PEC 3

Pablo Riutort Grande

May 28, 2023

M0.534 - Chaotic Dynamical Systems

Master's Degree in Computational and Mathematical Engineering

1

Suppose $f(z)$ has a fixed point at z_0. Then z_0 is [1]:

1. Attracting if $|f'(z_0)| < 1$
2. Repelling if $|f'(z_0)| > 1$
3. Neutral if $|f'(z_0)| = 1$

Plot the Dynamical Plane (Julia and Fatou sets) of $Q_c(z) = z^2 + c$ for the following values of c.

(a) $c_0 = -\frac{1}{2} - \frac{1}{10}i$. [Fig.3]. Prove that Q_{c_0} has an attracting fixed point.
To prove that $Q_{c_0}(z) = z^2 + c_0$ has an attrracting fixed point, we need to find a value of z such that $Q_{c_0}(z) = z$ and the derivative of Q_{c_0} evaluated at that point is < 1.

$$\begin{aligned}
z^2 + c_0 &= z \\
z^2 + -z + c_0 &= 0 \\
z &= \frac{-(-1) \pm \sqrt{-1^2 - 4(1)(c_0)}}{2} \\
z &= \frac{1 \pm \sqrt{1 - 4\left(-\frac{1}{2} - \frac{1}{10}i\right)}}{2} \\
z &= \frac{1 \pm \sqrt{1 + 2 + \left(\frac{4}{10}i\right)}}{2} \\
z &= \frac{1 \pm \sqrt{3 + \left(\frac{2}{5}i\right)}}{2} \\
z_1 &= \frac{1 - \sqrt{3 + \left(\frac{2}{5}i\right)}}{2} \\
z_2 &= \frac{1 + \sqrt{3 + \left(\frac{2}{5}i\right)}}{2}
\end{aligned} \tag{1}$$

$$
\begin{aligned}
Q'_{c0}(z) &= 2z \\
Q'_{c0}(z_1) &= 2\left(\frac{1-\sqrt{3+\left(\frac{2}{5}i\right)}}{2}\right) \\
Q'_{c0}(z_1) &= 1-\sqrt{3+\left(\frac{2}{5}i\right)} \\
Q'_{c0}(z_1) &= |-0.7358-0.1152i| < 0
\end{aligned} \tag{2}
$$

Hence z_1 is an attracting fixed point.

(b) $c_1 = \frac{1}{2}i$. [Fig.4]. Prove that the Q_{c_1} has an attracting fixed point.
To prove that $Q_{c_1}(z) = z^2 + c_1$ has an attrracting fixed point, we need to find a value of z such that $Q_{c_1}(z) = z$ and the derivative of Q_{c_1} evaluated at that point is < 1.

$$
\begin{aligned}
z^2 + c_1 &= z \\
z^2 + -z + c_1 &= 0 \\
z &= \frac{-(-1) \pm \sqrt{-1^2 - 4(1)(c_1)}}{2} \\
z &= \frac{1 \pm \sqrt{1-4\left(\frac{1}{2}i\right)}}{2} \\
z &= \frac{1 \pm \sqrt{1-2i}}{2} \\
z_1 &= \frac{1 - \sqrt{1-2i}}{2} \\
z_2 &= \frac{1 + \sqrt{1-2i}}{2}
\end{aligned} \tag{3}
$$

$$
\begin{aligned}
Q'_{c_1}(z) &= 2z \\
Q'_{c_1}(z_1) &= 2\left(\frac{1-\sqrt{1-2i}}{2}\right) \\
Q'_{c_1}(z_1) &= 1-\sqrt{1-2i} \\
Q'_{c_1}(z_1) &= |-0.272+0.786i| < 0
\end{aligned} \tag{4}
$$

Hence z_1 is an attracting fixed point.

(c) $c_2 = -1$. [Fig.5]. Prove that the Q_{c_2} has an attracting cycle of period two. To prove that Q_{c_2} has an attracting cycle of period two where $c_2 = -1$ we need to show the existence of two distinct points z_1, z_2 such that $Q_{c_2}(z_1) = z_2$ and $Q_{c_2}(z_2) = z_1$.
We can start by evaluating $Q_{c_2}(z_1)$ on c_2:

$$
\begin{aligned}
Q_{c_2}(z_1) &= z_1^2 - 1 \\
Q_{c_2}(z_1) &= (-1)^2 - 1 \\
Q_{c_2}(z_1) &= 0
\end{aligned} \tag{5}
$$

Now we can proceed by substituing the value of z_2 by z_1 in:

$$
\begin{aligned}
Q_{c_2}(z_2) &= z_2^2 - 1 \\
Q_{c_2}(z_2) &= (0)^2 - 1 \\
Q_{c_2}(z_2) &= -1
\end{aligned} \tag{6}
$$

This shows that $Q_{c_2}(z_1) = z_2$ and $Q_{c_2}(z_2) = z_1$ and so there is a cycle of two.
To see if this cycle is attracting we consider z_1 as a fixed point for $Q^2_{c_2}(z)$. We make use of the chain rule to compute $(Q^2_{c_2})'(z_1)$ [1]:

$$(Q^2_{c_2})'(z_1) = Q'_{c_2}(Q_{c_2}(z_1))Q'_{c_2}(z_1) = Q'_{c_2}(z_2)Q'_{c_2}(z_1) = 2(-1)2(0) = 0 < 1. \tag{7}$$

Which means the 2-cycle is attracting.

(d) $c_3 = \frac{1}{4}$. [Fig.6]. Prove that the Q_{c_3} has a neutral fixed point.
To prove that $Q_{c_3}(z) = z^2 + \frac{1}{4}$ has a neutral fixed point, we need to find a value of z such that $Q_{c_3}(z) = z$ and the derivative of Q_{c_3} evaluated at that point is equal to 1.

$$\begin{aligned} z^2 + \frac{1}{4} &= z \\ z^2 - z + \frac{1}{4} &= 0 \\ \left(z - \frac{1}{2}\right)^2 &= 0 \\ z - \frac{1}{2} &= 0 \\ z &= \frac{1}{2} \end{aligned} \tag{8}$$

We now have a fixed point at $z = \frac{1}{2}$, let's evaluate the derivative of Q_{c_3} at that point

$$\begin{aligned} Q'_{c_3}(z) &= 2z \\ Q'_{c_3}\left(\frac{1}{2}\right) &= 2\left(\frac{1}{2}\right) = 1 \end{aligned} \tag{9}$$

Then, $z = \frac{1}{2}$ is a neutral fixed point.

(e) $c_4 = -\frac{3}{4}$. [Fig.7]. Prove that the Q_{c_4} has a neutral cycle of period two.

$$\begin{aligned} Q_{c_4}(z_1) &= z_1^2 - \frac{3}{4} \\ Q_{c_4}(z_1) &= z^2 - \frac{3}{4} = z \\ Q_{c_4}(z_1) &= z^2 - z - \frac{3}{4} = 0 \\ z_{1_a} &= -\frac{1}{2} \\ z_{1_b} &= \frac{3}{2} \end{aligned} \tag{10}$$

$$z_2 = (z_1)^2 - \frac{3}{4} \tag{11}$$

$$\begin{aligned} Q_{c_4}\left(-\frac{1}{2}\right) &= \left(-\frac{1}{2}\right)^2 - \frac{3}{4} = \frac{1}{4} - \frac{3}{4} = -\frac{2}{4} = -\frac{1}{2} = z_{1_a} \\ Q_{c_4}\left(\frac{3}{2}\right) &= \left(\frac{3}{2}\right)^2 - \frac{3}{4} = \frac{9}{4} - \frac{3}{4} = \frac{6}{4} = \frac{3}{2} = z_{1_b} \end{aligned} \tag{12}$$

This shows that $Q_{c_4}(z_1) = z_2$ and $Q_{c_4}(z_2) = z_1$ and so there is a cycle of two. Similarly as in 7 we can make use of the chain rule to observe its behaviour in z_{1_a}:

$$Q'_{c_4}(z_2)Q'_{c_4}(z_1) = 2\left(-\frac{1}{2}\right)2\left(-\frac{1}{2}\right) = |(-1)(-1)| = 1 \tag{13}$$

Which means the 2-cycle is neutral.

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

def iterate_z(z, c, max_iterations, threshold=10):
    """
    Perform iteration of the quadratic polynomial function Qc(z) = z^2 + c.

    Parameters:
        z (complex): The complex number to iterate.
        c (complex): The parameter c in Qc(z) = z^2 + c.
        max_iterations (int): Maximum number of iterations.
        threshold (int): Point of escape.

    Returns:
        int: The number of iterations until the point escapes or max_iterations is
     reached.
    """
    for i in range(max_iterations):
        z = z**2 + c
        # stopping condition
        if abs(z) > threshold:
            return i
    return max_iterations

def plot_julia_set(c, N, max_iterations=100):
    """
    Plot the Julia set for the function Qc(z) = z^2 + c.

    Parameters:
        c (complex): The parameter c in Qc(z) = z^2 + c.
        N (int): Grid size for the complex plane.
        max_iterations (int): Maximum number of iterations.

    Returns:
        None
    """
    xmin, xmax, ymin, ymax = -2, 2, -2, 2

    julia_set = np.zeros((N, N))
    for l in range(N):
        for m in range(N):
            # Use the given expression to define the grid points
            z = -2 + 4 * l / N + (-2 + 4 * m / N) * 1j
            julia_set[l, m] = iterate_z(z, c, max_iterations)

    plt.imshow(julia_set.T, cmap='hot', extent=[xmin, xmax, ymin, ymax])
    plt.colorbar()
    plt.xlabel('Real')
    plt.ylabel('Imaginary')
    plt.savefig(f'Quadratic/julia_set_{c.real}_{c.imag}.png')
```

```python
    plt.close()

c_values = [
    -1/2 - 1/10j, # c0
    1/2j, # c1
    -1, # c2
    1/4, # c3
    -3/4 # c4
]

N = 500
# Iterate over c_values and plot/save Julia sets
for c in c_values:
    plot_julia_set(c, N)
```

Listing 1: Plot the Dynamical Plane.

2

Plot the Mandelbrot set $\mathcal{M}$ (two colors)

(a) Plot in color #0 the set of escaping parameters, i.e., parameters $c_{l,m}$ such that the critical orbit $Q^n_{c_{l,m}}(0)$ escapes to infinity.

(b) Plot in color #1 the set of bounded parameters, i.e., parameters $c_{l,m}$ such that the critical orbit $Q^n_{c_{l,m}}(0)$ is bounded.

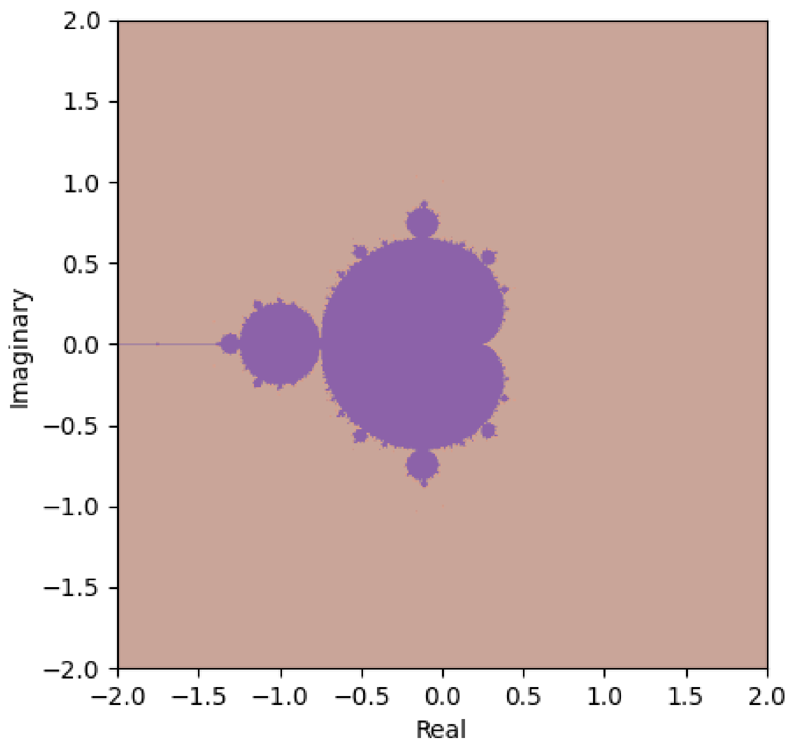

Figure 1: The Mandelbrot set with escaping (orange) and bounded (purple) parameters.

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


def iterate_z(z, c, max_iterations, threshold=2):
    """
    Perform iteration of the quadratic polynomial function Qc(z) = z^2 + c.

    Parameters:
        z (complex): The complex number to iterate.
        c (complex): The parameter c in Qc(z) = z^2 + c.
        max_iterations (int): Maximum number of iterations.
        threshold (int): Point of escape.

    Returns:
        int: The number of iterations until the point escapes or max_iterations is
```

```python
    reached.
    """
    iteration = 0
    for _ in range(max_iterations):
        z = z**2 + c
        # stopping condition
        if abs(z) > threshold:
            break
        iteration += 1
    return iteration

def mandelbrot_set(N, max_iterations=100):
    """
    Plot the Mandelbrot set for the function Qc(z) = z^2 + c.

    Parameters:
        N (int): Grid size for the complex plane.
        max_iterations (int): Maximum number of iterations.

    Returns:
        None
    """
    x = np.linspace(-2, 2, N)
    y = np.linspace(-2, 2, N)
    grid = np.array([complex(r, i) for r in x for i in y]).reshape(N, N)

    escape_iterations = np.zeros((N, N))

    for l in range(N):
        for m in range(N):
            c = -2 + 4*l/N + (-2 + 4*m/N)*1j
            z = 0  # Initialize the critical orbit
            escape_iterations[l, m] = iterate_z(z, c, max_iterations)

    bounded_mask = escape_iterations == 100
    escaping_mask = escape_iterations < 100
    cmap_escaping = 'Oranges'
    cmap_bounded = 'Purples'

    plt.imshow(escaping_mask.T, cmap=cmap_escaping, extent=(-2, 2, -2, 2))
    plt.imshow(bounded_mask.T, cmap=cmap_bounded, extent=(-2, 2, -2, 2), alpha
    =0.6)
    plt.xlabel('Real')
    plt.ylabel('Imaginary')
    plt.savefig('Quadratic/mandelbrot_2.png')
    plt.close()

N = 500
mandelbrot_set(N)
```

Listing 2: Plot the Mandelbrot set (two colors).

3

Plot the Mandelbrot set $\mathcal{M}$ (many colors)

(a) Plot in color #0 the set of escaping parameters.

(b) Plot in color #1 the set of parameters $c_{l,m}$ converging towards a fixed point.

(c) Plot in color #2 the set of parameters $c_{l,m}$ converging towards a periodic orbit of period two.

(d) Plot in color #3 the set of parameters $c_{l,m}$ converging towards a periodic orbit of period three.

(e) Plot in color #4 the set of parameters $c_{l,m}$ converging towards a periodic orbit of period four.

(f) Plot in color #5 the set of parameters $c_{l,m}$ converging towards a periodic orbit of period five.

With a few modifications on the previous program [Lst.1] we can accomplish the desired result [Lst.3].

```
escape_mask = escape_iterations == 100
fixed_point_mask = escape_iterations == 1
# converging towards a periodic orbit of two
period_two_mask = np.logical_and(escape_iterations > 1, escape_iterations % 2
== 0)
# converging towards a periodic orbit of three
period_three_mask = np.logical_and(escape_iterations > 1, escape_iterations %
3 == 0)
# converging towards a periodic orbit of four
period_four_mask = np.logical_and(escape_iterations > 1, escape_iterations % 4
 == 0)
# converging towards a periodic orbit of five
period_five_mask = np.logical_and(escape_iterations > 1, escape_iterations % 5
 == 0)

cmap_escape = 'Greys'   # Color #0
cmap_fixed_point = 'inferno'   # Color #1
cmap_period_two = 'Blues'   # Color #2
cmap_period_three = 'Greens'   # Color #3
cmap_period_four = 'Oranges'   # Color #4
cmap_period_five = 'Purples'   # Color #5

plt.imshow(escape_mask.T, cmap=cmap_escape, extent=(-2, 2, -2, 2))
plt.imshow(fixed_point_mask.T, cmap=cmap_fixed_point, extent=(-2, 2, -2, 2),
alpha=0.3)
plt.imshow(period_two_mask.T, cmap=cmap_period_two, extent=(-2, 2, -2, 2),
alpha=0.3)
plt.imshow(period_three_mask.T, cmap=cmap_period_three, extent=(-2, 2, -2, 2),
 alpha=0.3)
plt.imshow(period_four_mask.T, cmap=cmap_period_four, extent=(-2, 2, -2, 2),
alpha=0.3)
plt.imshow(period_five_mask.T, cmap=cmap_period_five, extent=(-2, 2, -2, 2),
alpha=0.3)
```

Listing 3: Plot the Mandelbrot set (many colors).

(a) The set of escaping parameters is grey.

(b) The set of parameters $c_{l,m}$ converging towards a fixed point is red.

(c) The set of parameters $c_{l,m}$ converging towards a periodic orbit of period two is blue.

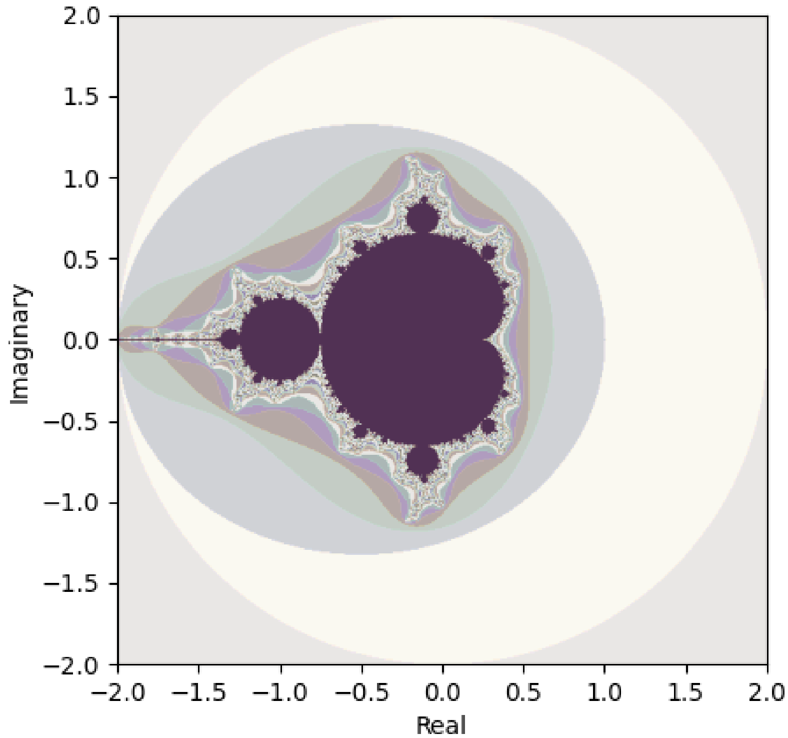

Figure 2: The Mandelbrot set with many colors*.

(d) The set of parameters $c_{l,m}$ converging towards a periodic orbit of period three is green.

(e) The set of parameters $c_{l,m}$ converging towards a periodic orbit of period four is orange.

(f) The set of parameters $c_{l,m}$ converging towards a periodic orbit of period five is purple.

* *Apologies if the color palette is not soothing, the author is colorblind.*

4

A Siegel disk is a Fatou component where points rotate around an indifferent fixed point. We denote the Golden mean by $\varphi = \frac{\sqrt{5}-1}{2} = 0.6180339887...$ and consider the parameter $c_\varphi = \frac{1}{2}e^{2\pi i\varphi} - \frac{1}{4}e^{4\pi i\varphi}$.

(a) Plot de Dynamical plane of $Q_c(z) = z^2 + c\varphi$. [Fig.8]

(b) Compute the fixed points of Q_c and their multipliers.
To compute the fixed points of the quadratic map $Q_c(z) = z^2+c\varphi$ and their multipliers, we can start by setting $z = Q_c(z)$ and solving for z with the quadratic formula. The resulting values of z will be the fixed points and with those we can compute their multipliers by evaluating of $Q'_c(z)$ at each fixed point. The multiplier is given by the absolute value of the derivative [Lst.4]:

```python
import numpy as np

def compute_fixed_points_and_multipliers(a, b, c):
    """
    Quadratic formula implementation with multipliers

    Parameters:
        a (int): First coefficient
        b (int): Second coefficient
        c (complex): The parameter c in Qc(z) = z^2 + c_phi

    Returns:
        z_1 (int): First fixed point
        multiplier_1 (np.abs): absolute values of the derivatives
        z_2 (int): Second fixed point
        multiplier_2 (np.abs): absolute values of the derivatives
    """
    z_1 = (-b + np.sqrt(b**2 - 4 * a * c)) / (2 * a)
    z_2 = (-b - np.sqrt(b**2 - 4 * a * c)) / (2 * a)
    derivative_1 = 2 * z_1
    derivative_2 = 2 * z_2
    return z_1, np.abs(derivative_1), z_2, np.abs(derivative_2)

phi = (np.sqrt(5) - 1) / 2
c_phi = 1/2 * np.exp(2 * np.pi * 1j * phi) - 1/4 * np.exp(4 * np.pi * 1j * phi
    )

print("Fixed point 1 {}\nMultiplier 1 {}\nFixed point 2: {}\nMultiplier 2: {}"
    .format(*compute_fixed_points_and_multipliers(1, -1, c_phi)))
```

Listing 4: Compute the fixed points

Results:

```
        Fixed point 1 (1.36868443903916+0.3377451471307618j)
        Multiplier 1 2.819481426133763
        Fixed point 2: (-0.36868443903915993-0.3377451471307618j)
        Multiplier 2: 1.0

```

(c) Plot in the dynamical plane the orbit of several (five) points near the indifferent fixed point. [Fig.9]

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

phi = (np.sqrt(5) - 1) / 2
c_phi = 1/2 * np.exp(2 * np.pi * 1j * phi) - 1/4 * np.exp(4 * np.pi * 1j * phi)

def iterate_z(z, c, max_iterations, threshold=10):
    """
    Perform iteration of the quadratic polynomial function Qc(z) = z^2 + c.

    Parameters:
        z (complex): The complex number to iterate.
        c (complex): The parameter c in Qc(z) = z^2 + c.
        max_iterations (int): Maximum number of iterations.
        threshold (int): Point of escape.

    Returns:
        int: The number of iterations until the point escapes or max_iterations is reached.
    """
    for i in range(max_iterations):
        z = z**2 + c
        # stopping condition
        if abs(z) > threshold:
            return i
    return max_iterations

def plot_dynamical_plane(c, N, max_iterations=1000, points=""):
    """
    Plot the Dynamical Plane set for the function Qc(z) = z^2 + c.

    Parameters:
        c (complex): The parameter c in Qc(z) = z^2 + c * phi
        N (int): Grid size for the complex plane.
        max_iterations (int): Maximum number of iterations.
        points (list): Arbitraty points near the indifferent fixed point

    Returns:
        None
    """
    xmin, xmax, ymin, ymax = -2, 2, -2, 2
    x = np.linspace(xmin, xmax, N)
    y = np.linspace(ymin, ymax, N)
    X, Y = np.meshgrid(x, y)
    Z = X + 1j * Y

    dynamical_plane = np.zeros((N, N))
    for l in range(N):
        for m in range(N):
            z = Z[l, m]
            dynamical_plane[l, m] = iterate_z(z, c, max_iterations)

    plt.imshow(dynamical_plane.T, cmap='hot', extent=[xmin, xmax, ymin, ymax])
    plt.colorbar()
    plt.xlabel('Real')
    plt.ylabel('Imaginary')

    for point in points:
        orbit = [point]
```

```python
        for _ in range(max_iterations):
            z = orbit[-1]
            next_z = z**2 + c
            orbit.append(next_z)
            if abs(next_z) > 10:
                break
        plt.plot(np.real(orbit), np.imag(orbit), 'w-', linewidth=1)

    plt.savefig(f'Quadratic/siegel_{"2" if points else "1"}.png')
    plt.close()

N = 500
points = [0.5 + 0.1j, 0.51 + 0.1j, 0.49 + 0.11j, 0.5 + 0.09j, 0.52 + 0.1j]
plot_dynamical_plane(c_phi, N, points=points)
```

Listing 5: Plot the Dynamical Plane of $Q_c(z) = z^2 + c\varphi$ with optional points param.

References

[1] Dr. Kuennen, Eric.
University of Winsconsin-Stout.
GRAPHICAL ANALYSIS, AND ATTRACTING AND REPELLING FIXED POINTS.
http://www.uwosh.edu/faculty_staff/kuennene/Chaos/ChaosNotes2.pdf.

A Figures

A.1 Dynamical Plane

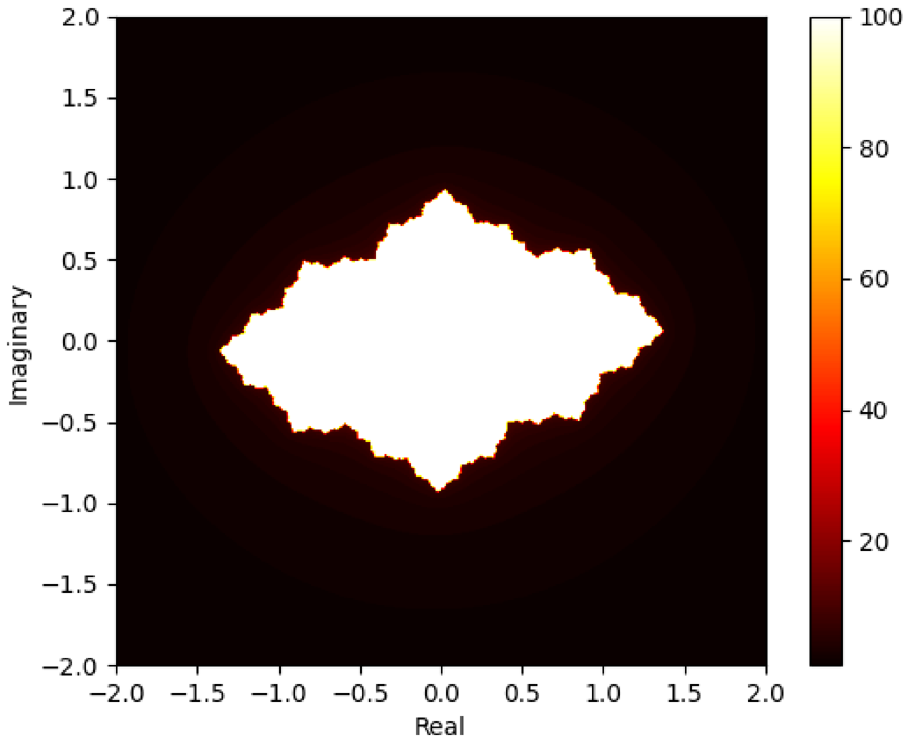

Figure 3: Dynamical plane of c_0.

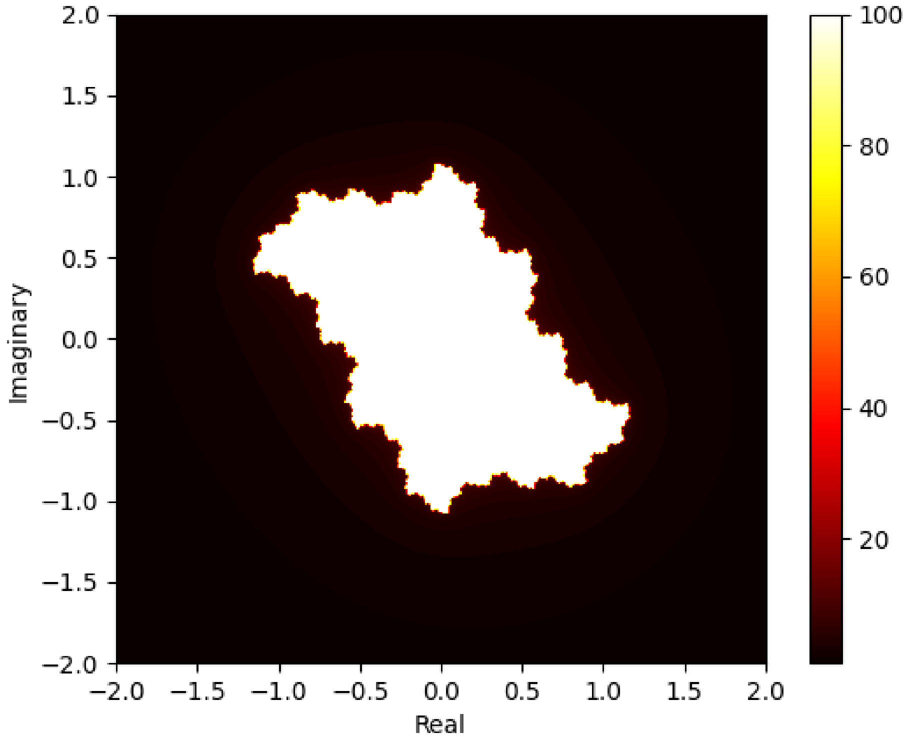

Figure 4: Dynamical plane of c_1.

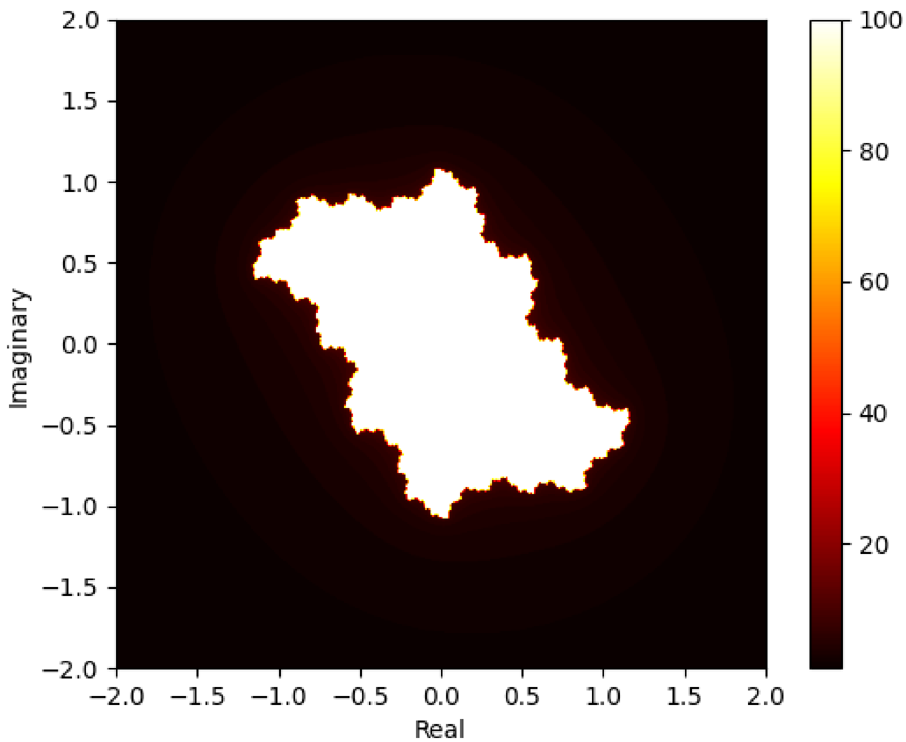

Figure 5: Dynamical plane of c_2.

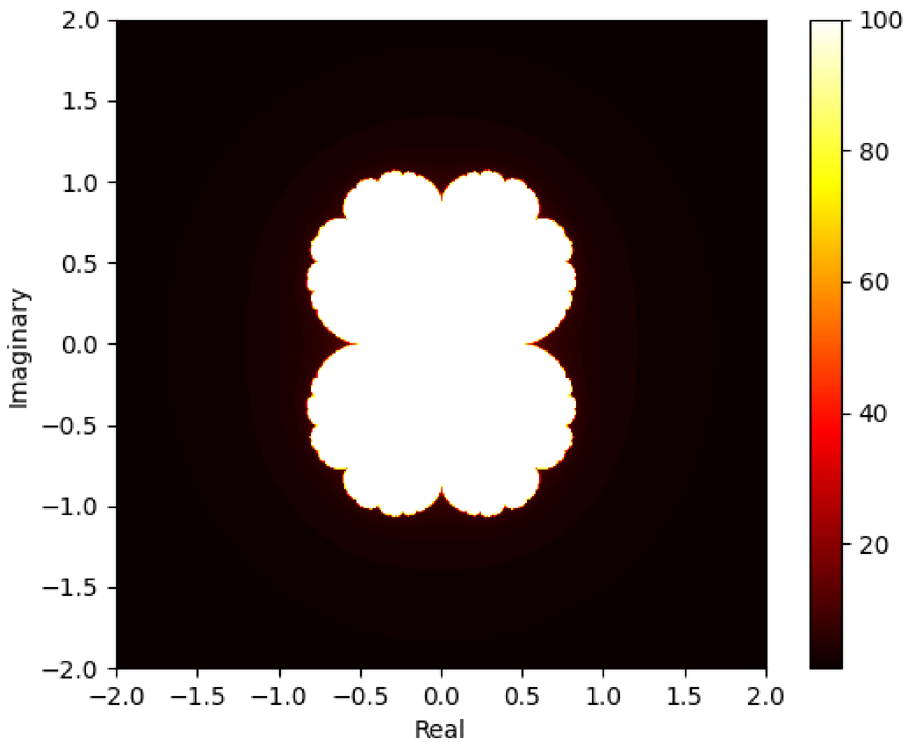

Figure 6: Dynamical plane of c_3.

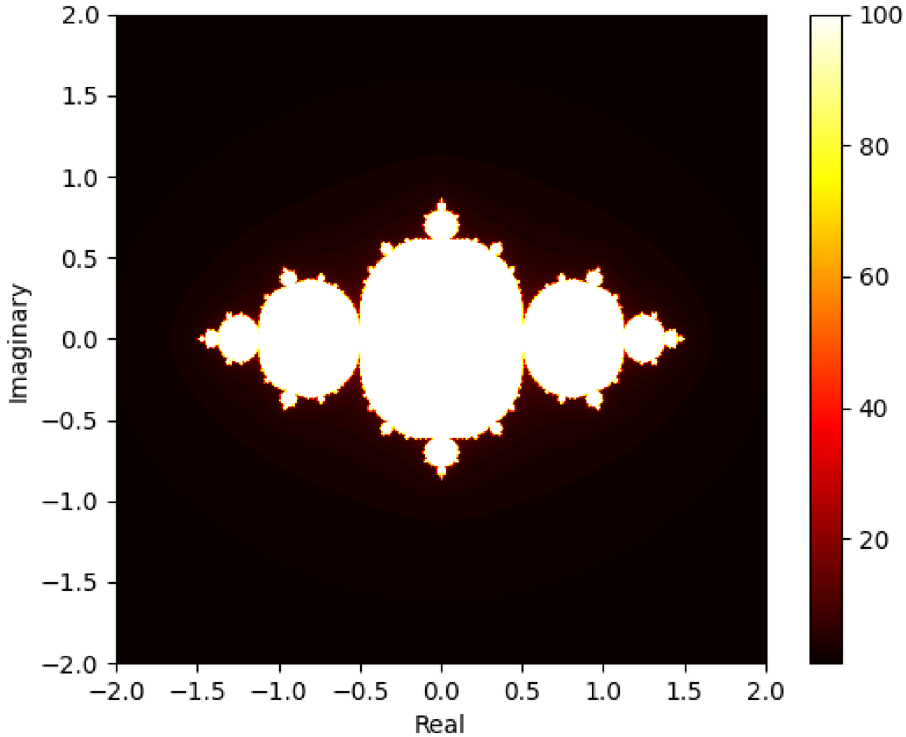

Figure 7: Dynamical plane of c_4.

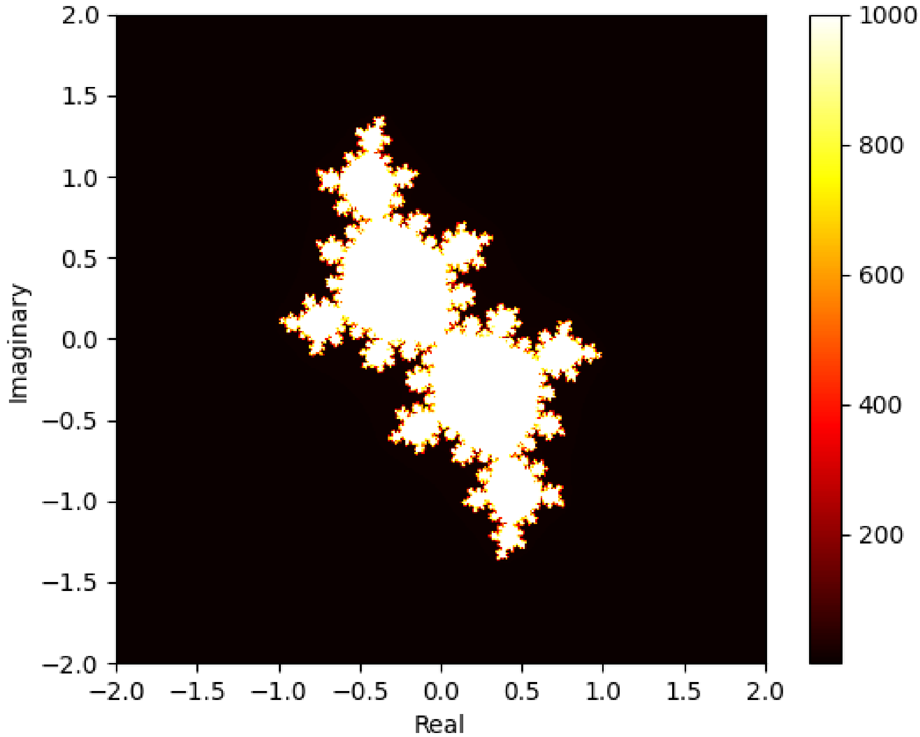

Figure 8: Dynamical plane of $Q_c(z) = z^2 + c_\varphi$.

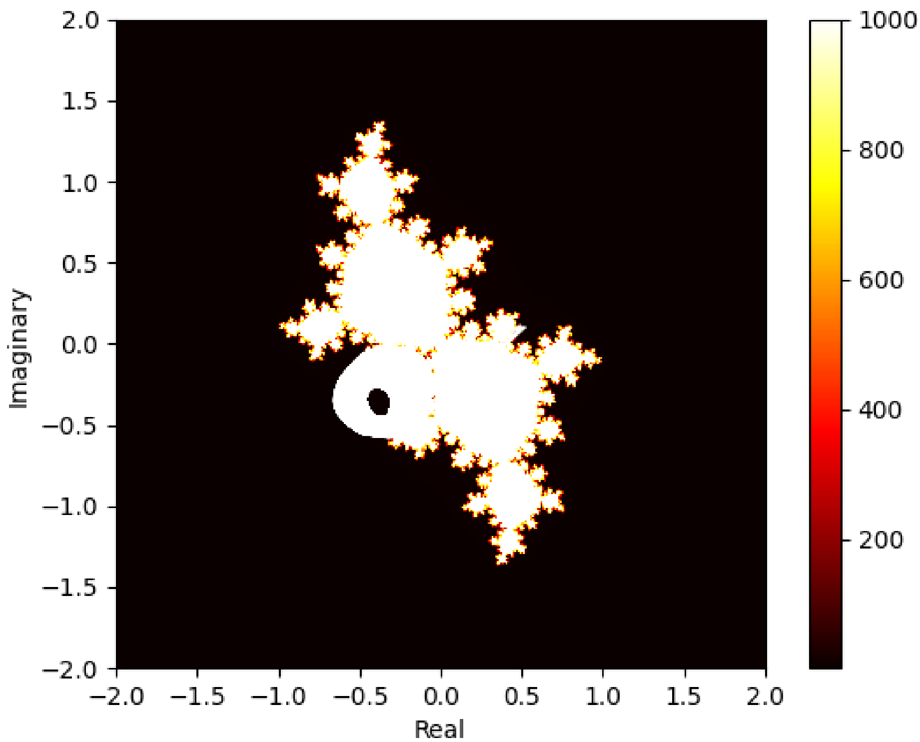

Figure 9: Dynamical plane of $Q_c(z) = z^2 + c_\varphi$ with several points near the indifferent fixed point.