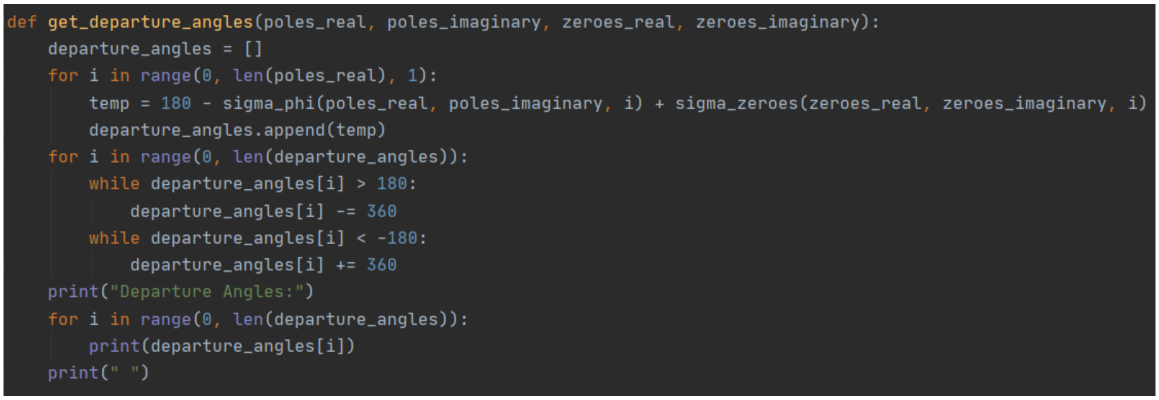
Algorithms description:

1- Drawing poles and the asymptotes.
2- Finding breakaway points and the intersections with the imaginary axis using Routh.
3- Finding Departure angles for poles
4- drawing approximated curves that start at a complex pole or the breakaway point and across the intersection with the imaginary axis.

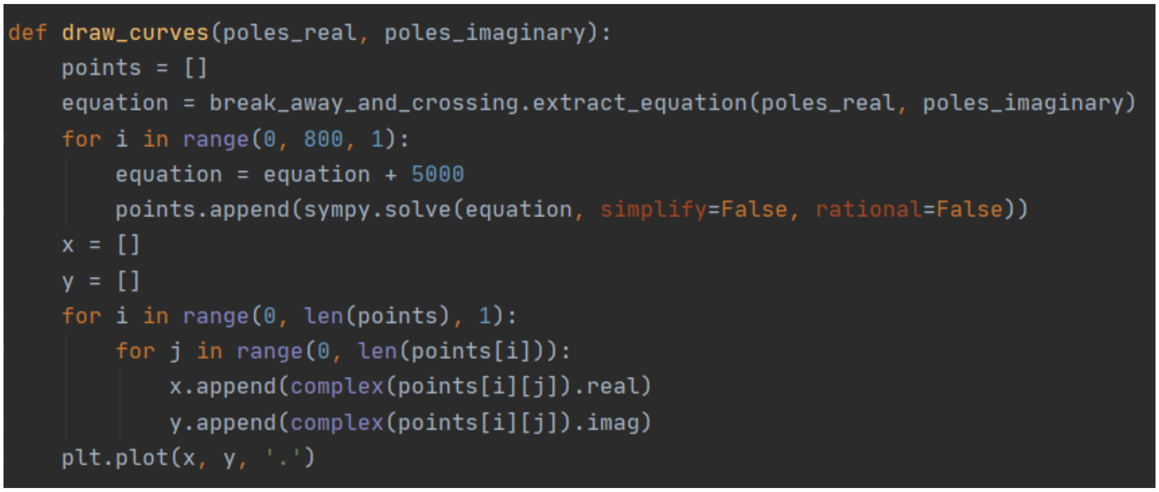
Code Snippets:



```
def get_sigma_a(poles_real, zeroes_real):
    sigma = 0
    for i in range(0, len(poles_real), 1):
        sigma += poles_real[i]
    for i in range(0, len(zeroes_real), 1):
        sigma -= zeroes_real[i]
    print("\u03C3 = " + str(sigma / (len(poles_real) - len(zeroes_real))) + "\n")
    return sigma / (len(poles_real) - len(zeroes_real))
```

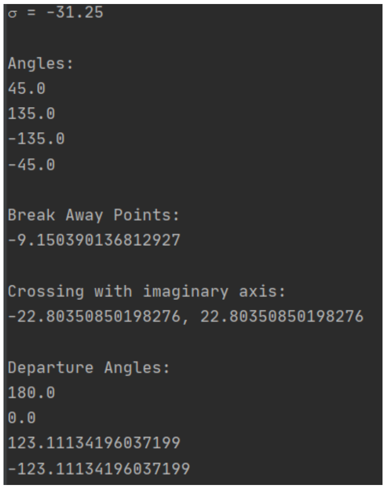


```python
def get_departure_angles(poles_real, poles_imaginary, zeroes_real, zeroes_imaginary):
    departure_angles = []
    for i in range(0, len(poles_real), 1):
        temp = 180 - sigma_phi(poles_real, poles_imaginary, i) + sigma_zeroes(zeroes_real, zeroes_imaginary, i)
        departure_angles.append(temp)
    for i in range(0, len(departure_angles)):
        while departure_angles[i] > 180:
            departure_angles[i] -= 360
        while departure_angles[i] < -180:
            departure_angles[i] += 360
    print("Departure Angles:")
    for i in range(0, len(departure_angles)):
        print(departure_angles[i])
    print(" ")
```



```python
def draw_curves(poles_real, poles_imaginary):
    points = []
    equation = break_away_and_crossing.extract_equation(poles_real, poles_imaginary)
    for i in range(0, 800, 1):
        equation = equation + 5000
        points.append(sympy.solve(equation, simplify=False, rational=False))
    x = []
    y = []
    for i in range(0, len(points), 1):
        for j in range(0, len(points[i])):
            x.append(complex(points[i][j]).real)
            y.append(complex(points[i][j]).imag)
    plt.plot(x, y, '.')
```

Sample Run:



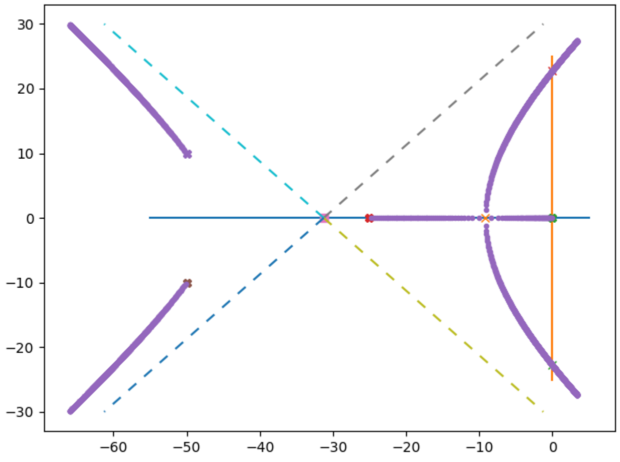
```
σ = -31.25

Angles:
45.0
135.0
-135.0
-45.0

Break Away Points:
-9.150390136812927

Crossing with imaginary axis:
-22.80350850198276, 22.80350850198276

Departure Angles:
180.0
0.0
123.11134196037199
-123.11134196037199
```



Project Link:

https://github.com/wzattout/root-locus