Julia Sets of Complex Polynomials and Their Implementation on the Computer

Diplomarbeit

zur Erlangung des akademischen Grades eines

Diplom-Ingenieurs

in der Studienrichtung

Technische Mathematik

Eingereicht von

Christoph Martin Stroh

Eingereicht bei

Ass.-Prof. Univ.-Doz. Dipl-Ing. Dr. Franz Peherstorfer

Angefertigt am

Institut für Mathematik

der Technisch-Naturwissenschaftlichen Fakultät der

Johannes Kepler Universität Linz

Linz, März 1997

Contents

Chapter 1

Basic Concepts

1.1 Introduction

In the last few decades, especially with the advent of computers powerful enough to visualize their beauty, the concepts of chaos, fractals and dynamics have become quite popular. There are many books and articles explaining the mathematical background of the various fields in this area. The books by Gulick ([Gul]) and Holmgren ([Hol]) give a good introduction to discrete dynamical systems. They cover the real case in detail and also give a brief introduction to the complex case. The latter is explained in more detail in the book of Devaney ([De1]), but the discussion of Julia sets is still restricted to polynomials. One of the first papers on the iteration of rational functions was the paper by Brolin ([Bro]). The books by Carleson/Gamelin ([CG]) and by Beardon ([Bea]) provide a complete discussion of the iteration of complex rational functions. A good introduction and overview of the theory of rational iteration can also be found in the thesis of Inninger ([Inn]). Papers on a variety of topics concerning complex dynamical systems can be found in the Proceedings of Symposia in Applied Mathematics ([DK] and [De3]).

In this thesis we will only concentrate on one topic — Julia sets of complex rational functions, and primarily on Julia sets of complex polynomials. In the first chapter we give an introduction to the concept of iteration and provide the first results concerning the Julia set. It is mainly based on a lecture given by Peherstorfer in 1996 in Linz and on [De1].

In Chapter 2 we will discuss the dynamics on the components of the filled Julia set. This chapter is mainly based on [CG] with some ideas taken from [Bl1].

Chapter 3 deals with a very special topic — escape and converge radii. This topic is not addressed in the literature. We have developed some results that are useful when programming Julia sets on the computer.

The computer implementation itself and many ideas, hints, and tricks are covered in Chapter 4. The basic principles of the computer implementation can be found in [De2]. Further ideas are partly based on [PR] and [PS]; some ideas are new. An example of a real-world implementation is given in the Appendix.

Chapter 5 gives a palette of color images demonstrating the theory and the beauty of complex dynamics as well.

1.2 Preliminaries

1.2.1 Notation

In this thesis we will mostly deal with polynomials and rational functions. To that end we will use the following notation:

- P always denotes a nonlinear complex polynomial, i. e.,

$$P(z) = \sum_{j=0}^{d} a_j z^j \qquad \text{where } a_j \in \mathbb{C},\ a_d \neq 0,\ \deg P = d \geq 2.$$

  If the coefficients are real this will be stated explicitly.

- R always denotes a rational function, i. e., $R = P/Q$, where P and Q are complex polynomials. The degree of R is defined as

$$\deg R = \max\{\deg P, \deg Q\}.$$

  We will only consider nonlinear rational functions, i. e., $\deg R \geq 2$.

- The *iterates* of R are defined by

$$R^0(z) = z, \quad R^1(z) = R(z), \quad R^{k+1}(z) = R(R^k(z)), \quad k = 1, 2, \ldots$$

- The *orbit* of $z \in \bar{\mathbb{C}}$ (under iteration of R) is the set $\{R^n(z) \mid n \in \mathbb{N}_0\}$.

- The *backward iterate* is defined as $R^{-1}(z) = \{w \in \bar{\mathbb{C}} \mid R(w) = z\}$. Unless stated otherwise, $R^{-1}(z)$ always denotes *all* the inverse branches of $R(z)$ while $R^{-1}_{(\nu)}(z)$ denotes the ν^{th} branch. Note that R^{-1} is *not* a *function*, whereas the branches $R^{-1}_{(\nu)} : \bar{\mathbb{C}} \to \bar{\mathbb{C}}$ are functions.

Below we will give a list of conventions we will use throughout the thesis, and we will specify which definition for some mathematical terms we will use. The definitions are those most commonly found in the literature.

- Unless stated otherwise, $R^n(z) \to a$ means $R^n(z) \xrightarrow{n\to\infty} a$.

- A function $f : \bar{\mathbb{C}} \to \bar{\mathbb{C}}$ is called *univalent* if it is analytic and one-to-one.

- Let $(z_n)_{n\in\mathbb{N}}$ be a sequence of complex numbers. The *cluster set* of this sequence is defined as

$$\Big\{z \in \bar{\mathbb{C}} \;\Big|\; \bigwedge_{\varepsilon>0} \bigwedge_{n_0\in\mathbb{N}} \bigvee_{n\geq n_0} |z - z_n| \leq \varepsilon\Big\}.$$

  (To cover infinity, we have to use the chordal metric here. For details on this we refer to [Inn] or [LR, p. 6].)

- A *neighborhood* of a point z is an open, simply connected set containing z.

- A *domain* is an open and connected set.

- A connected subset C of $A(\subset \bar{\mathbb{C}})$ is called a *component* of A if there is no connected subset B of A such that $C \subsetneqq B$.
- A set $A(\subset \bar{\mathbb{C}})$ is called *totally disconnected* if each component C of A consists of exactly one point.
- A closed set consisting of accumulation points only is called a *perfect set*.
- If a set is perfect and totally disconnected we will call it a *Cantor set*.

In this thesis, especially in Section 2.3, we will use the symbol $\mathcal{O}$. Let V be a topological space, X a Banach space with norm $\|.\|$, $f, g : W \subset V \to X$ and $\tilde{z} \in W$. We say that $f(z) = \mathcal{O}(g(z))$ (for $z \to \tilde{z}$) if there is a neighborhood U of $\tilde{z}$ in W such that

$$\bigvee_{c \in \mathbb{R}} \bigwedge_{z \in U \setminus \{\tilde{z}\}} \|f(z)\| \leq c\|g(z)\|.$$

Mostly, X will be $\bar{\mathbb{C}}$ or $\mathbb{C}$, and V will be $\bar{\mathbb{C}}$, $\mathbb{C}$ or $\mathbb{N}$. The point $\tilde{z}$ will be (in most cases) either ∞ or the origin. We will not specify $\tilde{z}$ since it will be clear from the context which point is considered.

1.2.2 Definitions

Definition 1.1. We say that $\omega \in \bar{\mathbb{C}}$ is a *successor* of order n of z with respect to R if there is an $n \in \mathbb{N}$ such that $\omega = R^n(z)$. z is called a *predecessor* of order n of ω.

Definition 1.2. If $R(\alpha) = \alpha$, $\alpha \in \bar{\mathbb{C}}$, we call α a *fixed point* of R. If $R^n(\alpha) = \alpha$ and $R^p(\alpha) \neq \alpha$ for $p = 1, 2, \ldots, n-1$ we say that α is a *period-n point* of R or a *periodic point* with period n. The set $\{\alpha, R(\alpha), \ldots, R^{n-1}(\alpha)\}$ is called *cycle* of order n. If $\alpha \in \mathbb{C}$ is a period-n point of R the derivative $(R^n)'(\alpha)$ is called the *multiplier* of α. If $\alpha = \infty$ then the multiplier of α is defined by $\left(\frac{1}{R(1/z)}\right)'(0)$. A point $z \in \bar{\mathbb{C}}$ is called *preperiodic* with respect to R if there is a $j \in \mathbb{N}$ such that $R^j(z)$ is a periodic point of R. A preperiodic point z that is not periodic is called *strictly preperiodic*.

Remark. A period-1 point is, by definition, a fixed point. If α is a period-n point then all its successors are period-n points too. All elements in a cycle have the same multiplier, since

$$\begin{aligned}
(R^n)'(R^p(\alpha)) &= \prod_{k=0}^{n-1} R'(R^k(R^p(\alpha))) \\
&= \prod_{k=0}^{n-1} R'(R^{k+p}(\alpha)) = \prod_{j=p}^{n+p-1} R'(R^j(\alpha)) \\
&= \Big(\prod_{j=p}^{n-1} R'(R^j(\alpha))\Big) \cdot \Big(\prod_{j=n}^{n+p-1} R'(R^j(\alpha))\Big) \\
&= \Big(\prod_{k=p}^{n-1} R'(R^k(\alpha))\Big) \cdot \Big(\prod_{k=0}^{p-1} R'(\underbrace{R^{k+n}(\alpha)}_{=R^k(\alpha)})\Big) \\
&= \prod_{k=0}^{n-1} R'(R^k(\alpha)) = (R^n)'(\alpha),
\end{aligned}$$

where we have used the chain-rule for the first and the last identity.

Definition 1.3. A period-n point $\alpha \in \mathbb{C}$ is called *attracting* (or *attractive*), *indifferent* (or *neutral*), or *repelling* (or *repulsive*) if $|(R^n)'(\alpha)| < 1$, $= 1$, or > 1, respectively. If $(R^n)'(\alpha) = e^{2\pi i \cdot p/q}$, $p, q \in \mathbb{N}$, we say that α is *rationally indifferent.* If $|(R^n)'(\alpha)| = 0$ we say that α is *superattracting.* A cycle of order n is called *(super)attracting* (*indifferent, repelling*) if the corresponding period-n point is (super)attracting (indifferent, repelling).

Lemma 1.4. *Let $\alpha \neq \infty$ be an attracting fixed point of R. Then there exists a neighborhood U of α such that $R^n(z) \to \alpha$ for all $z \in U$.*

Proof. α is an attracting fixed point, i. e., $|R'(\alpha)| = \lambda < 1$. Hence, there exists a $\rho < 1$ such that $|R'(\alpha)| < \rho < 1$ (e. g. choose $\rho = (\lambda + 1)/2$). By definition of the derivative this is equivalent to

$$\lim_{z \to \alpha} \frac{|R(z) - R(\alpha)|}{|z - \alpha|} < \rho.$$

Thus there exists a $\delta > 0$ such that for all z with $|z - \alpha| < \delta$ the inequality $|R(z) - R(\alpha)| = |R(z) - \alpha| < \rho|z - \alpha|$ holds. By induction it follows that

$$\bigwedge_{\substack{z \\ |z-\alpha|<\delta}} |R^n(z) - \alpha| < \rho^n |z - \alpha| \xrightarrow{n \to \infty} 0.$$

Setting $U = \{z \in \mathbb{C} \mid |z - \alpha| < \delta\}$ the assertion is proven. ☐

Definition 1.5. A set E is said to be *invariant* under R if $R(E) = E$, and *completely invariant* if $R^{-1}(E) = E = R(E)$.

Definition 1.6. Let $D \subset \bar{\mathbb{C}}$ be a domain, and for each $n \in \mathbb{N}$ let $f_n : D \to \bar{\mathbb{C}}$ be a meromorphic function. Then $\{f_n\}_{n \in \mathbb{N}}$ is called a *normal family* on D if every infinite subset of $\{f_n\}$ contains a subsequence which converges uniformly on every compact subset of D or converges uniformly to ∞ on D. Note that the uniform convergence to ∞ of a family of functions $\{g_n\}_{n \in \mathbb{N}}$ on a compact set D means that

$$\bigwedge_{M>0} \bigvee_{n_0} \bigwedge_{n>n_0} \bigwedge_{z \in D} |g_n(z)| > M.$$

A family $\{f_n\}_{n \in \mathbb{N}}$ is *normal* at z if there exists a domain U with $z \in U$ such that $\{f_n\}_{n \in \mathbb{N}}$ is normal in U.

Sequences of analytic functions have a very important property:

Theorem 1.7. *Suppose that f_n is a sequence of analytic functions which converges compactly on a domain U to a map f. Then f is analytic in U and*

$$\bigwedge_{k \in \mathbb{N}_0} \lim_{n \to \infty} f_n^{(k)}(z) = f^{(k)}(z),$$

where $f^{(k)}$ denotes the k^{th} derivative of f.

Remark. This theorem is sometimes referred to as Theorem of Weierstraß (see e. g. [Peh] or [Rem, p. 195]).

Definition 1.8. 1. Let α be an attracting fixed point of R. The *basin of attraction* or *attractive set* $A(\alpha)$ of α is defined by

$$A(\alpha) = \{z \in \bar{\mathbb{C}} \,|\, R^n(z) \to \alpha\}.$$

The *immediate basin of attraction* or *immediate attractive set* $A^*(\alpha)$ of α is the component of $A(\alpha)$ that contains α.

2. Let $\{\alpha_k\} = (\alpha_0, \alpha_1, \ldots, \alpha_{m-1})$ be an attracting cycle of order m. Then the *basin of attraction* or *attractive set* $A(\{\alpha_k\})$ of the cycle is defined by

$$A(\{\alpha_k\}) = \{z \in \bar{\mathbb{C}} \mid \{a_k\} \text{ is the cluster set of } \{R^n(z)\}\}.$$

The *immediate basin of attraction* or *immediate attractive set* $A^*(\{\alpha_k\})$ of the cycle is defined by

$$A^*(\{\alpha_k\}) = \bigcup_{k=0}^{m-1} A_m^*(\alpha_k),$$

where $A_m^*(\alpha_k)$ is the component of $A(\{\alpha_k\})$ containing α_k.

1.2.3 Conjugation

A powerful tool for proving theorems or showing the dynamics of a function is the method of conjugating a function to another one:

Definition 1.9. A function $f : U(\subset \bar{\mathbb{C}}) \to U$ is *(conformally) conjugate* to a function $g : V \to V$ if there is a conformal map $\varphi : U \to V$ such that $g = \varphi \circ f \circ \varphi^{-1}$, i. e.,

$$\varphi(f(z)) = g(\varphi(z)).$$

Recall that a map $\varphi : U \to V$ (where U and V are domains) is called *conformal*, if it is analytic, one-to-one, and onto.

The two maps f and g can be regarded as the same map viewed in different coordinate systems. From the definition it is also clear that $g^n = \varphi \circ f^n \circ \varphi^{-1}$ and that (if the inverse functions are defined) $g^{-1} = \varphi \circ f^{-1} \circ \varphi^{-1}$. Moreover, if z is a fixed point of f then $\varphi(z)$ is a fixed point of g; furthermore, for a fixed point z of f

$$f'(z) = \underbrace{(\varphi^{-1})'\Big(\underbrace{g(\varphi(z))}_{\varphi(z)}\Big)}_{\frac{1}{\varphi'(z)}} g'(\varphi(z))\varphi'(z) = g'(\varphi(z)),$$

so that the multipliers of the corresponding fixed points are the same. Hence the dynamics of the periodic points is preserved under the conjugation.

Example 1.10. One of the most widely used conjugations is the conjugation of a quadratic polynomial to another quadratic polynomial of the standard form $P_c(z) = z^2+c$ by a linear conjugating function φ; so we want to conjugate $P(z) = a_0+a_1z+a_2z^2$ to $P_c(z) = z^2 + c$ by $\varphi(z) = uz + v$, i. e.,

$$\varphi \circ P = P_c \circ \varphi. \tag{1.1}$$

If we evaluate (1.1) and compare coefficients we obtain

$$c = \frac{2a_1 - a_1^2 + 4a_0a_2}{4}, \quad u = a_2, \quad v = \frac{a_1}{2}.$$

This conjugation is the reason why in almost every book on complex dynamics or on Julia sets the investigation of quadratic polynomials is restricted to the investigation of the family of polynomials of the form $P_c(z) = z^2 + c$. Moreover, these polynomials form the basis for the theory of the famous Mandelbrot set. $\diamondsuit$

Remark. The principle of conjugation plays an important role in the investigation of the dynamics of one-dimensional maps (see e. g. [Gul, Chapter 2.3]) and even more in the investigation of the dynamics of (systems of) differential equations. A famous theorem in this direction is the Theorem of Hartman-Grobman (see [Per, p. 119]).

1.3 First Results on the Julia Set

Definition 1.11. The *Julia set* $\mathcal{J}_R$ of a rational function R is defined as the set of points at which the family $\{R^n\}$ is not normal. The complement of the Julia set is called the *Fatou set* $\mathcal{F}_R$, i. e., $\mathcal{F}_R = \bar{\mathbb{C}} \setminus \mathcal{J}_R$. If it is evident which rational function R or polynomial P is considered we will only write $\mathcal{J}$ ($\mathcal{F}$) instead of $\mathcal{J}_R$ or $\mathcal{J}_P$ ($\mathcal{F}_R$ or $\mathcal{F}_P$).

Remark. Lemma 1.4 shows that every attracting periodic point lies in $\mathcal{F}$.

In the remaining part of this chapter we will only consider complex polynomials; so $\mathcal{J} = \mathcal{J}_P$. For polynomials infinity is always a superattracting fixed point and hence in $\mathcal{F}$. This can be seen as follows: It is clear that ∞ is a fixed point of every polynomial. Moreover, it is also obvious that $|(P^n(z))| \to \infty$ if $|z|$ is sufficiently large. Using Definition 1.2 we can calculate the multiplier:

$$\begin{aligned}
P'(\infty) &= \left(\frac{1}{P(\frac{1}{z})}\right)'\Bigg|_{z=0} = -\frac{1}{P(\frac{1}{z})^2} \cdot P'\left(\frac{1}{z}\right) \cdot \left(-\frac{1}{z^2}\right)\Bigg|_{z=0} \\
&= \frac{1}{\left(\sum_{j=0}^d a_j \left(\frac{1}{z}\right)^j\right)^2} \left(\sum_{j=1}^d j a_j \left(\frac{1}{z}\right)^{j-1}\right) \cdot \left(\frac{1}{z}\right)^2\Bigg|_{z=0} \\
&= \frac{d a_d \left(\frac{1}{z}\right)^{d+1} + \mathcal{O}\left(\left(\frac{1}{z}\right)^d\right)}{a_d^2 \left(\frac{1}{z}\right)^{2d} + \mathcal{O}\left(\left(\frac{1}{z}\right)^{2d-1}\right)}\Bigg|_{z=0} = \frac{d a_d z^{2d-(d+1)} + \mathcal{O}(z^d)}{a_d + \mathcal{O}(z)}\Bigg|_{z=0} = 0.
\end{aligned}$$

The last identity is true because we only consider polynomials with $d \geq 2$. Thus ∞ is really a superattracting fixed point and hence in $\mathcal{F}$. Henceforth, with this in mind, we will consider $\mathbb{C}$ instead of $\bar{\mathbb{C}}$.

For polynomials we can define the filled Julia set:

Definition 1.12. The *filled Julia set* (or *filled-in Julia set*) $\mathcal{K}_P$ of a polynomial P is the complement of the basin of attraction of infinity, i. e.,

$$\mathcal{K}_P = \{z \in \mathbb{C} \mid |P^n(z)| \not\to \infty\}.$$

Again we will write $\mathcal{K}$ instead of $\mathcal{K}_P$ if it is obvious which polynomial is considered.

Let us return to the Julia set itself. We now present the fundamental properties of $\mathcal{J}$.

Lemma 1.13. *The set $\mathcal{J}$ is compact.*

Proof. Since the Fatou set $\mathcal{F}$ is the set of points where $\{P^n\}$ is normal $\mathcal{F}$ is by Definition 1.6 open. Since $\mathcal{J}$ is the complement of $\mathcal{F}$, $\mathcal{J}$ is closed. So we have to show that $\mathcal{J}$ is bounded. Obviously, for every polynomial there is a radius $r > 0$ such that for every z with $|z| > r$ $P^n(z) \to \infty$. Hence $\{P^n\}$ is normal at z and thus $z \in \mathcal{F}$. (For a detailed discussion of such an r see Section 3.1). $\square$

Remark. "$\mathcal{J}$ is bounded" is one of the few results that do not hold for rational maps. In fact, there are rational maps for which $\mathcal{J} = \bar{\mathbb{C}}$. Examples can be found in [CG] or [Bl1], among others.

Theorem 1.14. $\mathcal{J} \neq \emptyset$.

Proof. Let us assume that $\mathcal{J} = \emptyset$. Then $\{P^n\}$ is normal on every domain U. Therefore, we can choose an $r > 0$ sufficiently large such that there exist $z, w \in B(0, r)$ with $|P^n(z)| \to \infty$ and $P(w) = w$. But this is a contradiction to our assumption that $\{P^n\}$ is normal on $B(0, r)$ since a subsequence of $\{P^n\}$ cannot, at the same time, converge on $B(0, r)$ to an analytic function (because of $P^n(w) = w$) and to infinity. $\square$

Theorem 1.15. *$\mathcal{J}$ is completely invariant under P.*

Proof. As $\mathcal{F} = \mathbb{C} \setminus \mathcal{J}$ it is sufficient to show that $\mathcal{F}$ is completely invariant under P.

Let $z \in \mathcal{F}$ and let U be a neighborhood of z such that $\{P^n\}$ is normal on U. As P is continuous $P(U)$ and $P^{-1}(U)$ are also open sets. Hence, if a subsequence $\{P^{n_i}\}$ converges uniformly on every compact subset of U then $\{P^{n_i - 1}\}$ converges uniformly on every compact subset of $P(U)$ and $\{P^{n_i+1}\}$ converges uniformly on every compact subset of $P^{-1}(U)$. So, if $\{P^n\}$ is normal at z then it is also normal at $P(z)$ and $P^{-1}(z)$. Thus, $P(\mathcal{F}) \subset \mathcal{F}$ and $P^{-1}(\mathcal{F}) \subset \mathcal{F}$. Using the fact that $P^{-1}(P(\mathcal{F})) \supset \mathcal{F}$ one finds $P^{-1}(\mathcal{F}) = \mathcal{F} = P(\mathcal{F})$. $\square$

Theorem 1.16. *The Julia sets for P and for P^m ($m \in \mathbb{N}$) are the same.*

Proof. Let $S = P^m$. Since $\{S^n\}$ is a subset of $\{P^n\}$ it is clear that if $\{P^n\}$ is normal at z then also $\{S^n\}$ is normal at z. Thus we have proven

$$\mathcal{F}_P \subset \mathcal{F}_S. \tag{1.2}$$

Next we suppose that $\{S^n\}$ is normal at z. Then the families $\{P^k S^n\}_{n\in\mathbb{N}} = \{P^{nm+k}\}_{n\in\mathbb{N}}$ for $k = 0, 1, \dots, m-1$ are also normal in z. Now we take any infinite subset $\{P^{n_i}\}_{i\in\mathbb{N}}$ of $\{P^n\}_{n\in\mathbb{N}}$. Then there must be a $k \in \{0, 1, \dots, m-1\}$ such that there is a subsequence of $\{P^{n_i}\}_{i\in\mathbb{N}}$ which is contained in $\{P^k S^n\}_{n\in\mathbb{N}}$; but we know that $\{P^k S^n\}_{n\in\mathbb{N}}$ is normal at z. Hence, if $\{S^n\}$ is normal at z then also $\{P^n\}$ is normal at z and thus $\mathcal{F}_S \subset \mathcal{F}_P$ which in combination with (1.2) yields $\mathcal{F}_S = \mathcal{F}_P$. Since the Fatou set $\mathcal{F}$ is the complement of the Julia set this proves the assertion. $\square$

To investigate more properties of the Julia set we need some knowledge about normal families. If a family of functions fails to be normal at a point z then in any neighborhood of z it takes on almost every value. More precisely, we have by the famous theorem of Montel:

Theorem 1.17 (Montel). *Let $\{f_n\}$ be a family of analytic functions defined on a domain $U \subset \mathbb{C}$. Suppose there exist $a, b \in \mathbb{C}$, $a \neq b$ such that $f_n(z) \neq a$ and $f_n(z) \neq b$ for all $n \in \mathbb{N}$ and for every $z \in U$. Then $\{f_n\}$ is a normal family on U.*

This theorem has important consequences for our investigations:

Corollary 1.18. *Let $w \in \mathcal{J}$ and let U be a neighborhood of w. Then*

$$\bigcup_{n=0}^{\infty} P^n(U) = \mathbb{C} \setminus E,$$

where $E = \{a\}$ or $E = \emptyset$. If $E = \{a\}$ then

$$P(z) = a + \lambda(z-a)^k$$

for some $\lambda \in \mathbb{C}$ and some integer k. Moreover, $a \in \mathcal{F}$, and a does not depend on w or U.

Remark. The set E is called *exceptional set*, the point a is called *exceptional point.* For rational maps the exceptional set E can consist of two elements, since we have to deal with $\bar{\mathbb{C}}$ instead of $\mathbb{C}$, and infinity is treated as an element of the extended complex plane like any other point. This is reflected in the original version of Montel's Theorem: *Let $\mathcal{A}$ be a family of meromorphic functions on a domain D. If there are three fixed values that are omitted by every $f \in \mathcal{A}$, then $\mathcal{A}$ is a normal family.* A proof of this theorem can be found in [CG, p. 10].

Proof of Corollary 1.18. The first part is a direct consequence of Theorem 1.17. Now let $E = \{a\}$. Suppose $P(b) = a$. Then b is an exceptional point for P, because otherwise there would be a z in U which is mapped to b and then to a, which is a contradiction to $a \in E$. Now $|E| = 1$ implies that $a = b$. Hence, $P(a) = a = P^{-1}(a)$. Since P is a polynomial this yields

$$P(z) - a = \lambda(z-a)^k,$$

where $k = \deg P \geq 2$. If we calculate the derivative we obtain $P'(z) = k \cdot \lambda(z-1)^{k-1}$ and thus $P'(a) = 0$. Therefore, a is a superattracting fixed point. Lemma 1.4 shows that there exists a neighborhood of a on which $\{P^n\}$ converges uniformly to the constant function $f(z) \equiv a$; hence, $a \in \mathcal{F}$. Obviously a does not depend on w or U. □

Example 1.19. The easiest example for a polynomial with a nonempty exceptional set is the polynomial $P(z) = z^2$. The Julia set is the unit circle, and if we iterate a small domain meeting $\mathcal{J}$ (i. e., the unit circle) we reach every point in $\mathbb{C}$ except the origin.

In this example the Julia set is quite simple — it is a circle. This is not only true for $P(z) = z^2$ but for all polynomials that have an exceptional point, or, in other words, (by Corollary 1.18) for all polynomials of the form $P(z) = a + \lambda(z-a)^d$ where $d = \deg P$. Indeed, let μ be any $(d-1)^{\text{st}}$ root of λ and define $h(z) = \mu(z-a)$. Then an easy calculation shows that $Q \circ h = h \circ P$ with $Q(z) = z^d$. Thus P is conjugated to Q. The Julia set of Q is the unit circle again, and $\mathcal{J}_P = h^{-1}(\mathcal{J}_Q)$ is also a circle. ◊

Theorem 1.20. *1. Let U be open, $\mathcal{J} \cap U \neq \emptyset$ and $z \notin E$. Then there exists a sequence (n_j), $n_1 < n_2 < \ldots$, $n_j \in \mathbb{N}$ such that $P^{-n_j}(z) \cap U \neq \emptyset$, i. e., every point in $\mathcal{J}$ is an accumulation point of the backward iterates of z.*

2. *If $z \in \mathcal{J}$ then*

$$\mathcal{J} = \overline{\bigcup_{n=0}^{\infty} P^{-n}(z)}.$$

Proof. 1. Let $w \in \mathcal{J} \cap U \neq \emptyset$ and $U_1 = U$. We will show that w is an accumulation point of the backward iterates of z. By Corollary 1.18 there exists a $z_1 \in U_1$ and an n_1 such that $P^{n_1}(z_1) = z$. Inductively we choose $U_{j+1} \subset U_j$ such that $w \in U_{j+1}$ and $z_j \notin U_{j+1}$. Again there must be a $z_{j+1} \in U_{j+1}$ and an n_{j+1} such that $P^{n_{j+1}}(z_1) = z$. With this sequence (n_j) the first part of the theorem holds.

2. Using 1. we know that $\mathcal{J} \subset \overline{\bigcup_{n=0}^{\infty} P^{-n}(z)}$. On the other hand we know by Theorem 1.15 that $\mathcal{J}$ is completely invariant; from this we obtain that $\bigcup_{n=0}^{\infty} P^{-n}(z) \subset \mathcal{J}$, and by Lemma 1.13 we deduce that $\mathcal{J}$ is closed. Hence $\overline{\bigcup_{n=0}^{\infty} P^{-n}(z)} \subset \mathcal{J}$, which completes the proof.

□

Theorem 1.21. $\overset{\circ}{\mathcal{J}} = \emptyset$.

Proof. Suppose there is an open set $U \subset \mathcal{J}$. As $\mathcal{J}$ is completely invariant $\mathcal{J} \supset P^n(U)$ for every $n \in \mathbb{N}$. Hence, by Corollary 1.18,

$$\mathcal{J} \supset \bigcup_{n=0}^{\infty} P^n(U) = \mathbb{C} \setminus E.$$

Since $\mathcal{J}$ is closed $\mathcal{J} = \mathbb{C}$ which is a contradiction to Lemma 1.13 which says that $\mathcal{J}$ is bounded. □

Remark. As mentioned earlier, $\mathcal{J}$ is not necessarily bounded for rational maps. But the proof shows that if $\overset{\circ}{\mathcal{J}} \neq \emptyset$ then $\mathcal{J} = \bar{\mathbb{C}}$.

Theorem 1.22. *$\mathcal{J}$ is a perfect set, i. e., it is closed and contains no isolated points.*

Proof. Since $\mathcal{J}$ is closed it remains to demonstrate that $\mathcal{J}$ contains no isolated points. Take $z_0 \in \mathcal{J}$ and let U be a neighborhood of z_0. We have to show that $(U \setminus \{z_0\}) \cap \mathcal{J} \neq \emptyset$.

First assume that z_0 is not periodic and choose z_1 with $P(z_1) = z_0$. Then $P^n(z_0) \neq z_1$ for every $n \in \mathbb{N}$. Since $\mathcal{J}$ is completely invariant, $z_1 \in \mathcal{J}$ and thus, by Theorem 1.20, the backward iterates of z_1 are dense in $\mathcal{J}$. Hence, there is a $\zeta \in U$ with $P^m(\zeta) = z_1$. Thus $\zeta \in \mathcal{J} \cap U$ and $\zeta \neq z_0$.

Next suppose z_0 is a periodic point. As $\mathcal{J}_P = \mathcal{J}_{P^n}$ we can assume that z_0 is a fixed point, i. e., $P(z_0) = z_0$. If z_0 were the only solution of $P(z) = z_0$ then $P(z) = z_0 + \lambda(z - z_0)^d$, $d \geq 2$, and thus z_0 would be a superattracting fixed point contradicting $z_0 \in \mathcal{J}$. Hence, there is a $z_1 \neq z_0$ with $P(z_1) = z_0$. Furthermore, $z_1 = P^{-1}_{(\nu)}(z_0) \in \mathcal{J}$. Again, by the denseness of the backward iterates, there is a point $P^{-n}_{(\mu)}(z_1) \in U$ for some $n \in \mathbb{N}$ where $P^{-n}_{(\mu)}(z_1) \neq z_0$ because otherwise $z_1 = P^n(P^{-n}_{(\mu)}(z_1)) = P^n(z_0) = z_0$. □

Theorem 1.23. *The set of repelling and rationally indifferent periodic points is a subset of $\mathcal{J}$.*

Proof. As $\mathcal{J}_P = \mathcal{J}_{P^m}$ for any $m \in \mathbb{N}$ it is sufficient to consider only repelling fixed points and fixed points with multiplier 1.

First let z_0 be a repelling fixed point. Suppose $\{P^n\}$ is normal on a neighborhood U of z_0. Then $P^n(z)$ does not converge to ∞ on U since $P^n(z_0) = z_0$ for every $n \in \mathbb{N}$. Thus $\{P^n\}$ has a subsequence $\{P^{n_i}\}$ which converges uniformly to an analytic function G on every compact subset of U. By Theorem 1.7 $\left|(P^{n_i})'(z_0)\right| \to \left|G'(z_0)\right|$. But as z_0 is repelling we know that

$$\left|(P^{n_i})'(z_0)\right| = \left|P'(z_0)\right|^{n_i} \to \infty.$$

This contradiction shows that $\{P^n\}$ is not normal at z_0.

Now let z_0 be a rationally indifferent fixed point. As mentioned above we can assume that the multiplier is 1 because if it were $\lambda = e^{2\pi i p/q}$ we could take P^q instead of P for which the fixed point had multiplier 1. Moreover, we can shift the fixed point to the origin by a linear conjugating function. Hence P assumes the form $P(z) = z + a_k z^k + \cdots$ where $a_k \neq 0$. Since $P^n(z) = z + n \cdot a_k z^k + \cdots$ we have $(P^n)^{(k)}(0) = n \cdot a_k \cdot k!$ and thus

$$\lim_{n\to\infty} \left|(P^n)^{(k)}(0)\right| = \infty.$$

As before, this shows that $\{P^n\}$ cannot be normal at $z_0 = 0$. □

Theorem 1.24. *$\mathcal{J}$ is contained in the closure of the set of periodic points of P.*

Proof. Let us consider any open set U which meets $\mathcal{J}$. We have to show that there is a periodic point in U. We choose a point $w \in \mathcal{J} \cap U$. As there are only finitely many critical values of P^2 we may assume that w is none of them. This, together with $\deg P \geq 2$, implies that there are at least four distinct points in $P^{-2}(w)$. Now we choose two of them, say w_1 and w_2 with $w_i \neq w$, and construct neighborhoods U_0, U_1 and U_2 (with pairwise disjoint closures) around w, w_1, w_2, respectively, such that $U_0 \subset U$ and that P^2 is a homeomorphism from each U_j $(j = 1, 2)$ onto U_0. Now let $S_j : U_0 \to U_j$ be the inverse branch of P^2 which maps U_0 onto U_j, $j = 1, 2$.

Suppose that

$$\bigwedge_{z\in U_0} \bigwedge_{j=1,2} \bigwedge_{n\in\mathbb{N}} \quad P^n(z) \neq S_j(z),$$

then, by Corollary 1.18, $\{P^n\}$ is normal in U_0 which contradicts the fact that U_0 meets $\mathcal{J}$ (since $w \in U_0$ and $w \in \mathcal{J}$). Hence, there is some $z \in U_0$, some $j \in \{1, 2\}$, and some $n \in \mathbb{N}$ such that $P^n(z) = S_j(z)$. This implies that

$$P^{n+2}(z) = P^2(S_j(z)) = z,$$

and so z is the required periodic point. □

To prove one of the main results of this section, namely Theorem 1.26, we need one more theorem.

Theorem 1.25. *A polynomial P of degree $d \geq 2$ has at most $3d - 3$ non-repelling periodic points.*

Proof. See section 2.3.1. □

Remark. The upper bound of $3d-3$ non-repelling cycles is sufficient for our purposes (basically we only need the fact that there are *finitely* many non-repelling periodic points), but it is not optimal. Shishikura showed the following:

A rational map R of degree d, $d \geq 2$, has at most $2d-2$ non-repelling cycles.

The proof of this theorem is long and complicated and can be found in [Bea, pp. 211–223]. For a polynomial ∞ is an attracting fixed point of order $d-1$. This suggests that for polynomials the following theorem holds:

A polynomial P of degree d has at most $d-1$ non-repelling cycles in the finite plane.

A proof of this sharp bound, that uses the fact that there are only finitely many non-repelling cycles, can be found in [CG, pp. 100–101].

In some books the Julia set is defined as the closure of the set of repelling periodic points. We are now able to show that this definition is equivalent to Definition 1.11.

Theorem 1.26. *$\mathcal{J} = cl\{z \in \mathbb{C} \mid z$ is a repelling periodic point for $P\}$.*

Proof. Let $A = cl\{z \in \mathbb{C} \mid z \text{ is a repelling periodic point for } P\}$. By Theorem 1.23 and since $\mathcal{J}$ is closed we know that $A \subset \mathcal{J}$. Since there are only finitely many non-repelling cycles for P and since $\mathcal{J}$ is a perfect set Theorem 1.24 shows that $\mathcal{J} = A$. ☐

Theorem 1.27. *Let v be an attracting periodic point of a polynomial P or let $v = \infty$. Then*

$$\partial A(v) = \mathcal{J}.$$

Proof. Again it is sufficient to consider a fixed point v of P.

First we will show that $\mathcal{J} \subset \partial A(v)$. If $z \in \mathcal{J}$ then by Theorem 1.15 $P^n(z) \in \mathcal{J}$ for all $n \in \mathbb{N}$. This implies that $P^n(z)$ cannot converge to v since $v \in \mathcal{F}$ (see the Remark following Definition 1.11). Thus $z \notin A(v)$. We only have to show that $z \in \overline{A(v)}$, i. e.,

$$\bigwedge_{\varepsilon>0} \quad U_\varepsilon(z) \cap A(v) \neq \emptyset. \tag{1.3}$$

Since $z \in \mathcal{J}$ Corollary 1.18 yields

$$\bigwedge_{\varepsilon>0} \bigvee_{n\in\mathbb{N}} \quad P^n(U_\varepsilon(z)) \cap A(v) \neq \emptyset,$$

of which (1.3) is a direct consequence.

Next, we will show that $\partial A(v) \subset \mathcal{J}$. Suppose there is a $z \in \partial A(v)$ with $z \notin \mathcal{J}$. Then there is an open neighborhood U of z such that $\{P^n\}$ is normal on U, i. e., there is a subsequence $\{P^{n_i}\}$ that converges uniformly either to an analytic function G or to infinity. Since $\{P^n\}$ converges uniformly to the point v on $U \cap A(v)$ it follows by the Identity Theorem for analytic functions that $G(z) \equiv v$ on U which is a contradiction to $z \in \partial A(v)$. ☐

Corollary 1.28. $\partial \mathcal{K} = \mathcal{J}$.

Proof. By Definition 1.12 $\mathcal{K} = \mathbb{C} \setminus A(\infty)$. Therefore, with Theorem 1.27,

$$\partial \mathcal{K} = \partial A(\infty) = \mathcal{J}.$$

☐

Another important consequence of Theorem 1.27 is presented in the following corollary.

Corollary 1.29. *$A(\infty)$ is connected.*

Proof. Suppose that $A(\infty)$ is not connected. Then there is a component U of $A(\infty)$ not equal to $A^*(\infty)$; moreover, $\overline{U}$ is bounded. By Theorem 1.27 $\partial U \subset \partial A(\infty) = \mathcal{J}$ and since, by Lemma 1.13, $\mathcal{J}$ is bounded and, by Theorem 1.15, $\mathcal{J}$ is invariant under P it follows that P^n is uniformly bounded on ∂U and thus is uniformly bounded on $\overline{U}$ by the maximum principle for analytic functions. But this is a contradiction to $P^n(z) \xrightarrow{n\to\infty} \infty$ for all $z \in U$. $\square$

At the end of this introductory section on Julia sets we present two more theorems that, in some cases, give a characterization of the Julia set. Proofs can be found in [CG, pp. 65–67] and in [Inn].

Theorem 1.30. *The Julia set $\mathcal{J}$ is connected if and only if there is no finite critical point of P in $A(\infty)$, i. e., if and only if the forward orbit of each critical point that is not equal to ∞ is bounded.*

Theorem 1.31. *If $|P^n(q)| \to \infty$ for each critical point q then the Julia set J is totally disconnected.*

Example 1.32. Consider the family of polynomials $P_\mu(z) = \mu z(1-z)$ with $\mu \in \mathbb{R}$, $\mu > 4$. The first three iterates for $\mu = 4.2$ for the real case are shown in Figure 1.1. Obviously, 0 is a repelling fixed point and hence (by Theorem 1.23) an element of the Julia set. Moreover, it can easily be seen that $P^{-1}(z) \subset [0,1]$ for any $z \in [0,1]$. This together with $\mathcal{J} = \overline{\bigcup_{n=0}^{\infty} P^{-n}(0)}$ (by Theorem 1.20) shows that $\mathcal{J} \subset [0,1]$. Moreover, since $\mathcal{J} = \partial\mathcal{K}$ because of Corollary 1.28 it is obvious that in this case $\mathcal{J} = \mathcal{K}$.

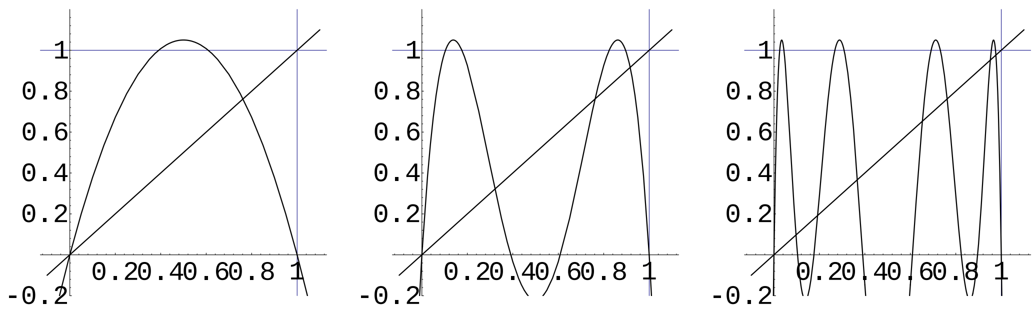

Figure 1.1: First three iterates of $P(z) = 4.2z(1-z)$.

If any iterate of a point $z \in \mathbb{R}$ is outside the interval $[0,1]$, graphical analysis shows that in this case $P^n(z) \to -\infty$. The only critical point of this polynomial is $q = 1/2$. As $P(1/2) \notin [0,1]$ we conclude that $P^n(q) \to -\infty$ from which we obtain by Theorem 1.31 that $\mathcal{J}$ is totally disconnected. Figure 1.1 gives a hint how this can happen: From what we have derived above we can see that

$$\mathcal{J} = \mathcal{K} = \{z \in [0,1] \mid \bigwedge_{n\in\mathbb{N}} P^n(z) \in [0,1]\}.$$

Figure 1.1 shows that there are two subintervals of $[0,1]$ on which $P(z)$ still remains inside $[0,1]$. Each subinterval again has two subintervals on which $P^2(z)$ remains inside $[0,1]$. Between those subintervals $P^n(z)$ escapes to $-\infty$. This process continues inductively and leaves a totally disconnected set for $\mathcal{J}$. ♢

Remark. The polynomial we used in Example 1.32 has a real Julia set. In [Inn] there is a complete characterization of polynomials with real Julia sets.

1.4 Further Results

Lemma 1.33. *Let $P(z)$ be a polynomial of degree $n \geq 2$ with distinct zeros $\zeta_1, \ldots, \zeta_n$. Then*

$$\sum_{i=1}^{n} \frac{1}{P'(\zeta_i)} = 0. \tag{1.4}$$

Proof. The proof is by induction. For $n = 2$ we have $P(z) = (z - \zeta_1)(z - \zeta_2)$, $P'(z) = 2z - (\zeta_1 + \zeta_2)$ and hence

$$\frac{1}{P'(\zeta_1)} + \frac{1}{P'(\zeta_2)} = \frac{1}{\zeta_1 - \zeta_2} + \frac{1}{\zeta_2 - \zeta_1} = 0.$$

For $n > 2$ let $P(z) = (z - \zeta_n)Q(z)$ with $Q(z) = \prod_{i=1}^{n-1}(z - \zeta_i)$. The partial fraction expansion

$$\frac{1}{Q(z)} = \sum_{i=1}^{n-1} \frac{1}{Q'(\zeta_i)(z - \zeta_i)},$$

yields

$$\frac{1}{Q(\zeta_n)} = \sum_{i=1}^{n-1} \frac{1}{Q'(\zeta_i)(\zeta_n - \zeta_i)}.$$

As $P'(z) = (z - \zeta_n)Q'(z) + Q(z)$ we have $P'(\zeta_n) = Q(\zeta_n)$ and $P'(\zeta_i) = (\zeta_i - \zeta_n)Q'(\zeta_i)$ for $i = 1, \ldots, n-1$. Hence,

$$\sum_{i=1}^{n} \frac{1}{P'(\zeta_i)} = \frac{1}{Q(\zeta_n)} + \sum_{i=1}^{n-1} \frac{1}{(\zeta_i - \zeta_n)Q'(\zeta_i)} = \frac{1}{Q(\zeta_n)} - \frac{1}{Q(\zeta_n)} = 0,$$

which proves the assertion. □

Theorem 1.34. *Let P be a polynomial of degree $n \geq 2$. Then either*

1. *P has a fixed point q with $P'(q) = 1$,*
2. *P has a fixed point q with $|P'(q)| > 1$.*

Proof. Let $\tilde{P}(z) = P(z) - z$. Then the roots of $\tilde{P}$ are the fixed points of P. If the roots of $\tilde{P}$ are not all distinct, then there exists a ζ with $\tilde{P}(\zeta) = 0$ and $\tilde{P}'(\zeta) = 0$ which implies that $P(\zeta) = \zeta$ and $P'(\zeta) = 1$. So for the following we can assume that the roots $\zeta_1, \ldots, \zeta_n$ of $\tilde{P}$ are all distinct.

By Lemma 1.33 we have

$$\sum_{i=1}^{n} \frac{1}{P'(\zeta_i) - 1} = \sum_{i=1}^{n} \frac{1}{\tilde{P}'(\zeta_i)} = 0.$$

Now suppose that $|P'(\zeta_i)| \leq 1$ and $P'(\zeta_i) \neq 1$ for $i = 1, \ldots, n$. Then $P'(\zeta_i) - 1$ lies in the circle $|z+1| \leq 1$ minus the origin. Hence $1/(P'(\zeta_i) - 1)$ is well defined and lies in the left half-plane $\{\Re(z) < 0\}$. On the other hand, we have

$$\sum_{i=1}^{n} \frac{1}{P'(\zeta_i) - 1} = 0,$$

which implies that at least one of the $P'(\zeta_i) - 1$ must lie in the region where $\Re(z) \geq 0$, which is the desired contradiction. $\square$

Remark. So far, we have only known by Theorem 1.14 that $\mathcal{J} \neq \emptyset$. Theorem 1.34 combined with the fact that $\mathcal{J}$ contains all repelling and rationally indifferent periodic points (see Theorem 1.23) tells us how to find, in a "simple way", at least one element from $\mathcal{J}$, which will be important for drawing the Julia set via an inverse iteration algorithm (see Section 4.2.2).

Chapter 2

On the Components of the Fatou Set

In this chapter we will focus on the components of the Fatou set $\mathcal{F}$ and on the behavior of a rational function R or a polynomial P of degree $d \geq 2$ on these components. We will deal with attracting regions, with parabolic regions, Siegel disks and Herman rings, and we will see how the iterates of R (or P) behave on such regions. Our main goal will be to give an insight in what can happen, such that one can get a feeling how the dynamics of R (or P) looks like on the components of $\mathcal{F}$. Hence, we will omit some (mostly technical) proofs. On the other hand we will prove all those results that are necessary to fill the gap we have left in our effort to prove Theorem 1.26, i. e., we will prove Theorem 1.25.

2.1 Sullivan's Theorem and the Classification Theorem

For the following it is important to realize that the image of a component of $\mathcal{F}$ is again a component of $\mathcal{F}$ and that the inverse image of a component of $\mathcal{F}$ is the disjoint union of at most d components of $\mathcal{F}$. This follows from the so-called Open Mapping Theorem in complex analysis (see e. g. [Rem, pp. 201–202]) since the components of $\mathcal{F}$ are open as $\mathcal{F}$ itself is open. In Definition 1.2 we have defined fixed points and (pre)periodic points. Let us give corresponding definitions for components:

Definition 2.1. Let U be a component of $\mathcal{F}$.

1. If $R(U) = U$, then U is called a *fixed component* of $\mathcal{F}$, sometimes also called a *(forward) invariant component* of $\mathcal{F}$.

2. If $R^n(U) = U$ for some $n \in \mathbb{N}$, then U is called a *periodic component* of $\mathcal{F}$. The minimal n for which $R^n(U) = U$ is the *period* of the component.

3. If $R^m(U)$ is periodic for some $m \geq 0$, then U is called a *preperiodic component* of $\mathcal{F}$. A preperiodic component that is not periodic is called *strictly preperiodic*.

4. If all iterates $\{R^n(U)\}$ are distinct, then U is called a *wandering domain*.

Remark. A periodic component with period 1 is, of course, a fixed component. As in Definition 1.5, if $R^{-1}(U) = U = R(U)$, then U is called *completely invariant*. Note that there are no other possibilities for a component than those given in Definition 2.1.

Example 2.2. A component U of a rational function R which contains a fixed point $\tilde{z}$ is a fixed component. This can be seen as follows: The image of any component of $\mathcal{F}$ is again a component of $\mathcal{F}$. Since different components of $\mathcal{F}$ are disjoint and $R(\tilde{z}) = \tilde{z}$ we conclude that $R(U) = U$. A preimage of such a U that is not U itself is a strictly preperiodic component since $R^1(U)$ is periodic (in this special case with period 1).

Examples concerning periodic components can be found in a similar way when dealing with periodic points. $\diamondsuit$

The following theorems give a first insight in what can happen with such components. The proofs can be found in [CG] starting at page 70.

Theorem 2.3. *If U is a completely invariant component of $\mathcal{F}_R$, then $\partial U = \mathcal{J}_R$, and every other component of $\mathcal{F}_R$ is simply connected. There are at most two completely invariant components of $\mathcal{F}_R$.*

Remark. For a polynomial P the basin of attraction of infinity $A(\infty)$ is completely invariant. Hence, all other components of $\mathcal{F}_P$ are simply connected.

Theorem 2.4. *The Fatou set $\mathcal{F}_R$ has one, two or infinitely many components, and all cases occur.*

Corollary 2.5. *If a polynomial P has at least two finite attracting periodic points then $\mathcal{F}_P$ has infinitely many components. Moreover if P has at least two finite attracting fixed points then* each *basin of attraction of a fixed point z_i consists of infinitely many components.*

Proof. Since $\mathcal{J}_P = \mathcal{J}_{P^m}$ we can assume that we have two finite fixed points z_1 and z_2. Then $\mathcal{F}_P$ contains $A(\infty)$, $A^*(z_1)$ and $A^*(z_2)$ as components, thus it has at least 3 components. Together with Theorem 2.4 this shows that there are infinitely many components for $\mathcal{F}_P$. Since by Theorem 1.27 $\partial\mathcal{F}_P = \mathcal{J}_P = \partial A(z_i)$ and $\mathcal{F}_P$ has infinitely many components we conclude that $A(z_i)$ has infinitely many components, too. $\square$

Theorem 2.6 (Sullivan). *A rational map has no wandering domains; in other words, every component of $\mathcal{F}_R$ is preperiodic.*

So we now know that there are no wandering domains. As a fixed component is a special case for a periodic component and preperiodic components are iterated to periodic ones, our next goal is to further characterize periodic components. First we need some definitions:

Definition 2.7. Let U be a periodic component of $\mathcal{F}_R$ with period n.

1. U is called *parabolic*, if there is on its boundary a neutral fixed point ζ for R^n with multiplier 1, such that all points from U converge to ζ under iteration by R^n, i. e.,

$$\bigwedge_{z\in U} R^{nm}(z) \xrightarrow{m\to\infty} \zeta.$$

   The domains $U, R(U), \ldots, R^{n-1}(U)$ form a so-called *parabolic cycle*, and their union is the immediate basin of attraction associated with an attracting petal at ζ (see Section 2.3.1).

2. U is called a *Siegel disk* if it is simply connected and R^n is conjugate to an irrational rotation on U, i. e.,

$$g = \varphi \circ R^n \circ \varphi^{-1} \quad \text{on } U,$$

where $g(z) = e^{2\pi i\theta} z$ with an irrational $\theta \in [0, 1]$ and φ is a conformal mapping.

3. U is called a *Herman ring* (or *Arnold ring*) if it is doubly connected and R^n is conjugate to either a rotation on an annulus or to a rotation followed by an inversion.

Siegel disks and Herman rings are referred to as *rotation domains*.

With these definitions we are ready to formulate the following important theorem (see [CG, pp. 74–79]):

Theorem 2.8 (Classification Theorem). *Let U be a periodic component of the Fatou set $\mathcal{F}_R$. Then one of the following statements is true:*

1. *U contains an attracting periodic point.*
2. *U is parabolic.*
3. *U is a Siegel disk.*
4. *U is a Herman ring.*

Remark. We will see later (in Section 2.3.2) that for polynomials Herman rings cannot occur.

2.2 Attracting Cycles

First we give a linearization theorem which is due to G. Koenigs (1884). It is valid not only for polynomials or rational functions, and so we present a general formulation.

Theorem 2.9. *Suppose an analytic function f has an attracting fixed point at z_0 with multiplier λ satisfying $0 < |\lambda| < 1$. Then there is a conformal map $\zeta = \varphi(z)$ of a neighborhood of z_0 onto a neighborhood of 0 which conjugates $f(z)$ to the linear function $g(\zeta) = \lambda\zeta$.*

Proof. For simplicity, we can assume that $z_0 = 0$. Now define $\varphi_n(z) = \lambda^{-n} f^n(z) = z + \cdots$. Then φ_n satisfies the equation

$$\varphi_n \circ f = \lambda^{-n} f^{n+1} = \lambda \varphi_{n+1}.$$

Thus, if we can show that φ_n converges to some function φ, this would yield $\varphi \circ f = \lambda\varphi$ or $\varphi \circ f \circ \varphi^{-1} = \lambda\zeta$ with the desired conjugation map φ.

To show convergence we first note that for a sufficiently small $\delta > 0$ there is a $C > 0$ such that

$$|f(z) - \lambda z| \le C|z|^2, \quad |z| \le \delta. \tag{2.1}$$

Using the second triangle inequality we get

$$|f(z)| \le |\lambda||z| + C|z|^2 \le (|\lambda| + C\delta)|z|,$$

and by induction

$$|f^n(z)| \le (|\lambda| + C\delta)^n |z|, \quad |z| \le \delta.$$

Now we choose δ so small that $\rho = (|\lambda| + C\delta)^2/|\lambda| < 1$ and observe that (2.1) holds if we replace z by $f^n(z)$ (since 0 is an attracting fixed point). So we obtain

$$|\varphi_{n-1}(z) - \varphi_n(z)| = \left| \frac{f^n(f(z)) - \lambda f^n(z)}{\lambda^{n+1}} \right| \le \frac{C|f^n(z)|^2}{|\lambda|^{n+1}} \le \frac{\rho^n C|z|^2}{|\lambda|}$$

for $|z| \le \delta$. Hence, $\varphi_n(z)$ converges uniformly for $|z| \le \delta$ and the conjugation exists. □

Remark. It can be shown that the conjugation function φ is unique, up to multiplication by a nonzero scale factor. The function φ constructed above is normalized, i. e., $\varphi'(z_0) = 1$.

Theorem 2.10. *If z_0 is an attracting periodic point of the polynomial P, then the immediate basin of attraction $A^*(z_0)$ contains at least one critical point. If the periodic point is attracting but not superattracting, then this critical point has infinite orbit.*

Proof. In the superattracting case there is nothing to prove.

First we assume that z_0 is an attracting fixed point with multiplier λ satisfying $0 < |\lambda| < 1$. Let $U_0 = U_\varepsilon(z_0)$ be a small, forward invariant (i. e., $P(U_0) \subset U_0$) neighborhood of z_0 on which the analytic branch S of P^{-1} with $S(z_0) = z_0$ is defined. S maps U_0 into $A^*(z_0)$ and is one-to-one. Hence, $U_1 = S(U_0)$ is simply connected and, furthermore, $U_1 \supset U_0$. This can be seen as follows: Suppose that $U_1 = S(U_0) \subset U_0$. Then also $P(S(U_0)) \subset P(U_0)$ which yields $U_0 \subset P(U_0)$ which is a contradiction to the forward invariance of U_0, unless equality occurs in all relations. Inductively we construct a sequence of sets $U_{n+1} = S(U_n) \supset U_n$ and extend S analytically to U_{n+1} if possible. If this process does not terminate, we obtain a family $S^n : U_0 \to U_n$ of analytic functions on U_0 which omit $\mathcal{J}$, since $U_n \subset A^*(z_0)$ which follows inductively by $U_n = S(U_{n-1}) \subset S(A^*(z_0)) = A^*(z_0)$. Thus, by Montel, $\{S^n\}$ is normal on U_0. But this contradicts the fact that z_0 is a repelling fixed point for S. Thus there is a U_{n+1} to which we finally cannot analytically extend S. There must be then at least one, w. l. o. g. exactly one, critical point p such that $P(p) \in U_n$. Furthermore, p must be in $A^*(z_0)$ since otherwise there would be a $\tilde{p} \in A^*(z_0)$ with $P(\tilde{p}) = P(p)$, $P'(\tilde{p}) \neq 0$, and we could extend S using the branch with $\tilde{p}$. The construction above also shows that this critical point must have distinct iterates.

Next, suppose that z_0 is an attracting periodic point with period $n > 1$, i. e., $0 < |(P^n)'(z_0)| < 1$. The argument above shows that each component of $A^*(z_0)$ contains at least one critical point of P^n. Since $(P^n)'(z) = \prod_{j=0}^{n-1} P'(P^j(z))$ we conclude that $A^*(z)$ also contains a critical point of P. □

Since a polynomial of degree d has $d-1$ critical points we immediately get the following corollary:

Corollary 2.11. *A polynomial of degree d has at most $d-1$ attracting cycles.*

Proof. The statement is obvious from Theorem 2.10. □

Remark. In the case of real functions $f : U \subset \mathbb{R} \to U$ there is a similar theorem that limits the number of attracting cycles by the number of critical points. It is called Singer's Theorem. More about this can be found in [Gul, Chapter 1.8].

Example 2.12. Let us consider the polynomial $P(z) = -2z^3 + 3z^2$. The three fixed points are $z_0 = 0$, $z_1 = 0.5$ and $z_2 = 1$. z_1 is repelling since $|P'(z_1)| = 3/2 > 1$, the two others are superattracting fixed points. Thus the two critical points are the fixed points themselves.

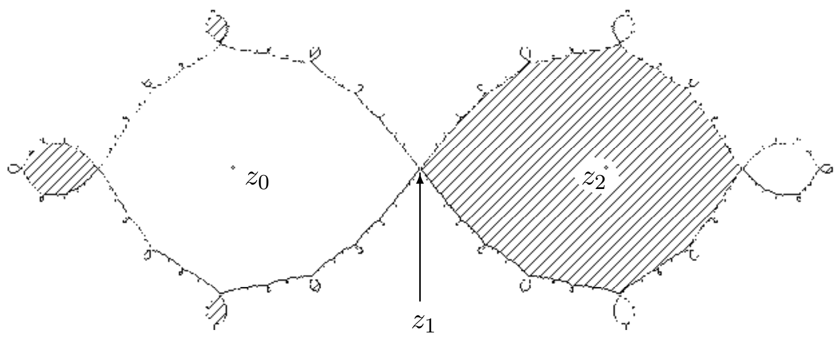

Figure 2.1: Filled Julia set for $P(z) = -2z^3 + 3z^2$. The basin of attraction of the superattracting fixed point $z_2 = 1$ is marked.

By Corollary 2.11 we know that there cannot be any additional attracting cycles. Theorem 1.27 shows that for each of the two fixed points the boundary of its basin of attraction must coincide with the Julia set, and Corollary 2.5 shows that $\mathcal{F}$ and $A(z_i)$, $i = 0, 2$, consist of infinitely many components. To get a feeling how this is possible we have drawn the basin of attraction of z_2 in red. The result can be seen in Figure 5.1. In Figure 2.1 the basin of attraction is hatched. ♢

2.3 Neutral Periodic Points

In our effort to prove that the two definitions of the Julia set are equivalent we used the fact that there are only finitely many non-repelling periodic points. In the previous section we showed that there are at most $d-1$ attracting cycles. The following theorem now covers most neutral cycles. For this theorem we need a one-parameter family of polynomials defined by

$$P(z, w) = (1 - w)P(z) + w.$$

Theorem 2.13. *Suppose that the polynomial P of degree $d \geq 2$ has N neutral periodic points $z_1, \ldots, z_N$ with multiplier $s_j \neq 1$, $j = 1, \ldots, N$. Then there exists an $\varepsilon > 0$ and a direction $\theta \in [0, 1)$ in the w-plane such that, if $0 < \rho < \varepsilon$, the polynomial*

$$P_\rho(z) = P(z, \rho e^{2\pi i\theta})$$

has at least $N/2$ attracting periodic points.

Proof. For each periodic point z_j let n_j denote its period. The pair $(z_j, 0)$ satisfies the equation

$$F_j(z, w) = P^{n_j}(z, w) - z = 0.$$

If the implicit function theorem (see e. g. [Mark, p. 109]) is applied to $F_j(z, w)$ there exist neighborhoods $U_\delta(0)$ of 0 and $U(z_j)$ of z_j and analytic functions $z_j(w) : U_\delta(0) \to U(z_j)$ such that

$$F_j(z_j(w), w) = 0 \quad \text{and} \quad z_j(0) = z_j.$$

Further let us set for $w \in U_\delta(0)$

$$s_j(w) = \frac{\partial P^{n_j}}{\partial z}(z_j(w), w), \quad j = 1, \dots, N,$$

which is an analytic function since $P^{n_j}(z, w)$ is a polynomial in z and w and $z_j(w)$ is an analytic function.

To find the desired angle θ let us expand $s_j(w)$ in a series at $w = 0$:

$$s_j(w) = s_j + a_j w^{k_j} + \cdots \quad \text{where } a_j \neq 0,\ k_j \in \mathbb{N}.$$

Since we consider only polynomials of degree $d \geq 2$ the existence of a nonzero higher order term is guaranteed. Next let us investigate the polynomial

$$\tilde{s}_j(w) = s_j + a_j w^{k_j}.$$

We are going to find an ε_1 and a θ such that at least half of the $\tilde{s}_j$ satisfy

$$|\tilde{s}_j(\rho e^{2\pi i\theta})| < 1 \quad \text{for } 0 < \rho < \varepsilon_1.$$

This immediately gives the existence of an ε such that the same condition holds for the s_j.

Let 2^l be the highest power of two that divides any of the k_j. For each s_j on the unit circle consider the tangent direction, and let T denote the set of those tangent directions (see Figure 2.2).

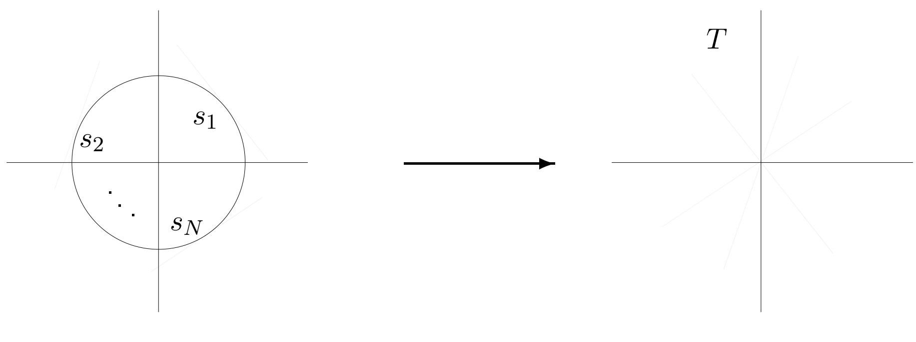

Figure 2.2: Tangent directions at s_j build the set T.

We define the set Φ by

$$\Phi = \Big\{\phi \in [0, 1) \mid \bigvee_{n \in \{0,1,\dots,2^l\}} \bigvee_{j \in \{1,\dots,N\}} 2\pi(k_j\phi + n/2^{l+1}) \in T\Big\}.$$

Note that Φ is a finite set. Now choose any angle $\varphi \notin \Phi$. The desired angle θ will be

$$\theta = \varphi + \psi$$

with

$$\psi = \frac{1}{2}(b_0 + b_1/2 + \cdots + b_l/2^l),$$

where $b_\nu \in \{0, 1\}$ are determined inductively.

Consider all periodic points z_j for which the corresponding k_j's are divisible by 2^l. Then $k_j = 2^l p_j$ for some odd integer p_j and hence, since $b_\nu \in \{0, 1\}$,

$$\begin{aligned}(\rho e^{2\pi i\theta})^{k_j} &= \rho^{k_j} e^{2\pi i k_j\varphi} \underbrace{e^{2\pi i 2^l p_j \psi}}_{=e^{\pi i b_l p_j} = e^{\pi i b_l}} \\ &= \rho^{k_j} e^{2\pi i(k_j\varphi + b_l/2)}\end{aligned}$$

and thus

$$\tilde{s}_j(\rho e^{2\pi i\theta}) = s_j + a_j \rho^{k_j} e^{2\pi i(k_j\varphi + b_l/2)}.$$

Since $k_j\varphi$ is a direction that is not tangent to the unit circle in s_j, we choose $b_l \in \{0, 1\}$ such that at least half of the $\tilde{s}_j(w)$'s go inside the unit circle.

To determine b_{l-1} we repeat this calculation for all periodic points z_j for which k_j is divisible by 2^{l-1} but not by 2^l. Then

$$\tilde{s}_j(\rho e^{2\pi i\theta}) = s_j + a_j \rho^{k_j} e^{2\pi i(k_j\varphi + \pi b_{l-1}/2 + (q/4)b_l)},$$

where q is some odd integer. Since $k_j\varphi + (q/4)b_l$ is not tangent to the unit circle we choose b_{l-1} such that at least half of these $\tilde{s}_j(w)$'s go inside the circle. We just continue this induction until the angle θ is determined.

The value ε_1 has to be chosen such that those $\tilde{s}_j(\rho e^{2\pi i\theta})$'s that enter the unit circle remain inside the unit circle for $0 < \rho < \varepsilon_1$. $\square$

2.3.1 Parabolic Domains

In this section we deal with rationally neutral cycles. So far we know by Theorem 1.23 that rationally neutral cycles belong to $\mathcal{J}$. To describe more precisely what happens in this case we can assume that P has a rationally neutral fixed point and this fixed point is the origin. So our polynomial is of the form

$$P(z) = \lambda z + a_{p+1}z^{p+1} + \cdots + a_d z^d, \quad \lambda = e^{2\pi i l/q},\ l, q \in \mathbb{N},\ a_{p+1} \neq 0.$$

In the case of an attracting fixed point with multiplier λ Theorem 2.9 showed that P could be conjugated to the linear function λz in a neighborhood of the origin. We will encounter a similar result for many irrationally indifferent fixed points when we will deal with the *Schröder equation* (2.17) (see Section 2.3.2). In the case of a rationally indifferent fixed point it is *not* possible to find a conjugation to the linear function λz; instead we will find a different conjugation which gives a useful result to describe the dynamics around the origin. The behavior now strongly depends on p, i. e., on the first non-zero coefficient. So we will consider three cases:

1. $\lambda = 1,\ p = 1,$
2. $\lambda = 1,\ p > 1,$
3. $\lambda^n = 1,\ \lambda \neq 1.$

Case 1: In this case we have $P(z) = z + a_2 z^2 + \cdots$ with $a_2 \neq 0$. Using the conjugating function $\varphi(z) = a_2 z$ we can assume that $a_2 = 1$. Next we move the origin to ∞ by the inversion $z \to -1/z$ and let $g(z)$ be the new function, hence,

$$g(z) = -\frac{1}{P(-\frac{1}{z})} = -\frac{1}{-\frac{1}{z} + \frac{1}{z^2} + \cdots} = \frac{z^d}{z^{d-1} - z^{d-2} + \cdots} = z + 1 + \frac{b}{z} + \cdots, \qquad (2.2)$$

or in other words

$$g(z) = z + 1 + \mathcal{O}\left(\frac{1}{z}\right), \qquad (2.3)$$

which means that there is a real constant $c > 0$ such that

$$|g(z) - (z+1)| \leq \frac{c}{|z|}, \qquad (2.4)$$

if $|z|$ is sufficiently large. Thus, if $|z|$ is sufficiently large $g(z)$ is essentially a shift by one unit to the right.

Our next aim is to prove the following lemma:

Lemma 2.14. *Let g be defined as in* (2.2). *Then there is a constant $C_0 > 0$ and a univalent function φ conjugating g to the translation $z \to z + 1$ on the half-plane $\{\Re(z) \geq C_0\} \subset \mathbb{C}$.*

Proof. First we observe that for a sufficiently large constant $C_0 > 0$ the half-plane $\{\Re(z) \geq C_0\}$ satisfies

$$g(\{\Re(z) \geq C_0\}) \subset \{\Re(z) \geq C_0\}. \qquad (2.5)$$

Furthermore, it can easily be shown by induction using (2.3) and (2.4) that

$$\Re(g^n(z)) > \Re(z) + \frac{n}{2}, \quad \Re(z) \geq C_0,\ n \geq 1, \qquad (2.6)$$

and

$$\frac{n}{2} \leq |g^n(z)| \leq |z| + 2n, \quad \Re(z) \geq C_0,\ n \geq 1. \qquad (2.7)$$

Let us demonstrate the technique to prove the upper bound in (2.7):

$$\begin{aligned} |g(z)| &= |z + 1 + \mathcal{O}(1/z)| \leq |z| + 1 + \mathcal{O}(1/z) \\ &\leq |z| + 1 + \frac{c}{|z|} \leq |z| + 2 \quad \text{for } |z| \geq c \end{aligned}$$

and with induction

$$|g^{n+1}(z)| \leq \underbrace{|g^n(z)|}_{\leq |z| + 2n} + 2 \leq |z| + 2(n+1).$$

The proofs for the lower bound in (2.7) and for (2.6) are analogous.

We now define functions φ_n, $n \in \mathbb{N}$, by

$$\varphi_n(z) = g^n(z) - n - b\ln n \quad \text{for } \Re(z) \geq C_0. \tag{2.8}$$

Our aim is to show the convergence of this sequence of functions. Using equation (2.2) for $g^k(z)$ instead of z and (2.5) we obtain for $\Re(z) \geq C_0$

$$g^{k+1}(z) = g^k(z) + 1 + \frac{b}{g^k(z)} + \mathcal{O}\left(\frac{1}{k^2}\right),$$

where we have used (2.7) to show that

$$\mathcal{O}\left(\frac{1}{|g^k(z)|^2}\right) = \mathcal{O}\left(\frac{1}{k^2}\right).$$

From this we obtain

$$\begin{aligned}\varphi_{k+1}(z) &= g^{k+1}(z) - (k+1) - b\ln(k+1)\\ &= g^k(z) - k - b\ln(k+1) + \frac{b}{g^k(z)} + \mathcal{O}\left(\frac{1}{k^2}\right)\end{aligned}$$

and hence, by (2.8)

$$\varphi_{k+1}(z) - \varphi_k(z) = b\big(\ln k - \ln(k+1)\big) + \frac{b}{g^k(z)} + \mathcal{O}\left(\frac{1}{k^2}\right) = \mathcal{O}\left(\frac{1}{k}\right), \tag{2.9}$$

where we have used the Taylor expansion of the logarithm for

$$\ln k - \ln(k+1) = -\ln\left(\frac{k+1}{k}\right) = -\ln\left(1 + \frac{1}{k}\right) = -\frac{1}{k} + \mathcal{O}\left(\frac{1}{k^2}\right). \tag{2.10}$$

Note that these estimates are independent of z. As $\sum_{k=1}^{n-1} \frac{1}{k} \approx \ln n$ it follows, with the help of (2.4) and (2.8), that

$$|\varphi_n(z) - z| \leq |\varphi_1(z) - z| + \sum_{k=1}^{n-1} |\varphi_{k+1}(z) - \varphi_k(z)| = \mathcal{O}(\ln n) \tag{2.11}$$

for $\Re(z) \geq C_0$. As we want to prove the convergence of the φ_n's we make the following estimate:

$$\begin{aligned}\varphi_{n+1}(z) - \varphi_n(z) &= -\frac{b}{n} + \frac{b}{g^n(z)} + \mathcal{O}\left(\frac{1}{n^2}\right)\\ &= b\left(\frac{1}{n + b\ln n + \varphi_n(z)} - \frac{1}{n}\right) + \mathcal{O}\left(\frac{1}{n^2}\right)\\ &= \frac{b}{n^2}\mathcal{O}\Big(|b\ln n + \varphi_n(z)|\Big) + \mathcal{O}\left(\frac{1}{n^2}\right)\\ &= \mathcal{O}\left(\frac{\ln n}{n^2}\right).\end{aligned} \tag{2.12}$$

This estimate needs some explaining: For the first equality we have used equations (2.9) and (2.10). To show that the third equality is correct observe that

$$\begin{aligned} \frac{n^2}{n + b\ln n + \varphi_n(z)} - n &= \frac{n^2 - n^2 - nb\ln n - n\varphi_n(z)}{n + b\ln n + \varphi_n(z)} \\ &= \underbrace{\frac{-n}{n + b\ln n + \varphi_n(z)}}_{=\mathcal{O}(1) \text{ because of } (2.7)} (b\ln n + \varphi_n(z)). \end{aligned}$$

Finally, using (2.11), we obtain for a given $z \in \{\Re(z) \geq C_0\}$

$$\mathcal{O}(\varphi_n(z)) = \mathcal{O}(\varphi_n(z) - z) = \mathcal{O}(\ln n). \tag{2.13}$$

With the help of (2.12) it is now easy to see that $\sum_{n=1}^{\infty} |\varphi_{n+1}(z) - \varphi_n(z)| < \infty$ which implies the convergence of φ_n to a function φ. Moreover, since (2.13) is the only estimate that depends on z, the convergence is locally uniform on $\{\Re(z) \geq C_0\}$. From the definition of φ_n in (2.8) we obtain

$$\varphi_n(g(z)) = \varphi_{n+1}(z) + 1 + b\ln(1 + 1/n)$$

and hence, taking the limit,

$$\varphi \circ g = \varphi + 1, \tag{2.14}$$

so φ is the desired conjugation of g to the translation $z \to z + 1$.

Since the sequence (φ_n) is locally uniformly convergent and since all the φ_n's are analytic it follows by [Rem, p. 75] that (φ_n) is compactly convergent. Hence, by Theorem 1.7, φ is analytic on $\{\Re(z) \geq C_0\}$, too. Moreover g has only finitely many critical points since g is a rational function. So, if we choose C_0 sufficiently large then $g'(z) \neq 0$ for all $z \in \{\Re(z) \geq C_0\}$. This implies that in this half-plane g is one-to-one, which is based on the following theorem of complex analysis (see e. g. [CM, p. 54]):
Let f: $U \to \mathbb{C}$ be analytic, $z_0 \in U$ with $f'(z_0) \neq 0$. Then there is an open neighborhood U_1 of z_0 such that $f|_{U_1}$ is one-to-one and hence a homeomorphism from U_1 onto $V = f(U_1)$, where V is open in $\mathbb{C}$.
Since $\{\Re(z) \geq C_0\}$ is simply connected we can connect the local homeomorphisms to just one homeomorphism on $\{\Re(z) \geq C_0\}$. Suppose now that g^n is univalent on $\{\Re(z) \geq C_0\}$. To show that g^{n+1} is univalent we observe that

$$\begin{aligned} & g^{n+1}(z_1) = g^{n+1}(z_2) \\ \iff \quad & g(g^n(z_1)) = g(g^n(z_2)) \\ \iff \quad & g^n(z_1) = g^n(z_2) \\ \iff \quad & z_1 = z_2. \end{aligned}$$

We have used the fact that g^n is univalent, g is univalent and $g^n(z_i) \in \{\Re(z) \geq C_0\}$. Hence, all g^n's are univalent and thus, by (2.8), the φ_n's are univalent, too. An analytic function $f : U \to V$ that is one-to-one and onto has the property that $f'(z) \neq 0$ for all $z \in U$ (see e. g. [CM, p. 55]), which implies that $\varphi_n'(z) \neq 0$ for all $z \in \{\Re(z) \geq C_0\}$. Now consider any closed curve Γ in $\{\Re(z) \geq C_0\}$. Using the argument principle (see

e. g. [Neh, p. 143]) we can count the number of zeros of an analytic function f inside a given curve as

$$N = \frac{1}{2\pi i}\int_\Gamma \frac{f'(z)}{f(z)}dz.$$

We now apply this fact to the derivative of φ_n from which we receive

$$0 = \frac{1}{2\pi i}\int_\Gamma \frac{\varphi_n''(z)}{\varphi_n'(z)}dz.$$

As φ_n converges locally uniformly to φ we have (using the fact that we can interchange the limit and the integral; see e. g. [Rem, p. 141])

$$0 = \lim_{n\to\infty}\frac{1}{2\pi i}\int_\Gamma \frac{\varphi_n''(z)}{\varphi_n'(z)}dz = \frac{1}{2\pi i}\int_\Gamma \lim_{n\to\infty}\frac{\varphi_n''(z)}{\varphi_n'(z)}dz = \frac{1}{2\pi i}\int_\Gamma \frac{\varphi''(z)}{\varphi'(z)}dz,$$

and thus φ has no critical points inside Γ. As Γ was arbitrarily chosen in $\{\Re(z) \geq C_0\}$ we conclude that φ is univalent on $\{\Re(z) \geq C_0\}$. □

Our next aim is to extend φ to a larger domain. Using the functional equation $\varphi(g(z)) = \varphi(z) + 1$ we can extend φ analytically to any simply connected domain Ω provided that g is defined on Ω, satisfies $g(\Omega) \subset \Omega$ and $g^n(z) \to \infty$ for $z \in \Omega$. (From (2.3) we see that if $g^n(z) \to \infty$ then also $\Re(g^n(z)) \to \infty$.) The latter conditions guarantee that all the steps that we have done in the proof of Lemma 2.14 can also be done with Ω instead of $\{\Re(z) \geq C_0\}$. Furthermore, if we choose Ω such that $\min_{z\in\Omega}|z|$ is sufficiently large then we can also prove the univalence of φ as in the proof of Lemma 2.14. Thus we have proven

Lemma 2.15. *Let g be defined as in* (2.2) *and let Ω be a simply connected domain such that g is defined on Ω and satisfies $g(\Omega) \subset \Omega$ and $g^n(z) \to \infty$ for $z \in \Omega$. Then there is a univalent function φ conjugating g to the translation $z \to z + 1$ on Ω.*

Remark. We will need the function φ later when we will prove Theorem 2.20.

As an example consider a domain $\tilde{\Omega}$ defined by $\tilde{\Omega} = \{z = x + iy \mid |y| > -kx + d\}$. For any given $k > 0$ it is possible to find a sufficiently large d such that $g(\tilde{\Omega}) \subset \tilde{\Omega}$. To make the curve smooth even where the two lines intersect we connect them with an appropriate segment of a circle (see left part of Figure 2.3). We call the new domain constructed with this method Ω. It is drawn shaded in Figure 2.3. By taking d sufficiently large we can again guarantee that φ is univalent on Ω.

Originally, we wanted to describe the dynamics of $\{P^n\}$ near the indifferent fixed point at the origin. Using the function g defined in (2.2) and Lemma 2.15 we can now give answers. From (2.3) and Lemma 2.15 we know that g is essentially a shift to the right on Ω. g was defined as $g = h \circ P \circ h^{-1}$ where h is the inversion $z \to -1/z$ (in this case $h^{-1} = h$). h now takes Ω to the cardioid-shaped region displayed in the right part of Figure 2.3. As $P^n = h \circ g^n \circ h$ we can simply apply h to the orbits of $\{g^n\}$ on Ω if we want to investigate the direction of the orbits of $\{P^n\}$. The arrows in the right part of Figure 2.3 indicate the direction in which iterates of P move. Points to the right of the origin are first repelled, but most of them make a circle and are eventually attracted from the left of the origin. We call such a region $h(\Omega)$ an *attracting petal* for the fixed point. The angle of the "mouth" at the origin can even be made smaller if we use a

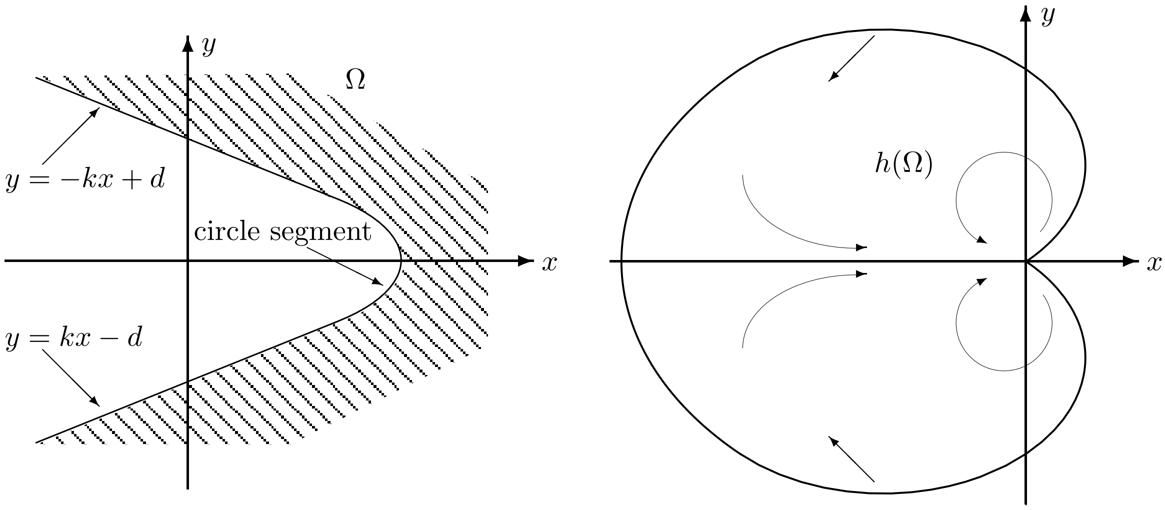

Figure 2.3: Part of parabolic attracting region for g near ∞ (left) and attracting petal for P at 0 (right).

more sophisticated region for Ω (see e. g. [CG, p. 37]). For these regions it can even be shown that at the cusp of the petal the boundary curves are tangent to the real axis.

If we use $\{\Re(z) \geq C_0\}$ (as in Lemma 2.14) instead of Ω and apply h, then we only see that points to the left of the origin are locally attracted. But using the same techniques we can also show that points to the right of the origin are locally repelled (although most of them make a circle and return from the left of the origin, as seen in the right of Figure 2.3). This is stated in the following proposition (which can be found in [De1, p. 301]):

Proposition 2.16. *Let $P(z) = z + z^2 + a_3z^3 + \cdots + a_nz^n$. There exists a real $\mu > 0$ such that*

1. *all points in the interior of the circle of radius μ centered at $-\mu$ are attracted to the origin.*

2. *all points in the interior of the circle of radius μ centered at μ are repelled from the origin.*

The question arises if *all* points that are repelled first are finally attracted from the left of the origin. But this cannot be, as otherwise there would be a neighborhood of the origin such that $P^n(z) \to 0$ for all z in this neighborhood. This implies that $\{P^n\}$ is normal at the origin which is a contradiction to the fact that rationally indifferent fixed points are elements of the Julia set (see Theorem 1.23). Moreover, since (for polynomials) the Julia set is the boundary of $A(\infty)$ (Theorem 1.27), there must be points in the neighborhood of the origin that tend to infinity under iteration.

Case 2: Suppose now that $z' = P(z) = z + a_{p+1}z^{p+1} + \cdots$ with $a_{p+1} \neq 0$ and $p > 1$. As before we can assume that $a_{p+1} = 1$. We now make a change of variables by $z = \zeta^{1/p}$ and $z' = \zeta'^{1/p}$ for $\arg\zeta, \arg\zeta' \in (0, 2\pi)$ and a restriction of z and z' to an

appropriate segment of angle $2\pi/p$. With this we have

$$\zeta' = \zeta\Big(1 + \zeta + \mathcal{O}(|\zeta|^{1+1/p})\Big)^p$$

and thus

$$\zeta' = \zeta + p\zeta^2 + \mathcal{O}(|\zeta|^{2+1/p}).$$

Again we can renormalize to make the coefficient of ζ^2 equal to 1. Then we change variables by $\zeta = -1/z$ and $\zeta' = -1/z'$, i. e.,

$$\begin{aligned}
-\frac{1}{z'} &= -\frac{1}{z} + \frac{1}{z^2} + \mathcal{O}\left(\left|\frac{1}{z}\right|^{2+1/p}\right) \\
z' &= \frac{1}{\frac{1}{z} - \frac{1}{z^2} + \mathcal{O}\left(\frac{1}{|z|^2}\cdot\frac{1}{|z|^{1/p}}\right)} = \frac{z^2}{z - 1 + \mathcal{O}\left(|z|^{-1/p}\right)} \\
&= z + 1 + \mathcal{O}\left(|z|^{-1/p}\right).
\end{aligned}$$

If we call the resulting function g this yields

$$z' = g(z) = z + 1 + \mathcal{O}(|z|^{-1/p}).$$

This looks very much like (2.3), but it is *not*! The problem is that we only have terms of $\mathcal{O}(|z|^{-1/p})$ and not of $\mathcal{O}(1/|z|)$. Thus the method used in Case 1 cannot be applied directly to Case 2. The following lemma will fix this problem:

Lemma 2.17. *A polynomial of the form $P(z) = z + a_{p+1}z^{p+1} + a_{p+2}z^{p+2} + \cdots$ can be locally conjugated to a polynomial of the form $\tilde{P}(z) = z + z^{p+1} + b_{2p+1}z^{2p+1} + b_{2p+2}z^{2p+2} + \cdots$.*

Proof. The conjugation will be done step by step. In the first step we can make the coefficient of z^{p+1} equal to 1 by choosing the conjugation $\varphi(z) = a_{p+1}^{1/p}z$.

Now we eliminate gradually all the coefficients of z^{p+k} for $1 < k < p+1$. Let us demonstrate one such "reduction": We want to conjugate a polynomial

$$P(z) = z + z^{p+1} + a_{p+k}z^{p+k} + \cdots$$

to a polynomial of the form

$$Q(z) = z + z^{p+1} + b_{p+k+1}z^{p+k+1} + \cdots$$

with a conjugating function of the form $H(z) = z + \alpha z^k$, i. e.,

$$P \circ H = H \circ Q. \tag{2.15}$$

If we want to write this in the form

$$H^{-1} \circ P \circ H = Q$$

we must keep in mind that H^{-1} means the branch that satisfies $H^{-1}(0) = 0$.

Evaluating (2.15) we obtain

$$\begin{aligned} P \circ H(z) &= z + \alpha z^k + (z + \alpha z^k)^{p+1} + a_{p+k}(z + \alpha z^k)^{p+k} + \cdots \\ &= z + \alpha z^k + z^{p+1} + \alpha(p+1)z^{p+k} + a_{p+k}z^{p+k} + \mathcal{O}\big(|z|^{p+k+1}\big) \end{aligned}$$

and

$$\begin{aligned} H \circ Q(z) &= z + z^{p+1} + \mathcal{O}\big(|z|^{p+k+1}\big) + \alpha\Big(z + z^{p+1} + \mathcal{O}\big(|z|^{p+k+1}\big)\Big)^k \\ &= z + z^{p+1} + \alpha z^k + \alpha k z^{p+k} + \mathcal{O}\big(|z|^{p+k+1}\big) \end{aligned}$$

The coefficients of z, z^k and z^{p+1} are the same in both equations. As the coefficients b_{p+k+m} with $m \geq 1$ can be chosen arbitrarily, we only have to make sure that the coefficients of z^{p+k} coincide, thus

$$\alpha k = \alpha(p+1) + a_{p+k}, \quad \text{i. e.,} \quad \alpha = \frac{a_{p+k}}{k-(p+1)}.$$

This is possible as long as $k < p+1$. Hence, the first coefficient that may not disappear is the coefficient of z^{2p+1} which is exactly what we wanted to prove. □

Thus, by the above lemma, we may replace $P(z) = z + a_{p+1}z^{p+1} + a_{p+2}z^{p+2} + \cdots$ by $\tilde{P}(z) = z + z^{p+1} + b_{2p+1}z^{2p+1} + b_{2p+2}z^{2p+2} + \cdots$, and, with the same transformations as at the beginning of Case 2, we now again obtain $g(z) = z + 1 + \mathcal{O}(1/|z|)$ and thus the existence of φ, where the proof runs as in Case 1.

As we make conjugations of the form $z = \zeta^{1/p}$ we have p different branches to choose from. So we not only have one attracting petal, instead we have p attracting petals.

Figure 2.4 shows the Julia set for the polynomial $P(z) = z + z^4$ and the forward orbits of some points. The three rays tending to the origin denote the postcritical set, i. e., the forward orbit of the critical points (see Definition 2.32). In this example they are exactly on the *attracting directions*. Generally, the attracting directions bisect the sections bounded by two consecutive *repelling directions* that are given by

$$\theta_k = -\frac{\arg a_{p+1}}{p} + \frac{2\pi k}{p}, \quad 0 \leq k \leq p-1.$$

Note that these directions are those angles for which $|1 + a_{p+1}z^p|$ is largest and therefore $P(z) = z(1 + a_{p+1}z^p) + \mathcal{O}(|z|^{p+2})$ expands the most. So, in our example the repelling directions are given by the angles 0, $2\pi/3$ and $4\pi/3$. The repelling and attracting directions always denote the local directions in a neighborhood of the indifferent fixed point currently investigated in which points are repelled or attracted.

Case 3: In the third case we have $P(z) = \lambda z + \cdots$ where $\lambda = e^{2\pi i k/n}$, $k < n$, i. e., λ is a primitive n^{th} root of unity with $n > 1$. Then $P^n(z)$ belongs to either Case 1 or Case 2. It can be shown that in this case P^n assumes the form $P^n(z) = z + b_{nl+1}z^{nl+1} + \cdots$ for some $l \in \mathbb{N}$ and thus has nl attracting petals. P permutes those petals in cycles of length n. Let us demonstrate these facts by the following example:

Example 2.18. Consider the polynomial $P(z) = -z + z^2 - z^3$. Here we have $n = 2$; so we need to consider the second iterate which is $P^2(z) = z + 4z^5 - 6z^6 - 3z^8 - z^9$ and thus in this case $l = 2$. There must be then 4 attracting petals. Figure 2.5 shows the

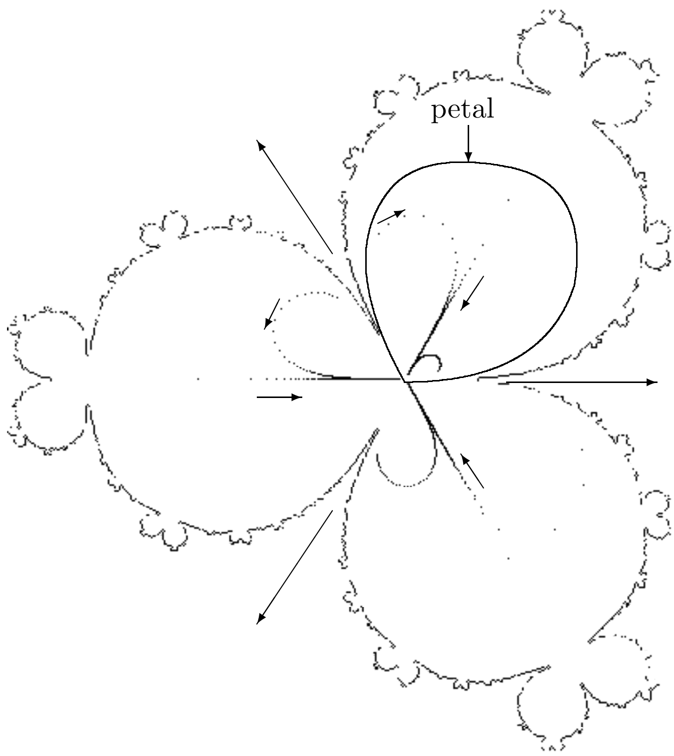

Figure 2.4: Parabolic attracting region with three attracting petals for $P(z) = z + z^4$.

Julia set of P together with the postcritical set. The four rays of the postcritical set are tangent to the four attracting directions at the origin. The picture might be a little bit misleading: In reality the rays do reach the origin, but due to rounding errors the iteration process stalls at a certain distance from the origin (or at least takes a very long time to proceed). ♢

In the case of an attracting periodic point the definition of the basin of attraction was quite obvious. For rationally indifferent periodic points the situation is a little bit different. Note that if a polynomial has a rationally indifferent fixed point then the discussion above shows that there is always at least one attracting petal for this fixed point.

Definition 2.19. Let P be a complex polynomial with a rationally indifferent fixed point z_0 and let S be an attracting petal at z_0. The *basin of attraction associated with* S is defined as

$$A_S(z_0) = \Big\{ z \in \mathbb{C} \;\Big|\; \bigvee_{n \in \mathbb{N}} P^n(z) \in S \Big\}.$$

The *immediate basin of attraction associated with* S $A_S^*(z_0)$ is the component of $A_S(z_0)$ containing S. The *basin of attraction* of z_0 is defined as

$$A(z_0) = \{ z \in \mathbb{C} \mid P^n(z) \to z_0 \} = \bigcup_{S \text{ petal at } z_0} A_S(z_0).$$

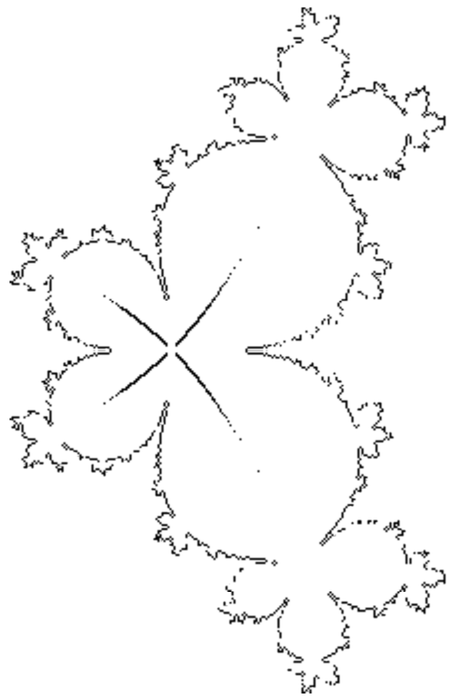

Figure 2.5: Parabolic attracting region with four attracting petals for $P(z) = -z + z^2 - z^3$.

The *immediate basin of attraction* $A^*(z_0)$ is the union of all those components of $A(z_0)$ that contain a petal, i. e.,

$$A^*(z_0) = \bigcup_{S \text{ petal at } z_0} A^*_S(z_0).$$

Let $\{z_0, \ldots, z_{n-1}\}$ be a rationally indifferent cycle of length n such that the multiplier of P^n at z_0 is a primitive m^{th} root of unity and let S be an attracting petal for P^{nm} at z_0. The *basin of attraction associated with* S is again defined as

$$A_S(\{z_j\}) = \Big\{ z \in \mathbb{C} \;\Big|\; \bigvee_{k \in \mathbb{N}} P^k(z) \in S \Big\}.$$

In this case the sets $P^j(S)$, $j = 0, \ldots, nm-1$, are disjoint attracting petals (there are m petals at each of the points z_k, $k = 0, \ldots, n-1$; see the short discussion in Case 3 before Example 2.18). Hence, the *immediate basin of attraction associated with* S $A^*_S(\{z_j\})$ for the cycle is defined as the union of those components of $A_S(\{z_j\})$ containing the $P^j(S)$'s, $j = 0, \ldots, nm-1$. The *basin of attraction* of the cycle is defined as

$$A(\{z_j\}) = \{ z \in \mathbb{C} \mid \{z_j\} \text{ is the cluster set of } \{P^n(z)\} \} = \bigcup_{k=0}^{n-1} \bigcup_{S \text{ petal at } z_k} A_S(\{z_j\}).$$

The *immediate basin of attraction* $A^*\{(z_j\})$ is the union of all those components of $A(\{z_j\})$ that contain a petal, i. e.,

$$A^*(\{z_j\}) = \bigcup_{k=0}^{n-1} \bigcup_{S \text{ petal at } z_k} A^*_S(\{z_j\}).$$

Remark. As in the case of a basin of attraction of an attracting periodic point, the basin of attraction of a rationally neutral periodic point is again an open subset of the Fatou set $\mathcal{F}$ whose boundary coincides with the Julia set $\mathcal{J}$.

Theorem 2.20. *If z_0 is a rationally neutral periodic point for a polynomial P then each immediate basin of attraction associated with a petal S contains a critical point.*

Proof. Replacing P by P^N we can assume that z_0 is a fixed point with multiplier 1. (As in the proof of Theorem 2.10 we conclude that if we have a critical point of P^N then we also have a critical point of P in the immediate basin of attraction, since $(P^N)'(z) = \prod_{i=0}^{N-1} P'(P^i(z))$.)

Now let $A_S^*(z_0)$ be the immediate basin of attraction associated with the petal S. Furthermore let h be the function that conjugates P to the function g that we introduced in (2.2), i. e.,

$$g = h \circ P \circ h^{-1}.$$

If we are in the situation of Case 1 then $h(z) = -1/z$. The beginning of Case 2 shows how the function h is assembled in this case. Now we define a function ψ as

$$\psi = \varphi \circ h,$$

where φ is the function we obtain from Lemma 2.15. Hence, ψ is univalent on S and conjugates P to a translation

$$\psi(P(z)) = \psi(z) + 1. \tag{2.16}$$

For every $z \in A_S^*(z_0)$ there is an $n \in \mathbb{N}$ such that $P^n(z) \in S$. Since all elements in S tend to the origin this guarantees that $\Re(g^n(h(z))) \to +\infty$. Furthermore, $P\big(A_S^*(z_0)\big) = A_S^*(z_0)$. As the proof of Lemma 2.15 shows, these two facts allow us to extend ψ analytically by the functional equation (2.16) to $A_S^*(z_0)$ where it also satisfies (2.16) but where it is not necessarily univalent.

$\psi(S)$ now approximately looks like the domain Ω in Figure 2.3 shifted to the right by one, so it covers a right half-plane. Hence, the functional equation (2.16) and $P\big(A_S^*(z_0)\big) = A_S^*(z_0)$ show that ψ maps $A_S^*(z_0)$ onto the entire complex plane. This can be seen as follows: Since S is a subset of $A_S^*(z_0)$ we know that $\psi(A_S^*(z_0))$ at least covers a half-plane. Let w be an arbitrarily chosen point in this half-plane and $z_1 \in A_S^*(z_0)$ such that $\psi(z_1) = w$. As $P\big(A_S^*(z_0)\big) = A_S^*(z_0)$ there is a $z_2 \in A_S^*(z_0)$ with $P(z_2) = z_1$ and thus (using (2.16))

$$w = \psi(z_1) = \psi\big(P(z_2)\big) = \psi(z_2) + 1,$$

which yields

$$\psi(z_2) = w - 1.$$

Thus we can inductively cover the whole complex plane.

Now suppose that ψ has no critical point in $A_S^*(z_0)$. Then we can define a branch of the inverse function ψ^{-1} which would be an analytic function on the entire complex plane. Moreover, the range of this branch would be bounded since $\partial A_S^*(z_0) \subset \mathcal{J}$ and $\mathcal{J}$ is bounded. But a bounded analytic function defined on the whole complex plane must

be constant. (This is known as Liouville's Theorem; see e. g. [Peh] or [Rem, p. 192].) This is of course a contradiction. Hence, ψ must have a critical point w_0 in $A_S^*(z_0)$.

For sufficiently large n $P^n(w_0)$ is in S, where ψ has no critical points. By replacing w_0 by the largest iterate that is a critical point we can assume that $P(w_0)$ is not a critical point of ψ. From the functional equation (2.16) we then have $\psi'(P(w_0))P'(w_0) = \psi'(w_0) = 0$ which implies that w_0 is a critical point of P. □

Corollary 2.21. *The number of attracting cycles plus the number of rationally neutral cycles of a polynomial of degree $d \geq 2$ is at most $d-1$.*

Proof. Since a polynomial P of degree d has at most $d-1$ critical points, the proof follows immediately from Theorem 2.10 and Theorem 2.20. □

Corollary 2.22. *The number of attracting cycles plus the number of indifferent cycles of a polynomial of degree $d \geq 2$ is at most $3d-3$.*

Proof. From Corollary 2.21 we know that the number of attracting cycles plus the number of rationally indifferent cycles is at most $d-1$. By Theorem 2.13 half of the neutral cycles with multiplier $\lambda \neq 1$ become attracting cycles of a perturbed polynomial of the same degree. Hence, there are at most $2d-2$ neutral cycles with multiplier not equal to 1 which gives a total of $3d-3$ non-repelling cycles. □

We are now ready to fill the gap we left in Chapter 1 when we wanted to show the equivalence of the two definitions of the Julia set. Remember that we wanted to show that there are only finitely many non-repelling cycles. In particular, we can now prove the upper bound of $3d-3$ non-repelling cycles.

Proof of Theorem 1.25: A non-repelling periodic point is either attracting or indifferent. Hence, Corollary 2.22 gives the desired result. □

Remark. Using algebraic functions one can prove Theorem 2.13 for *all* neutral periodic points (even for those with multiplier $s_j = 1$). Moreover, the attracting periodic points of the original polynomial remain attracting periodic points for the perturbed polynomial. Hence, the number of attracting periodic points plus half the number of neutral periodic points is at most $d-1$ which gives an upper bound of $2d-2$ for the total number of non-repelling periodic points.

2.3.2 Siegel Disks and Herman Rings

Let us now first concentrate on Siegel disks. By definition we know that on a Siegel disk R^n is conjugate to an irrational rotation, i. e.,

$$\varphi(R^n(z)) = \lambda\varphi(z), \quad \text{where } \lambda = e^{2\pi i\theta},\ \theta \in \mathbb{R} \setminus \mathbb{Q}$$

with (without loss of generality) $\varphi'(0) = 1$. Thus we obtain

$$R^n(\varphi^{-1}(\zeta)) = \varphi^{-1}(\lambda\zeta), \qquad (\varphi^{-1})'(0) = 1, \tag{2.17}$$

which is a so-called *Schröder equation* for the function φ. A Schröder equation has the general form

$$f(h(\zeta)) = h(\lambda\zeta), \qquad h'(0) = 1. \tag{2.18}$$

The importance of this equation can be seen in the following theorem:

Theorem 2.23. *Let z_0 be a neutral fixed point. Then $z_0 \in \mathcal{F}$ if and only if* (2.17) *has an analytic solution in some neighborhood of z_0.*

Thus the existence of a Siegel disk is equivalent to the existence of an analytical solution of the Schröder equation 2.17. For a proof we refer to [Bl1, p. 98].

The question now arises when (2.17) or the more general form (2.18) has a solution. C.L. Siegel gave a result that guarantees the existence of a solution to (2.18) for a wide range of values for θ. First we need a definition:

Definition 2.24. A real number θ is called *diophantine* if it is badly approximable by rational numbers, in the sense that

$$\bigvee_{c\in\mathbb{R}^+} \bigvee_{\substack{\mu\in\mathbb{R}^+\\ \mu<\infty}} \bigwedge_{\substack{p,q\in\mathbb{Z}\\ q\neq 0}} \left|\theta - \frac{p}{q}\right| \geq \frac{c}{q^\mu},$$

which is equivalent to

$$\bigvee_{c\in\mathbb{R}^+} \bigvee_{\substack{\mu\in\mathbb{R}^+\\ \mu<\infty}} \bigwedge_{n\in\mathbb{N}} |\lambda^n - 1| \geq c\cdot n^{1-\mu},$$

where $\lambda = e^{2\pi i\theta}$.

With this definition we can now present the important theorem due to Siegel (for a proof we refer to [CG, pp. 43–46]):

Theorem 2.25 (Siegel). *Let θ be diophantine, and let f have a fixed point at* 0 *with multiplier $\lambda = e^{2\pi i\theta}$. Then there exists a solution to the Schröder equation* (2.18)*, that is, f can be conjugated near* 0 *to an irrational rotation by λ.*

So if we have a diophantine θ Theorem 2.25 guarantees the existence of a Siegel disk. But when is a number diophantine? The criterion given in Definition 2.24 is not very useful for practical purposes. J. Liouville proved a result that makes it easy to show, for a wide range of numbers, that a number is diophantine . Recall that a real (or complex) number is called *algebraic* if it is a root of any algebraic equation of the form

$$a_n z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0 = 0, \tag{2.19}$$

where $a_i \in \mathbb{Z}$ for $i = 0, \ldots, n$. An algebraic number has *degree n* if it is a root of an equation of the form (2.19) with degree n and no root of any equation of the form (2.19) with degree less than n. Hence, an algebraic number of degree n with $n > 1$ cannot be rational, since a rational number p/q is always a root of $qz - p = 0$. Liouville now proved the following:

Theorem 2.26 (Liouville). *Let $x \in \mathbb{R}$ be an algebraic number of degree $n > 1$. Then the following holds:*

$$\bigvee_{c>0} \bigwedge_{\substack{p,q\in\mathbb{Z}\\ q\neq 0}} \left|x - \frac{p}{q}\right| \geq \frac{c}{q^n}.$$

In particular x is diophantine.

A proof of (a slightly modified version of) this theorem can be found in [CR, pp. 83–85]. There one can also find a method to construct numbers that are not algebraic.

Example 2.27. Let us consider the polynomial $P(z) = \lambda(z - z^3/3)$ with $\lambda = e^{2\pi i\theta}$ and $\theta = \frac{\sqrt{5}-1}{2}$. Obviously, θ is a root of the algebraic equation $z^2 + z - 1 = 0$, therefore it is an algebraic number of degree 2. By Theorem 2.26 it is also diophantine, and Theorem 2.25 guarantees the existence of a Siegel disk. Figure 2.6 shows the Julia set and the border of the Siegel disk. Figure 5.2 shows a color image of the filled Julia set. In this picture the color of a point depends on the value of $|P^n(z)|$ for $n = 30$. It can be clearly seen that the absolute value of points inside the Siegel disk changes only marginally under iteration which suggests that they rotate around the origin. $\diamondsuit$

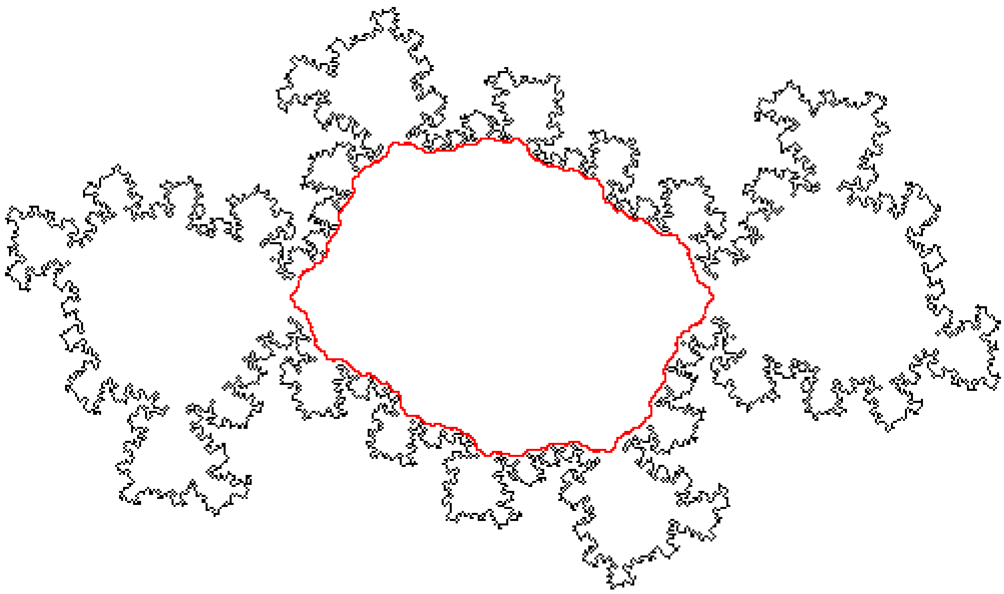

Figure 2.6: Julia set for $P(z) = \lambda(z - z^3/3)$ with $\lambda = \exp(2\pi i(\sqrt{5}-1)/2)$ with a Siegel disk.

The question now arises if there is any irrational θ such that there is no Siegel disk. The answer is yes. One result in this direction is the following (see [CG, p. 42]):

Theorem 2.28. *There exists a* $\lambda = e^{2\pi i\theta}$, $\theta \in \mathbb{R} \setminus \mathbb{Q}$, *such that the Schröder equation* (2.18) *has no solution for any polynomial* f.

In some cases one can say even more:

Theorem 2.29. *Suppose* $P(z) = z^d + \cdots + \lambda z$ *where* $\lambda = e^{2\pi i\theta}$, $\theta \in \mathbb{R}\setminus\mathbb{Q}$. *Furthermore suppose that* θ *satisfies*

$$\bigvee_{\substack{\alpha>0\\ C>0}} \bigwedge_{m\in\mathbb{N}} \bigvee_{\substack{n>m\\ n\in\mathbb{N}}} |\lambda^n - 1| \leq \left(\frac{C}{n^\alpha}\right)^{d^n-1}. \tag{2.20}$$

Then 0 *is an irrationally neutral fixed point that belongs to the Julia set* $\mathcal{J}$.

Proof. In order to prove that 0 belongs to $\mathcal{J}$ we will prove that in any arbitrary neighborhood of 0 there is a periodic point of P. Since, by Theorem 1.25, there are only finitely many non-repelling periodic points this implies that there is a repelling periodic point in every neighborhood of 0. As $\mathcal{J}$ is the closure of the repelling periodic points it follows that $0 \in \mathcal{J}$.

To find such a periodic point we observe that the equation $P^n(z) - z = 0$ is of the form

$$z^{d^n} + \cdots + (\lambda^n - 1)z = 0.$$

As 0 is one root, we can divide by z. Let $\zeta_1, \ldots, \zeta_{d^n-1}$ denote the roots of the new equation. For these roots we know by Vieta that

$$|\zeta_1| \cdot \ldots \cdot |\zeta_{d^n-1}| = |\lambda^n - 1|.$$

Therefore at least one of the ζ_i's must satisfy

$$|\zeta_i| \leq \sqrt[d^n-1]{|\lambda^n - 1|} \leq \frac{C}{n^\alpha} \xrightarrow{n\to\infty} 0,$$

where in the last inequality we have used assumption (2.20). Hence, in any neighborhood of 0 there is a periodic point. □

Remark. It is not obvious whether or not there are any irrational numbers for which (2.20) holds. By means of continued fractions the existence of such numbers can be shown. An example with $C = \alpha = 1$ is given in [De1, pp. 307–308].

For Herman rings the theory is much more complicated. It was only in 1984 that the existence of such domains was proven by M. Herman. On the other hand, it is not very difficult to show that for polynomials such Herman rings can never occur:

Theorem 2.30. *A polynomial P has no Herman rings.*

Proof. As a Herman ring U_H is a component of the Fatou set $\mathcal{F}$ and for any component U of $\mathcal{F}$ $\partial U \subset \partial\mathcal{F} = \mathcal{J}$, we have (using Theorem 1.27)

$$\partial U_H \subset \mathcal{J} = \partial A(\infty).$$

As a Herman ring is doubly connected, this contradicts the fact that for polynomials $A(\infty)$ is connected (Corollary 1.29). □

Remark. Another proof of Theorem 2.30 uses Theorem 2.3: As stated in the remark after Theorem 2.3 all components of $\mathcal{F}$ except $A(\infty)$ must be simply connected. As a Herman ring is doubly connected we conclude that a polynomial cannot have a Herman ring.

At the end of this section we present some further general results that are valid not only for polynomials but for any rational function with degree $d \geq 2$. First recall that on the Fatou set $\mathcal{F}_R$, where $\{R^n\}$ is normal, subsequences of $\{R^n\}$ converge uniformly to functions which are called *limit functions*.

Theorem 2.31. *Let U be an invariant component of the Fatou set $\mathcal{F}_R$, and suppose that the set of limit functions for R^n on U contains non-constant functions. Then U is either a Siegel disk or a Herman ring.*

A proof of this theorem can be found in [Bea, pp. 167–170].

Another theorem, which is quite useful for drawing a Siegel disk or a Herman ring, needs an additional definition:

Definition 2.32.

1. Let CP be the set of the critical points of the rational function R. The *postcritical set* is defined as

$$\bigcup_{n\in\mathbb{N}_0} R^n(CP).$$

2. The closure of the postcritical set is denoted by CL.

With this definition the following theorem holds (cf. [CG, p. 82]):

Theorem 2.33. *If U is a Siegel disk or a Herman ring, then*

$$\partial U \subset CL.$$

Example 2.34. Let us again consider the polynomial $P(z) = \lambda(z - z^3/3)$ with $\lambda = e^{2\pi i\theta}$ and $\theta = \frac{\sqrt{5}-1}{2}$. The critical points are -1 and 1. Hence, the postcritical set is

$$\bigcup_{n\in\mathbb{N}_0} P^n(-1) \cup P^n(1).$$

In this case both critical points lie in the boundary of the Siegel disk and thus we have $CL = \partial U_S$ where U_S denotes the Siegel disk. Figure 2.7 shows the postcritical set for this polynomial. It delineates exactly the Siegel disk in Figure 2.6. $\diamondsuit$

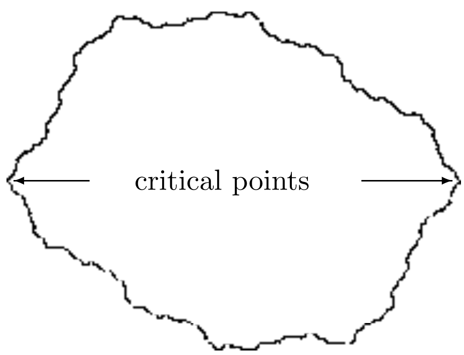

Figure 2.7: Postcritical set for $P(z) = \lambda(z - z^3/3)$ delineates a Siegel disk.

Chapter 3

Escape and Converge Radius

3.1 Escape Radius

In Lemma 1.13 we have seen that for polynomials the Julia set $\mathcal{J} = \mathcal{J}_P$ is bounded. In this section we now want to specify more precisely this boundedness. We will try to find a disk centered at the origin in which $\mathcal{J}$ is contained. Since $\mathcal{J} = \partial\mathcal{K}$ and $\mathcal{K} = \{z \in \mathbb{C} \mid |P^n(z)| \not\to \infty\}$ this leads to the following definition:

Definition 3.1. A radius r is called an *escape radius* if

$$\bigwedge_{\substack{z\in\mathbb{C}\\ |z|>r}} |P^n(z)| \xrightarrow{n\to\infty} \infty. \tag{3.1}$$

The smallest escape radius is called *optimal escape radius* r_{opt}, i. e.,

$$r_{\text{opt}} = \min\{r \mid r \text{ satisfies condition (3.1)}\}. \tag{3.2}$$

Thus if r is an escape radius then $\mathcal{J}$ is contained in $\overline{B(0,r)}$. $\overline{B(0,r_{\text{opt}})}$ is the smallest disk centered at the origin in which $\mathcal{J}$ is contained.

In almost every book on this topic you find one estimate for an escape radius for quadratic polynomials of the form $P_c(z) = z^2 + c$; they use

$$r(c) = \max\{2, |c|\}.$$

But, as we will see later, this is only a rough estimate. We will provide much better results. For polynomials of higher degree most books and articles do not give any estimate whatsoever. It is only in [Dou] where we find a hint that

$$r = \frac{1 + |a_d| + \cdots + |a_0|}{|a_d|}$$

can be used as an escape radius. Again, we will derive much better results.

Theorem 3.2. *Let P be a complex polynomial of degree $d \geq 2$, and let us define $\mathcal{E} = \{z \in \mathbb{C} \mid |P(z)| = |z|\}$ and $r_{\mathcal{E}} = \max_{z\in\mathcal{E}} |z|$. Then $r_{\mathcal{E}} \geq r_{opt}$, i. e., $r_{\mathcal{E}}$ is an escape radius.*

Proof. Let $z_1 \in \mathbb{C}$ be arbitrary but fixed with $|z_1| > r_{\mathcal{E}}$. We will first prove the following assertion:

$$|P(z_1)| > |z_1|. \tag{3.3}$$

Case 1: suppose $|P(z_1)| = |z_1|$; then $z_1 \in \mathcal{F}$, in contradiction to $|z_1| > r_{\mathcal{E}}$.
Case 2: suppose $|P(z_1)| < |z_1|$; we now define a function $f : \mathbb{C} \to \mathbb{R}$ by $f(z) = |P(z)| - |z|$. By definition of $r_{\mathcal{E}}$ we have

$$\bigwedge_{\substack{z \\ |z|>r_{\mathcal{E}}}} f(z) \neq 0.$$

Since f is continuous and $f(z_1) < 0$ it follows that

$$\bigwedge_{\substack{z \\ |z|>r_{\mathcal{E}}}} f(z) < 0. \tag{3.4}$$

On the other hand, for every polynomial Q of degree ≥ 2 it is obvious that $\lim\limits_{z\to\infty} \left|\frac{Q(z)}{z}\right| = \infty$. In particular, we have

$$\bigwedge_{r>0} \bigvee_{\substack{z_2 \\ |z_2|>r}} |Q(z_2)| > |z_2|.$$

If we set $Q = P$ this guarantees the existence of a point w with $|w| > r_{\mathcal{E}}$ and $f(w) > 0$, which is a contradiction to (3.4). So the assumptions in cases 1 and 2 were wrong, which proves (3.3).

Next we will show that

$$|P^n(z_1)| \to \infty. \tag{3.5}$$

To prove this we consider the sequence $(|P^n(z_1)|)_{n=0}^{\infty}$. Because of (3.3) this sequence is strictly increasing. So we only have to show that it cannot converge. Now suppose that it does converge, i. e., that there exists a $c > r_{\mathcal{E}}$ such that $|y_n| \xrightarrow{n\to\infty} c$, where $y_n = P^n(z_1)$. But then $(y_n)_{n=0}^{\infty}$ is a bounded sequence in $\mathbb{C}$. Hence, there exists a convergent subsequence (y_{n_k}), i. e., there is a $y^* \in \mathbb{C}$ with $y_{n_k} \xrightarrow{k\to\infty} y^*$ and $|y^*| = c$. As P is continuous, $(P(y_{n_k})) = (y_{n_k+1})$ is also a convergent subsequence with $P(y_{n_k}) \to P(y^*)$ and $|P(y_{n_k})| \to c$, i. e., $|P(y^*)| = c$. Therefore $|P(y^*)| = |y^*|$, hence, $y^* \in \mathcal{E}$. But this is a contradiction to $|y^*| = c > r_{\mathcal{E}}$, which proves (3.5). As z_1 was arbitrarily chosen, the prove is complete. □

In general, it is not easy to compute $r_{\mathcal{E}}$. For this reason we derive some upper estimates for $r_{\mathcal{E}}$ that are easier to compute. One result is the following:

Corollary 3.3. *Let $P(z) = \sum\limits_{j=0}^{d} a_j z^j$. For real $r > 0$ we define*

$$Q(r) = |a_d| r^d - \sum_{j=0}^{d-1} |a_j| r^j. \tag{3.6}$$

Let r_Q be the largest real fixed point of Q. Then $r_Q \geq r_{\mathcal{E}}$, i. e., r_Q is an escape radius.

Proof. First we have to show the existence of a real fixed point. As $Q(0) = -|a_0| \leq 0$, $Q(r) \geq r$ for sufficiently large r and as Q is continuous it is obvious that such a fixed point exists and that

$$\bigwedge_{r>r_Q} Q(r) > r. \tag{3.7}$$

Next, we will transform our problem into polar coordinates, i. e., we will use the representation $z = re^{i\varphi}$. So we have

$$P(z) = a_d r^d e^{id\varphi} + a_{d-1} r^{d-1} e^{i(d-1)\varphi} + \cdots + a_0.$$

If $r > r_Q$ then, by (3.7), the following inequalities hold:

$$|P(z)| \geq |a_d| r^d - \sum_{j=0}^{d-1} |a_j| r^j > r = |z|.$$

Hence, for *every* z with $|z| > r_Q$ we have $|P(z)| > |z|$. Therefore, r_Q is greater than or equal to the escape radius $r_{\mathcal{E}}$ defined in Theorem 3.2; this proves the corollary. □

The fixed point r_Q of Q defined in Corollary 3.3 can be calculated much easier than $r_{\mathcal{E}}$ (at least numerically) and is often quite close to it. Sometimes such an r_Q can even be calculated by hand as we will see in Example 3.7.

For certain polynomials the radius r_Q is even the optimal escape radius as shown in the following corollary:

Corollary 3.4. *Let* $P(z) = \sum_{j=0}^{d} a_j z^j$ *with* $a_i \in \mathbb{R}$ *for* $i = 0, \ldots, d$, $a_d > 0$ *and* $a_j \leq 0$ *for* $j = 0, \ldots, d-1$. *Then* $r_Q = r_{opt}$.

Proof. In this special case we have $Q(r) = P(r)$ with Q defined as in (3.6). Thus

$$P(r_Q) = Q(r_Q) = r_Q.$$

In particular we have

$$\bigwedge_{n\in\mathbb{N}} P^n(r_Q) = r_Q.$$

On the other hand, since by Corollary 3.3 r_Q is an escape radius,

$$\bigwedge_{\varepsilon>0} P^n(r_Q + \varepsilon) \xrightarrow{n\to\infty} \infty.$$

It is now obvious that (P^n) can't be normal at r_Q. Hence, r_Q must be an element of $\mathcal{J}$. On the other hand r_Q can be chosen as an escape radius. This implies that $r_Q = r_{\text{opt}}$. □

To quickly get an estimate of the escape radius r_Q we can use the following lemma (see [Sto, p. 270]):

Lemma 3.5. *For all zeros ξ_i of a polynomial*

$$p(z) = b_d z^d + b_{d-1} z^{d-1} + \cdots + b_0$$

the following inequalities hold:

$$
\begin{aligned}
|\xi_i| &\le r_1(b) = \max\left\{\left|\frac{b_0}{b_d}\right|, 1 + \left|\frac{b_1}{b_d}\right|, \ldots, 1 + \left|\frac{b_{d-1}}{b_d}\right|\right\} \\
|\xi_i| &\le r_2(b) = \max\left\{1, \sum_{i=0}^{d-1}\left|\frac{b_i}{b_d}\right|\right\} \\
|\xi_i| &\le r_3(b) = \max\left\{\left|\frac{b_0}{b_1}\right|, 2\left|\frac{b_1}{b_2}\right|, \ldots, 2\left|\frac{b_{d-1}}{b_d}\right|\right\} \\
|\xi_i| &\le r_4(b) = \sum_{i=1}^{d}\left|\frac{b_{i-1}}{b_i}\right| \\
|\xi_i| &\le r_5(b) = 2 \cdot \max\left\{\left|\frac{b_{d-1}}{b_d}\right|, \sqrt{\left|\frac{b_{d-2}}{b_d}\right|}, \sqrt[3]{\left|\frac{b_{d-3}}{b_d}\right|}, \ldots, \sqrt[d]{\left|\frac{b_0}{b_d}\right|}\right\}
\end{aligned}
$$

If we combine the results of Corollary 3.3 and Lemma 3.5 we immediately get the desired estimate for r_Q in terms of the coefficients of the polynomial P:

Corollary 3.6. *Let $P(z) = \sum_{j=0}^{d} a_j z^j$ and let $b_d = |a_d|$, $b_i = -|a_i|$ for $i = 0, 2, 3, \ldots, d-1$ and $b_1 = -|a_1| - 1$. Then the following value for r can be chosen as an escape radius:*

$$\min_{1 \le i \le 5}\{r_i(b)\} \ge r_Q \ge r_{\mathcal{E}}, \tag{3.8}$$

with $r_i(b)$ defined as in Lemma 3.5.

Proof. Let Q be as in Corollary 3.3. Finding the largest fixed point of $Q(r)$ is equivalent to finding the largest root of $Q(r) - r$. Now apply Lemma 3.5 to $Q(r) - r$. □

To illustrate the quality of these estimates for the escape radius $r_{\mathcal{E}}$ let us consider the following example:

Example 3.7. We consider the complex polynomial

$$P(z) = \frac{1}{16} z^5 + \frac{5}{4} z$$

Using Corollary 3.3 we obtain

$$
\begin{aligned}
Q(r) = \frac{1}{16} r^5 - \frac{5}{4} r &\overset{!}{=} r \\
\frac{1}{16} r^5 - \frac{9}{4} r &= 0 \quad | \cdot 16 \\
r(r^4 - 36) &= 0 \\
r(r^2 - 6)(r^2 + 6) &= 0.
\end{aligned}
$$

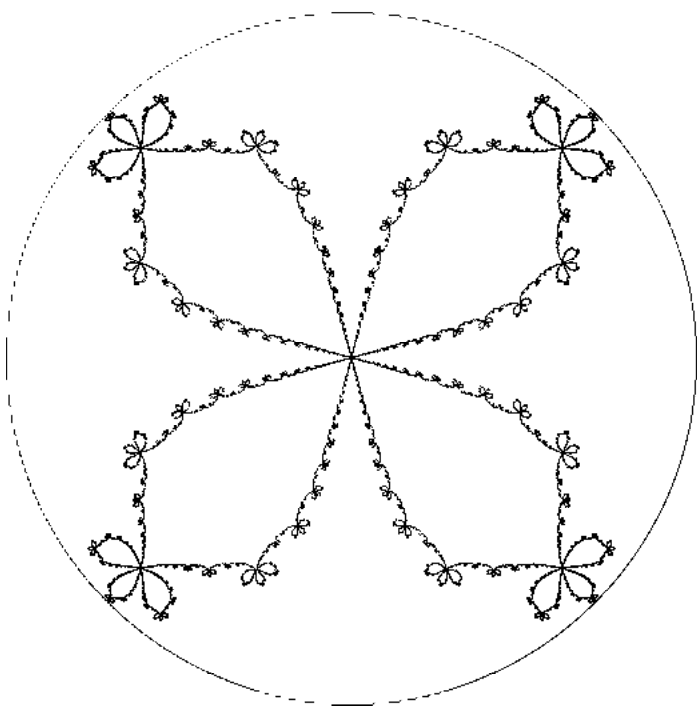

Figure 3.1: Julia set and optimal escape radius for $P(z) = \frac{1}{16}z^5 + \frac{5}{4}z$.

The largest real root is obviously $\sqrt{6}$. In this example $r_Q = \sqrt{6}$ is identical with the optimal escape radius r_{opt} because if we let $z = \sqrt{6}\, e^{i\pi/4}$, then $P(z) = -z$ and $P^2(z) = z$. As $|(P^2)'(z)| = |P'(z)| \cdot |P'(-z)| > 1$, z is a repelling periodic point and is therefore an element of $\mathcal{J}$ (see Figure 3.1).

Now let us use the escape radius of Corollary 3.6: In this case we have $b_0 = b_2 = b_3 = b_4 = 0$, $b_1 = -\frac{5}{4} - 1 = -\frac{9}{4}$ and $b_5 = \frac{1}{16}$. Computing r_i we obtain

$$\begin{aligned}
r_1(b) &= \max\{0, 1 + 16 \cdot \tfrac{9}{4}, 1, 1, 1\} = 37, \\
r_2(b) &= \max\{1, 16 \cdot \tfrac{9}{4}\} = 36, \\
r_3(b) &= r_4(b) = \infty, \\
r_5(b) &= 2 \cdot \max\{0, 0, 0, \sqrt[4]{36}, 0\} = 2\sqrt{6}.
\end{aligned}$$

Obviously, the minimum of these values is $2\sqrt{6}$, which is a useful approximation of the optimal escape radius $\sqrt{6}$. $\diamondsuit$

For quadratic polynomials the escape radius r_Q and its estimate in Corollary 3.6 can be calculated easily. Even $r_{\mathcal{E}}$ can be calculated by hand as we will see in the following example. As every quadratic polynomial is conjugate with a linear function to a polynomial of the form $P_c(z) = z^2 + c$ (see Example 1.10) it is sufficient to consider only such polynomials.

Example 3.8. We consider the family of polynomials

$$P_c(z) = z^2 + c$$

with $c \in \mathbb{C}$. The estimates of r_Q in Corollary 3.6 lead to the following values ($b_0 = -|c|$, $b_1 = -1$, $b_2 = 1$):

$$\begin{aligned}
r_1(b) &= \max\{|c|, 2\},\\
r_2(b) &= \max\{1, 1+|c|\} = 1+|c|,\\
r_3(b) &= \max\{|c|, 2\} = r_1(b),\\
r_4(b) &= |c|+1 = r_2(b),\\
r_5(b) &= 2\cdot\max\{1, \sqrt{|c|}\}.
\end{aligned}$$

Although these results are already quite useful, we get much better results when we calculate r_Q. Using (3.6) in Corollary 3.3 we get

$$\begin{aligned}
Q(z) = z^2 - |c| &\overset{!}{=} z\\
z^2 - z - |c| &= 0.
\end{aligned}$$

As we are interested in the largest real fixed point, the result is

$$r_Q = \frac{1}{2} + \sqrt{\frac{1}{4} + |c|}. \tag{3.9}$$

For a comparison of r_Q and $r_i(b)$ for different values of c see Table 3.1.

c	$r_1(b) = r_3(b)$	$r_2(b) = r_4(b)$	$r_5(b)$	$\min r_i(b)$	r_Q
0	2	1	2	1	1
0.25	2	1.25	2	1.25	1.2071
-1	2	2	2	2	1.61803
3	3	4	3.4641	3	2.3028

Table 3.1: Estimates for the escape radius for $P(z) = z^2 + c$.

Now we will try to calculate $r_\mathcal{E}$. To do so we introduce real variables x, y, u, v with $z = x + iy$ and $c = u + iv$. First we take a closer look at the set $\mathcal{E}$ defined in Theorem 3.2:

$$\begin{aligned}
& & |P_c(z)| &= |z|\\
&\iff & |P_c(z)|^2 &= |z|^2\\
&\iff & |(x+iy)^2 + (u+iv)|^2 &= |x+iy|^2\\
&\iff & |x^2 - y^2 + u + i(2xy+v)|^2 &= |x+iy|^2\\
&\iff & (x^2 - y^2 + u)^2 + (2xy+v)^2 &= x^2 + y^2 \qquad (3.10)
\end{aligned}$$

To compute $r_\mathcal{E}$ we have to maximize $r = \sqrt{x^2+y^2}$ under the constraint that r satisfies equation (3.10). This leads to a constrained optimization problem, and, in general, it is only possible to solve this numerically (although it is possible to calculate $r_\mathcal{E}$ in a completely different way as we will see later). But for certain values of c we can compute further results by hand.

Let us now assume that c is real. Then equation (3.10) becomes (as $u = c$)

$$(x^2 - y^2 + c)^2 + 4x^2y^2 = x^2 + y^2. \tag{3.11}$$

This equation can be rewritten as

$$c^2 + (x^2 + y^2)^2 + (2x^2 - 2y^2)c = x^2 + y^2 \tag{3.12}$$

or (as $r^2 = x^2 + y^2$)

$$c^2 + r^4 + (2x^2 - 2y^2)c = r^2. \tag{3.13}$$

First we consider the case $c \geq 0$. It is now reasonable that $x = 0$ if we want to maximize r under restriction (3.13). Using equation (3.12) we get

$$y^4 - (2c + 1)y^2 + c^2 = 0.$$

Taking into account that in this situation $y^2 = r_{\mathcal{E}}^2$ and using equation (3.9) we finally get

$$r_{\mathcal{E}}^2 = y^2 = \frac{2c+1}{2} + \sqrt{\frac{(2c+1)^2}{4} - c^2} = c + \frac{1}{2} + \sqrt{c + \frac{1}{4}} = \left(\frac{1}{2} + \sqrt{\frac{1}{4} + c}\right)^2 = r_Q^2.$$

If $c < 0$ the situation is nearly the same. If we let $y = 0$ we also get $r_{\mathcal{E}} = r_Q$.

This result can also be established the other way around. Again, we first consider the case $c \geq 0$. By equation (3.9) we have $r_Q = \frac{1}{2} + \sqrt{\frac{1}{4} + c}$. Now we choose $z_0 = i \cdot r_Q$. Then

$$P(z_0) = -r_Q^2 + c = -\left(\frac{1}{4} + \sqrt{\frac{1}{4} + c} + \frac{1}{4} + c\right) + c = -\left(\frac{1}{2} + \sqrt{\frac{1}{4} + c}\right).$$

Hence, $|P(z_0)| = |z_0| = r_Q$; therefore z_0 is an element of $\mathcal{E}$ (defined in Theorem 3.2). But then (by definition of $r_{\mathcal{E}}$) $r_{\mathcal{E}} \geq |z_0| = r_Q$. On the other hand, we know, by Corollary 3.3, that $r_Q \geq r_{\mathcal{E}}$ which now gives the desired result that $r_Q = r_{\mathcal{E}}$.
If $c < 0$ we let $z_0 = r_Q = \frac{1}{2} + \sqrt{\frac{1}{4} - c}$ and obtain

$$P(z_0) = r_Q^2 + c = \frac{1}{4} + \sqrt{\frac{1}{4} - c} + \frac{1}{4} - c + c = \frac{1}{2} + \sqrt{\frac{1}{4} - c} = z_0. \tag{3.14}$$

This again leads to $r_Q = r_{\mathcal{E}}$.

In summary, the following assertion is proved:

$$P(z) = z^2 + c \quad \wedge \quad c \in \mathbb{R} \quad \Longrightarrow \quad r_{\mathcal{E}} = r_Q. \tag{3.15}$$

In equation (3.14) we saw that for real $c < 0$ $P(z_0) = z_0$ for $z_0 = r_Q$. As $P'(z_0) = 2z_0 = 1 + \sqrt{1 - 4c} > 1$ we conclude that z_0 is a repelling fixed point of P and therefore an element of $\mathcal{J}_P$. Hence, $r_Q = r_{\text{opt}}$. This equality is obviously also true for $c = 0$, i. e., we have

$$P(z) = z^2 + c \quad \wedge \quad c \in \mathbb{R}_0^- \quad \Longrightarrow \quad r_{\mathcal{E}} = r_Q = r_{\text{opt}}. \tag{3.16}$$

For $c > 0$ this is mostly not true (see Figure 3.2).

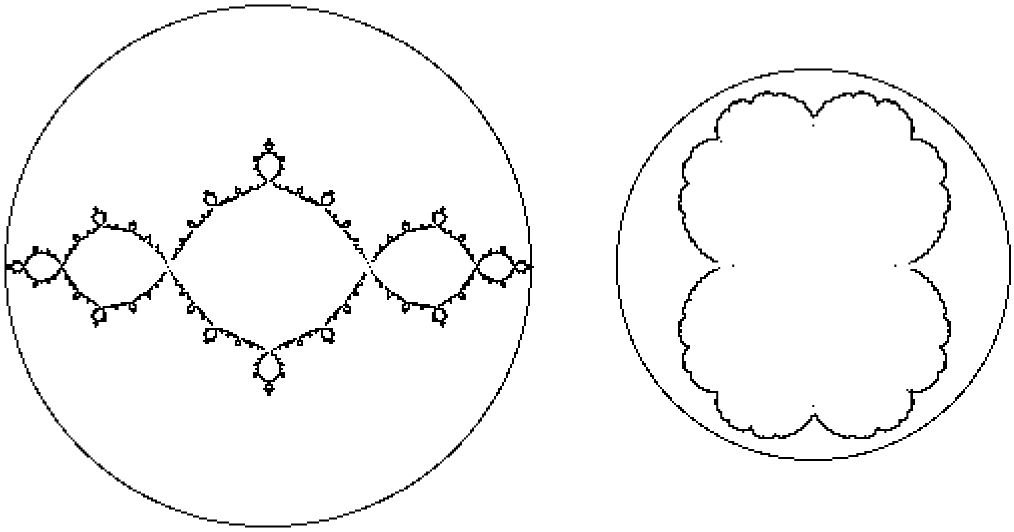

Figure 3.2: Julia set and $r_{\mathcal{E}} = r_Q$ for $P(z) = z^2 - 1$ and $P(z) = z^2 + \frac{1}{4}$.

It was quite instructive to consider the case when c is real, but it is possible to prove equation (3.15) for arbitrary $c \in \mathbb{C}$. To do so we use polar coordinates. Let $c = |c| \cdot e^{i\varphi}$ and let $z_0 = r_Q e^{i\psi}$ where $r_Q = \frac{1}{2} + \sqrt{\frac{1}{4} + |c|}$ by equation (3.9). Hence,

$$P(z_0) = z_0^2 + c = \left(\frac{1}{2} + |c| + \sqrt{\frac{1}{4} + |c|} \right) e^{i2\psi} + |c| e^{i\varphi}.$$

Now we choose ψ such that $e^{i2\psi} = -e^{i\varphi}$, i. e., $2\psi = \varphi + \pi$ or $\psi = \frac{\varphi + \pi}{2}$. Then we have

$$P(z_0) = e^{i2\psi} \left(\frac{1}{2} + |c| + \sqrt{\frac{1}{2} + |c|} - |c| \right) = e^{i2\psi} r_Q.$$

Hence, $|P(z_0)| = |z_0|$. As in the real case this leads to the equation $r_{\mathcal{E}} = r_Q$. In summary, we have

$$P(z) = z^2 + c, \quad c \in \mathbb{C} \quad \Longrightarrow \quad r_{\mathcal{E}} = r_Q. \tag{3.17}$$

$\diamondsuit$

Up to now, we have only considered *upper* bounds for the optimal escape radius. Now we will also give a lower bound. For a proof of the following lemma see [Mard, p. 123].

Lemma 3.9. *Let* $p(z) = \sum_{i=0}^{d} b_i z^i$ *and let* z_1 *be the zero of* p *with largest modulus. Then*

$$\alpha = \max_{1 \le k \le d} \left| \frac{b_{d-k}}{b_d} \binom{d}{k}^{-1} \right|^{1/k} \le |z_1|. \tag{3.18}$$

An immediate consequence is

Corollary 3.10. *Let $P(z) = \sum_{i=0}^{d} a_i z^i$ and define $b_i = a_i$ for $i = 0, 2, \ldots, d$ and $b_1 = a_1 - 1$. Then α as defined in (3.18) is a lower bound for every escape radius for P.*

Proof. α is a lower bound for the zero of $p(z) = \sum_{i=0}^{d} b_i z^i$ with the largest modulus and thus a lower bound for the fixed point of P with largest modulus. Since every fixed point is an element of $\mathcal{K}$ and $\mathcal{J} = \partial\mathcal{K}$ we have

$$\alpha \leq \max_{z \in \mathcal{J}} |z| \leq r_{\text{opt}}.$$

$\square$

Remark. As $\mathcal{J}_P = \mathcal{J}_{P^m}$, Corollary 3.10 can be applied to any iterate of P which mostly gives better results, although calculating higher iterates soon yields polynomials of very high degrees.

Example 3.11. Let us again consider the quadratic polynomials $P_c(z) = z^2 + c$. First we apply Corollary 3.10 to P_c itself. Here we have $b_0 = c, b_1 = -1$ and $b_2 = 1$ and thus $(b_d = b_2 = 1)$

$$\alpha = \max_{1 \leq k \leq 2} \left| b_{2-k} \binom{2}{k}^{-1} \right|^{1/k} = \max\left\{ \frac{1}{2}, \sqrt{|c|} \right\}.$$

As suggested in the remark above we can apply Corollary 3.10 to any iterate of P; we will do so for the second iterate:

$$P_c^2(z) = (z^2 + c)^2 + c = z^4 + 2cz^2 + c^2 + c,$$

resulting in the following values for the b_i's:

$$b_4 = 1, \quad b_3 = 0, \quad b_2 = 2c, \quad b_1 = -1, \quad b_0 = c^2 + c.$$

If we calculate the new value α_1 we obtain

$$\alpha_1 = \max\left\{ 0, \sqrt{\frac{|c|}{3}}, \sqrt[3]{\frac{1}{4}}, \sqrt[4]{|c^2 + c|} \right\}.$$

A combination of α and α_1 shows that

$$\alpha_2 = \max\{\alpha, \alpha_1\} = \max\left\{ \sqrt{|c|}, \sqrt[3]{\frac{1}{4}}, \sqrt[4]{|c^2 + c|} \right\} \tag{3.19}$$

is a lower bound for the optimal escape radius. Table 3.2 gives an overview of these estimates for some values of c in comparison to the escape radius r_Q (see Example 3.8). $\diamondsuit$

3.2 Converge Radius

In the previous section we computed values for an escape radius and we then knew that outside a disk with such a radius no point in the complex plane could be an element of $\mathcal{J}_P$ or $\mathcal{K}_P$. Now we want to compute domains that must be contained in $\mathcal{K}_P$. This leads to the following definition:

c	$\sqrt[3]{\frac{1}{4}}$	$\sqrt{\lvert c\rvert}$	$\sqrt[4]{\lvert c^2+c\rvert}$	α_2	r_Q
0	0.63	0	0	0.63	1
0.25	0.63	0.5	0.7477	0.7477	1.2071
-1	0.63	1	0	1	1.61803
3	0.63	1.732	1.8612	1.8612	2.3028
i	0.63	1	1.0905	1.0905	1.61803

Table 3.2: Lower bounds for the escape radius for $P_c(z) = z^2 + c$.

Definition 3.12. Let P be a complex polynomial and let $\tilde{z}$ be an attracting fixed point. A radius $\tilde{r}$ that satisfies

$$B(\tilde{z}, \tilde{r}) \subset A^*(\tilde{z}) \tag{3.20}$$

is called a *converge radius* for the fixed point $\tilde{z}$. The largest converge radius is called *optimal converge radius* $\tilde{r}_{\text{opt}}$, i. e.,

$$\tilde{r}_{\text{opt}} = \max\{\tilde{r} \mid \tilde{r} \text{ satisfies condition (3.20)}\}. \tag{3.21}$$

Remark. We will always use a tilde to distinguish a converge radius, $\tilde{r}$, from an escape radius, r.

Now we are ready to formulate the first result:

Theorem 3.13. *Let P be a complex polynomial of degree ≥ 2 and let $\tilde{z}$ be an attracting fixed point of P. In addition let us set $\mathcal{G} = \{z \in \mathbb{C} \mid |P(z + \tilde{z}) - \tilde{z}| = |z|\}$ and $\tilde{r}_{\mathcal{G}} = \min_{z \in \mathcal{G}\setminus\{0\}} |z|$. Then $\tilde{r}_{\mathcal{G}}$ satisfies equation (3.20), i. e., $\tilde{r}_{\mathcal{G}}$ is a converge radius for $\tilde{z}$.*

Proof. First we will prove the assertion for the case when $\tilde{z} = 0$ is an attracting fixed point of P. In this case the set $\mathcal{G}$ is the same as the set $\mathcal{E}$ in Theorem 3.2. As in the proof of Theorem 3.2 let $f(z) = |P(z)| - |z|$. Since 0 is an attracting fixed point, $|P'(0)| < 1$. Therefore, we have:

$$|P'(0)| = \lim_{\substack{z\to 0\\ z\in\mathbb{C}}} \left|\frac{P(z)}{z}\right| < 1 \quad \Longrightarrow \quad \bigvee_{\varepsilon_0>0} \bigwedge_{\substack{z\\ |z|\leq\varepsilon_0}} \left|\frac{P(z)}{z}\right| < 1,$$

i. e., $f(z) < 0$ for $z \in \overline{B(0, \varepsilon_0)}$. This leads to (see Lemma 1.4)

$$\bigvee_{\tilde{\varepsilon}>0} \bigwedge_{z\in U_{\tilde{\varepsilon}}(0)} P^n(z) \to 0 \quad \wedge \quad f(z) < 0, \tag{3.22}$$

which also shows that $\tilde{r}_G$ is strictly greater than 0. By definition of $\tilde{r}_{\mathcal{G}}$ we have

$$\bigwedge_{\substack{z\neq 0\\ |z|<\tilde{r}_{\mathcal{G}}}} f(z) \neq 0.$$

Along with the continuity of f and (3.22) this implies

$$\bigwedge_{\substack{z\neq 0\\ |z|<\tilde{r}_{\mathcal{G}}}} f(z) = |P(z)| - |z| < 0. \tag{3.23}$$

Therefore $P(B(0,\tilde r_{\mathcal G})) \subset B(0,\tilde r_{\mathcal G})$ and thus $B(0,\tilde r_{\mathcal G}) \subset \mathcal K$.

Now we claim

$$z_1 \in B(0,\tilde r_{\mathcal G})\backslash\{0\} \quad\Longrightarrow\quad P^n(z_1) \to 0. \tag{3.24}$$

Indeed, by (3.23) $(|P^n(z_1)|)_{n=1}^{\infty}$ is a strictly decreasing sequence with lower bound 0. Therefore it converges. As in the proof of Theorem 3.2 one can show that this sequence cannot converge to any value strictly greater than 0. Thus it must converge to 0. But this obviously implies $P^n(z_1) \to 0$, which proves our claim. The assertion of the theorem for $\tilde z = 0$ follows now immediately from (3.24).

The case when $\tilde z \neq 0$ can be transformed to the case $\tilde z = 0$ as follows: Let $\tilde z \neq 0$ be an attracting fixed point for P (i. e., $P(\tilde z) = \tilde z$ and $|P'(\tilde z)| < 1$), and let us set

$$\tilde P(z) = P(z+\tilde z) - \tilde z. \tag{3.25}$$

With these settings we have

$$\begin{aligned} \tilde P(0) &= P(\tilde z) - \tilde z = 0, \\ |\tilde P'(0)| &= \left| P'(z+\tilde z)\big|_{z=0} \cdot 1 \right| = |P'(\tilde z)| < 1. \end{aligned}$$

Therefore, we can now apply our already proven result to the polynomial $\tilde P$ and $\mathcal G = \left\{ z \in \mathbb C \mid |\tilde P(z)| = |z| \right\}$ which gives, see (3.24),

$$\bigwedge_{z\in B(0,\tilde r_{\mathcal G})} \tilde P^n(z) \to 0.$$

Since

$$\tilde P^n(z) = P^n(z+\tilde z) - \tilde z,$$

which can easily be shown by induction, we obtain

$$\bigwedge_{z\in B(0,\tilde r_{\mathcal G})} P^n(z+\tilde z) \to \tilde z,$$

which is obviously equivalent to the assertion of the theorem. □

As in the previous section, where we tried to calculate a good escape radius, it is again very difficult to compute $\tilde r_{\mathcal G}$. So we will give a lower estimate for $\tilde r_{\mathcal G}$ that is easier to compute.

Corollary 3.14. *Let $P(z) = \sum_{i=0}^{d} a_i z^i$, and let $\tilde z$ be an attracting fixed point of P. Furthermore, let $\tilde P(z) = \sum_{i=0}^{d} b_i z^i = P(z+\tilde z) - \tilde z$ and*

$$\tilde Q(r) = \Big(\sum_{i=1}^{d} |b_i| r^{i-1}\Big) - 1. \tag{3.26}$$

Then $\tilde r_{\tilde Q}$ defined by

$$\tilde r_{\tilde Q} = \min\{r \in \mathbb R^+ \mid \tilde Q(r) = 0\} \tag{3.27}$$

satisfies $\tilde r_{\tilde Q} < \tilde r_{\mathcal G}$, i. e., $\tilde r_{\tilde Q}$ is a converge radius for the fixed point $\tilde z$.

Proof. $\tilde{z}$ is a fixed point of P, therefore 0 is a fixed point of $\tilde{P}$, i. e., $\tilde{P}(0) = 0$. Hence, b_0 must be 0. When we calculate $\tilde{r}_{\mathcal{G}}$ as in Theorem 3.13 we consider the set $\mathcal{G} = \{z \in \mathbb{C} \mid |\tilde{P}(z)| = |z|\}$ and $\tilde{r}_{\mathcal{G}} = \min_{z \in \mathcal{G} \setminus \{0\}} |z|$. Since 0 is an attracting fixed point of $\tilde{P}$ we know that $|\tilde{P}'(0)| < 1$. From this we can see that

$$\bigwedge_{0<|z|<\tilde{r}_{\mathcal{G}}} |\tilde{P}(z)| < |z|.$$

Therefore, we get a radius less than or equal to $\tilde{r}_{\mathcal{G}}$ when we use an upper estimate for $|\tilde{P}(z)|$ instead of $|\tilde{P}(z)|$:

$$|\tilde{P}(z)| \le \sum_{i=1}^{d} |b_i||z^i| \stackrel{!}{=} |z|$$

So we are interested in the smallest real fixed point strictly greater than 0 of the polynomial $\sum_{i=1}^{d} |b_i| r^i$. This is obviously equivalent to finding the smallest positive root of the polynomial $\tilde{Q}(r) = \left(\sum_{i=1}^{d} |b_i| r^{i-1}\right) - 1$. This proves the corollary. $\square$

Example 3.15. Let us consider the family of polynomials $P_c(z) = z^2 + c$. We will consider only those values for c, for which P_c has an attracting fixed point, i. e., c is inside the main cardioid of the Mandelbrot set $\mathcal{M}$ (see e. g. [CG, p. 126]). The two fixed points are given by

$$\tilde{z}_c = (1 \pm \sqrt{1-4c})/2. \tag{3.28}$$

It depends on c which one of the two fixed points is attracting.

As it was the case with the escape radius, we can hardly compute $\tilde{r}_{\mathcal{G}}$ directly from its definition. But we can compute $\tilde{r}_{\tilde{Q}}$. First of all, let us compute $\tilde{P}_c(z)$:

$$\begin{aligned}
\tilde{P}_c(z) &= P_c(z + \tilde{z}_c) - \tilde{z}_c \\
&= z^2 + 2\tilde{z}_c z + \tilde{z}_c^2 + c - \tilde{z}_c \\
&= z^2 + (1 \pm \sqrt{1-4c})z + \frac{1}{4}(1 + (1-4c) \pm 2\sqrt{1+4c}) - \frac{1}{2}(1 \pm \sqrt{1-4c}) + c \\
&= z^2 + (1 \pm \sqrt{1-4c})z.
\end{aligned}$$

From this we can immediately derive $\tilde{Q}$:

$$\tilde{Q}(r) = r + \left|1 \pm \sqrt{1-4c}\right| - 1 \stackrel{!}{=} 0. \tag{3.29}$$

Hence, the desired result is

$$\tilde{r}_{\tilde{Q}} = 1 - \underbrace{\left|1 \pm \sqrt{1-4c}\right|}_{=|P_c'(\tilde{z}_c)|<1} > 0. \tag{3.30}$$

Next we will show that for the polynomials P_c the converge radius $\tilde{r}_{\tilde{Q}}$ is the same as the converge radius $\tilde{r}_{\mathcal{G}}$. The proof can be done as follows: Let $\tilde{z}_c = \frac{1}{2}\left|1 \pm \sqrt{1-4c}\right| \cdot e^{i\varphi}$

(polar representation of the fixed point $\tilde{z}_c$). Then choose $z_0 = \tilde{r}_{\tilde{Q}} \cdot e^{i\varphi}$. With these settings we have

$$\begin{aligned} |\tilde{P}_c(z_0)| &= |P_c(z_0+\tilde{z}_c) - \tilde{z}_c| = |z_0^2 + 2\tilde{z}_c z_0| = |z_0| \cdot |z_0 + 2\tilde{z}_c| \\ &= |z_0| \cdot \Big| \Big(1 - |1 \pm \sqrt{1-4c}|\Big) e^{i\varphi} + |1 \pm \sqrt{1-4c}| e^{i\varphi} \Big| \\ &= |z_0| \cdot |e^{i\varphi}| = |z_0|. \end{aligned}$$

Therefore z_0 is an element of $\mathcal{G}$. Hence, $\tilde{r}_{\mathcal{G}} \le |z_0| = \tilde{r}_{\tilde{Q}}$. On the other hand we know by Corollary 3.14 that $\tilde{r}_{\tilde{Q}} \le \tilde{r}_{\mathcal{G}}$, which yields $\tilde{r}_{\tilde{Q}} = \tilde{r}_{\mathcal{G}}$. In summary we have proven

$$c \in W \quad \Longrightarrow \quad \tilde{r}_{\tilde{Q}} = \tilde{r}_{\mathcal{G}}, \tag{3.31}$$

where W denotes the main cardioid of the Mandelbrot set $\mathcal{M}$.

For certain values of c we can show even more than (3.31). Suppose c is an element of the real interval $[0, \frac{1}{4}]$. Then $1 - 4c > 0$ and thus

$$\tilde{r}_{\tilde{Q}} = 1 - \Big|1 - \underbrace{\sqrt{1-4c}}_{<1}\Big| = \sqrt{1-4c}.$$

Therefore we have

$$\begin{aligned} P_c(\tilde{z}_c + \tilde{r}_{\tilde{Q}}) &= P_c\Big(\frac{1-\sqrt{1-4c}}{2} + \sqrt{1-4c}\Big) = \Big(\frac{1+\sqrt{1-4c}}{2}\Big)^2 + c \\ &= \frac{1}{2} + \frac{1}{2}\sqrt{1-4c} = \tilde{z}_c + \tilde{r}_{\tilde{Q}}. \end{aligned}$$

Hence, $\tilde{z}_c + \tilde{r}_{\tilde{Q}}$ is a fixed point. On the other hand, for every positive ε (ε small enough) $\tilde{z}_c + \tilde{r}_{\tilde{Q}} - \varepsilon$ tends to $\tilde{z}_c$, because $\tilde{r}_{\tilde{Q}}$ is a converge radius. Therefore, $\tilde{z}_c + \tilde{r}_{\tilde{Q}}$ can't be an attracting fixed point, so it must be either repelling or rationally indifferent. But this implies that it is an element of $\mathcal{J}_{P_c}$. Obviously, the converge radius cannot be increased in this case. So we have proven

$$c \in [0, \tfrac{1}{4}] \quad \Longrightarrow \quad \tilde{r}_{\tilde{Q}} = \tilde{r}_{\mathcal{G}} = \tilde{r}_{\text{opt}}. \tag{3.32}$$

In Figure 3.3 the Julia sets with escape and converge radius are drawn for $c = -\frac{1}{2}$ and $c = \frac{1}{8}$. In the first case the escape radius is optimal but the same cannot be said of the converge radius ($= 2 - \sqrt{3} \approx 0.268$). In the second case the escape radius is not optimal, but the converge radius ($= 1/\sqrt{2} \approx 0.7071$) is — as it must be because of (3.32). ♢

Example 3.16. Let us consider the polynomial

$$P(z) = -\frac{3}{2}z^5 + 15z^4 - \frac{115}{2}z^3 + 105z^2 - 90z + 30.$$

This polynomial has attracting real fixed points for $z = 1, 2, 3$. If we calculate the escape radius we obtain a (numerical) value for r_Q of about 13.3, which is far from optimal (computer graphics suggest that $r = 4$ is an escape radius that can even be made smaller). But if we want to calculate a converge radius for e. g. the fixed point $z = 2$, the situation is quite different. In this case we have

$$\tilde{P}(z) = P(z+2) - 2 = -\frac{3}{2}z^5 + \frac{5}{2}z^3$$

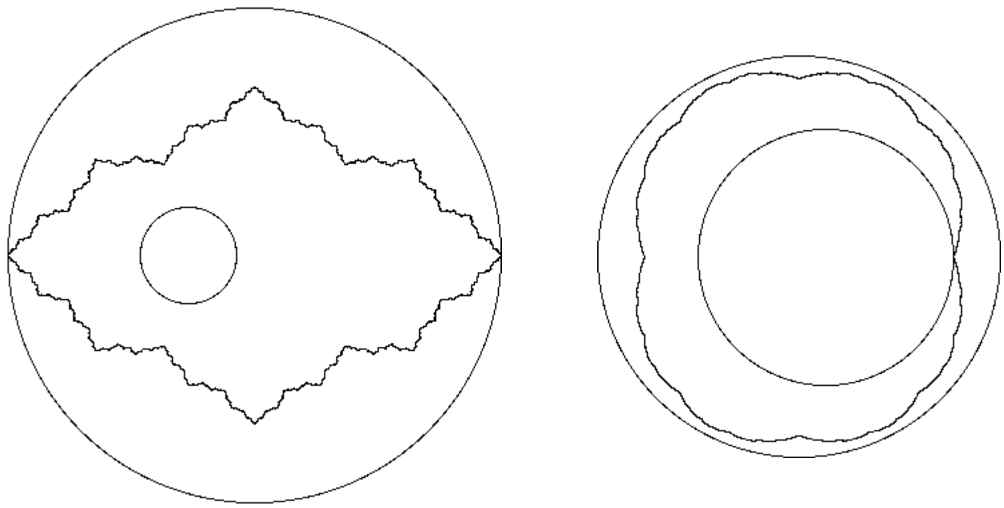

Figure 3.3: Julia set, r_Q and $\tilde{r}_{\tilde{Q}}$ for $P(z) = z^2 - \frac{1}{2}$ and $P(z) = z^2 + \frac{1}{8}$.

and

$$\tilde{Q}(r) = \frac{3}{2}r^4 + \frac{5}{2}r^2 - 1.$$

The roots of $\tilde{Q}$ are given by $\pm i\sqrt{2}$ and $\pm 1/\sqrt{3}$. Hence, the smallest positive real root is obviously $1/\sqrt{3} \approx 0.577$. This value is optimal, since

$$P(2 + 1/\sqrt{3}) = 2 - 1/\sqrt{3} \quad \text{and} \quad P(2 - 1/\sqrt{3}) = 2 + 1/\sqrt{3},$$

which shows that $\{2 + 1/\sqrt{3}, 2 - 1/\sqrt{3}\}$ is a repelling period-2 cycle. (The multiplier of this cycle is 11.111.) Figure 3.4 shows the Julia set and the converge radius. $\diamondsuit$

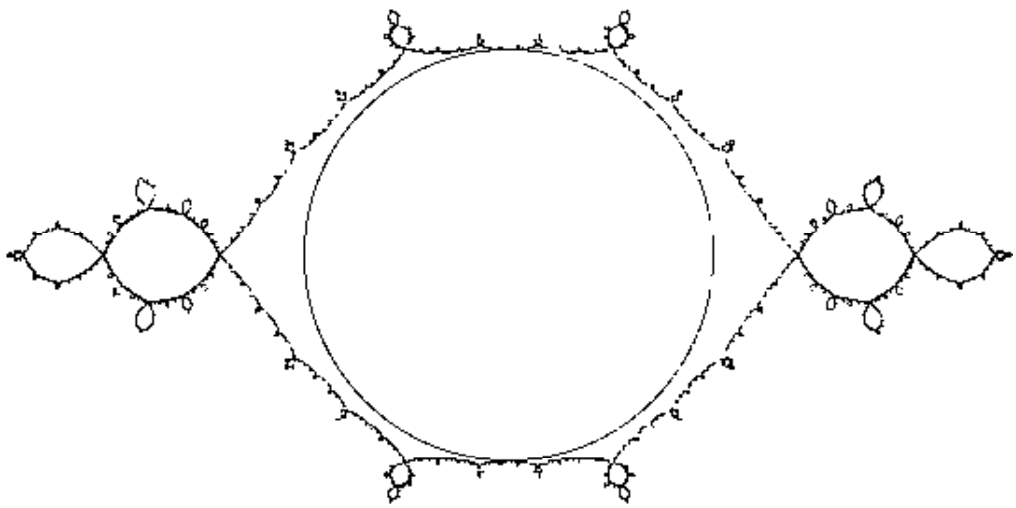

Figure 3.4: Julia set and $\tilde{r}_{\tilde{Q}}$ for $P(z) = -\frac{3}{2}z^5 + 15z^4 - \frac{115}{2}z^3 + 105z^2 - 90z + 30$.

Chapter 4

The Computer Program

In the previous chapters we have investigated some of the interesting aspects of Julia sets of complex polynomials. One reason why this field has become so increasingly popular in the last decades is the ability to visualize many of these theoretical results on the computer. An easy, entry-level introduction to these computer-related topics can be found in [De2]. Most shareware and freeware programs that float around on the internet have one big problem: They concentrate mainly on the beauty of the pictures and not on the mathematical background. Hence, they support thousands of colors, they can animate the colors, the pictures can be stored and printed, but they only deal with the Mandelbrot set and Julia sets of polynomials of the form $P(z) = z^2 + c$. And — with only a few exceptions — the pictures and figures in the books on Julia sets either deal again with polynomials of the form $P(z) = z^2 + c$, or they are taken from [PR] or [PS]. In these two books there are hundreds of beautiful pictures and there are also many ideas and suggestions on how to implement such programs on the computer. However even in [PS] most (interesting!) algorithms again only deal with the quadratic family, i. e., with polynomials of the form $P(z) = z^2 + c$.

But we wanted to be able to visualize many topics not covered by shareware programs (there are almost no commercial programs in this field), e. g. higher degree polynomials, postcritical sets, escape and converge radius, forward/backward orbit of any given point, etc. Therefore we decided to write our own program. All the computer pictures dealing with Julia sets in this thesis are generated with our program. It is written in LS FORTRAN on an Apple PowerMacintosh.

This chapter now shows how the implementation of all those results can be done.

4.1 General Aspects

When we draw the Julia set or something like that onto the screen, then the screen always represents a part of the complex plane. As the image on the monitor consists of pixels, each pixel corresponds to a certain point in the complex plane and vice versa. So we introduce two functions *pixeltopoint* and *pointtopixel* that compute the corresponding point in the complex plane for a given pixel on the screen and the corresponding pixel on the screen for a given point in the complex plane, respectively.

To be as versatile as possible, let *width* be the number of pixels horizontally and *height* the number of pixels vertically. Also, let $x_{\min}$ ($x_{\max}$) denote the lower (upper) bound for the real part of the sector of the complex plane that we want to investigate.

Further let $y_{\min}$ be the lower bound for the imaginary part. If we want to have an undistorted picture, we obtain

$$y_{\max} = y_{\min} + (x_{\max} - x_{\min}) \cdot \frac{height}{width}.$$

In practice it is not necessary to calculate $y_{\max}$; instead we compute a new variable

$$factor = width/(x_{\max} - x_{\min}),$$

so $1/factor$ is the "width of a pixel". We are now ready to give a pseudo-code for the functions *pixeltopoint* and *pointtopixel*:

function $pixeltopoint(i, j)$
begin
    $\Re(z) = i/factor + x_{\min}$
    $\Im(z) = (height - j)/factor + y_{\min}$
    return z
end

function $pointtopixel(z)$
begin
    $i = (\Re(z) - x_{\min}) \cdot factor$
    $j = height - (\Im(z) - y_{\min}) \cdot factor$
    return (i, j)
end

A first sensible way to get bounds $x_{\min}$, $x_{\max}$, and $y_{\min}$ would be to choose them in such a way that the escape radius just fits onto the screen. This guarantees that the whole Julia set is visible. The calculation of the escape radius r uses the definition of the escape radius r_Q in Corollary 3.3. With the help of *findroots* (see later) we calculate the zeros of the polynomial Q and then select the largest real one. We will use r as a global variable.

Besides r, $x_{\min}$, $x_{\max}$, $y_{\min}$, *width*, and *height* we will use additional global variables. First of all, d denotes the degree of the polynomial we are considering. The coefficients $a_0, a_1, \ldots, a_d$ of the polynomial $P(z) = a_d z^d + \cdots a_1 z + a_0$ are stored in the $(d+1)$-dimensional vector a.

To evaluate the polynomial P we always use Horners scheme, i. e., if we write $p = P(z)$ the following procedure is executed:

$p = a(d)$
do $i = d - 1, 0, -1$
    $p = p \cdot z + a(i)$
enddo
return p

In the course of the program we will often need to compute the roots of a polynomial. In this case it is advisable to use one of the routines available in many commercial or non-commercial libraries of mathematical functions, e. g. the routine `zroots` in [PTVF, p. 367]. In our pseudo-code we will call this routine *findroots*, i. e., if we write

result = *findroots*(a, d), then *result* is a d-dimensional vector consisting of the d roots of the polynomial whose coefficients are stored in a.

During calculation it is a good idea to store the color of the pixels in a matrix which we call *cmatrix*. On the one hand, this makes it easier to redraw a picture without recalculating everything, but it can also be used to speed up the program or do some other nice things which we will consider later. But this method is quite memory-intensive. Let us assume that each color corresponds to an integer and suppose there are not more than 32768 colors. Then we need a two-byte integer for each pixel. Common monitor resolutions usually do not exceed 1024×768 pixels; subtracting some pixels for a frame this leaves us with a total memory requirement of

$$2 \cdot 1000 \cdot 750 \text{ B } \approx 1.5 \text{ MB},$$

which might cause some difficulties in programming under MS-DOS.

4.2 Basic Algorithms

4.2.1 Drawing the Filled Julia Set $\mathcal{K}$

When we draw the filled Julia set we want to assign a color to each pixel on the screen. Therefore the outer loop looks like this:

```
subroutine drawimage
begin
    do j = 1,height
        do i = 1,width
            z = pixeltopoint(i, j)
            color = getcolor(z, r)
            cmatrix(i, j) = color
            pset(i, j, color)
        enddo
    enddo
end
```

The function *getcolor* now depends on what we actually want to draw. Usually we want to draw the filled Julia set $\mathcal{K}$ itself in black and the points outside $\mathcal{K}$ in a color corresponding to their "escape velocity". Thus the function looks like this:

```
function getcolor(z, r)
comment K black, C \ K in color
begin
    k = 0
    over = false
    do while not over
        if k = maxit or |z|^2 > r^2 then
            over = true
        else
            z = P(z)
```

$k = k + 1$
endif
enddo
if $k = maxit$ **then**
color = *black*
else
color = $f(k)$
endif
return *color*
end

First of all, note that we wrote $|z|^2 > r^2$ and not $|z| > r$ which would be mathematically equivalent; but in the first case you only have to compute $\Re(z)^2 + \Im(z)^2$ whereas in the second case you have to compute an additional square root. For the whole algorithm, the computer time difference is about 15–20% — not insignificant!!

With the choice of the function f you can control the appearance of the picture. If $f \equiv white$ you have a simple black and white picture where $\mathcal{K}$ is black and everything else is white. But even on a black and white monitor it is a better idea to set *color* to white for even k and to black for odd k or vice versa, depending on whether or not *maxit* is odd. *maxit* is a global variable and denotes the maximum number of iterations. Usually, a choice of $maxit = 30$ already gives quite a good impression of how the Julia set looks like. But be careful, in some cases it is necessary to compute hundreds or even thousands of iterations to get an idea of how the Julia set really looks like! The classical example for this effect is the polynomial $P(z) = z^2 + c$ with $c \in \mathbb{R}$ around 0.25. For $c \leq 0.25$ the Julia set is connected, for $c > 0.25$ it is totally disconnected. So if we want to draw the Julia set for $P(z) = z^2 + 0.25 + \varepsilon$ for very small $\varepsilon > 0$ we need many iterations to get a good picture. Figure 4.1 shows images of $\mathcal{K}$ for the polynomial $P(z) = z^2 + 0.251$ drawn with 30 and 100 iterations. Figure 5.3 shows color images of $\mathcal{K}$ for the same polynomial with 100 and 500 iterations.

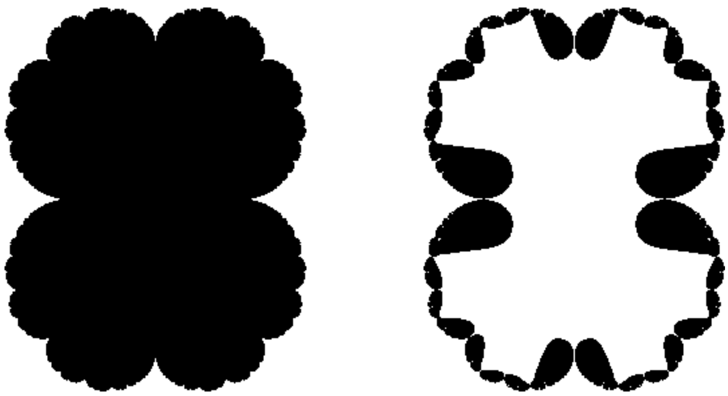

Figure 4.1: Filled Julia set for $P(z) = z^2 + 0.251$ drawn with 30 and 100 iterations.

Another possibility for the function *getcolor* is the following: This time we are not interested in how quickly a point in $A(\infty)$ escapes, instead we are interested in what happens with the points in $\mathcal{K}$. The pseudo-code changes only marginally, but the outcome is quite different:

function *getcolor*(z, r)

```
comment K in color, C \ K white
begin
    k = 0
    over = false
    do while not over
        if k = maxit or |z|^2 > r^2 then
            over = true
        else
            z = P(z)
            k = k + 1
        endif
    enddo
    if k = maxit then
        color = f(|z|)
    else
        color = white
    endif
    return color
end
```

With this method it is quite easy to visualize basins of attraction of cycles. As an example consider the polynomial $P(z) = z^2 - 1$. It is easy to see that there is a superattracting period-2 cycle consisting of 0 and -1 (and by Corollary 2.11 we know that there cannot be any further attracting cycle). Figures 5.4 and 5.5 show the filled Julia set for this polynomial drawn with the function *getcolor* as illustrated above. In Figure 5.4 we used 30 iterations, in Figure 5.5 we used 31. The 2-cycle can be seen clearly.

It is even more interesting to apply this method to a polynomial which contains a Siegel disk. If the center of the Siegel disk is at 0 then this method is perfectly suited for visualizing the conjugation to a rotation. An illustration for this can be found in Example 2.27 and in Figure 5.2. If the center is not at 0 then this method at least shows that the points inside the Siegel disk and its preimages do not converge to any specific point. See Figures 5.6 and 5.7.

Of course, it is possible to combine these two versions for the function *getcolor*. You then receive an image where $\mathcal{K}$ *and* $\mathbb{C} \setminus \mathcal{K}$ are colored. An example can be seen in Figure 5.8.

4.2.2 Drawing the Julia Set Itself

There are two main methods to draw the Julia set $\mathcal{J}$ itself. The first method we want to describe here is the so-called *boundary scanning method (BSM)*. The name already tells us the main principle of this method: For each pixel on the screen (and thus for the corresponding point in the complex plane) we investigate if it is on the boundary of $\mathcal{K}$ and thus in $\mathcal{J}$. There are several ways to implement this (see e. g. [PR, p. 38] or [PS, p. 178]). But since we have stored the color matrix *cmatrix* one implementation is particularly simple. The first step is to draw the filled Julia set $\mathcal{K}$ as described in the previous section (with a function for *getcolor* that returns black if and only if the point

is in $\mathcal{K}$). Actually, you don't have to *draw* $\mathcal{K}$, you "only" have to compute *cmatrix*. Then the algorithm is as follows:

```
subroutine BSM
begin
    do j = 2, height − 1
        do i = 2, width − 1
            if cmatrix(i, j) = black then
                if (cmatrix(i + 1, j) = white or cmatrix(i − 1, j) = white
                    or cmatrix(i, j + 1) = white or cmatrix(i, j − 1) = white) then
                    pset (i, j, black)
                endif
            endif
        enddo
    enddo
end
```

So you draw those points that are in $\mathcal{K}$ and that have a neighbor that is in $A(\infty)$. It is not difficult to adopt this routine such that it works even when $\mathcal{K}$ (or *cmatrix*) was computed with a different version of *getcolor*.

Of course the outcome depends on *maxit* since the calculation of *cmatrix* did. The subroutine *BSM* itself is — of course — very fast. The time it takes to draw the Julia set with this method almost only depends on the way how you calculate *cmatrix*. The results of this method are usually quite satisfactory (see Figure 4.2).

Although the output of the boundary scanning method is usually quite good, there are situations where this method fails to give any useful result. Suppose that $\mathcal{J} = \mathcal{K}$, e. g. when $\mathcal{J}$ is a Cantor set or a dendrite. (The definition of a dendrite is not standardized in the literature; we say that $\mathcal{J}_P$ is a *dendrite* if it is connected and $\mathcal{F}_P$ consists of exactly one component, i. e., $\mathbb{C} = \mathcal{J}_P \cup A(\infty)$.) In this case it is very unlikely that one of the algorithms mentioned in Section 4.2.1 exactly hits a point of the Julia set. Using a color image of the Fatou set we can suspect where the Julia set itself is, but there will be only very few black pixels. Hence, the boundary scanning method will fail to give a good result. As an example consider the polynomial $P(z) = z^2 - 1.543689$. A color image is given in Figure 5.9. The image clearly suggest where the Julia set is but, except from one vertical bar, there are almost no black pixels in this picture. We need a completely different algorithm to cope with this problem.

The algorithm that will cover these cases is called *inverse iteration method (IIM)*. Again the name of the routine already tells us the main principle of this method. Recall that if we have a starting point $z_0 \in \mathcal{J}$ then (by Theorem 1.20)

$$\mathcal{J} = \overline{\bigcup_{n=0}^{\infty} P^{-n}(z_0)}.$$

The first question is how to choose the starting value z_0. From Theorem 1.34 we know that a polynomial has a repelling fixed point or a fixed point with multiplier 1. In either case this point is an element of the Julia set. Hence, we only have to compute the fixed points of the polynomial and choose the one with the largest multiplier.

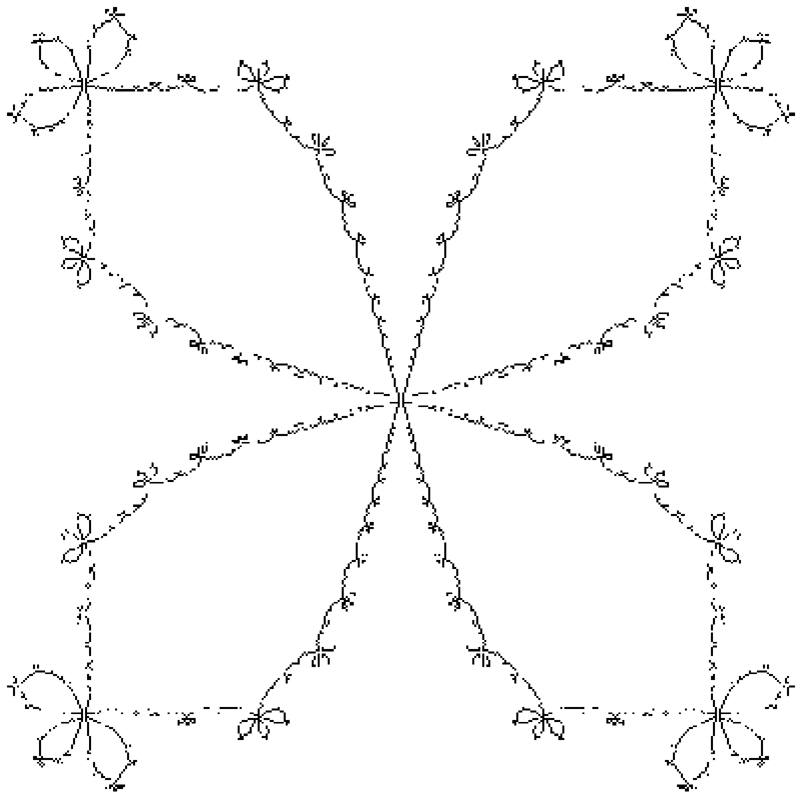

Figure 4.2: Julia set for $P(z) = \frac{1}{16}x^5 + \frac{5}{4}x$ drawn with the boundary scanning method. The color matrix *cmatrix* was computed with 100 iterations.

The second problem is that we cannot calculate and store *all* the backward iterates, since for the n^{th} level we would have d^n points. As a compromise for a given point we draw all the backward iterates and choose one of them to go on. Thus the algorithm looks like this:

subroutine *IIM*
begin
    $a_0 = a(0)$
    $cmatrix = white$
    calculate starting value z
    do while *true*
        if *button* **return**
        $a(0) = a(0) - z$
        $roots = findroots(a, d)$
        $a(0) = a_0$
        do $k = 1, d$
            $(i, j) = pointtopixel(roots(k))$
            if $inscreen(i, j)$ **then**
                pset $(i, j, black)$
                $cmatrix(i, j) = black$
            endif
        enddo

$\qquad\quad k = \lfloor n \cdot rnd(i) \rfloor + 1$
$\qquad\quad z = roots(k)$
$\quad$ **enddo**
end

A few comments on this pseudo-code: The function *button* gives *true* if the user has pressed the button of the mouse. This is only one possibility to terminate the procedure and it gives the user the ability to stop when he thinks the image is already good enough for his purposes. Other possibilities would be to set a time constraint or a given number of backward iterations. The function *inscreen* gives *true* if the pixel (i, j) is on the screen, i. e., if

$$1 \leq i \leq width, \qquad 1 \leq j \leq height.$$

Finally, the function *rnd* returns a random number in the interval $[0, 1)$.

Let us return to our example $P(z) = z^2 - 1.543689$. If we draw the Julia set of this polynomial with the inverse iteration method we obtain the picture in Figure 4.3.

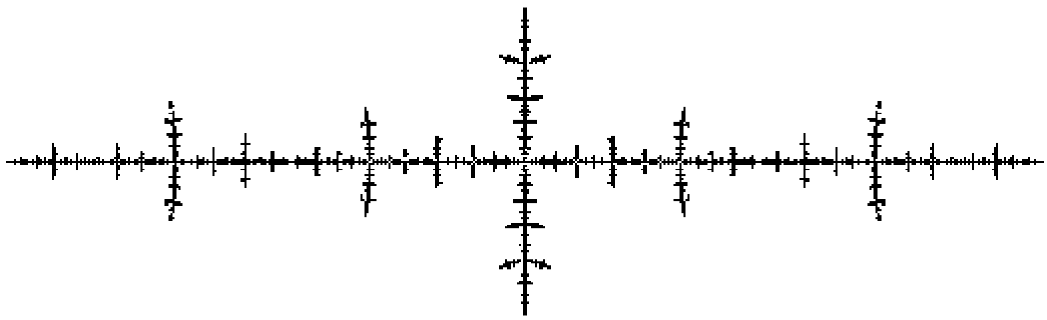

Figure 4.3: Dendrite drawn with inverse iteration method.

The inverse iteration method is extremely well-suited for drawing dendrites. But even for "normal" Julia sets it is, in most cases, the quickest way to get a *feeling* of how the Julia set looks like. But this method has big shortcomings, too. First of all, it is nearly impossible to draw details of the Julia set with this method, because you always calculate backward iterates, and they are spread all over the Julia set. Another big problem is that the backward iterates are not uniformly distributed on the Julia set. The distribution of the backward iterates is characterized by a measure μ. (For details we refer to [PR, p. 37] or [Bro, p. 140].) So, if you draw the Julia set as described above you will receive an image where some parts of the Julia set are left almost completely white, as it is very unlikely that a randomly chosen backward iterate hits those areas where μ is very small.

To cope with this problem we have to modify our algorithm. We will use the matrix *cmatrix* for our version of the *modified inverse iteration method (MIIM).* Furthermore, note that this version only works well if the whole Julia set is on the screen. (To that end it is necessary that you have a good estimate for an escape radius to guarantee that the whole Julia set is indeed on the screen.) We will explain later why this is necessary. During the process of creating the image we will use *cmatrix* not in the usual way for storing the color; instead, *cmatrix*(i, j) will denote how often the pixel (i, j) has been hit. When we compute the d backward iterates of a point we draw them and add 1 to *cmatrix* at each of the d corresponding pixels $(i_1, j_1), \ldots, (i_d, j_d)$. The idea is now

that we choose the new point $z = roots(k)$ not randomly; instead, we use the backward iterate $roots(k)$ for which the corresponding pixel (i_k, j_k) satisfies

$$cmatrix(i_k, j_k) = \min_{1 \le l \le d} cmatrix(i_l, j_l).$$

This means that we prefer points in those areas where, in comparison to other regions, not much has yet been drawn. When the user has stopped the iteration process we set

$$cmatrix(i,j) = \begin{cases} white & \text{if } cmatrix(i,j) = 0 \\ black & \text{if } cmatrix(i,j) > 0. \end{cases}$$

As an example we again consider the polynomial $P(z) = \frac{1}{16}x^5 + \frac{5}{4}x$. On the left of Figure 4.4 we have drawn the Julia set with the original unmodified inverse iteration method. It is quite obvious that the measure μ around the repelling fixed point at the origin is very small. Hence, there are almost no points in this area. The picture in the right side of Figure 4.4 is much better, and yet it took even less time to produce this image.

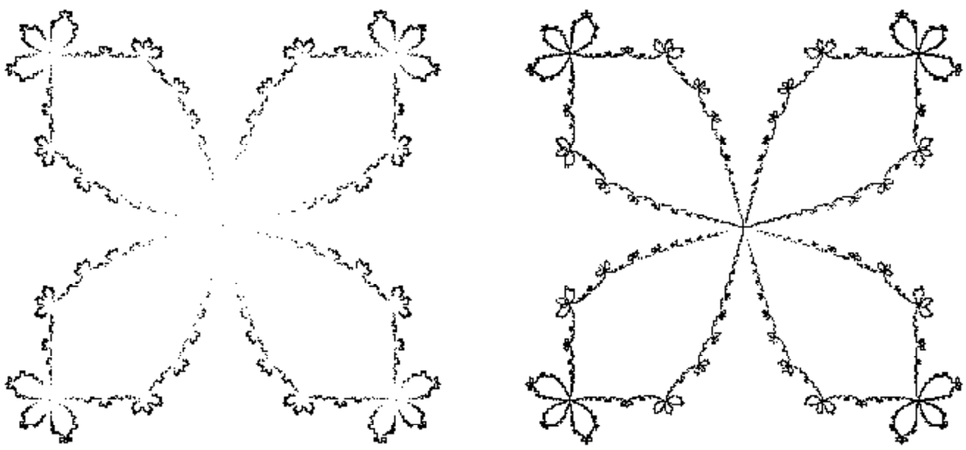

Figure 4.4: Julia set for $P(z) = \frac{1}{16}x^5 + \frac{5}{4}x$ drawn with IIM and MIIM.

While this first modification can, in many cases, lead to significantly better pictures we can still improve the algorithm. The idea is that we put each root for which $cmatrix(i_k, j_k)$ is very small (i. e., 1 or 2) into a sufficiently large bucket (sufficiently large means that we set aside storage for e. g. 1000 points). And as long as the bucket is not empty we take out an element of this bucket for going on in the algorithm instead of choosing one of the backward iterates. Hence, in the beginning of the drawing process the bucket rapidly fills with numbers; then, as the elements in *cmatrix* become larger and larger, only few elements are added to the bucket, and as we take out one element of the bucket in each loop of the algorithm the number of elements in the bucket decreases. If there is no element left in the bucket, we proceed as above (i. e., taking the root with smallest corresponding value in *cmatrix*). The advantage of this method is that we deal with *all* the points that cover "new" areas on the screen. This also suggests that we start with filling the bucket not right from the beginning, instead we "wait" for e. g. 1500 backward iterations until those areas where μ is big are covered and the bucket then only contains interesting points, i. e., points where μ is small.

This version of the modified inverse iteration method now gives, in nearly all situations, results that are very useful, especially when dealing with Cantor sets or attracting n-cycles with larger n. As an example, consider the polynomial $P(z) = z^2+0.32+0.043i$. With the first modification of the inverse iteration algorithm the result gives merely a hint of how the real Julia set might look like. With the new MIIM the image is tremendously better, as seen in Figure 4.5. Moreover, the left image in Figure 4.5 took 100.000 backward iterations, whereas the right image with the new MIIM only required 50.000 backward iterations.

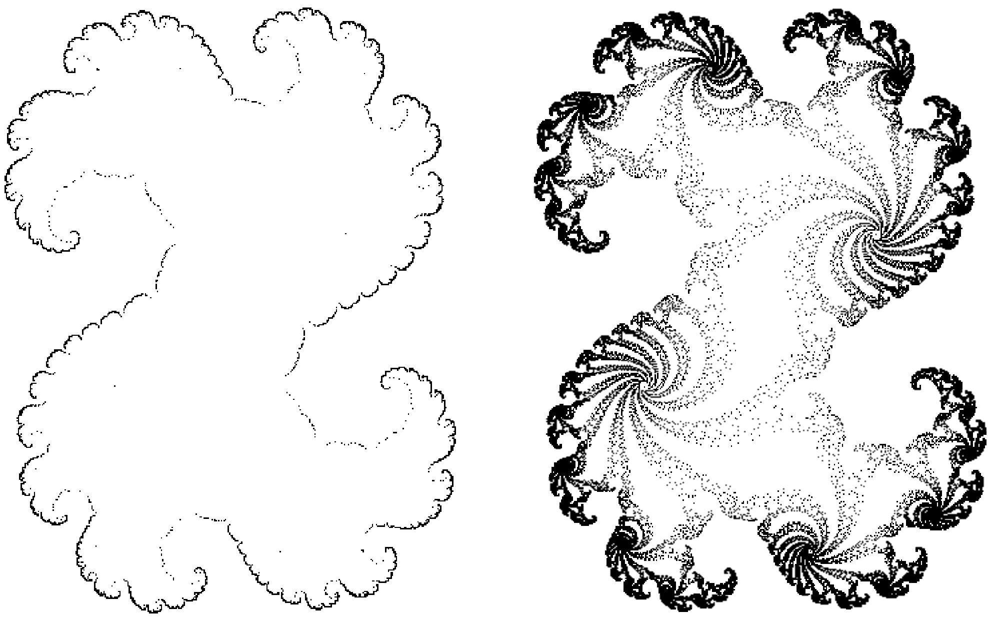

Figure 4.5: Julia set of $P(z) = z^2+0.32+0.043i$ drawn with different versions of MIIM.

As stated earlier, this version of MIIM only works if the whole Julia set is on the screen. This is because we use *cmatrix* for storing the information how often a certain pixel has been visited. Hence, if a backward iterate were outside the screen then there would not be a corresponding element of *cmatrix*. But usually MIIM is used to draw the whole Julia set, and for that purpose our version is well-suited. Nevertheless, it is not too difficult to write a version of MIIM that also handles the case when we draw only a part of the Julia set. To that end we lay a grid over the Julia set, e. g. a square of size $(2r) \times (2r)$ (where r is the escape radius) divided into 500×500 small subsquares. And then instead of counting how often a certain pixel has been visited we count how often one of these squares has been visited. The advantage of this version is that it is independent of the area actually displayed on the screen, the disadvantage is that it uses a lot of additional memory and might be less efficient (since those subsquares are usually larger than one pixel unless you make the subsquares very small; this, in turn, results in an even larger memory requirement).

4.3 Speeding up the Program

Although computers become more and more powerful it may still take quite some time to generate good pictures of Julia sets. In this section we present several tricks and

techniques to speed up the program. Some of these measures cause only a slight speed gain, others really speed up the program significantly.

4.3.1 Mathematical Measures

We have already encountered a little trick that speeds up the program more than one would expect. When checking if the absolute value of a complex number z is equal to (greater than, less than) some (positive) value r we always keep in mind that

$$\begin{array}{ccc} |z| \lesseqgtr r & \Longleftrightarrow & |z|^2 \lesseqgtr r^2 \\ \Updownarrow & & \Updownarrow \\ \sqrt{\Re(z)^2 + \Im(z)^2} \lesseqgtr r & \Longleftrightarrow & \Re(z)^2 + \Im(z)^2 \lesseqgtr r^2. \end{array}$$

On the right hand side you have one additional multiplication but you do not have to compute a square root. Furthermore, if r is the same for many equations (as in the function *getcolor*) it is a good idea to calculate r^2 once and store it as an additional variable.

For polynomials with attracting fixed points one can achieve big speed gains by using the theory of the *converge radius* as developed in Section 3.2. The first step is to compute all the attracting fixed points $z_1, \dots, z_m$ and the corresponding converge radii $\tilde{r}_1, \dots, \tilde{r}_m$. (To calculate a converge radius we use the definition of $\tilde{r}_{\tilde{Q}}$ in (3.27).) Then, inside the loop of *getcolor*, we check if z is inside the converge radius of any of the m attracting fixed points $z_1, \dots, z_m$. Thus the modified version of *getcolor* looks like this:

function $getcolor(z, r, m, \tilde{r})$
comment $\mathcal{K}$ black, $\mathbb{C} \setminus \mathcal{K}$ in color; modified version
begin
    $k = 0$
    $over = false$
    do while not $over$
        if $k = maxit$ **or** $|z|^2 > r^2$ **then**
            $over = true$
        else
            do $i = 1, m$
                if $|P(z) - z_i|^2 < \tilde{r}_i{}^2$ **then**
                    $k = maxit$
                    $over = true$
                endif
            enddo
            if not $over$ **then**
                $z = P(z)$
                $k = k + 1$
            endif
        endif
    enddo
    if $k = maxit$ **then**

        color = *black*
    else
        color = $f(k)$
    endif
    return *color*
end

The speed gain that can be achieved with this method strongly depends on the structure of the Julia set and on the number of iterations. If the filled Julia set almost only consists of basins of attractions of attracting fixed points and there is a high maximum number of iterations then the speed increase can be tremendous because this method provides a criterion to halt the iteration process for those points that are in $\mathcal{K}$. In the unmodified version you have to compute all *maxit* iterations for those points since they never exceed the escape radius. The best example is the polynomial $P(z) = z^2$. In this case the converge radius and the escape radius are the same. Hence, there is no need to calculate any iterate for points not on the unit circle itself, because they are either outside the escape radius or inside the converge radius.

The questions now arises if the above can also be applied to attracting cycles. The answer is yes — in theory. Of course, one can compute fixed points of higher iterates but the polynomials you have then to deal with are soon of very high degree, and numerical results are less accurate and quite time-consuming to get. So, except in special cases (e. g. a quadratic polynomial with a superattracting 2-cycle) it does not make sense to compute attracting periodic points and the corresponding converge radii. Instead, we can use a different method to check for (some) cycles:

If a point z tends to an attracting cycle of period n, then $P^{nm}(z)$ converges with $m \to \infty$. Thus a criterion for an attracting cycle would be

$$|P^{n+m}(z) - P^m(z)| \leq \varepsilon, \quad m \geq n_0, \tag{4.1}$$

where n_0 is a sufficiently large integer. The smaller you choose ε the better you can ascertain that you do not make a mistake, but the less effective the method as such becomes. A mistake can be made, for example, when there are points that escape extremely slowly. It is then possible that (4.1) suggests that there is a fixed point although, actually, the point escapes. Hence, this method should be applied where speed is more important than total precision.

Of course, you cannot check (4.1) for every n. A good compromise is to check it only for $n = 12$, as you then also check for fixed points and periodic points of periods 2, 3, 4 and 6.

4.3.2 Computer-related Measures

Relatively slow graphics routines (at least on some machines) can be a problem when you draw an image on a computer — especially, if you really draw each pixel with a function for drawing a single point (in our pseudo-code we have called it **pset**); for then the drawing procedure proper takes a considerable amount of time. (This can be seen quite clearly if you just redraw the current image using the values stored in *cmatrix*.)

To reduce the number of drawing operations as much as possible you can proceed as follows. First, if you want to draw a black and white picture, clear the whole screen and set *cmatrix* to *white*. Then you only draw those points that are not white. For

black and white pictures as well as for color images you then only draw something onto the screen if the color changes. As an example consider the filled Julia set of $P(z) = z^2$, i. e., the unit disk. Since the escape radius is 1 we only have two colors, say blue for outside $\mathcal{K}$ and black for $\mathcal{K}$. If you draw a horizontal line on the screen that meets $\mathcal{K}$ the color changes only twice, once when the line enters the unit disk and once when it leaves the unit circle. Thus it is sufficient to call a drawing routine that draws a line just three times. First draw a blue line from the left border of the screen to the unit disk, then a black line inside the unit disk, and finally again a blue line from the unit disk to the right border of the screen. In the original version we would have to call **pset** *width* times.

The speed gain that can be achieved with this method is quite considerable. A further improvement can be achieved with offscreen drawing. In this method the drawing routines do not draw on the screen, instead they draw a virtual picture that is then displayed all at once. This method is not easy to program, and it is machine-dependent.

If you do not want to draw very small details then it is also possible to use complex numbers with less precision (e. g. in FORTRAN you can use `COMPLEX*8` instead of `COMPLEX*16`). Operations with `COMPLEX*8`-numbers are usually faster than those with `COMPLEX*16`-numbers.

4.3.3 The Raster-mode Algorithm

With the exception of the inverse iteration methods we have, up to now, always used methods that draw an image line by line. The algorithm presented here — we call it *raster-mode algorithm* — works in a different way.

First, we lay a grid on the screen (*not* over the complex plane; hence, it depends on the actual area that the screen represents). The grid cells should be square, and their length should be a power of two and not be too large. A good length is 64 pixels. So we have a total of $\lceil width/64 \rceil \times \lceil height/64 \rceil$ square cells that cover the screen. To draw the contents of such a square we initially only calculate its boundary values. If they are all the same we fill the whole square with this color and investigate the next square. If there are two points on the boundary with different colors we divide the square into four subsquares of equal size and treat those subsquares like the original one. It is this process of dividing a square into four subsquares that makes it advantageous to make the length of the original square a power of two.

To illustrate this process take a look at Figure 4.6. Suppose the area to the left of the arc has one color and the area to the right of the arc has another color. The four images show step by step, which of the squares have to be divided again. Figure 5.10 shows how the raster-mode algorithm has divided the screen into squares of equal color for the polynomial $P(z) = \frac{1}{16}x^5 + \frac{5}{4}x$.

The raster-mode algorithm is usually the fastest method to draw color images, especially when there are larger parts of $\mathcal{K}$ (i. e., a larger black area which means calculation of many iterations) on the screen. And it is much more interesting to watch the process of creating an image by this algorithm than watching the screen when one line is drawn after the other, apart from the fact that the difference in execution time between the algorithms can really be tremendous. But be careful. It is possible that this algorithm does not give correct results!! Just imagine a totally disconnected Julia set and a black and white image. Then you would almost certainly encounter a

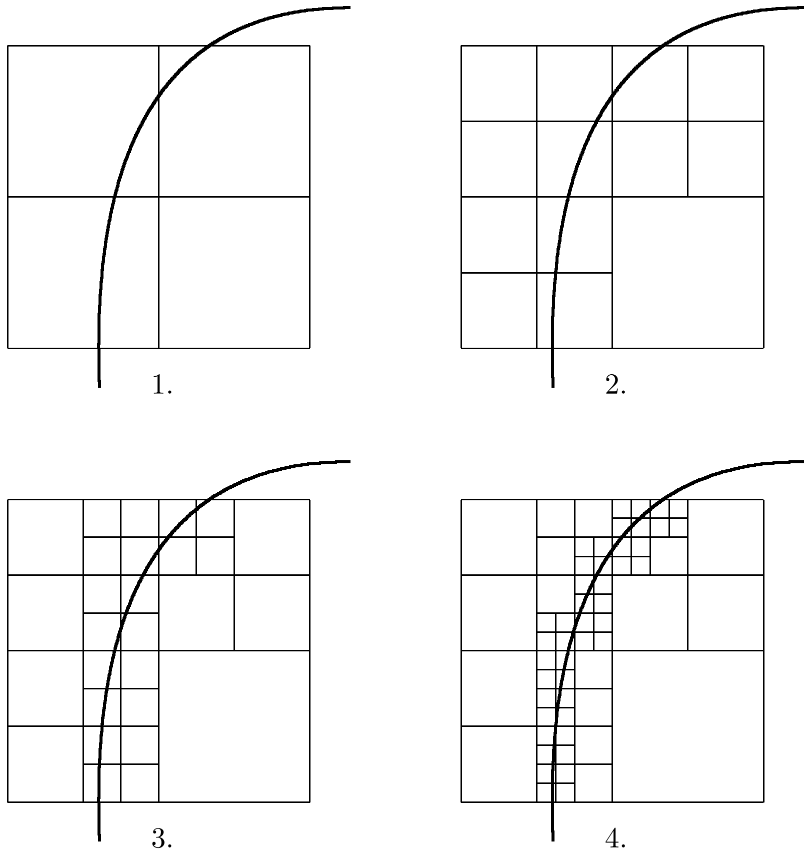

Figure 4.6: Illustration of how the raster-mode algorithm works.

(sub)square with white boundaries but where a part of the Julia set is inside the square. The raster-mode algorithm draws the whole square white. This is the reason why the raster-mode algorithm should only be applied when at least $\mathbb{C} \setminus \mathcal{K}$ is drawn in color. But even then there are situations when the raster-mode algorithm misses a part of the Julia set. For connected Julia sets (and for the Mandelbrot set, which is connected) the results are almost always identical to those obtained by our original algorithms.

Some thoughts on the implementation of this algorithm: First of all, when we investigate the boundaries of a square we put the color in the corresponding element of *cmatrix*. Hence, if the boundaries of the square are not all of the same color and we have to check the boundaries of the subsquares we do not have to recalculate the color of some points as they are already in *cmatrix*. This suggests that we set *cmatrix* to white at the beginning; if, while drawing, $cmatrix(i, j)$ is white we have to calculate the color, else we can take the color already stored in *cmatrix*. (It is, obviously, necessary that the final image does not contain any white points, which is the case when we draw the $\mathbb{C} \setminus \mathcal{K}$ in color and the filled Julia set either in color or black.) When we investigate the squares from the top to the bottom and from the left to the right, it is also possible to use the lower boundary of the square above the currently investigated one as the upper boundary of the currently investigated one (and the right boundary of the square to the left of the currently investigated one as the left boundary of the

currently investigated one), since those boundaries have already been investigated and *cmatrix* hence contains the appropriate values.

A further speed increase can be achieved if we only investigate every other (or even third) point on the boundary. This looks worse than it is. In fact, in most pictures you won't see any difference. And if you want complete accuracy you can always use the standard algorithm.

Appendix A shows the implementation of this raster-mode algorithm.

4.4 Other Visualization Topics

4.4.1 Mandelbrot Set

One of the most famous "objects" in the field of complex dynamics is the Mandelbrot set $\mathcal{M}$. In addition to the original publications by Mandelbrot himself there are uncountably many articles and books that include aspects of this set. There are also many shareware and freeware programs that show the beauty of this set and can be downloaded from the Internet. We will only provide the basic mathematical background that is necessary to understand and draw $\mathcal{M}$. For a more detailed treatment of the mathematical background of the Mandelbrot set we refer e. g. to [Bra].

The Mandelbrot set $\mathcal{M}$ is defined as

$$\mathcal{M} = \{c \in \mathbb{C} \mid \mathcal{J}_{P_c} \text{ is connected}\}, \tag{4.2}$$

where $P_c(z) = z^2 + c$. Hence, $\mathcal{M}$ is a set in the parameter plane. The definition of $\mathcal{M}$ in (4.2) is not suitable for drawing the Mandelbrot set. But recall that, by Theorem 1.30, the Julia set of a polynomial is connected if and only if the orbits of all finite critical points remain bounded. Since 0 is the only critical point of P_c we have

$$\mathcal{M} = \{c \in \mathbb{C} \mid P_c^n(0) \not\to \infty\}. \tag{4.3}$$

With (4.3) it is now easy to write a program that draws the Mandelbrot set. For the outer loop we can take a subroutine similar to *drawimage* which we introduced in Section 4.2.1:

subroutine *drawmandel*
begin
    do $j = 1, height$
        do $i = 1, width$
            $c = pixeltopoint(i, j)$
            $color = getmandelcolor(c)$
            $cmatrix(i, j) = color$
            pset$(i, j, color)$
        enddo
    enddo
end

As the polynomial P_c is different for each pixel we have to compute a different escape radius for each pixel. This makes it advisable to use a simple equation for the escape radius so that it can be calculated easily. We will use

$$r = \max\{2, |c|\},$$

which is a valid escape radius (see Example 3.8). Hence, the function *getmandelcolor* is as follows:

function *getmandelcolor*(c)
begin
  $k = 2$
  $z = c^2 + c$
  $r = \max\{2, |c|\}$
  $over = false$
  do while not $over$
    if $k = maxit$ **or** $|z|^2 > r^2$ **then**
      $over = true$
    else
      $z = P_c(z)$
      $k = k + 1$
    endif
  enddo
  if $k = maxit$ **then**
    $color = black$
  else
    $color = f(k)$
  endif
  return $color$
end

As neither 0 nor c is greater than r (taking the absolute value) we can start the iteration process with the second iterate which is $c^2 + c$.

The whole Mandelbrot set can be seen in Figure 5.11. One of the astonishing features of the Mandelbrot set is its (almost) self-similarity. There are small copies of $\mathcal{M}$ in the filigree branches of $\mathcal{M}$ one of which can be seen in the left of Figure 5.12. The right part of Figure 5.12 shows a magnification of a detail that unveils the beauty of this set. One can spend hours just zooming in and out and exploring the various regions and the gracefulness of $\mathcal{M}$.

4.4.2 Newton's Method

One of the first techniques to solve a real equation $f(x) = 0$ with differentiable f was Newton's method. It is well-known that for a given starting point x_0 one uses the iteration scheme

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

which converges to a zero $\bar{x}$ of f at least when x_0 is sufficiently close to $\bar{x}$.

Newton's method also works for complex functions. We now define the *Newton function* N_f of an analytic function f as

$$N_f(z) = z - \frac{f(z)}{f'(z)}. \tag{4.4}$$

Hence, for a given starting point z_0 the iterates in Newton's method are given by

$$z_n = N_f^n(z_0),$$

i. e., they are the orbit of z_0 under N_f. We will now again focus only on complex polynomials P of degree $d \geq 2$ with Newton function

$$N_P(z) = z - \frac{P(z)}{P'(z)}. \tag{4.5}$$

Note that in this case N_P is a rational function of degree d. The derivative of N_P is given by

$$N_P'(z) = \frac{P(z)P''(z)}{(P'(z))^2}. \tag{4.6}$$

We are now interested in the question which points of the complex plane do converge and, if they do, to which root of the polynomial. Suppose that

$$P(z) = (z-a)^k Q(z), \quad k \in \mathbb{N},\ Q(a) \neq 0.$$

Then, using (4.5) and (4.6), one can easily show that a is a fixed point of N_P with multiplier $(k-1)/k$. Hence, if $k = 1$ then a is a superattracting, else an attracting fixed point of N_P. As shown for attracting fixed points of polynomials in Theorem 1.27 it can also be shown that if v is an attracting fixed point of a rational function R then

$$\partial A(v) = \mathcal{J}_R.$$

This implies that the boundary of the set on which Newton's method converges to a certain root is the Julia set of the rational function N_P! Thus we have a method for drawing the Julia set for a certain class of rational functions, namely of those that are Newton functions of a polynomial. The algorithm now works as follows: The standard outer loop is

```
subroutine drawnewton
begin
    zeros = findroots(a, d)
    do j = 1,height
        do i = 1,width
            z = pixeltopoint(i, j)
            color = getnewtoncolor(z, zeros)
            cmatrix(i, j) = color
            pset(i, j, color)
        enddo
    enddo
end
```

The function *getnewtoncolor* looks like this:

```
function getnewtoncolor(z, zeros)
begin
    k = 0
```

```
    over = false
    do while not over
        do i = 1, d
            if |z − zeros(i)|^2 < ε^2 then
                over = true
                color = f(i, k)
            endif
        enddo
        if not over then
            k = k + 1
            if k > maxit then
                over = true
                color = black
            else
                z = z − P(z)/P'(z)
            endif
        endif
    enddo
    return color
end
```

Usually, it is not necessary to choose ε very small; a value of $\varepsilon = 10^{-3}$ should be small enough. The color function f depends on i and k. If you want to visualize *how fast* the iterates converge then it depends on k; if you want to visualize *to which root* the iterates converge, then it depends on i. Of course, it is possible to combine the two.

The program above draws all those points black that do not converge to a root of the polynomial. The question now is if those points are necessarily in the Julia set. The answer is no, since it is possible that N_P also has attracting periodic points or even rotation domains. To show how this can happen we consider a family of cubic polynomials

$$P_\rho(z) = z^3 + (\rho - 1)z - \rho.$$

(Most interesting cubic polynomials can be conjugated to this form; see e. g. [CGS] or [Bl2].)

Since basins of attraction contain critical points and the boundaries of rotation domains are subsets of the closure of the postcritical set (see Theorems 2.10, 2.20 and 2.33; the results there are also valid for rational functions) we see that if there is a set $A \subset \mathbb{C}$ with nonempty interior on which Newton's method fails to converge then there is also a critical point of N_P that does not converge to a root of P. Using (4.6) we see that the critical points of N_{P_ρ} are either roots of P_ρ (which obviously converge) or roots of P''_ρ. But the only root of P''_ρ is the origin. This suggests that we draw the set of points ρ for which $N^n_{P_\rho}(0)$ does not converge to a root of P_ρ. Note that — like the Mandelbrot set — this again is a set in the parameter plane. The outer loop is as usual, and the function to determine the color looks like this:

```
function getnewtonparametercolor(ρ)
begin
    k = 0
```

```
    z = 0
    zeros(1) = 1
    zeros(2) = (−1 − √(1 − 4ρ))/2
    zeros(3) = (−1 + √(1 − 4ρ))/2
    over = false
    do while not over
        do i = 1, 3
            if |z − zeros(i)|^2 < ε^2 then
                over = true
                color = f(i, k)
            endif
        enddo
        if not over then
            k = k + 1
            if k > maxit then
                over = true
                color = black
            else
                z = z − P_ρ(z)/P'_ρ(z)
            endif
        endif
    enddo
    return color
end
```

The results of this program can be seen in Figure 5.13. The amazing thing is that those parts where $N^n_{P_\rho}(0)$ does not converge have exactly the shape of the Mandelbrot set $\mathcal{M}$. The closeups in Figure 5.14 show this clearly. And if we take a parameter ρ in such a small copy of $\mathcal{M}$ and denote by c the corresponding parameter in the real Mandelbrot set, then the Julia set of N_{P_ρ}, which we can draw using Newton's method, contains tiny copies of $\mathcal{J}_{P_c}$ with the value c as above and $P_c(z) = z^2 + c$. An example with $\rho \approx 0.34741 + 1.64896i$ is depicted in Figure 5.15 and 5.16. More about the mathematical background of this remarkable results can be found in [Bl2].

4.4.3 Miscellaneous

In this section we present some minor aspects that can be of interest when writing a program to visualize aspects of Julia sets.

First of all, some comments on *drawing orbits*. Drawing the orbit of a given point z (which can be specified either by typing its coordinates or by clicking on the corresponding pixel on the screen) is nothing special, one simply iterates the point and draws the according pixels. The drawing process should stop if either any iterate is outside the escape radius or if the user presses a key or clicks the mouse. With this method you can also draw the postcritial set (see Definition 2.32). Just note that in this case the iteration process should stop only if the iterates of *all* critical points are outside the escape radius, but only those points that are still inside the escape radius are iterated further. Pictures that contain orbits of points can be found in Section 2.3.

Sometimes it is also interesting to *draw the backward iterates* of a point. One way to do so is to use an inverse iteration method (IIM or MIIM) as explained in Section

4.2.2; one just uses a specified point for starting the algorithm instead of letting the computer calculate a repelling fixed point. The pictures one receives with this method are essentially the same as when drawing the Julia set because if the starting point z is not an exceptional point then the backward iterates tend to the Julia set (see Theorem 1.20). It is more interesting to study a certain backward iterate of a set of points. As an example consider the polynomial $P(z) = 2z^4 - 1$. The Julia set for this polynomial is a dendrite. Moreover, the interval $[-1, 1]$ is a subset of $\mathcal{J}$. To see how the dendrite develops we draw $P^{-i}([-1, 1])$ for $i = 1, 2, 3$. Note that we really draw *all* backward iterates, not only a randomly chosen branch. To draw the backward iterates of an interval we choose sufficiently many points in the interval, e. g. in our example the points $-1 + 2 * j/5000$, $j = 0, 1, \ldots, 5000$, and calculate P^{-i} for each point. The problem is that the backward iterates are far from being uniformly distributed. Hence, there are some points left white in the picture that should be drawn black. The result together with the Julia set can be seen in Figure 4.7.

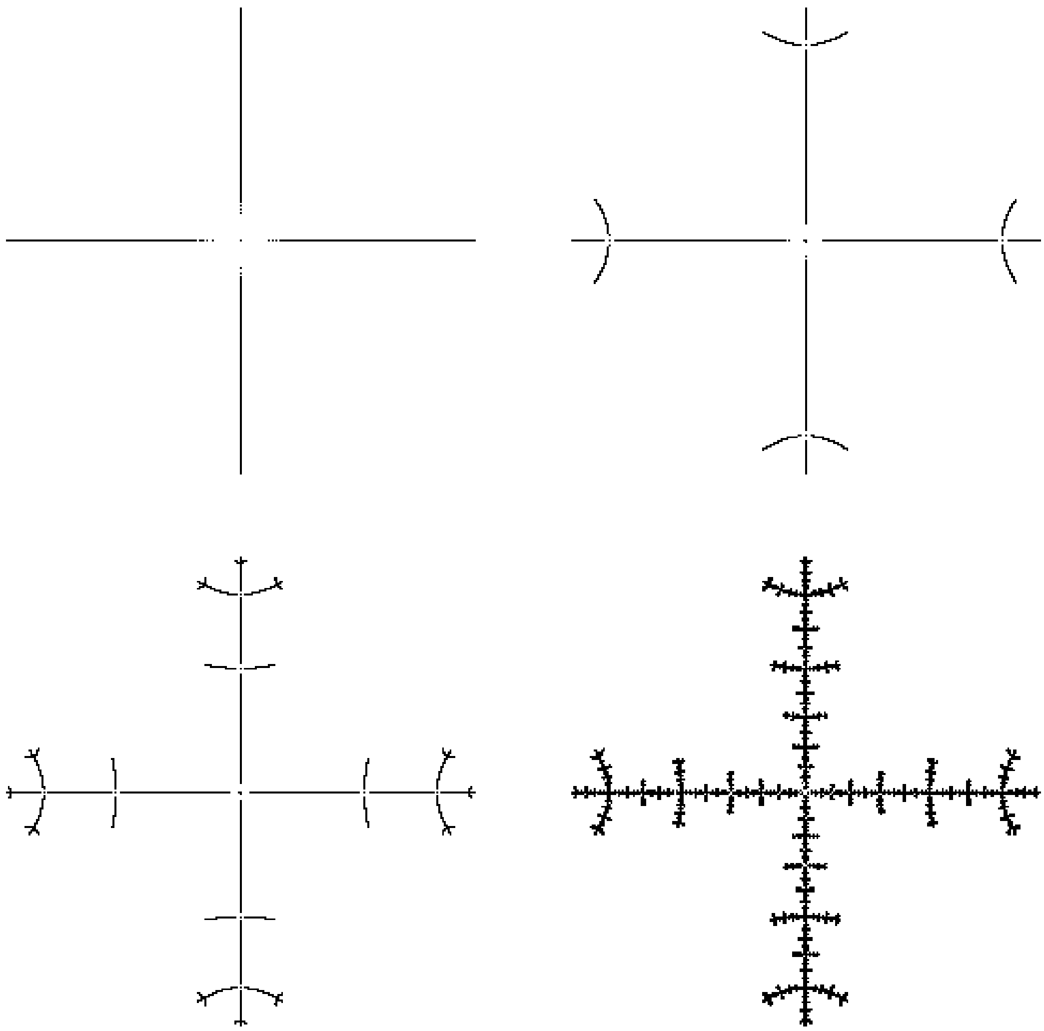

Figure 4.7: $P^{-i}([-1, 1])$ for $i = 1, 2, 3$ and $\mathcal{J}$ for the polynomial $P(z) = 2z^4 - 1$.

Sometimes it is also of interest to *draw the set where* $P(z)$ *is real.* There are two ways of accomplishing this. The first method is to choose a sufficiently small ε and

then check for each point that corresponds to a pixel if

$$|\Im(P(z))| < \varepsilon. \tag{4.7}$$

The results are not satisfactory, since if we choose ε too small then it is unlikely that a point that corresponds to a pixel satisfies (4.7). If we choose ε too big then there are too many points that satisfy (4.7) although they should not. Therefore, we better choose the second method. Obviously, the set where $P(z)$ is real can be written as $P^{-1}(\mathbb{R})$. Thus it makes sense to draw $P^{-1}([a,b])$ for a sufficiently large interval $[a,b]$ with the method described above. A good choice for the interval is $[-r_Q, r_Q]$ where r_Q is the escape radius as in Corollary 3.3.

When *drawing the sets $\mathcal{E}$ and $\mathcal{G}$* that we used in Chapter 3 (see Theorem 3.2 and Theorem 3.13) we encounter problems similar to those above. For example, if we want to draw $\mathcal{E}$ the obvious check for a point corresponding to a pixel would be to test if

$$\big||P(z)| - |z|\big| < \varepsilon.$$

Unfortunately, there is no easy method to avoid the problems mentioned above.

Chapter 5

Color Plates

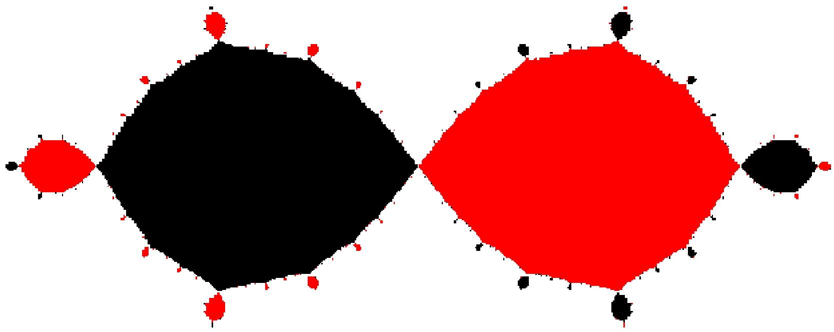

Figure 5.1: Filled Julia set for $P(z) = -2z^3 + 3z^2$. The basin of attraction of the superattracting fixed point $z_2 = 1$ is drawn in red.

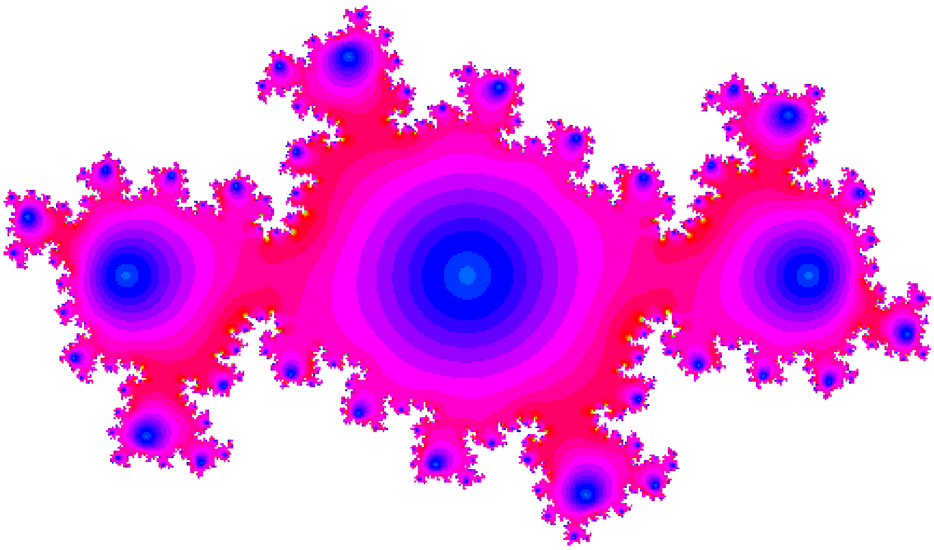

Figure 5.2: Filled Julia set for $P(z) = \lambda(z - z^3/3)$ with $\lambda = \exp(2\pi i(\sqrt{5} - 1)/2)$ with a Siegel disk. The colors denote the absolute value of the 30$^{\text{th}}$ iterate of the specified point. The conjugation to a rotation can be seen.

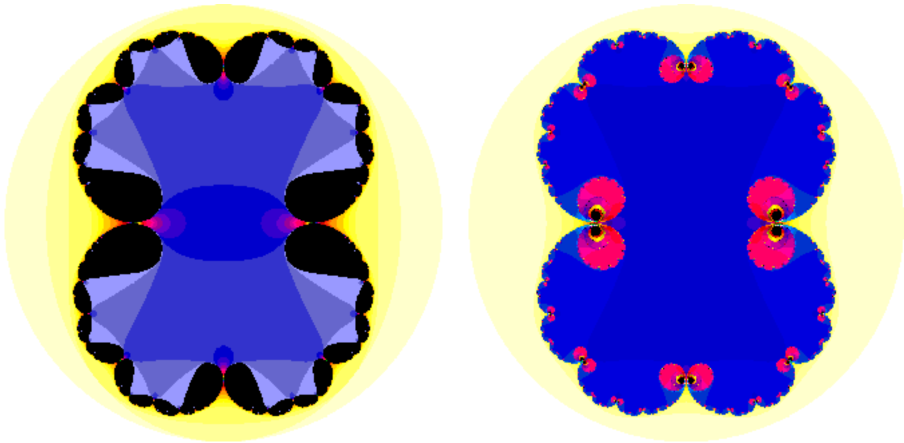

Figure 5.3: Filled Julia set for $P(z) = z^2 + 0.251$ drawn with 100 and 500 iterations.

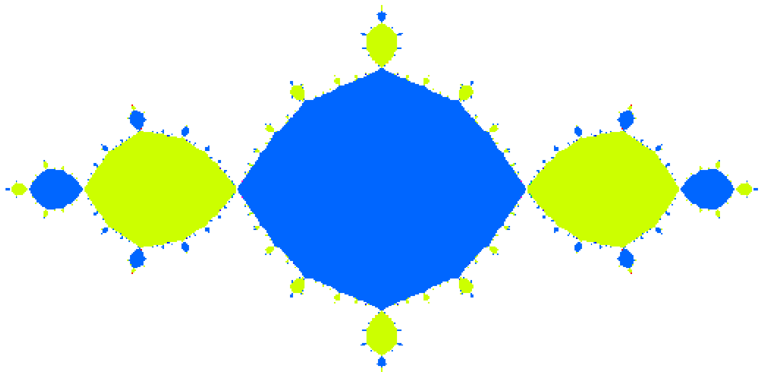

Figure 5.4: Filled Julia set in color for $P(z) = z^2 - 1$ drawn with 30 iterations.

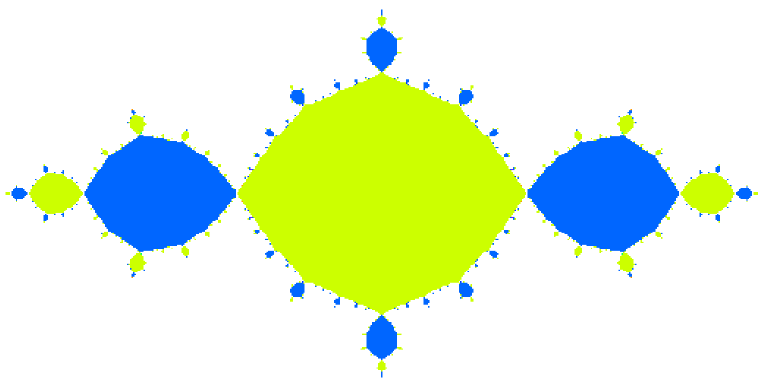

Figure 5.5: Filled Julia set in color for $P(z) = z^2 - 1$ drawn with 31 iterations.

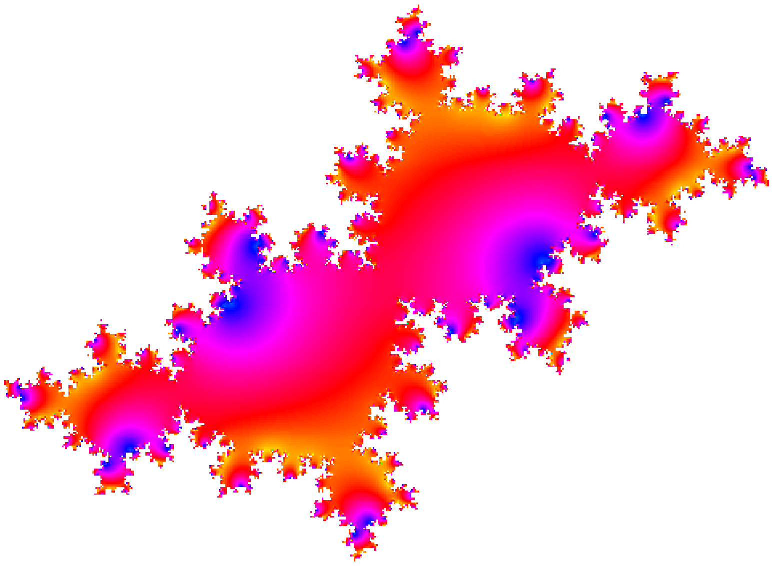

Figure 5.6: Filled Julia set in color for a quadratic polynomial with Siegel disk drawn with 100 iterations.

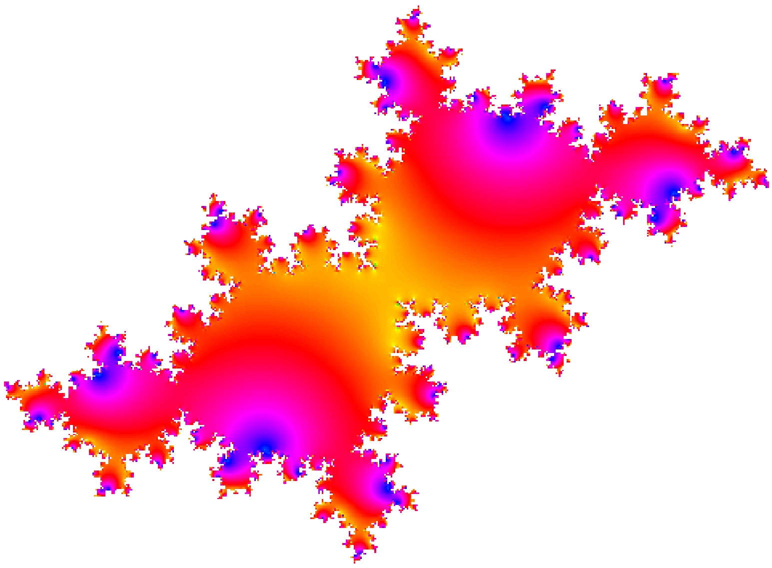

Figure 5.7: Filled Julia set in color for a quadratic polynomial with Siegel disk drawn with 101 iterations. The points in $\mathcal{K}$ do not converge to any specific point.

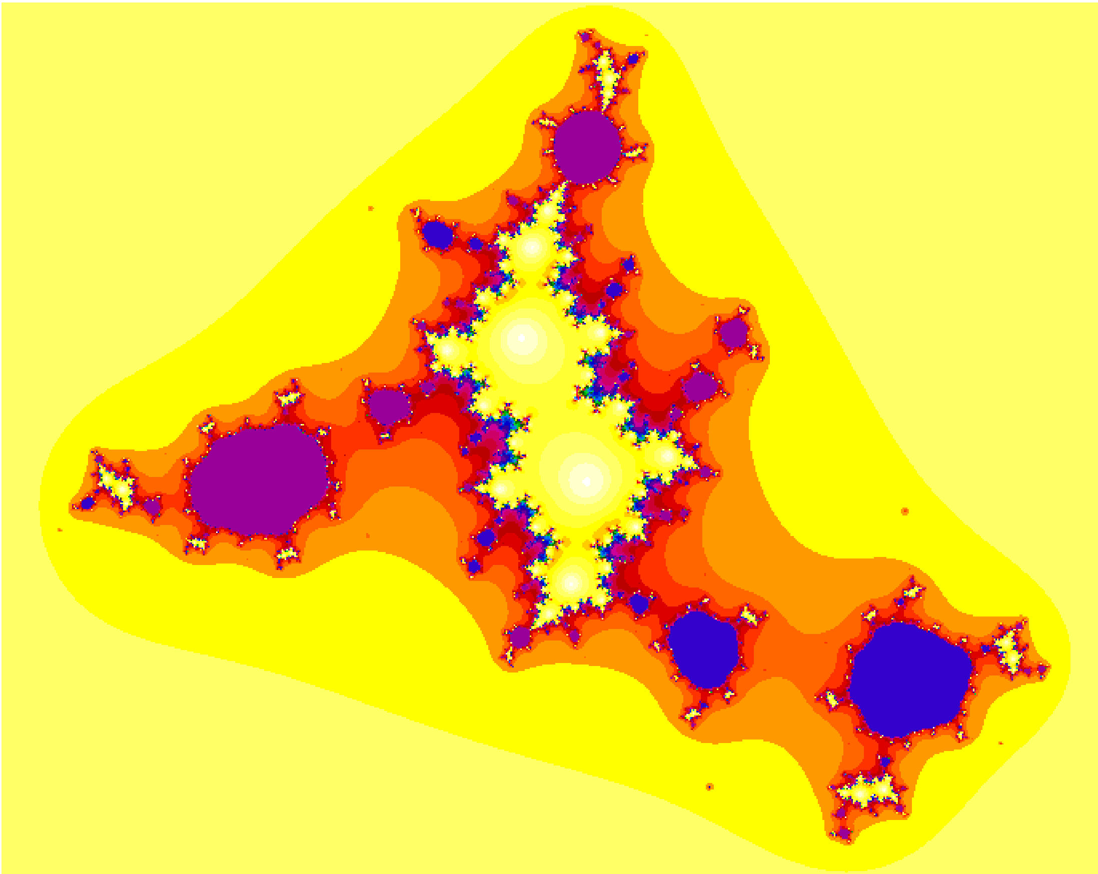

Figure 5.8: Filled Julia set and Fatou set in color for a polynomial of degree 7 with Siegel disk, attracting fixed point and attracting period-2 cycle.

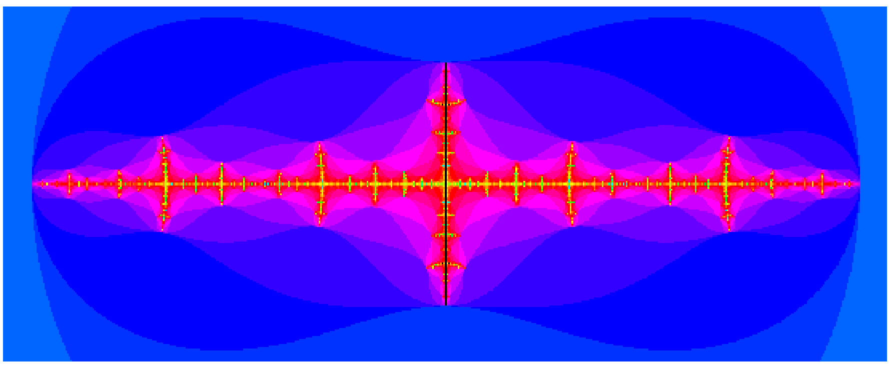

Figure 5.9: Fatou set in color for $P(z) = z^2 - 1.543689$.

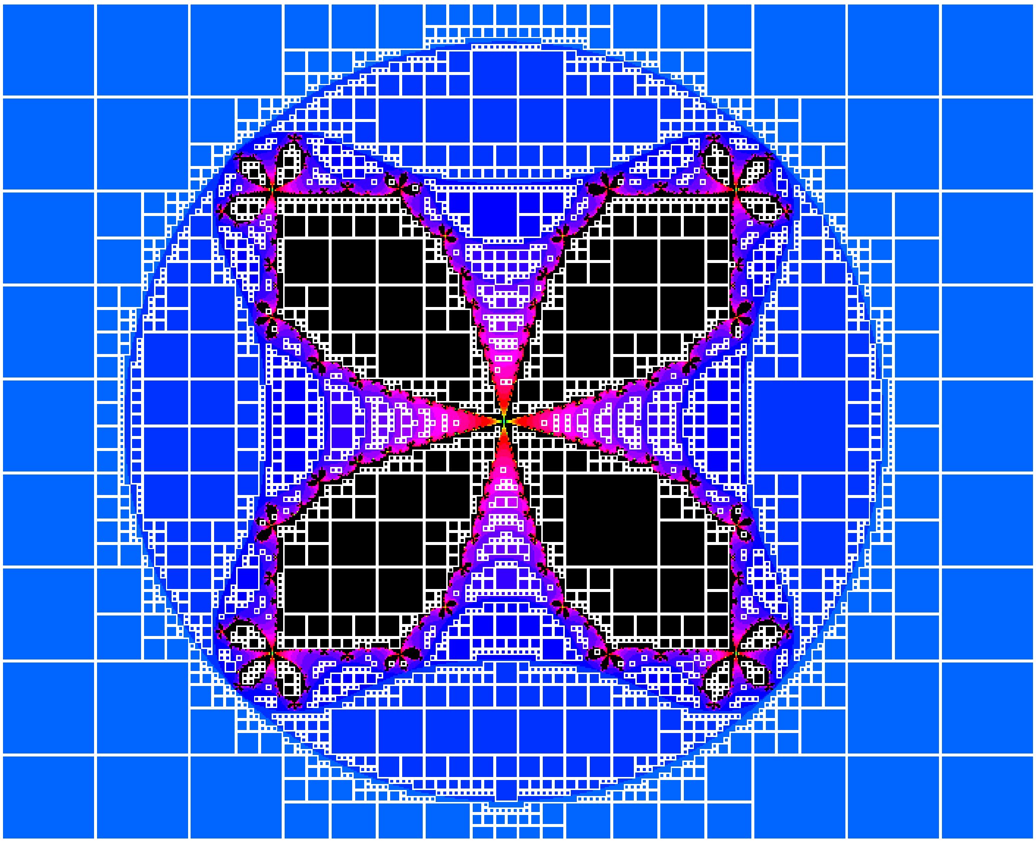

Figure 5.10: Squares used for drawing in raster-mode.

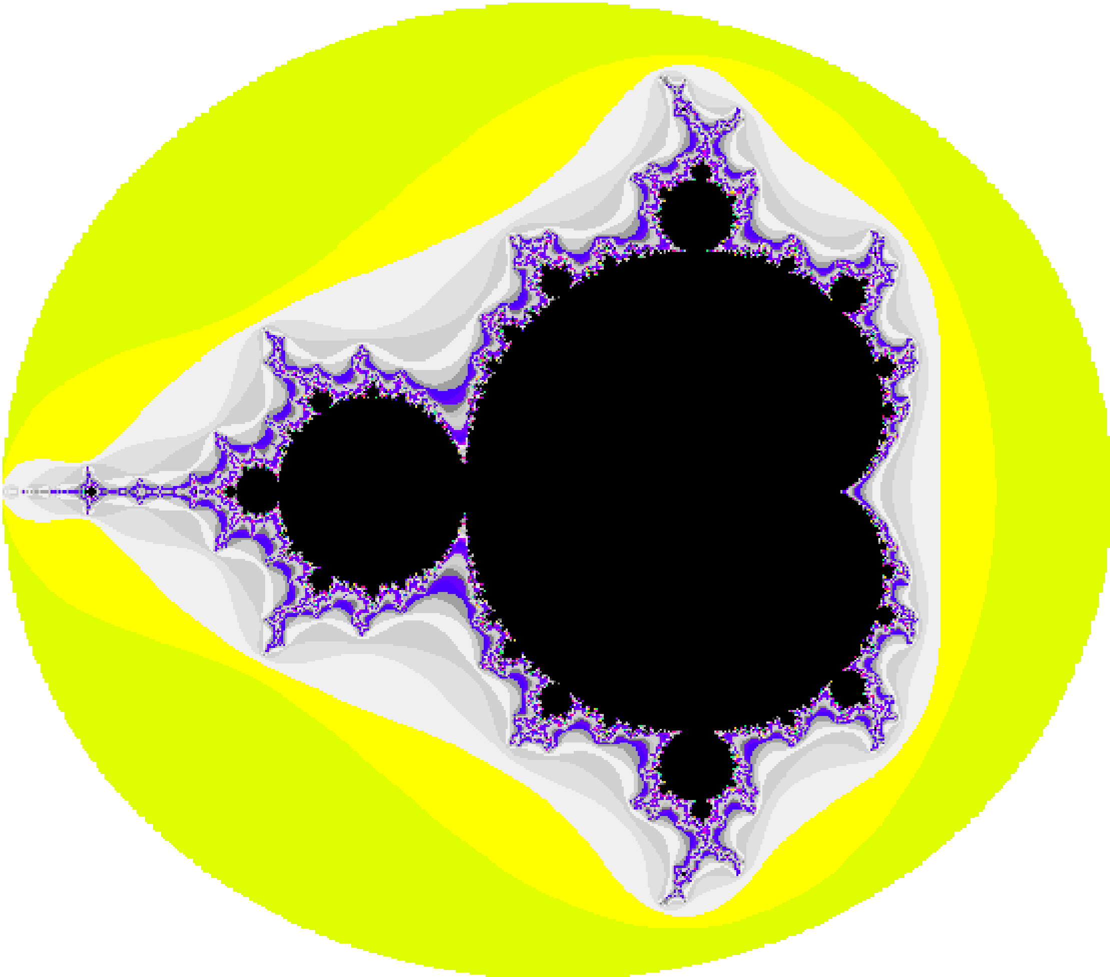

Figure 5.11: Complete Mandelbrot set $\mathcal{M}$.

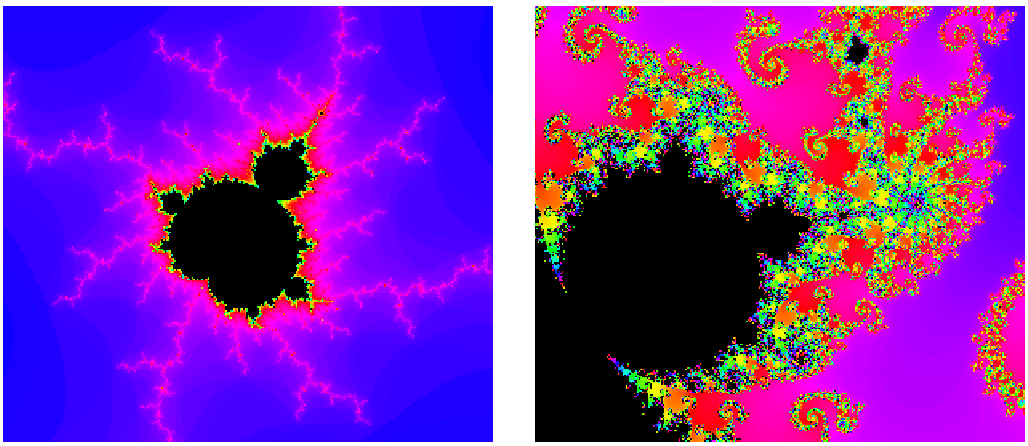

Figure 5.12: Small copy of $\mathcal{M}$ in a branch of $\mathcal{M}$ and a detail of $\mathcal{M}$.

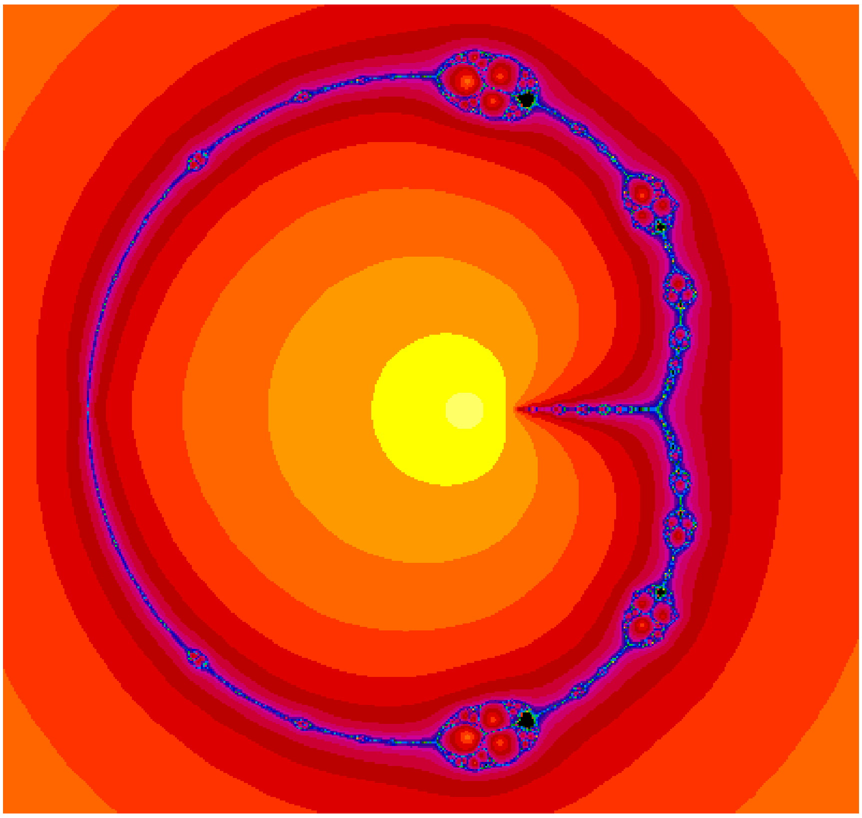

Figure 5.13: Set of parameters ρ for which $N^n_{P_\rho}(0)$ does not converge.

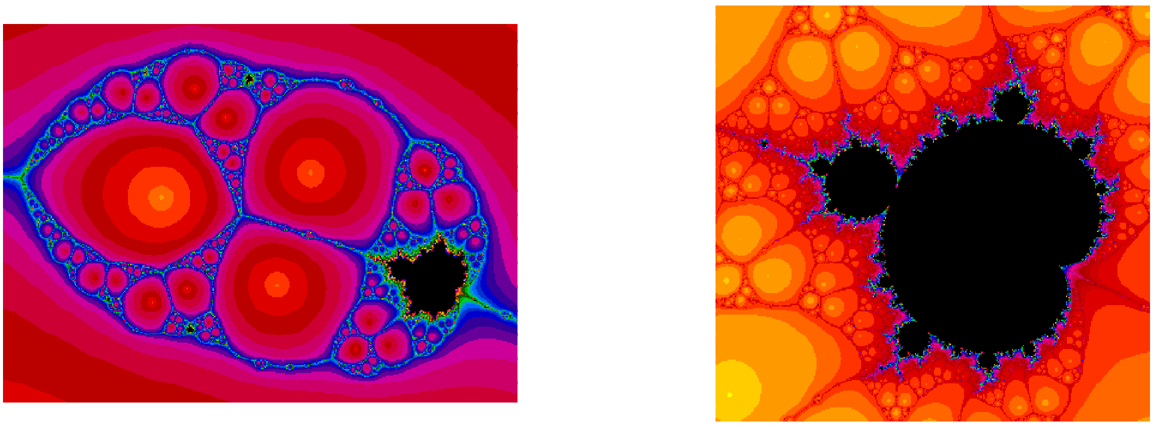

Figure 5.14: Enlargements of Figure 5.13.

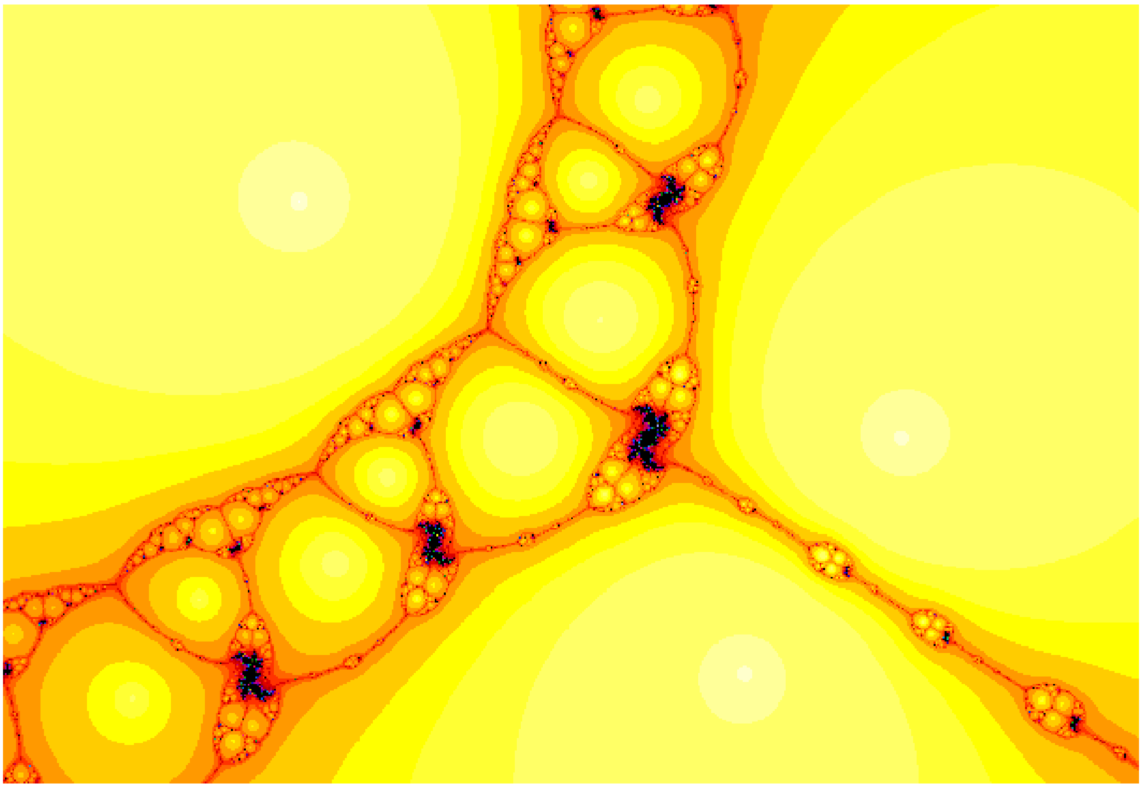

Figure 5.15: Newton method for P_ρ with $\rho \approx 0.34741 + 1.64896i$.

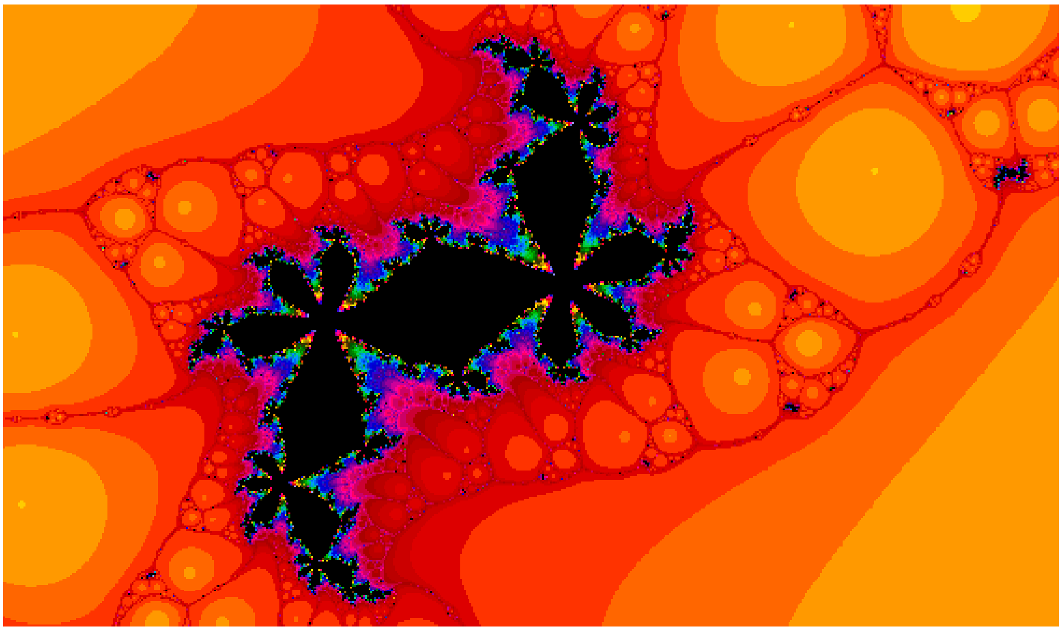

Figure 5.16: Enlargement of Figure 5.15.

Appendix A

Sample Source Code

As an example for a real-world implementation we present here the source code of the implementation of the raster-mode algorithm (see Section 4.3.3). The subroutines and functions presented here are part of our program which has been used to draw all the graphics containing Julia sets in this thesis. The program is written in LS Fortran for PowerMacintosh from Fortner Research, a FORTRAN 77 compiler with some FORTRAN 90 extensions. The graphic routines are highly machine-dependent, in this case the Macintosh Toolbox is used. The names of the variables are essentially the same as those used in the pseudo-codes in Chapter 4. There are just two differences: The degree d of the polynomial is the variable `n` and the maximum number of iterations *maxit* is denoted by `iterations`. The subroutine `zroots` computes the zeros of a polynomial.

```
c   Load the Toolbox definition files
!!G Toolbox.finc
!!MP PPCInlines
!!SETC USINGINCLUDES = .FALSE.


********************************************************************************
    subroutine DRAWRASTER
*   subroutine to draw the filled in julia set in raster mode.
********************************************************************************
    implicit    none
    integer*2   xx, yy, x, y
    integer*2   level, index(6)
    integer*2   length
    integer*2   i, j
    integer     n, width, height, numoffix
    global      width, height
    integer*2   color, cmatrix(0:1000,0:750)
    global      cmatrix
    logical     monocolor
    real        xmin, xmax, ymin, factor, r
    real        originalr
    real        radius(100), RTILDE
    external    RTILDE
    real        EVALDERIV
    external    EVALDERIV
    complex*16  a(0:100), fixpoint(100), roots(100)
    record /Rect/ myrect
    record /WindowPtr/ mywindow
```

```
    common /limits/ xmin, xmax, ymin
    global      r, a, n, mywindow

    DATA cmatrix /751751*0/        ! sets color matrix to white
*--------------------------------------------------------------------------------

    call SelectWindow (mywindow)
    call SetPort (mywindow)

*   -- store original escape radius and compute square
    originalr = r
    r = r*r

    factor = width/(xmax-xmin)

*   -- search for attracting fixed points
*   -- modify polynomial so you can search for zero instead of fixed point
    a(1) = a(1) - CMPLX(1)
    call zroots(a,n,roots,.true.)
*   -- restore original polynomial
    a(1) = a(1) + CMPLX(1)
    numoffix = 0
    do i = 1,n
        if (ABS(EVALDERIV(a,n,roots(i))) < 0.95) then
            numoffix = numoffix + 1
            fixpoint(numoffix) = roots(i)
            radius(numoffix) = RTILDE(a,n,fixpoint(numoffix))
*           -- store square of converge radius
            radius(numoffix) = radius(numoffix)*radius(numoffix)
        end if
    end do

    do while (button)
    end do

    do xx = 0,width,64
        do yy = 0,height,64
*           -- click => restore original escape radius and return
            if (button) then
                r = originalr
                return
            end if
            level = 1
            index(1) = 1
            length = 64
            do while (level .ne. 0)
                do while (index(level) = 5)
                    level = level - 1
                    length = length*2
                    index(level) = index(level) + 1
                end do
                if (index(1) = 2) exit
*               -- calculate current coordinates
                x = xx
                y = yy
                do i = 2, level
                    y = y + 2**(6-i+1)*(index(i)/3)
                    if (MOD(index(i),2) = 0) x = x + 2**(6-i+1)
                end do
*               -- square consists only of four points => draw directly
                if (length = 2) then
                    call FOURPOINT (factor, x, y, fixpoint, radius, numoffix)
                    index(level) = index(level) + 1
                else
*                   -- check if boundary of rectangle is all one color
                    call CHECKER (factor,x,y,length,color,monocolor,fixpoint,radius,numoffix)
                    if (monocolor) then
                        do i = x,x+length-1
```

```
                        do j = y,y+length-1
                            cmatrix(i,j) = color
                        end do
                    end do
                    myrect.top = y
                    myrect.bottom = y + length
                    myrect.left = x
                    myrect.right = x + length
                    call PmForeColor (color)
                    call PaintRect (myrect)
                    index(level) = index(level) + 1
                else
                    level = level + 1
                    length = length/2
                    index(level) = 1
                end if
            end if
        end do
    end do
end do

*   -- restore original escape radius
    r = originalr

    return
    end


********************************************************************************
    subroutine FOURPOINT(factor, x, y, fixpoint, radius, numoffix)
*   subroutine to draw four points (2x2 square).
********************************************************************************
    implicit    none
    integer*2   i, j, x, y
    integer     n, height, width, numoffix
    global      height, width
    integer*2   GETCOLOR
    integer*2   color, cmatrix(0:1000,0:750)
    global      cmatrix
    real        factor, xmin, xmax, ymin, zre
    complex*16  a(0:100), z, fixpoint(numoffix)
    global      a, n
    real        radius(numoffix)
    common /limits/ xmin, xmax, ymin
*-------------------------------------------------------------------------------

    if (x > width .or. y > height) then
        cmatrix(x,y) = 1
        cmatrix(x+1,y) = 1
        cmatrix(x,y+1) = 1
        cmatrix(x+1,y+1) = 1
        return
    end if
    do i = x, x+1
        zre = i/factor + xmin
        do j = y, y+1
            if (cmatrix(i,j) .ne. 0) then
                color = cmatrix(i,j)
            else
                z = CMPLX(zre, (height-j)/factor + ymin)
                color = GETCOLOR(a,n,z,fixpoint,radius,numoffix)
            end if
            call PmForeColor (color)
            call MoveTo(i,j)
            call LineTo(i,j)
            cmatrix(i,j) = color
        end do
    end do
```

```
      end


******************************************************************************
      subroutine CHECKER (factor,x,y,length,oldcolor,monocolor, fixpoint, radius, numoffix)
*     subroutine that checks if a square with upper left corner (x,y) and side
*     length "length" is all one color.
******************************************************************************
      implicit    none
      integer*2   x, y
      integer     numoffix
      integer*2   length
      integer     n, i, height
      global      height
      integer*2   oldcolor, newcolor, GETCOLOR
      integer*2   cmatrix(0:1000,0:750)
      global      cmatrix
      real        xmin, xmax, ymin, factor, dummy1, dummy2
      complex*16  a(0:100), z, fixpoint(numoffix)
      global      a, n
      real        radius(numoffix)
      logical     monocolor
      common /limits/ xmin, xmax, ymin
*-----------------------------------------------------------------------------

      monocolor = .true.
      dummy1 = (height-y)/factor + ymin
      if (x .ne. 0 .and. y .ne. 0) then
          oldcolor = cmatrix(x-1,y-1)
      else
          z = CMPLX(x/factor + xmin, dummy1)
          oldcolor = GETCOLOR(a,n,z,fixpoint,radius,numoffix)
          cmatrix(x,y) = oldcolor
      end if

*     -- checking of the upper horizontal
      if (y = 0) then
          do i = x, x + length - 1, 2
              z = CMPLX(i/factor + xmin, dummy1)
              newcolor = GETCOLOR(a,n,z,fixpoint,radius,numoffix)
              cmatrix(i,y) = newcolor
              if (newcolor .ne. oldcolor) then
                  monocolor = .false.
                  return
              end if
          end do
      else
          do i = x, x + length - 1, 2
              newcolor = cmatrix(i,y-1)
              if (newcolor .ne. oldcolor) then
                  monocolor = .false.
                  return
              end if
          end do
      end if

*     -- checking of the left vertical
      dummy1 = x/factor + xmin
      if (x = 0) then
          do i = y, y + length - 1, 2
              z = CMPLX(dummy1, (height-i)/factor + ymin)
              newcolor = GETCOLOR(a,n,z,fixpoint,radius,numoffix)
              cmatrix(x,i) = newcolor
              if (newcolor .ne. oldcolor) then
                  monocolor = .false.
                  return
              end if
          end do
```

```
      else
          do i = y, y + length - 1, 2
              newcolor = cmatrix(x-1,i)
              if (newcolor .ne. oldcolor) then
                  monocolor = .false.
                  return
              end if
          end do
      end if

*     -- checking of the lower horizontal
      dummy2 = (height-y-length+1)/factor + ymin
      do i = x, x + length - 1, 2
          if (cmatrix(i,y+length-1) .ne. 0) then
              newcolor = cmatrix(i,y+length-1)
          else
              z = CMPLX(i/factor + xmin, dummy2)
              newcolor = GETCOLOR(a,n,z,fixpoint,radius,numoffix)
              cmatrix(i,y+length-1) = newcolor
          end if
          if (newcolor .ne. oldcolor) then
              monocolor = .false.
              return
          end if
      end do

*     -- checking of the right vertical
      dummy2 = (x+length-1)/factor + xmin
      do i = y, y + length - 1, 2
          if (cmatrix(x+length-1,i) .ne. 0) then
              newcolor = cmatrix(x+length-1,i)
          else
              z = CMPLX(dummy2, (height-i)/factor + ymin)
              newcolor = GETCOLOR(a,n,z,fixpoint,radius,numoffix)
              cmatrix(x+length-1,i) = newcolor
          end if
          if (newcolor .ne. oldcolor) then
              monocolor = .false.
              return
          end if
      end do
      end



********************************************************************************
      integer*2 function GETCOLOR(a,n,z,fixpoint,radius,numoffix)
*     subroutine to get the color corresponding to the point z.
********************************************************************************
      implicit    none
      integer     i, n, k, numoffix
      integer*4   iterations
      real        r, radius(numoffix)
      real        SQUAREABS
      complex*16  a(0:n), z, EVALPOLY, fixpoint(numoffix)
      external    EVALPOLY, SQUAREABS
      global      r, iterations
*-------------------------------------------------------------------------------

      abbruch = .false.
      k = 0
      do while (.true.)
          if (SQUAREABS(z) > r) then
              if (iterations < 224) then
                  GETCOLOR = int2((223./iterations)*k + 16)
              else
                  GETCOLOR = MOD(k,224) + 16
              end if
```

```
            return
        end if
        k = k + 1
        if (k > iterations) then
            GETCOLOR = 1
            return
        end if
        do i = 1, numoffix
            if (SQUAREABS(z-fixpoint(i)) < radius(i)) then
                GETCOLOR = 1
                return
            end if
        end do
        z = EVALPOLY(a,n,z)
    end do
    end


********************************************************************************
    complex*16 function EVALPOLY(a,n,z)
*   subroutine to evaluate a polynomial via Horner's scheme
********************************************************************************
    integer n, i
    complex*16 z, a(0:n), result
*-------------------------------------------------------------------------------

    result = (0,0)
    do i = n,1,-1
        result = (result + a(i))*z
    enddo
    result = result + a(0)

    EVALPOLY = result
    end


********************************************************************************
    real function EVALDERIV(a,n,z)
*   subroutine to evaluate the abs of the first derivative of a polynomial in z.
********************************************************************************
    integer n, i
    complex*16 z, a(0:n), result, b(0:99)
*-------------------------------------------------------------------------------

    result = (0,0)
    do i = 0,n-1
        b(i) = (i+1)*a(i+1)
    enddo
    result = EVALPOLY(b,n-1,z)

    EVALDERIV = ABS(result)
    end


********************************************************************************
    real function SQUAREABS(z)
*   computes |z|^2
********************************************************************************
    complex*16 z
*-------------------------------------------------------------------------------

    SQUAREABS = DREAL(z)*DREAL(z) + DIMAG(z)*DIMAG(z)

    end


********************************************************************************
    real function RTILDE(a,n,fixpt)
```

```
*   subroutine to calculate the converge radius for the fixed point fixpt
********************************************************************************
    integer     i, n, k, l
    integer*4   BINOMIAL
    real        rt
    complex*16  a(0:100), fixpt
    complex*16  b(0:100), roots(0:99)
*-------------------------------------------------------------------------------
    do i = 0,n
        b(i) = DCMPLX(0)
    enddo

*   -- calculate coefficients of P(z+fixpt) - fixpt
    do l = 0,n
        do k = 0,l
            b(l-k) = b(l-k) + a(l)*BINOMIAL(l,k)*fixpt**k
        enddo
    enddo

*   -- divide polynomial by z and subtract 1
    do i = 0,n-1
        b(i) = DCMPLX(CDABS(b(i+1)))
    enddo
    b(0) = b(0) - DCMPLX(1)

*   -- find smallest positive root of polynomial generated above
    call zroots(b,n-1,roots,.true.)
    rt = 0
    i = 0
    do while (rt = 0)
        if (DREAL(roots(i)) > 0 .and. DABS(DIMAG(roots(i))) < 1d-7) rt = DREAL(roots(i))
        i = i+1
        if (i = n) then
            call AlertBox('Calculation of rtilde failed.')
            rt = -1
        endif
    enddo
    RTILDE = rt
    end


********************************************************************************
    integer*4 function BINOMIAL(n,k)
*   subroutine to the binomial (n k) = n!/((n-k)!*k!)
********************************************************************************
    integer     i, n, k
    integer*4   result
*-------------------------------------------------------------------------------

    result = 1
    do i = n,n-k+1,-1
        result = result*i
    enddo
    do i = 1,k
        result = result/i
    enddo

    BINOMIAL = result
    end
```

List of Figures

List of Symbols

$\mathbb{C}$	set of complex numbers
$\bar{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$	extended complex plane
$\mathbb{N}$	set of positive integers
$\mathbb{Q}$	set of rational numbers
$\mathbb{R}$	set of real numbers
$\mathbb{Z}$	set of integers
$\mathcal{E}$	$\{z \in \mathbb{C} \mid \lvert P(z)\rvert = \lvert z\rvert\}$
$\mathcal{F}$, $\mathcal{F}_P$, $\mathcal{F}_R$	Fatou set of a complex polynomial or rational function
$\mathcal{G} = \mathcal{G}(\tilde{z})$	$\{z \in \mathbb{C} \mid \lvert P(z+\tilde{z}) - \tilde{z}\rvert = \lvert z\rvert\}$
$\mathcal{J}$, $\mathcal{J}_P$, $\mathcal{J}_R$	Julia set of a complex polynomial or rational function
$\mathcal{K}$, $\mathcal{K}_P$	filled Julia set of a complex polynomial
$\mathcal{M}$	Mandelbrot set
$\Im(z)$	imaginary part of the complex number z
$\Re(z)$	real part of the complex number z
$cl\{\dots\} = \overline{\{\dots\}}$	closure of a set
$A(z_0)$	basin of attraction of z_0
$A^*(z_0)$	immediate basin of attraction of z_0
$B(z_0, r)$	open disk of radius r around z_0
CL	closure of the postcritical set
N_f	Newton function for the analytic function f
P	a complex polynomial
$P_c(z)$	$z^2 + c$
$P_\rho(z)$	$z^3 + (\rho - 1)z - \rho$
R	a complex rational function

Bibliography

[Bea] Alan F. Beardon: *Iteration of Rational Functions*, Springer-Verlag New York, 1991

[Bl1] Paul Blanchard: *Complex Analytic Dynamics on the Riemann Sphere*, Bull. Amer. Math. Soc., 11, 1984, pp. 85–141

[Bl2] Paul Blanchard: *The Dynamics of Newton's Method*, in: [De3], pp. 139–154

[Bra] Bodil Branner: *The Mandelbrot Set*, in: [DK], pp. 75–105

[Bro] Hans Brolin: *Invariant sets under iteration of rational functions*, Arkiv för Matematik Band 6 nr 6, 1965

[CG] Lennart Carleson, Theodore W. Gamelin: *Complex Dynamics*, Springer-Verlag New York, 1993

[CGS] James H. Curry, Lucy Garnett, Dennis Sullivan: *On the Iteration of a Rational Function: Computer Experiments with Newton's Method*, Communications in Mathematical Physics 91, pp. 267–277, 1983

[CM] James B. Cooper, Paul Müller: *Skriptum zur Funktionentheorie*, Linz, 1994

[CR] Richard Courant, Herbert Robbins: *Was ist Mathematik?*, Dritte Auflage, Springer-Verlag Berlin, Heidelberg, New York, 1973

[De1] Robert L. Devaney: *An Introduction to Chaotic Dynamical Systems*, 2nd ed., Menlo Park, Addison-Wesley, 1989

[De2] Robert L. Devaney: *Chaos, Fractals, and Dynamics*, Menlo Park, Addison-Wesley, 1990

[De3] Robert L. Devaney (editor): *Complex Dynamical Systems: The Mathematics behind the Mandelbrot and Julia Sets*, Proceedings of Symposia in Applied Mathematics, Vol. 49, Amer. Math. Soc., 1994

[DK] Robert L. Devaney, Linda Keen (editors): *Chaos and Fractals: The Mathematics Behind the Computer Graphics*, Proceedings of Symposia in Applied Mathematics, Vol. 39, Amer. Math. Soc., 1988

[Dou] Adrien Douady: *Does a Julia set depend continuously on the Polynomial?*, in: [De3], pp. 91–138

[Gul] Denny Gulick: *Encounters with Chaos*, McGraw-Hill International Editions, New York, Singapur, 1992

[Hol] Richard A. Holmgren: *A First Course in Discrete Dynamical Systems*, Springer-Verlag New York, 1994

[Inn] Clemens Inninger: *Julia and Fatou Sets of Rational Functions and a Characterization of Polynomials with Real Julia Sets*, Diplomarbeit, JKU Linz, 1997

[LR] Daniel H. Luecking, Lee A. Rubel: *Complex Analysis: A Functional Analysis Approach*, Springer-Verlag New York, Berlin, Heidelberg, Tokyo, 1984

[Mard] Morris Marden: *Geometry of Polynomials*, American Mathematical Society, Providence, Rhode Island, 1966

[Mark] A. I. Markushevich: *Theory of Functions of a Complex Variable*, Vol. II, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1965

[Neh] Zeev Nehari: *Introduction to Complex Analysis*, Revised Edition, Allyn and Bacon, Inc., Boston, 1968

[Peh] Franz Peherstorfer: *Skriptum zur Vorlesung Funktionentheorie*, Linz, 1996

[Per] Lawrence Perko, *Differential Equations and Dynamical Systems*, Springer-Verlag New York, 1991

[PR] Heinz-Otto Peitgen, Peter H. Richter: *The Beauty of Fractals*, Springer-Verlag Berlin, Heidelberg, New York, Tokyo, 1986

[PS] Heinz-Otto Peitgen, Dietmar Saupe: *The Science of Fractal Images*, Springer-Verlag Berlin, Heidelberg, New York, 1988

[PTVF] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery: *Numerical Recipes in FORTRAN*, Cambridge University Press 1986, 1992

[Rem] Reinhold Remmert: *Funktionentheorie 1*, 4. Auflage, Springer-Verlag Berlin, Heidelberg, 1995

[Sto] Josef Stoer: *Numerische Mathematik*, 5. Auflage, Springer-Verlag Berlin, Heidelberg, New York, 1989