1. $$\frac{\left(1 - i\sqrt{3}\right)^{6/5}}{(1-i)^{12/5}} = \left(\frac{2e^{-\frac{1}{3}\pi i}}{\left(\sqrt{2}\, e^{-1/4\,\pi i}\right)^2}\right)^{6/5}$$

$$= \left(\frac{2e^{-1/3\,\pi i}}{2e^{-\pi/2\, i}}\right)^{6/5}$$

$$= \left(1\, e^{\pi/6\, i}\right)^{6/5}$$

$$= e^{\pi/5\, i}$$

$$= \cos(\pi/5) + i\sin(\pi/5)$$

$$= 0.809 + i\,0.588$$

2 a) $\cos(e^z)$

$$e^z = e^x e^{iy}$$
$$= e^x(\cos y + i\sin y)$$

$$\cos(e^z) = \cos(e^x(\cos y + i\sin y))$$
$$= \cos(e^x\cos y)\cos(e^x i\sin y) - \sin(e^x\cos y)\sin(e^x i\sin y)$$
$$= \cos(e^x\cos y)\cosh(e^x\sin y) - i\sin(e^x\cos y)\sinh(e^x\sin y)$$

Real part $\cos(e^x\cos y)\cosh(e^x\sin y)$

Imaginary part $-\sin(e^x\cos y)\sinh(e^x\sin y)$

2b) $\cos(z) = \cos(x+iy)$

$= \cos x \cos iy - \sin x \sin iy$

$= \cos x \cosh y - i \sin x \sinh y$

$e^{\cos x \cosh y - i \sin x \sinh y}$

$= e^{\cos x \cosh y} e^{-i \sin x \sinh y}$

$= e^{\cos x \cosh y}\left(\cos(-\sin x \sinh y) + i \sin(-\sin x \sinh y)\right)$

$= e^{\cos x \cosh y}\left(\cos(\sin x \sinh y) - i \sin(\sin x \sinh y)\right)$

Real part $e^{\cos x \cosh y} \cos(\sin x \sinh y)$

Imaginary part $-e^{\cos x \cosh y} \sin(\sin x \sinh y)$

3a) $\sum_{K=0}^{n-1} w_K = 0 \quad , \quad w_K = e^{2\pi i K/n}$

$= w_0 + w_1 + w_2 + \dots w_n$

$= 1 + e^{\frac{2\pi i}{n}} + e^{\frac{4\pi i}{n}} + e^{6\pi i/n} + e^{8\pi i/n} + \dots + e^{\frac{2(n-1)\pi i}{n}}$

$= 1 + \left(e^{\frac{2\pi i}{n}}\right)^1 + \left(e^{\frac{2\pi i}{n}}\right)^2 + \left(e^{\frac{2\pi i}{n}}\right)^3 + \dots + \left(e^{\frac{2\pi i}{n}}\right)^{(n-1)}$

$\alpha = w_1^0 + w_1^1 + w_1^2 + w_1^3 + \dots + w_1^{(n-1)}$

$w_1 \alpha = w_1 + w_1^2 + w_1^3 + w_1^4 + \dots + w_1^n$

$\alpha - w_1 \alpha = w_1^0 - w_1^n$

$\alpha(1 - w_1) = w_1^0 - w_1^n$

$\alpha = \frac{w_1^0 - w_1^n}{1 - w_1}$

$\alpha = \frac{1 - w_1^n}{1 - w_1}$ $\qquad w_1 \neq 0$

$\alpha = \frac{1 - \left(e^{\frac{2\pi i}{n}}\right)^n}{1 - e^{\frac{2\pi i}{n}}}$ $\qquad w_K = e^{2\pi i K/n} \neq 0$

$\Rightarrow w_1 \neq 0$

$\alpha = \frac{1 - 1^n}{1 - w_1}$

$\alpha = 0 \quad \therefore \sum_{K=0}^{n-1} w_K = 0$

3 b) $$\prod_{K=0}^{n-1} w_K = \prod_{K=0}^{n-1} e^{\frac{2\pi i K}{n}}$$

$$= e^{\frac{2\pi i}{n}\left(0+1+2+\cdots+(n-1)\right)}$$

$$0+1+2+\cdots+(n-1) = \sum_{K=0}^{n-1} K$$

$$2\sum_{K=0}^{n-1} K = 1+2+\cdots+(n-2)+(n-1)+(n-1)+(n-2)+\cdots+2+1$$

$$= n(n-1)$$

$$\Rightarrow \sum_{K=0}^{n-1} K = \frac{n(n-1)}{2}$$

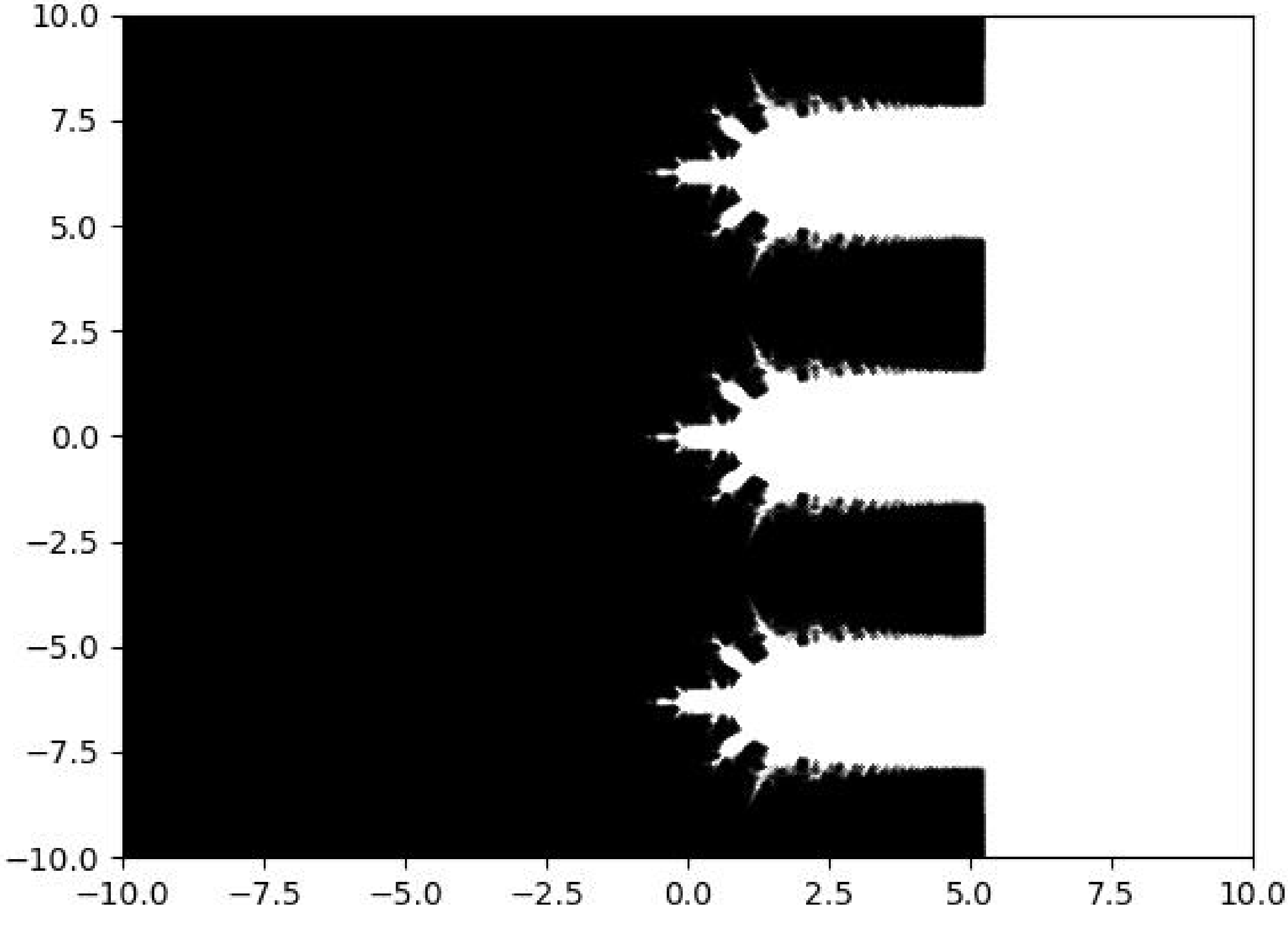
$$\prod_{K=0}^{n-1} w_K = e^{\frac{2\pi i}{n}\left(\frac{n(n-1)}{2}\right)}$$

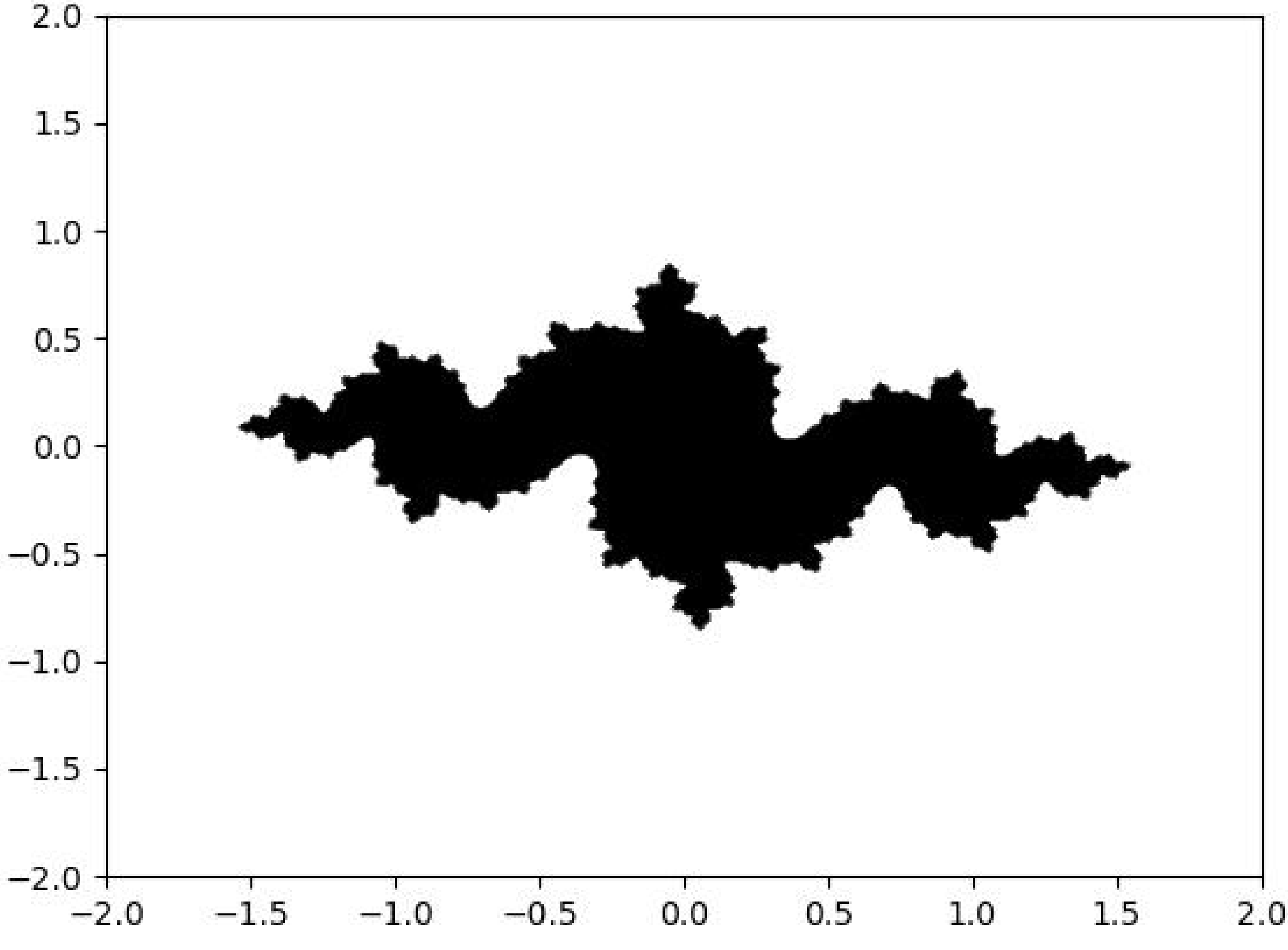
$$= e^{i\pi(n-1)}$$

$$= \left(e^{i\pi}\right)^{(n-1)}$$

$$= (-1)^{(n-1)}$$

$$\therefore \prod_{K=0}^{n-1} w_K = (-1)^{(n-1)}$$


10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0



2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

```python
# mandelbrot.py
import numpy as np
import matplotlib.pyplot as plt


def mandelbrotEsq(a, b, n, thresh):
    xn = 0
    yn = 0
    for i in range(n):
        x = np.exp(xn)*np.cos(yn)+a
        y = np.exp(xn)*np.sin(yn)+b
        if np.linalg.norm([x, y], 2) > thresh:
            return False
        xn = x
        yn = y
    return True


n = 1000
a = np.linspace(-10, 10, n)
b = np.linspace(-10, 10, n)

# check if the point is in the mandelbrot set and if it is plot it on the graph
# 15 iterations of the function are done and it is removed from the set if the
# magnitude of the point is greater than 500
for i in range(n):
    for j in range(n):
        if mandelbrotEsq(a[i], b[j], 15, 500):
            plt.plot(a[i], b[j], 'k.', markersize=0.3)
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.show()
plt.savefig('mandelbrot.png')

```

```python
# julia.py
import numpy as np
import matplotlib.pyplot as plt


def julia(x, y, a, b, n, thresh):
    for i in range(n):
        #zn = zn**2 + c
        xn = x**2 - y**2 + a
        yn = 2*x*y + b
        if np.linalg.norm([xn, yn], 2) > thresh:
            return False
        x = xn
        y = yn
    return True


a = -0.83
b = 0.18

n = 1000
x = np.linspace(-2, 2, n)
y = np.linspace(-2, 2, n)
# check if the point is in the julia set and if it is plot it on the graph
# 15 iterations of the function are done and it is removed from the set if the
# magnitude of the point is greater than 1000
for i in range(n):
    for j in range(n):
        if julia(x[i], y[j], a, b, 15, 1000):
            plt.plot(x[i], y[j], 'k.', markersize=0.3)

plt.xlim(-2, 2)
plt.ylim(-2, 2)
plt.show()
plt.savefig('julia.png')

```