EXPERIMENTAL APPROACHES ON PROBLEMS CONCERNING THE GEOMETRY OF POLYNOMIALS

by

Jens Happe

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the Department
of
Mathematics and Statistics

SIMON FRASER UNIVERSITY

February 1995

ISBN 0-612-06673-8

APPROVAL

Name: Jens Happe

Degree: Master of Science

Title of thesis: Experimental Approaches on Problems concerning the Geometry of Polynomials

Examining Committee: Dr. G.A.C. Graham
Chair

Dr. Peter Borwein
Senior Supervisor

Dr. Jonathan Borwein

Dr. Brian Thomson

Dr. Manfred Trummer
External Examiner

Department of Mathematics and Statistics
SIMON FRASER UNIVERSITY

Date Approved: February 8, 1995

Partial Copyright License

Title of Thesis/Project/Extended Essay:

Experimental Approaches on Problems concerning the

Geometry of Polynomials

Author: ______________________

(signature)

Jens Happe

(name)

19/02/1995

(date)

Abstract

Many interesting conjectures have been made about the location of zeros of a given polynomial p and the location of zeros of related structures. The first relationship known to be investigated systematically is between nonreal zeros of p and the so-called Fourier critical points of p. Gauss dealt with this problem in a vague way in 1833. His work was continued by Pólya. We will state and prove rigorously that such a relationship can be found, except for certain limiting cases. A more recent conjecture by Craven, Csordas and Smith about the zeros and the zeros of the Wronskian of p is cited and investigated.

This thesis follows a new approach to these problems: A graphical package has been programmed that, running on graphical workstations, lets one interactively design and modify polynomials by their zeros: it also allows one to compute related structures, and visualize them with various internal and external graphing tools. Its usage and applications are also described in this thesis.

Quotation

Are you still taking as much delight in the quest for truth as you were? Truly, it is not the knowledge, but the acquistion, not the presence, but the addition, which renders the most delight. When I clarified and exhausted a topic completely, I leave it, in order to go back into the dark. Of such a strange kind is the insatiable man: Once he completed a building, it is not for him to live in it, but to commence building another one. This I imagine must be the feeling of an emperor who, no sooner than one kingdom is conquered, already reaches out for the next.

Carl Friedrich Gauss, in a letter to F. Bolyai, Sept. 02, 1808

Macht Dir das Nachforschen der Wahrheit noch ebenso viel Freude wie sonst? Wahrlich, es ist nicht das Wissen, sondern das Erwerben, nicht das Dasein, sondern das Hinzukommen, was den größten Genuß gewährt. Wenn ich eine Sache ganz ins Klare gebracht und erschöpft habe, so wende ich mich davon weg, um wieder ins Dunkle zu gehen; so sonderbar ist der nimmersatte Mensch: Hat er ein Gebäude vollendet, so ist es nicht, um darin zu wohnen, sondern um ein anderes anzufangen. So, stelle ich mir vor, muß einem Welteroberer zu Mute sein, der, nachdem ein Königreich kaum bezwungen ist, schon wieder nach andern seine Arme ausstreckt.

Carl Friedrich Gauß, an F. Bolyai, 2.9. 1808

Acknowledgements

I would like to express my gratitude to Dr. Peter Borwein and Dr. Jonathan Borwein, for introducing me into the world of experimental mathematics, and for their guidance and encouragement in my research.

I acknowledge the excellent technical support I have received from Dr. Loki Jörgenson throughout my programming. But my very special thanks go to Jack Ho, for the countlessly many hours he spent with me finding my programming mistakes.

Thanks also to all present and former members of the Centre for Experimental and Constructive Mathematics, for their companionship and their good ideas.

I am deeply grateful to Dr. Karl Dilcher, without whom I would certainly not have had the opportunity to do my M.Sc. degree in this form.

Contents

List of Figures

Chapter 1

Introduction

This thesis is based on two interesting, and incompletely answered, questions related to the geometry of real or complex polynomials. A quick overview of the two conjectures is given here, at this point without all the necessary definitions. A full discussion of them is reserved for Chapters 2 and 3.

Let p be a polynomial with real coefficients.

Conjecture 2.16 (Gauss, 1833)[1]: There exists a definite one-to-one relationship between the pairs of conjugate non-real zeros of p and the Fourier critical points of p.

Conjecture 3.11 (Csordas, Craven, Smith, 1990)[2]: The number of real zeros of the Wronskian of p does not exceed the number of nonreal zeros of p.

The first conjecture (in a modified version) will be proved in this thesis, and an approach to the second conjecture will be outlined.

When one looks at articles in the field of the Geometry of Polynomials, especially at articles from the turn of the century and earlier, one finds only a few graphical illustrations. Topics are dealt with and results are presented in a purely analytic way, and one would naïvely assume that graphical illustrations of these problems only existed in the heads of the ingenious people who worked on them. But of course, there is a different reason: Until recently, it required a tremendous effort to draw exact graphs. Even apart from drawing,

[1] Gauss did not conjecture this so much as question it, as will be shown in Chapter 2.

[2] This conjecture has been incorrectly attributed to Gauss, as we will remark in Chapter 3.

computing points with sufficient exactness to produce a smooth graph was a tedious and time-consuming job. Gauss complained explicitly about the effort he took in drawing a simple-looking diagram in his Doctor's thesis ([9], see also Figure A.1 in the Appendix). In a letter to his friend Schumacher [11], he wrote that he would most certainly not be able to spend the immense amounts of time necessary to draw graphs for all cases of Conjecture 2.16. It appears that this was the major reason that prevented him from pursuing his conjecture further.

Nowadays, tools for graphing and illustrating are readily available since graphical terminals turned the previously text-oriented computer world into one with colourful windows, icons and images. Similarly, high-level tools for symbolic computation have been developed and are continuously expanding, doing all the computations for plotting graphs in a few seconds.

However, these tools have not been exploited very much to find proofs for conjectures like these. One reason might be that they are still somewhat rudimentary. While output can be graphed in almost any imaginable way, the formulae and data to generate them still have to be typed in as text (or at least loaded from a text file). We found it desirable to possess a completely graphical user interface, that not only allows us to input and modify data on a graphical screen, but also responds graphically at the same place, thus enabling user-computer interaction in its true meaning.

A step in this direction taken in this thesis is the development of the attached package, *xzero*. *xzero* lets one create and manipulate polynomials up to a multiplicative factor, by defining their zeros in a "virtual" complex plane on the screen. Computations on these polynomials can then be done by calling the implemented operations. Depending on the nature of the results, they will be graphed in the same area, allowing direct comparison with the input data. Or they will be graphed in a separate window, or simply displayed as plain text. It is also possible to define one's own procedures to generate specific results.

The spectrum of applications of *xzero* is restricted to polynomials, but otherwise quite general. As examples, the two conjectures cited above have been examined. For the first question, illustrations generated with the aid of *xzero* will be presented along with the proof. Some experimental results towards Conjecture 3.11 obtained with *xzero* will be shown in Chapter 5.

In Chapter 4, we will describe the system requirements and general usage of *xzero*. This chapter may be referenced as a manual to the program. Technical details on configuring *xzero*. writing one's own procedures and programming strategies used in writing *xzero* are left to the appendix, and to the documentation within the *xzero* package.

Chapter 2

Gauss' original conjecture

One of the interesting things a student learns in a calculus course is that the coefficients of the derivative of a polynomial p can easily be determined from the coefficients of p: They are just multiples of the next-higher coefficients. However, when looking at the zeros of p and its derivative, one doesn't find such easy relationships (except that the number of complex zeros must decrease by one per derivative taken). One must wonder if there is a relationship at all. This question has, in some sense, been open for more than 160 years. We will rephrase this question in this chapter, and give a positive answer to the modified question. Throughout, we assume p to be a *real* polynomial, or equivalently: to have real coefficients, unless explicitly stated otherwise. We do not allow p to be the null polynomial. (We will write $\mathcal{P}$ for the set of all such polynomials.) The set $I\!N$ of natural numbers is understood to be the set of *positive* integers, thus excluding 0.

2.1 The coefficient sequence of a polynomial

The way in which many mathematicians of the 19th century viewed polynomials appears quite complicated nowadays. Instead of the derivatives of a polynomial, they studied the sequence of its coefficients, and particularly, sign changes occurring in this sequence. We will first examine this way of describing polynomials and explain how it is related to the modern, more comprehensive notation of the derivatives of a polynomial. This description essentially follows [24], pp. 28ff.

Definition 2.1 For $x \in \mathbb{R}$. construct

$$p(\xi + x) = \sum_{k=0}^{n} c_k(x)\xi^k$$

and define the vector

$$\sigma(x) = (\sigma_0(x) \dots \sigma_n(x)) = (\operatorname{sgn} c_k(x))_{k=0,\dots,n}$$

which contains the signs of all coefficients of p.

Using the notion of the derivative, this can be written in the following more familiar-looking form:

Corollary 2.2 *For every $k = 0, \dots, n$, and every $x \in \mathbb{R}$, we have $c_k(x) = \frac{1}{k!}p^{(k)}(x)$, and $\sigma(x)$ is elementwise equal to the vector $(\operatorname{sgn} p^{(k)}(x))_{k=0,\dots,n}$, where $p^{(0)} = p$.*

Proof: This is an easy calculus exercise. □

The vector $\sigma(x)$, regarded as a vector-valued function in x, provides a "ternary" description of p, in that its components take only the values -1, 0, and 1. The "points of interest" in the behaviour of $\sigma(x)$ are those points where some components of $\sigma(x)$ vanish, because they are the points of possible sign changes. The following theorem states this more precisely:

Theorem 2.3 *Let a and b be real numbers, and $a < b$. If $\sigma(a) \neq \sigma(b)$, then there exists a point $x \in [a, b]$ at which $\sigma(x)$ contains a zero component.*

Proof: If either $\sigma(a)$ or $\sigma(b)$ has a zero component, we have nothing to show. Otherwise, there exists an index k such that $\operatorname{sgn} p^{(k)}(a) = -\operatorname{sgn} p^{(k)}(b) \neq 0$. From the intermediate value theorem follows the existence of a point $x \in (a, b)$ where $p^{(k)}(x)$ is zero. □

For convenience, we denote as *zero points* all points where p or one of its first n derivatives is zero, i.e. where some $\sigma_k(x)$ are 0. In this setting, the preceding theorem (in its negated form) reads that in the open intervals between the (finitely many) zero points, $\sigma(x)$ does not change. In order to categorize these points further, we need some more terminology:

Definition 2.4 Let x be a non-zero point. We say, $\sigma(x)$ (or $p(x)$) has a *variation* at $\sigma_k(x)$ (or at $p^{(k)}(x)$) if $\sigma_k(x) = -\sigma_{k+1}(x)$. By $v_p(x)$ we denote the total number of variations[1] in $\sigma(x)$.

This definition has nothing to do with the definition of variance in Measure Theory. It is independent of the absolute value of p. In [7], Gauss is speaking of *sign changes.*

Now how do $\sigma(x)$ and $v_p(x)$ change at the zero points? To answer this for all possible cases, we recall two results in real polynomial calculus:

Proposition 2.5 *Let a be a point where $p(a) = 0$.*

(1) If p changes its sign at a from negative to positive, then there exists a positive ϵ so that p' is positive in the open interval $(a - \epsilon, a + \epsilon)$, except possibly at a where p' is nonnegative.

(2) If p does not change its sign at a, then $p'(a) = 0$, and there exists a positive ϵ so that p' is negative in $(a - \epsilon, a)$ and positive in $(a, a + \epsilon)$.

Both (1) and (2) also hold if the words "negative" and "positive" are exchanged.

Proof: On recalling that the zeros of p and p' form a discrete set, and that p' is continuous, these statements are an immediate consequence of the mean and intermediate value theorems. □

We apply this proposition to analyze how $\sigma(x)$ and $v_p(x)$ are affected as x goes through a zero point a. As above, all cases are stated only for positive $p(x)$, or for $p(x)$ changing from negative to positive, but remain valid if "negative" and "positive" are exchanged.

Case 1: If $p(a) = 0$, and $p(x)$ (and hence $\sigma_0(x)$) changes its sign in a: Then by (1), $p'(x)$ is positive and does not change its sign. Thus, the variation of $\sigma(x)$ at σ_0 disappears. If no other component of $\sigma(x)$ vanishes at a, then $v_p(x)$ diminishes by 1 as x goes through a.

[1] If x is a zero point, say, such that $\sigma_{k+1}(x) = \ldots = \sigma_{k+l}(x) = 0$, one can define that $\sigma(x)$ has a variation at $\sigma_k(x)$ when $\sigma_k(x) = -\sigma_{k+l+1}(x)$. This definition would be consistent to our results, and we will see that the corresponding value of $v_p(x)$ is just the limit of $v_p(\xi)$ as $\xi \searrow x$. We will study the behaviour of $v_p(x)$ in neighbourhoods of zero points, and speak of "changes in $v_p(x)$ as x goes through the zero point x_0".

Case 2: If p does not change its sign, (but possibly $p(a) = 0$) and p' changes from negative to positive: Then again by (1), p'' is positive and does not change its sign. Hence, the variations of $\sigma(x)$ both at $\sigma_0(x)$ and $\sigma_1(x)$ disappear. If no other component of $\sigma(x)$ vanishes at a, then $v_p(x)$ diminishes by 2 as x goes through a.

Case 3: If p does not change its sign, and p' changes from positive to negative: Note that by (2) this can only occur if $p(a) > 0$, so p has a positive maximum at a. By (1), p'' must be negative without a sign change. $\sigma(x)$ abandons its variation at $\sigma_1(x)$, but gains one at $\sigma_0(x)$. Hence, if no other component of $\sigma(x)$ vanishes at a, then $v_p(x)$ remains unchanged as x goes through a.

Case 4: Neither p nor p' change their sign. Then p cannot vanish at a, according to (1). If $\sigma(x)$ has a variation at $\sigma_0(x)$, it keeps it through a, so $v_p(x)$ remains unchanged as x goes through a, if no other component of $\sigma(x)$ vanishes at a.

Case 5: To each derivative of p that does not change its sign, apply the appropriate choice out of Cases 2 through 4, to determine the variation of the $\sigma_k(x)$ of higher indices k. The effects of all sign changes add up to the total decrease of $v_p(x)$ in a.

Case 5 allows us to handle multiple zero points at a independently, except for zeros in two subsequent derivatives which must be regarded simultaneously[2].

Corollary 2.6 *If $p(a) = p'(a) = \ldots = p^{(m-1)}(a) = 0$ and $p^{(m)}(a) \neq 0$, then the derivatives $p^{(m-2k)}(x), k = 1, \ldots, \lfloor \frac{m}{2} \rfloor$, don't change their sign at a, and the derivatives $p^{(m-2k+1)}(x), k = 1, \ldots, \lceil \frac{m}{2} \rceil$, change their sign at a. The derivatives $p(x), p'(x), \ldots, p^{(m)}(x)$ have alternating sign for a fixed x in some interval $(a - \epsilon, a)$, and constant sign for x in $(a, a + \epsilon)$, namely $\operatorname{sgn} p^{(m)}(a)$. Thus, $v_p(x)$ diminishes by m, due to this m-fold zero of p.*

Proof: Apply (1) and (2) of Proposition 2.5 alternatingly to the derivatives of p. □

Theorem 2.7

1. *The function $v_p(x)$ is monotonically decreasing.*

[2] We will use terms like "$v_p(x)$ diminishes by 2 due to $p^{(k)}(a)$" to emphasize that we disregard possible further defects of $v_p(x)$ due to zeros of other derivatives at a.

2. *For every sufficiently large negative a and large positive b*, $v_p(a) = n$ *and* $v_p(b) = 0$.

3. *The converse of Theorem 2.3 also holds, i.e.* $\sigma(x)$ *always changes as* x *goes through a zero point.*

Proof:

1. In any of the Cases above, $v_p(x)$ is either constant or decreases when passing a zero point. In the intermediate intervals, $\sigma(x)$ is unchanged, as is $v_p(x)$.

2. This follows from the fact that, for sufficiently large $|x|$, p and all its derivatives have the same sign as the polynomial $c_n x^n$ and its derivatives. For the latter, the statement holds for all $a < 0$ and $b > 0$, respectively, since the components of $\sigma(a)$ alternate, and the components of $\sigma(b)$ are consistently equal to $\operatorname{sgn} c_n$.

3. By giving the variations between the $k-1$st and kth component of $\sigma(x)$ a weight k, the so modified $v_p^{(w)}(x)$ decreases at all zero points, even in the Case 4 above. Therefore, if a and b are separated by one or more zero points, then $v_p^{(w)}(a) \neq v_p^{(w)}(b)$ and thus $\sigma(a) \neq \sigma(b)$.

□

2.2 Fourier Critical Points

As we have seen, describing the behaviour of $v_p(x)$ in terms of $\sigma(x)$ is rather complicated. Fourier took a different (and simpler) approach. He considered points with the following properties:

Definition 2.8 For a point $a \in \mathbb{R}$ where $p^{(l-1)}(a) \neq 0$, $p^{(l)}(a) = p^{(l+1)}(a) = \ldots = p^{(l+m-1)}(a) = 0$, $p^{(l+m)}(a) \neq 0$ for some $l, m \in \mathbb{N}$, let k be defined as:

$$k = \begin{cases} \frac{m}{2} & \text{if } m \text{ is even} \\ \frac{m+1}{2} & \text{if } m \text{ is odd and } p^{(l-1)}(a)p^{(l+m)}(a) > 0 \\ \frac{m-1}{2} & \text{if } m \text{ is odd and } p^{(l-1)}(a)p^{(l+m)}(a) < 0. \end{cases} \tag{2.1}$$

Then a is called a ***critical zero*** of p of ***order*** l, (ordinary) ***multiplicity*** m, and ***critical multiplicity*** k.

This is not Fourier's original definition, but rather a refined version due to Pólya [24]. We will see that the distinction in the case of odd m corresponds to Cases 2 and 3 of the discussion in the previous section. Note the following peculiarities in this definition:

- Unlike in real calculus, multiple zeros of p are not considered as critical zeros.
- Also unlike in real calculus, multiple critical zeros of p are considered only as critical zeros of order l, not as critical zeros of the next-higher derivatives.
- The case $k = 0$ is possible — namely, when $p^{(l)}(a) = 0$, and $p^{(l-1)}(a)$ and $p^{(l+1)}(a)$ are both non-zero and have different signs.
- In particular, the zero points that satisfy $k = 0$ and $l = 1$ are the simple negative minima and positive maxima of p.
- We speak of a *Fourier critical zero* (see also Definition 2.9) whenever k is strictly positive.

We will later see that real critical zeros fit much better into the theory when specifying them with their critical multiplicity, rather than with their ordinary multiplicity.

A point a may occur as a critical zero of p of several different orders, possibly with different critical multiplicities. Hence, a term that covers all occurrences of critical zeros at a is in order:

Definition 2.9 A point a is called *Fourier critical point* of p if it is a critical zero of p of some order with positive critical multiplicity.

The *(total) critical multiplicity* of a is defined as the sum of the critical multiplicities of all critical zeros at a.

Again we must distinguish between the common definition of critical points, and Fourier critical points. In particular, a point where all critical zeros have multiplicity 1 and critical multiplicity 0 is not considered a Fourier critical point. Furthermore, the ordinary multiplicity of a Fourier critical point is in general not equal to its critical multiplicity.

To illustrate the distinction between critical zeros and Fourier critical points, let us give an example of a polynomial with two critical zeros at 0.

Example 2.10 The polynomial $p(x) = (x^2+1)^2 = x^4 + 2x^2 + 1$ has 4 imaginary zeros, counting multiplicities. It can easily be seen that p, p'' and $p^{(4)}$ are strictly positive, whereas p' and p''' each have one root at $x = 0$. It follows that 0 is a critical zero of p' and p''', both with multiplicity 1 and critical multiplicity 1. Therefore, $a = 0$ is the only Fourier critical point of p; its total critical multiplicity is 2. Note that this is also the number of pairs of nonreal zeros of p. This equality always holds, as we will now show.

Theorem 2.11 *If a is a real zero of p with multiplicity m and a Fourier critical point of critical multiplicity k, then $v_p(a-\epsilon) - v_p(a+\epsilon) = m + 2k$ for a sufficiently small ϵ.*

Note that m and k are allowed to be 0 here. The case $k = 0$ means that no critical zero at a is of positive critical multiplicity. (This case will be included implicitly in the proof.) In the trivial case $m = k = 0$, $v_p(x)$ remains constant as x goes through a.

Proof: We refer to the case distinction in the previous section. If critical zeros of different orders, and perhaps a real zero, coincide at a, then Case 5 allows us to regard these zeros independently and to add up the defects in $v_p(x)$ due to all zeros. For a real zero of multiplicity m, the theorem follows from Corollary 2.6.

Now assume that a is an m-fold critical zero of order l and critical multiplicity k as in (2.1). Without loss of generality, we assume that $p^{(l+m)}(a)$ is positive. Applying the first part of Corollary 2.6 to $p^{(l)}(a)$, we obtain that the derivatives $p^{(l+m-2)}, \dots, p^{(l+m-2\lfloor\frac{m}{2}\rfloor)}$ are all positive in some interval around a (except at a itself). Now Case 2 applies to each of these $\lfloor\frac{m}{2}\rfloor$ derivatives, and the defect of $v_p(x)$ due to them amounts to $2\lfloor\frac{m}{2}\rfloor$. It now remains to check the derivatives $p^{(l-1)}$ and $p^{(l)}$ in all possible cases:

- If m is even (first line in (2.1)), then, as said above, $p^{(l+m-2\lfloor\frac{m}{2}\rfloor)} = p^{(l)}$ does not change its sign; nor does $p^{(l-1)}$. So Case 4 applies, in which $v_p(x)$ does not decrease any further. Hence the total defect of $v_p(x)$ at a is $m = 2k$.

- If m is odd, then $p^{(l)}(x)$ changes from negative to positive.
    - If further $p^{(l-1)}(a) > 0$ (this corresponds to the second line in (2.1)), then Case 2 applies, in which $v_p(x)$ diminishes by 2, due to $p^{(l-1)}$. Hence, the total defect of $v_p(x)$ at a is $2\lfloor\frac{m}{2}\rfloor + 2 = m + 1 = 2k$.

- Otherwise, if $p^{(l-1)}(a) < 0$ (third line in (2.1)), then Case 3 applies, in which $v_p(x)$ does not decrease any further. Hence, the total defect of $v_p(x)$ at a is $2\lfloor\frac{m}{2}\rfloor = m - 1 = 2k$.

In either case, $v_p(x)$ diminishes by twice the critical multiplicity of a, as specified in Definition 2.8. This completes the proof. □

Since every non-zero polynomial has only finitely many zero points, we can extend this "local" theorem to an (almost) arbitrary interval which may cover several zeros and critical points of p.

Theorem 2.12 *If a and b are non-zero points of p, $a < b$, then*

$$v_p(a) - v_p(b) = \sum_{a<x_z<b} \mu(x_z) + 2 \sum_{a<x_{cr}<b} k(x_{cr}) \tag{2.2}$$

where x_z and x_{cr} are the roots and Fourier critical points of p with multiplicities $\mu(x_z)$ and critical multiplicities $k(x_{cr})$, respectively.

This is commonly referred to as Fourier's theorem. In [24, 13] (see Appendix A), the case that a or b are themselves zero points is also dealt with. We will not need these extensions.

Proof: For every zero $x \in (a, b)$, a neighbourhood $(x-\epsilon, x+\epsilon)$ can be found so that Theorem 2.12 applies, giving $v_p(x-\epsilon) - v_p(x+\epsilon) = \mu(x) + 2k(x)$. From Theorem 2.3 it is known that $v_p(x)$ doesn't change in the intermediary intervals, and that $v_p(x)$ monotonically decreases. Therefore, the defects in $v_p(x)$ due to each zero point can simply be added together. Note that critical points which are neither roots nor Fourier critical points don't contribute to these sums and are automatically disregarded, because $k(x_{cr}) = \mu(x_{cr}) = 0$. □

From Equation (2.2) we obtain an inequality which will prove useful much later in this chapter:

Theorem 2.13 *With a and b as defined in Theorem 2.12, let m and m' be the number of real zeros of p and p' in (a, b), respectively, counting multiplicities. Let further c be the number of critical zeros of order 1 in (a, b), counting critical multiplicities. Then we have the inequality $m + 2c - 1 \le m' \le m + 2c + 1$.*

Proof: First, observe that the value of the first sum term in (2.2) is just m. Now apply (2.2) to the derivative of p. It is clear that $v_{p'}(x)$ counts the same sign changes as $v_p(x)$ does, except for a possible sign change at p, and thus may be lower than $v_p(x)$ by at most 1. On the right-hand-side of (2.2) for p', the first sum is exactly m', and the second sum is taken over all critical zeros in (a, b), except for those of order 1 with their critical multiplicities, which were accounted for as c. Hence (2.2) for p' reads:

$$v_{p'}(a) - v_{p'}(b) = m' + 2\left(\sum_{a<x_{cr}<b} k(x_{cr}) \; - c\right) \tag{2.3}$$

Now we take the difference between (2.2) for p, and (2.3). We obtain

$$(v_p(a) - v_{p'}(a)) - (v_p(b) - v_{p'}(b)) = m + 2c - m' \tag{2.4}$$

According to our previous observations, both left-hand terms can be either 0 or -1, so the value on the left-hand side of (2.4) lies between -1 and 1. This proves the theorem. □

Keeping in mind that, according to Part 2. of Theorem 2.7, there exist real values a and b such that $v_p(a) - v_p(b) = n$, we get the following

Corollary 2.14

$$n = \sum_{x_z} \mu(x_z) + 2\sum_{x_{cr}} k(x_{cr}) \tag{2.5}$$

where the summation is taken over all roots x_z and Fourier critical points x_{cr} of p, and $\mu(x_z)$ and $k(x_{cr})$ are their respective multiplicities and critical multiplicities.

The first sum in (2.5) is just the number q of real roots of p, counting multiplicities. If d denotes the number of pairs of non-real roots of p, then $n = q + 2d$, and (2.5) is equivalent to the following

Theorem 2.15 *The number of Fourier critical points of p, counting critical multiplicities, equals the number of pairs of nonreal zeros of p, counting multiplicities.*

Pólya has generalized this theorem for analytic functions of order less than $\frac{2}{3}$ with finitely many nonreal zeros, instead of polynomials [24]. He also conjectured that this can be extended to functions of order at most 1, which is now proved and extended to functions of order less than 2 [3]. But these generalizations are not of relevance to our further discussion.

2.3 Gauss' Conjecture

Fourier critical points were defined first in [6], a work which remained incomplete, due to Fourier's death. In his announcement of this work to the scholars in Göttingen, Gauss mentioned an ambiguity he had found. He said that from Fourier's work it followed that a polynomial "has as many pairs of imaginary roots as critical points" [7]. However, he pointed out that Fourier did not mean this as "each [critical point] belonged to a *specific* pair of imaginary roots" [7](or at least that he did not prove it). This question leads to the

Conjecture 2.16 *There exists a definite, natural, one-to-one relationship between the Fourier critical points and the pairs of conjugate nonreal zeros of a polynomial.*

This conjecture is very vague. One may say, too vague to be taken seriously. In their papers referenced here, neither Gauss nor Fourier suggested an algorithm that would define such a bijective relationship. More than anything else, Gauss was doubtful if such a relationship existed at all, and suggested the same about Fourier [10]. In general, the conjecture is indeed false. Any polynomial that has critical zeros of critical multiplicity greater than 1 provides a counterexample, because a multiple critical zero would have to be assigned to more than one pair of nonreal zeros. Also, Gauss failed to consider critical points of higher orders. One can only guess that with this too simplified view, the problems he encountered in examples he was checking led him to the conclusion that such a relationship in general does not exist. He mentioned his view in a letter [10] to his friend Schumacher. Three years later, he spent some more time on finding a relationship, and wrote to Schumacher that he found it "quite probable to find a connection" after all [11]. Then again, just three days later, he changed his opinion and suggested that "there is no common, natural, non-arbitrary relationship ... at all". He mentioned that in order to prove or disprove this, he would have to examine and visualize a large number of special cases, a task which he doubted he could accomplish [12]. (See also the remarks in Chapter 1.)

Apparently, Gauss gave up his investigations at this point. He didn't leave a summary of his investigations behind, so one can only hypothesize what kind of relationship he sought. Other authors suggested he meant some geometric relationship, and came up with partial results [3, 20, 24]). Most of these results restrict the possible locations of the Fourier critical points, depending on the location and density of the zeros. Some of them will be presented in the last section of this chapter. But none of them are general, in that they do not resolve the conjecture completely.

Instead of this conjecture, we will now state a more precise but general theorem. Let $Z(p)$ be the set of pairs of conjugate non-real zeros, and $F(p)$ the set of Fourier critical zeros of p of any order. We distinguish between critical zeros of different order at the same value.

Theorem 2.17

1. *There exists an algorithm which, given any value $c \in (0, \pi)$ and any polynomial $p \in \mathcal{P}$, determines a mapping $f_{p,c}$ from a partition of $Z(p)$ to subsets of $F(p)$. No class of this partition is mapped to the empty set, and every point in $F(p)$ is assumed at least once.*

2. *Consider a fixed polynomial p. If no nonreal zero or critical zero of p is multiple, then the function $f_{p,c}$ for all but a finite number of values $c \in (0, \pi)$ maps one-element sets to one-element sets, and can be identified with a surjective function $f_{p,c} : Z(p) \to F(p)$.*

3. *If all Fourier critical zeros are of critical multiplicity 1, then $f_{p,c}$ maps a partition of $Z(p)$ to classes of a partition of $F(p)$.*

4. *If the conditions in 2. and 3. hold, then all but a finite number of values $c \in (0, \pi)$ define an $f_{p,c}$ that maps one-element sets 1:1 to one-element sets, and $f_{p,c}$ can be identified with a bijective function $f_{p,c} : Z(p) \to F(p)$.*

The most common case in fact is 4, where we can find a family of mappings all but finitely many of which are bijective. After introducing some tools (Sections 2.4, 2.5, 2.6), we will outline the ideas of the proof by some examples (Section 2.7). The proof itself (Section 2.8) will be constructive, that is, we will describe an algorithm that, given a polynomial p and a value $c \in (0, \pi)$, lets one uniquely determine a mapping $f_{p,c}$. Also, we will be able to decide whether $f_{p,c}$ can be identified with a surjective, or even bijective function $f_{p,c} : Z(p) \to F(p)$. Note however that we don't claim to have an algorithm that determines a bijection, or even just a function $f_{p,c} : Z(p) \to F(p)$ for *every* polynomial $p \in \mathcal{P}$. In fact, for all values $c \in (0, \pi)$, one can find a polynomial (without multiple zeros or critical zeros) for which c provides an exception. This problem will be further discussed in Section 2.9.

2.4 The Logarithmic Derivative

The *logarithmic derivative* of a polynomial is widely used in the literature, as a helpful tool to examine properties of polynomials in the complex plane. It will also play an essential role in our proof.

In this section, p may be an arbitrary complex-valued polynomial of degree at least 1. Unless otherwise stated, we define n as the number of roots of p, *not* counting multiplicities. We define the *logarithmic derivative* of p pointwise as $(\log p)'(z)$, for any complex point z at which p does not vanish; $\log p$ is chosen as an analytic branch of the logarithm in a simply connected region containing z, but no zero of p. Likewise, we define the *second logarithmic derivative* of p as the second derivative of $\log p$ at z.

This definition explains the name "logarithmic derivative", but is too complicated to handle. Keeping in mind that the derivative of any analytic branch of $\log z$ is just $1/z$, and applying the chain and quotient rules of differentiation, we immediately obtain the following two relationships:

Corollary 2.18 *At any z where p does not vanish, we have*

$$(\log p)'(z) = \frac{p'(z)}{p(z)}. \tag{2.6}$$

$$(\log p)''(z) = \frac{p(z)p''(z) - (p'(z))^2}{p^2(z)}. \tag{2.7}$$

Because of its much simpler form, we will henceforth denote the logarithmic derivative as $\frac{p'}{p}$, and the second logarithmic derivative as $\left(\frac{p'}{p}\right)'$. From (2.6) we infer that the zeros of p are poles of $\frac{p'}{p}$; by its definition, $\frac{p'}{p}$ is analytic everywhere else. The zeros of $\frac{p'}{p}$ are exactly the zeros of p', except for the multiple zeros of p. Recalling that a k-fold zero of p is also a zero of p' of multiplicity $k-1$, we find that every pole of $\frac{p'}{p}$ is single. Hence, the degree of the denominator of $\frac{p'}{p}$ is n. From degree considerations we obtain that $\frac{p'}{p}$ has a single zero at ∞, and that $\frac{p'}{p}$ has $n-1$ zeros, counting multiplicities. Note that the zeros of $\frac{p'}{p}$ may have arbitrary multiplicity, as the example $p(z) = z^n - 1$ shows.

For the second logarithmic derivative we find the following analogous results:

- The poles of $\left(\frac{p'}{p}\right)'$ are exactly the zeros of p; their multiplicity is always 2.

- ∞ is a zero of $\left(\frac{p'}{p}\right)'$ of multiplicity 2.
- $\left(\frac{p'}{p}\right)'$ has $2n-2$ finite zeros, counting multiplicities.

The zeros of $\left(\frac{p'}{p}\right)'$ will be further studied in Chapter 3, although they will be of some interest in this chapter.

If we factorize p by its complex zeros, we obtain the following standard but useful result:

Corollary 2.19 *Let $p(z) = a(z-\alpha_1)^{k_1}\dots(z-\alpha_n)^{k_n}$ be a polynomial with roots $\alpha_1,\dots,\alpha_n$ of multiplicity $k_1,\dots,k_n$, respectively. Then we have*

$$\frac{p'}{p}(z) = \sum_{j=1}^{n} \frac{k_j}{z-\alpha_j} \tag{2.8}$$

and

$$\left(\frac{p'}{p}\right)'(z) = -\sum_{j=1}^{n} \frac{k_j}{(z-\alpha_j)^2} \tag{2.9}$$

Proof: Applying the logarithm, we get

$$(\log p)(z) = \log a + \sum_{j=1}^{n} k_j \log(z-\alpha_j)$$

where the logarithms on the right-hand side are suitably chosen. Differentiating once and twice yields the above two relationships, independent of the choice of the branches of the logarithms. □

A nice physical interpretation of (2.8) is the next theorem, apparently due to Gauss [8, 27]:

Theorem 2.20 *Let a field of force be given by n numbered particles $1,\dots,n$ of unit weight located at the (not necessarily distinct) points $\alpha_1,\dots,\alpha_n$. Let each particle j repel an object at a given point z with a force equal to the inverse distance vector $\frac{1}{\overline{z-\alpha_j}}$, and the total force $F(z)$ on the object be the sum of the forces due to each particle. Then the points of equilibrium ($F(z)=0$) are exactly the zeros of $\frac{p'}{p}$.*

Proof: This is just another way of writing (2.8). In order to handle a k-fold zero of p, one can either attach k particles of unit weight at the same point, or, equivalently, assign an integer weight to each zero, corresponding to its multiplicity. □

Gauss stated and proved this theorem in 1816, so when he studied Fourier critical points in 1833, he must have known that the logarithmic derivative was a potentially useful tool. So one may suggest that he tried to find a proof of Conjecture 2.16 that involves the logarithmic derivative.

For polynomials with real coefficients, we know that their zeros are either real, or come in conjugate-complex pairs. By a simple calculation, we show for each such pair $\alpha, \overline{\alpha}$ that:

$$\frac{1}{z-\alpha} + \frac{1}{z-\overline{\alpha}} = \frac{2z-\alpha-\bar{\alpha}}{(z-\alpha)(z-\overline{\alpha})} = \frac{2(z-\Re(\alpha))}{z^2-2z\Re(\alpha)+|\alpha|^2},$$

and hence, (2.8) can be rewritten as

$$\frac{p'}{p}(z) = \sum_{\alpha_j \text{ real}} \frac{k_j}{z-\alpha_j} + 2 \sum_{\substack{\alpha_j = x_j + y_j i \\ y_j > 0}} \frac{k_j(z-x_j)}{(z-x_j)^2+y_j^2} \tag{2.10}$$

This lets one qualify the behaviour of $\frac{p'}{p}$ on the real axis:

Corollary 2.21 *For $p \in \mathcal{P}$, let x_1 be the minimum real part and x_n be the maximum real part of all zeros of p. Then $\frac{p'}{p}$ is negative for $x < x_1$ and positive for $x > x_n$.*

Proof: This is because all terms in (2.10) are negative/positive, respectively. □

Before we return to the study of the logarithmic derivative, we will introduce the next definition for arbitrary rational functions.

2.5 Loci of points of prescribed argument

In this section, let r be a non-constant rational function, and c be a value in the interval $(-\pi, \pi]$[3]. Let further $Z(r)$ and $P(r)$ be the set of zeros and poles of r, respectively. We call a point z *w-point* (or *critical point*) if $r'(z) = 0$, but $r(z) = w \neq 0$. The *multiplicity* of a w-point is defined like the ordinary multiplicity of a critical zero of a polynomial.

[3] We define this as the range of the principal value of $\operatorname{Arg} z$ for $z \in \boldsymbol{C}$, $z \neq 0$.

Definition 2.22 The set

$$L_c(r) = \{z : \operatorname{Arg} r(z) = c\}$$

is called the *locus of points of r of constant argument c.*

We will simply call the sets $L_c(r)$ *loci*, since we will not deal with any other type of locus, and the choice of r and c is clear from the context. Lucas [16] and Walsh [27] contain a thorough description of these loci in the case that r is a polynomial. The properties stated there can be easily generalized:

Proposition 2.23

1. *Two different loci $L_{c_1}(r)$ and $L_{c_2}(r)$ are disjoint, and the union of the loci for all arguments $c \in (-\pi, \pi]$ is just $\boldsymbol{C} \setminus (Z(r) \cup P(r))$.*
2. *If z_0 is a point on $L_c(r)$, and $r'(z_0) \neq 0$, then in a neighbourhood of z_0, $L_c(r)$ consists of an analytic arc through z_0.*
3. *If z_0 is a w-point on $L_c(r)$ of multiplicity k, then in a neighbourhood of z_0, $L_c(r)$ consists of $k+1$ analytic arcs intersecting in z at angles of $\frac{\pi}{k+1}$.*
4. *If z_0 is a zero/pole of r with multiplicity k, then k arcs terminate in z_0 with angles $2\pi/k$ between adjacent arcs. Furthermore, the k arcs each of two loci $L_{c_1}(r)$ and $L_{c_2}(r)$ terminate in z_0 with angles $\frac{\pm(c_2-c_1)}{k}$ between an arc of $L_{c_1}(r)$ and an adjacent arc of $L_{c_2}(r)$. (The negative sign is to be used in case of a pole, the positive sign in case of a zero.)*
5. *No arc of $L_c(r)$ terminates in a point other than a zero, a pole, or the infinite point.*

It follows that every locus can be subdivided into a finite number of analytic Jordan arcs which connect poles, zeros, and possibly the infinite point. These arcs may intersect at w-points of r. We call divide these arcs of $L_c(r)$ into *segments*, such that every segment has two points of the above types as its endpoints, and does not contain any other w-points. We now assume that the degree of the denominator of r exceeds the degree of the numerator by some positive integer k. (Of course, the statements can be generalized to arbitrary rational functions, but that is not needed here.) The following specifies how many segments of $L_c(r)$ are infinite.

6. *If $r(z)$ has a k-fold zero at ∞, then every locus has exactly k infinite segments. These segments asymptotically behave as lines at angles of $2\pi/k$ to each other. The segments of two loci $L_{c_1}(r)$ and $L_{c_2}(r)$ asymptotically behave as lines with angles $\frac{c_1-c_2}{k}$ between a segment of $L_{c_1}(r)$ and an adjacent segment of $L_{c_2}(r)$.*

7. *Along a single segment of any locus, $|r(z)|$ is strictly monotonic.*

This allows us to define a direction on the segments of a locus. At a zero (including the point of infinity), pole, or w-point z_0, we call a segment *incoming* (*outgoing*) if $|r(z)|$ decreases (increases) as z tends to z_0 along the segment. In other words, if one passes a segment in the direction of decreasing $|r(z)|$, this segment is outgoing at its starting point, and incoming at its endpoint.

8. *All segments at a (finite or infinite) zero are incoming. All segments at a pole are outgoing.*

9. *The $2k+2$ segments at a w-point are alternately incoming and outgoing, i.e. two segments at an odd multiple of the angle $\frac{\pi}{k+1}$ have different directions. In particular, the number of incoming segments equals the number of outgoing segments.*

10. *No segment can both begin and end in either a zero or a pole, or in the same w-point.*

Note that despite 10., $L_c(r)$ may contain closed arcs consisting of two or more segments. Examples will be given in Chapter 5.

Proof: Remember that $L_c(r)$ is the inverse image (with respect to r) of a ray γ of points w with $\operatorname{Arg} w = c$. We will prove the Parts 1. to 10. in a slightly different order.

1. This follows immediately from the fact that at every point z except at zeros and poles, $r(z)$ has a well-defined argument.

2. As r has no poles in a neighbourhood of z_0, it is analytic around z_0. Since also $r'(z_0) \neq 0$, r is locally conformal and has a locally defined (also analytic) inverse which is likewise conformal (see [1], p. 132f.). So we find that r^{-1} maps a small piece of the ray γ through $r(z_0)$ into a single analytic arc of $L_c(r)$ through z_0.

4. To handle zeros and poles consistently, we use negative values to denote the multiplicity of poles. Since translations in z don't affect the angles between loci, we can assume that $z_0 = 0$.

   Let us first assume 0 is a single zero of r (i.e. $k = 1$). Then $r'(0) \neq 0$, and, as in 2., r has a locally defined, conformal inverse r^{-1} with $r^{-1}(0) = 0$. Let as above γ, γ_1 and γ_2 be the rays of points with arguments c, c_1, and c_2, respectively. Then r^{-1} maps a small closed segment of $\gamma \cup \{0\}$ with the zero point as one of its endpoints into a single analytic arc of $L_c(r)$, likewise with 0 as one endpoint. (Which gives the trivial angle of 2π between two segments of $L_c(r)$.) Furthermore, the angle $c_2 - c_1$ between γ_1 and γ_2 is preserved under r^{-1}, so $L_{c_1}(r)$ and $L_{c_2}(r)$ also have an angle $c_2 - c_1$ at 0.

   Now let 0 be a k-fold zero (or a $(-k)$-fold pole). Then r must be of the form

$$w = r(z) = z^k q(z) \tag{2.11}$$

   where q is a rational function that does not have the factor z either in its numerator or denominator. This implies that q is analytic, and $q(z) \neq 0$, in a neighbourhood $N(0)$. Then (2.11) can be rewritten as

$$v = w^{1/k} = zq^{1/k}(z). \tag{2.12}$$

   where any of the k possible roots can be chosen for $w^{1/k}$. In either case, the corresponding choice of the function $q^{1/k}(z)$ is analytic and nonzero in $N(0)$. Thus, $s(z) = zq^{1/k}(z)$ is also analytic, it has a single zero at 0, and hence, $s'(0) \neq 0$. So we can apply the first part of the proof to s, which shows that r^{-1} exists and can be written as the composition of the k-valued function $v = w^{1/k}$ and the locally conformal mapping $z = s^{-1}(v)$. Clearly, $v = w^{1/k}$ maps γ into $|k|$ rays of argument $\operatorname{Arg} v = \frac{c+2j\pi}{k}$, $j = 0, \ldots, k-1$, and shrinks the angle between γ_1 and γ_2 to $\frac{c_2-c_1}{k}$. The second mapping s^{-1} preserves the angles between the segments, which proves this part of the theorem.

5. Assume that an arc of $L_c(r)$ terminates in a finite point z_0 which is neither a pole nor a zero of r. Then there exists a neighbourhood $N(z_0)$ in which r is analytic, and $|r(z)| > \epsilon > 0$. Then $\operatorname{Arg}(r(z))$ is also analytic in $N(z_0)$. Pick a sequence of points (z_j) on $L_c(r) \cap N(z_0)$ that converges to z_0. Then $\operatorname{Arg}(r(z_j))$ must converge to $\operatorname{Arg}(r(z_0))$. But $\operatorname{Arg}(r(z_j)) = c$ for all j, so we must also have $\operatorname{Arg}(r(z_0)) = c$, and z_0 lies itself on $L_c(r)$. Now Parts 2. and 3. show that z_0 cannot be an endpoint of $L_c(r)$, which contradicts the assumption.

6. Consider $\hat{r}(z) = r(\frac{1}{z})$ in an open neighbourhood around 0. $\hat{r}$ is again a rational function and has a k-fold zero at 0. By the first part of 4., $L_c(\hat{r})$ consists of k analytic segments terminating in 0 at angles $2\pi/k$ of each other. The mapping $z \longrightarrow 1/z$ maps these segments conformally to infinite segments that behave asymptotically as straight lines. Angles are preserved. but change their orientation. This shows the first part of 6. Likewise, we obtain the second part. using the second part of 4.

7. Suppose $|r(z)|$ takes a local minimum (or maximum) on a segment λ of $L_c(r)$, say, at z_0. Then $\hat{r}(z) = r(z) - r(z_0)$ has a zero at z_0 at which two segments of $L_c(\hat{r})$ (or $L_{c\pm\pi}(\hat{r})$) terminate. at an angle of π. By 4., z_0 must be at least a double zero of $\hat{r}$, which implies $\hat{r}'(z_0) = r'(z_0) = 0$. So z_0 must be a w-point of r. This means, according to our definition of segments. that z_0 can only be an endpoint of λ. It follows that λ is strictly monotonic.

8. This follows simply from the fact that $|r(z)| \longrightarrow \infty$ as z approaches a pole, and $|r(z)| \longrightarrow 0$ as z approaches a zero.

3. and 9. Consider the function $\hat{r}(z) = r(z) - r(z_0)$ which has a $(k+1)$-fold zero at z_0; In some neighbourhood around z_0, we have $\operatorname{Arg}(r(z)) = c$ whenever $\operatorname{Arg}(\hat{r}(z)) = c$ or $\operatorname{Arg}(\hat{r}(z)) = c \pm \pi$. The first condition implies $|r(z)| > |r(z_0)|$, so $|r(z)|$ is increasing in outgoing direction (which means the segment is incoming). Likewise, the latter condition indicates that $|r(z)|$ is decreasing, so we have an outgoing segment. By Part 4., we know that two segments of the same direction have angles of $\frac{2\pi}{k+1}$ of each other, and two adjacent segments of opposite direction have angles of $\frac{\pi}{k+1}$ of each other. This shows both Parts 3. and 9.

10. This follows immediately from the monotonicity condition 7.

□

These statements prove that we can, beginning at a pole of r, follow the arcs of $L_c(r)$ in descending direction. If we reach a w-point of r, Part 9. assures that we can always find an outgoing segment on which to resume our journey. By Part 10. and by the monotonicity condition, we can never return to a previous point, and also, there is only a finite number of segments available. Therefore, we must finally come to a zero point, or approach the infinite point, where the tour ends. If we never come across a w-point of r, then one single segment of $L_c(r)$ gets us directly from the pole to a zero of r. We will see that this is the "normal" case.

We summarize more formally:

Corollary 2.24

1. *Every pole of* r *is connected to at least one zero of* r *(possibly the infinite point) through one or more segments of* $L_c(r)$, *and vice versa.*
2. *If* $L_c(r)$ *contains no* w*-point of* r, *then every segment of* $L_c(r)$ *connects exactly one pole to exactly one zero of* r *(possibly the infinite point).*

If in addition to 2., every zero, the infinite point, and every pole of r are single, then the segments define a bijective mapping between the poles on the one side, and the zeros and the infinite point on the other side. This makes Corollary 2.24 an essential part of the proof of Theorem 2.17.

2.6 Some Basic Notations

For the proof of Theorem 2.17, we need an extension of the terms *critical zero* and *critical point* to arbitrary complex points. We will later see that the generic definition suits our needs:

Definition 2.25 A point $z \in \mathbb{C}$ is called a *critical zero* of p of *order* l and *multiplicity* k, if $p^{(l-1)}(z) \neq 0$, $p^{(l)}(z) = p^{(l+1)}(z) = \ldots = p^{(l+k-1)}(z) = 0$, $p^{(l+k)}(z) \neq 0$ for some $k \in \mathbb{N}$.

We call z a *critical point* of p if it is a critical zero of p of some order. Its total *multiplicity* is defined as the sum of the multiplicities of all critical zeros at z.

In the nonreal case, the terms *critical zero* and *multiplicity* are unambiguous. There is no need to define an equivalent to Fourier critical zeros for nonreal points. For real critical zeros however, we recall that we must carefully distinguish between *ordinary* and *Fourier* critical zeros, and between *multiplicity* and *critical multiplicity.* In general, these values are different, as noted after Definition 2.8.

For simplicity, we will refer either to critical zeros of fixed order 1, or critical zeros of arbitrary order, as *critical zeros*, whenever the meaning is clear from the context.

Definition 2.26 We define

H_+ : the open half-plane above the real axis.

H_- : the open half-plane below the real axis.

We already introduced the identifiers $Z(p)$ and $F(p)$. Some more identifiers of this type will be introduced now for later use in the proof.

Definition 2.27 For a polynomial $p \in \mathcal{P}$ of degree $n \in I\!N$, and $k = 1, \ldots, n-1$, we define the following sets:

$Z(p)$:	the set of pairs of nonreal zeros of p.
$Z(p^{(k)})$:	the set of pairs of nonreal zeros of $p^{(k)}$,
$Z_k(p)$:	the set of nonreal critical zeros of order k of p,
$F_k(p)$:	the set of Fourier critical zeros of order k,
$C_k(p)$:	the union of $Z_k(p)$ and $F_k(p)$,
$Z_+(p)$:	the union of all $Z_k(p)$, $\quad k = 1, \ldots, n-1$,
$Z_*(p)$:	the union of $Z(p)$ and $Z_+(p)$,
$F(p)$:	the union of all $F_k(p)$, $\quad k = 1, \ldots, n-1$,
$C(p)$:	the union of $Z_+(p)$ and $F(p)$.

A few comments about these sets and about the fine but important distinctions between them are in order:

- The elements of these sets are either single real points, or pairs of conjugate-complex nonreal points. We will consistently refer to the elements as points. Whenever a statement concerns a pair of points, it is meant to apply to either point[4].

- In some aspects, the nonreal zeros of p behave just like a special kind of critical zeros of order 0; in others, they don't. Therefore, we have to distinguish between the two notations: Z_+ for all nonreal critical zeros, and Z_* for all nonreal critical zeros *and* zeros.

[4] However, if several instances of point pairs are involved in a statement, we refer consistently to either the representatives in H_+ or those in H_-.

- In general, $Z_k(p)$ and $Z(p^{(k)})$ are *not* identical. $Z_k(p)$ contains only those points in $Z(p^{(k)})$ for which $p^{(k-1)}(z) \neq 0$; whereas $Z(p^{(k)})$ contains all critical zeros of order k, and perhaps multiple zeros of derivatives of lower order.

  If all zeros and critical zeros of p are single, then $Z_k(p)$ and $Z(p^{(k)})$ are in fact equal for all k.

2.7 Examples

Prior to the (rather complicated) proof, we will motivate its steps by a few examples. These examples also have the virtue of clarifying the notation just introduced. Throughout this section, we let $c = \pi/2$, unless otherwise stated.

Example 2.28 $p(z) = (z^2 - 10z + 26)(z^2 + 10z + 26)(z - 1)(z + 2)$

The zeros of p are $5 \pm i$, $-5 \pm i$, 1 and -2; its critical zeros are approximately -4.5130, -3.8089, -0.4309, 3.2741, and 4.6454. It can be easily verified that the first and fifth of these are Fourier critical points. Hence we have

$$\begin{aligned} Z(p) &= \{-5 \pm i, 5 \pm i\} \\ F(p) &= F_1(p) = \{-4.5130, 4.6454\} \end{aligned}$$

We now consider the locus $L_c\left(\frac{p'}{p}\right)$ for $c = \pi/2$, this is, the locus of all points at which $\Re\left(\frac{p'}{p}(z)\right) = 0$, and $\Im\left(\frac{p'}{p}(z)\right) > 0$. The trace of the locus is outlined in Figure 2.1, by the small '+'-shaped points. The boxes and circles are the zeros and critical zeros of p, respectively[5]. We make the following observations:

- The arcs originate at the zeros and end at the critical zeros, or at ∞.
- Exactly one arc ends at ∞; it originates at one of the real zeros and converges to the lower half of the imaginary axis. It is contained entirely in H_-.

[5] We will keep this convention in all upcoming plots.

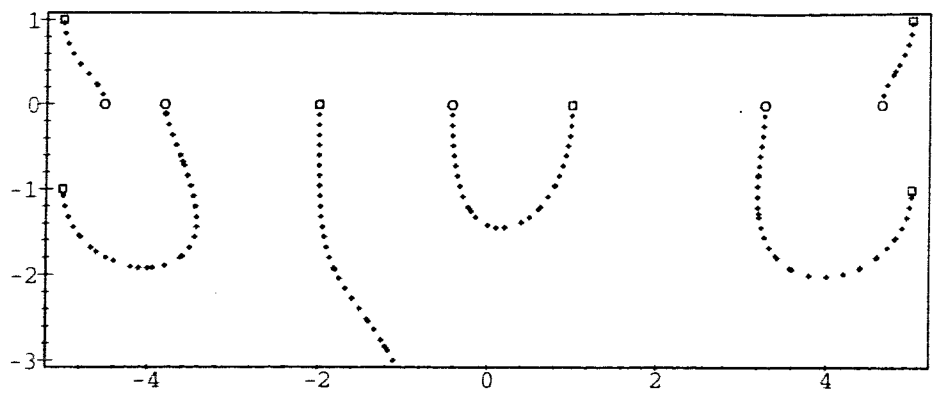

Figure 2.1: *The locus of* $p(z) = (z^2 - 10z + 26)(z^2 + 10z + 26)(z - 1)(z + 2)$

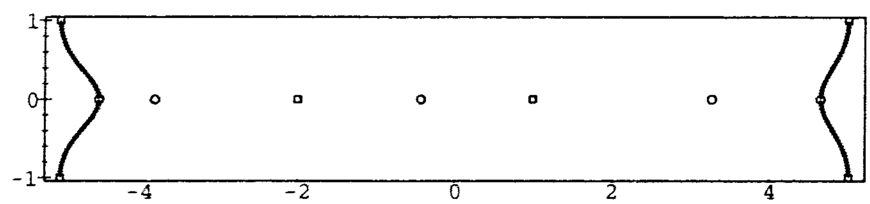

Figure 2.2: *The symmetric locus of* $p(z) = (z^2 - 10z + 26)(z^2 + 10z + 26)(z - 1)(z + 2)$

- The arcs which end at the two Fourier critical points originate in the two nonreal zeros in the upper half plane; the arcs themselves are contained in H_+.

- The other arcs connect the three remaining nonreal or real zeros to the three non-Fourier critical points. They are contained entirely in H_-.

We see that the arcs in the upper half plane behave "nicely" for our purposes, in that they connect nonreal zeros to Fourier critical zeros; whereas the arcs in the lower half plane are "superfluous". In order to get a nicer, and symmetric, picture, we replace $L_c\left(\frac{p'}{p}\right)$ in H_- by $L_{-c}\left(\frac{p'}{p}\right)$. Thus we obtain Figure 2.2.

Now the arcs are symmetric to the real axis; each nonreal zero is connected to some Fourier critical zero, and exactly one pair of conjugate-complex zeros is connected to each Fourier critical zero. Therefore, the arcs define a bijection from $Z(p)$ to $F(p)$, verifying Theorem 2.17 for this polynomial.

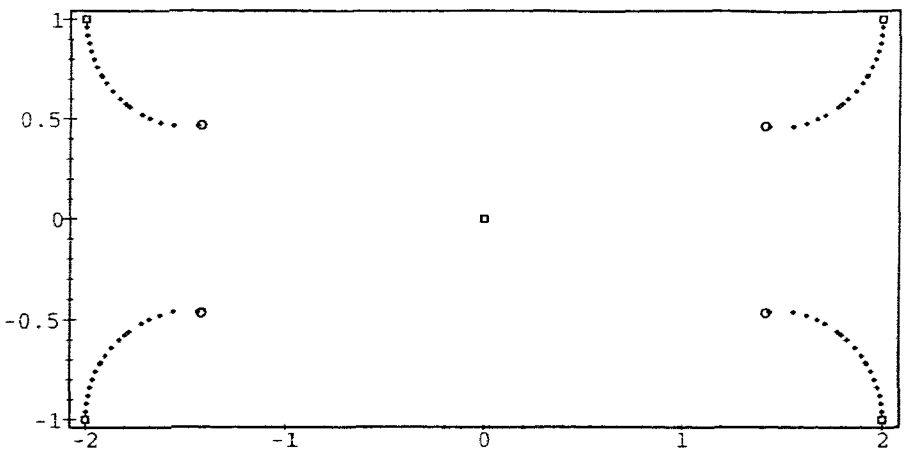

Figure 2.3: *The symmetric locus of* $p(z) = z(z^2 - 4z + 5)(z^2 + 4z + 5)$

We see the virtue of taking $L_{-c}\left(\frac{p'}{p}\right)$ in H_- in order to get a symmetric image on the entire complex plane. This is obvious and holds for arbitrary values of c, because $\frac{p'}{p}(\bar{z}) = \overline{\frac{p'}{p}(z)}$ for all $z \in \mathbb{C}$. Hence we introduce the following simplifying definition:

Definition 2.29 For a polynomial $p \in \mathcal{P}$, and a value $c \in (-\pi, \pi)$, we define the *symmetric locus* of argument c as:

$$S_c(p) = \left(L_c\left(\frac{p'}{p}\right) \cap H_+\right) \cup \left(L_{-c}\left(\frac{p'}{p}\right) \cap H_-\right).$$

Example 2.30 $\quad p(z) = z(z^2 - 4z + 5)(z^2 + 4z + 5)$

Here we have

$$\begin{aligned} Z(p) &= \{-5 \pm i, 5 \pm i\} \\ Z_1(p) &= \left\{\sqrt{\frac{1}{2}\sqrt{5} + \frac{9}{10}} \pm \sqrt{\frac{1}{2}\sqrt{5} - \frac{9}{10}}i, -\sqrt{\frac{1}{2}\sqrt{5} + \frac{9}{10}} \pm \sqrt{\frac{1}{2}\sqrt{5} - \frac{9}{10}}i\right\} \\ &\cong \{-1.4206 \pm 0.4669i, 1.4206 \pm 0.4669i\} \\ F(p) &= F_2(p) = \{-\frac{3}{5}\sqrt{5}, \frac{3}{5}\sqrt{5}\} \cong \{-1.3412, 1.3412\} \end{aligned}$$

In this example, all Fourier critical zeros are of order 2. So we cannot expect to connect zeros to Fourier critical zeros just through arcs of $S_c(p)$. Figure 2.3 shows that the arcs of

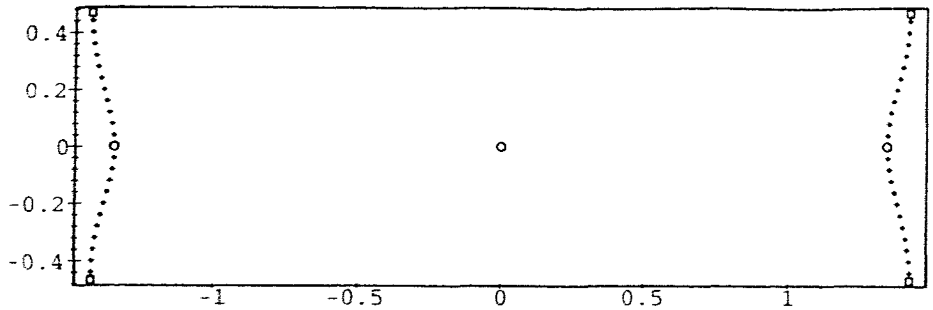

Figure 2.4: *The symmetric locus of* $p'(z) = 5z^4 - 18z^2 + 25$

$S_c(p)$ rather connect the zeros 1:1 to the nonreal critical zeros. (In every such case, we will say that two points are ***directly connected.***) This defines a bijection from $Z(p)$ to $Z_1(p)$.

To continue towards the Fourier critical zeros, we observe the equalities $F_1(p') = F_2(p)$ and $Z(p') = Z_1(p)$, and apply the same procedure to p'. Namely, we determine $S_c(p')$, as in Figure 2.4. We see that every nonreal critical zero is directly connected to a Fourier critical point, and that the arcs define a bijection from $Z_1(p)$ to $F(p)$. We now simply concatenate these two mappings in order to get the desired bijection from $Z(p)$ to $F(p)$.

In more general cases, we will have Fourier critical zeros of higher, and perhaps several different orders; in that case, the notion of concatenation is more difficult to handle. Therefore, we introduce a different notation: We call two zero points (e.g. a zero and a Fourier critical zero) *connected* if they are linked by a sequence of directly connected "intermediary" critical zeros. We will use the notation $z \mapsto \hat{z}$ for this. In this example we found, for instance, $5 \pm i \mapsto 1.3412$: these two points are connected via the point pair $1.4206 \pm 0.4669i$. We get the desired bijection by mapping the elements of $Z(p)$ to the respective Fourier critical zeros to which they are connected.

Example 2.31 $p(z) = z^4 + 6z^2 + 25$

This example shows that the loci are not always as simple as in the previous two examples. We have

$$\begin{aligned} Z(p) &= \{z_1, z_2\} = \{-1 \pm 2i, 1 \pm 2i\} \\ Z_1(p) &= \{c\} = \{\pm\sqrt{3}i\} \end{aligned}$$

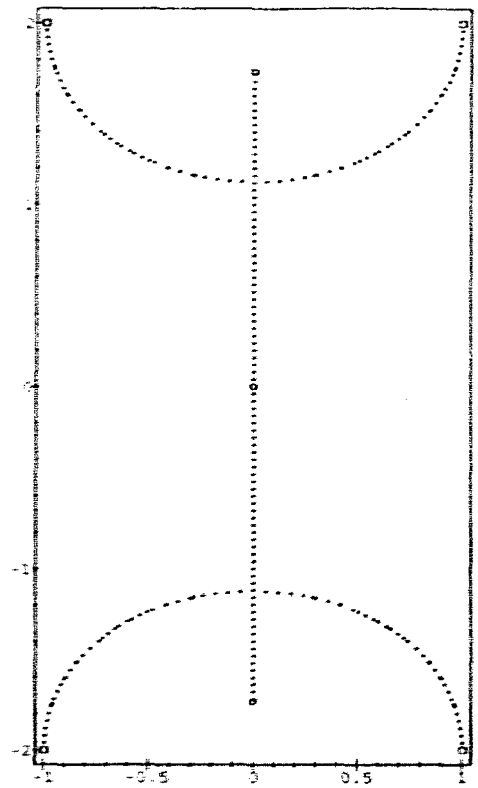

Figure 2.5: *The symmetric locus of* $p(z) = z^4 + 6z^2 + 25$

$$\begin{aligned} F_1(p) &= \{0\} \\ Z_2(p) &= \{\pm i\} \\ F_3(p) &= \{0\} \end{aligned}$$

The locus $S_c(p)$ is graphed in Figure 2.5. Observe that all points in $C_1(p) = Z_1(p) \cup F_1(p)$ are connected to points in $Z(p)$. However, we see that four arcs of $S_c(p)$ intersect each in $z_{cr}, \overline{z_{cr}} \cong \pm 1.1251i$, which are w-points of $\frac{p'}{p}$. The two arcs in imaginary direction are outgoing and connect to the critical zeros, while the horizontal arcs are incoming, originating from the zeros. It is impossible to assign either of them in a non-arbitrary way to either of the critical zeros. One would rather say that both zeros are directly connected to both critical zeros, and also to each other. (We will use the notation $z_1 \sim z_2$ for this.)

We can resolve this conflict by grouping all those interconnected zeros into *classes*. Likewise, we group all those critical zeros which are directly connected to the zeros of a certain class into another class, which we assign to the former. This mapping from a partition of $Z(p)$ to a partition of $Z_1(p)$ is again bijective.

In the general case, this situation can occur in any, possibly even in several derivatives of p. Therefore, we need to "backtrace" connected zeros of, say $p^{(k)}$, $z_k \sim \hat{z}_k$ to their respective "ancestors" $z, \hat{z} \in Z(p)$. (We already introduced the more formal notation $z \mapsto z_k$ and $\hat{z} \mapsto \hat{z}_k$ for this.) Then we will also call z and $\hat{z}$ *connected*. We write this as $z \simeq \hat{z}$.

In this example, $c = \pi/2$ is the only value for which this problem exists. For any other value of c in $(0, \pi)$, the arcs of $S_c(p)$ are separated and thus give rise to a well-defined bijection. It can be shown, however, that on choosing $c < \pi/2$, one would obtain a different bijection than for a value $c > \pi/2$.

Example 2.32 $\quad p(z) = 3z^5 + 10z^3 + 15z$

This will be the most complicated example, and it will reveal several problems. Note that

$$p'(z) = 15(z^2+1)^2$$

is essentially the polynomial discussed in Example 2.10, and we showed that 0 is assumed twice as a Fourier critical zero. As remarked, we distinguish between several critical zeros at the same point.

$$\begin{aligned}
Z(p) &= \{z_1, z_2\} = \left\{\sqrt{\frac{1}{2}\sqrt{5}-\frac{5}{6}} \pm \sqrt{\frac{1}{2}\sqrt{5}+\frac{5}{6}}i, -\sqrt{\frac{1}{2}\sqrt{5}-\frac{5}{6}} \pm \sqrt{\frac{1}{2}\sqrt{5}+\frac{5}{6}}i\right\} \\
&\cong \{-0.5335 \pm 1.3969i, 0.5335 \pm 1.3969i\} \\
Z_1(p) &= Z(p') = \{\pm i\} \\
Z_2(p) &= \emptyset \\
Z(p'') &= \{\pm i\} \\
F_2(p) &= \{0\} \\
Z_3(p) &= \{\pm \frac{1}{3}\sqrt{3}i\} \\
F_1(p) &= \{0\}
\end{aligned}$$

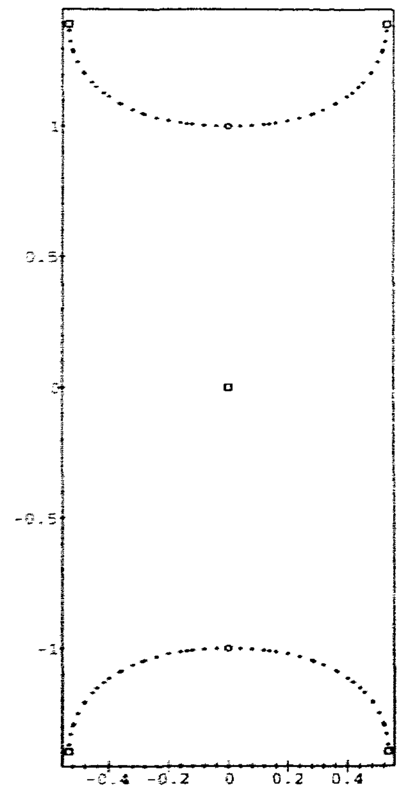

Figure 2.6: *The symmetric locus of* $p(z) = 3z^5 + 10z^3 + 15z$

The points $\pm i$ are critical zeros of multiplicity 2. Corresponding to this, the graph of $S_c(p)$ (see Figure 2.6) shows two pairs of arcs incoming at $\pm i$ from the two pairs of nonreal zeros. Otherwise, these arcs are separated, so they define a mapping from $Z(p)$ to $Z_1(p)$; but this mapping is not injective.

Let us now take a look at the first derivative. $\pm i$, though it is a double zero of p', is just a single pole of $\frac{p'}{p}$. So $S_c(p')$ consists of only one pair of arcs going from $\pm i$ to 0. (These arcs are just finite segments of the imaginary axis, so we did not plot $S_c(p')$.) This defines a bijection from $Z(p')$ to $F_2(p)$.

Now $\pm i$ is also a zero of p'', so it is connected through $S_c(p'')$ to the critical zero in Z_3, and further through $S_c(p''')$ to 0, the second Fourier critical zero. This defines a bijection from $Z(p'')$ to $F_4(p)$.

We saw that the mappings in every step are well-defined, and, except in the first step, they are bijective. The problem arises from putting the steps together and finding a mapping from $Z(p)$ to $F(p)$. The points $\pm i$ are the images of two points in $Z(p)$, and in turn, they are mapped to two points in $F(p)$. But there is no non-arbitrary way of composing these two mappings to a bijection.

We resolve this problem in a similar way as in Example 2.31. We group all those zeros of p into a class, which are connected to the same *intermediary* point (which in general is a point in Z_+). As before, we will write $z_1 \simeq z_2$ for this. Further, we map this class to the set of all points in $F(p)$ which are connected to this intermediary point. In this example, we would thus map the class $[z_1, z_2]$ to $\{0_2, 0_4\}$. (The indices on the 0's are to underline that we distinguish between these two Fourier critical zeros.)

In the general case, we will take greater care in defining the relation $\simeq$. It will be defined in a way that makes it an equivalence relation, which matches the special cases outlined in Examples 2.31 and 2.32, even if they occur in multiple instances in a single polynomial.

We conclude with the remark that Fourier critical zeros can themselves be multiple, in which case several different zeros (or, more generally, classes of zeros) in $Z(p)$ may be mapped to them. However, there is no need to group all these zeros into a class, as we did in the above case of a multiple *nonreal* zero. The situation resembles the first step in this example: Multiply connected Fourier critical zeros do give rise to a well-defined, non-arbitrary mapping, except that this mapping is not injective.

2.8 The proof of the conjecture

As in the examples, we dicuss the symmetric loci of a polynomial $p \in \mathcal{P}$.

Lemma 2.33 *Let $c \in (0, \pi)$. Then the segments of $S_c(p)$ have the following properties:*

- *The arcs of $S_c(p)$ connect all nonreal zeros of p, all (nonreal or Fourier) critical zeros of p of order 1, and w-points of $\frac{p'}{p}$ with* $\operatorname{Arg} w = c$.
- *Every zero of p has exactly one outgoing arc.*
- *Every nonreal critical zero has a number of incoming arcs equal to its multiplicity.*

- *Every real critical zero has a number of incoming arcs in H_+ and H_- each equal to its* critical *multiplicity.*
- *No zero of p and no real critical zero of critical multiplicity 0 is connected.*

Proof: We can restrict ourselves to the positive half-plane. The proof for the negative half-plane is analogous, due to symmetry. In H_+, we identify $S_c(p)$ with $L_c\left(\frac{p'}{p}\right)$. Hence, we can apply all results of Section 2.5.

First, we recall that every zero of p is a single pole of $\frac{p'}{p}$, and a (real or nonreal) critical zero of p is a zero of $\frac{p'}{p}$ with the same (ordinary) multiplicity. Therefore, most of the propositions follow directly from the corresponding parts of Proposition 2.23. All we need to prove is the statements about the nonreal and the real critical zeros, and that no arc leaves H_+ or is infinite.

We observe that no arc can cross the real axis (and thus leave or enter H_+) at any point $x \in I\!R$ other than a zero or pole of $\frac{p'}{p}$, because $\operatorname{Arg}\left(\frac{p'}{p}(x)\right)$ is either 0 or π. Secondly, since ∞ is a single zero of $\frac{p'}{p}$, $L_c\left(\frac{p'}{p}\right)$ has exactly one infinite arc. By Part 6. of Proposition 2.23., this arc has asymptotically an angle of $-c$ to the infinite segment of $L_0\left(\frac{p'}{p}\right)$. However, we infer from Corollary 2.21 that $\frac{p'}{p}(z)$ is positive for large positive z, so this segment is just part of the positive real axis. The points on the infinite segment of $L_c\left(\frac{p'}{p}\right)$ must then satisfy $\operatorname{Arg} z \to -c$ as $z \to \infty$. Consequently, this segment must lie in H_-, which leaves all segments in H_+ to be finite.

Now let z_0 be a real zero of p. Then $L_c\left(\frac{p'}{p}\right)$ has exactly one outgoing arc, at an angle of $-c$ to the outgoing arc of $L_0\left(\frac{p'}{p}\right)$. From the Equation (2.10) for the logarithmic derivative on the real axis, we see that $\lim_{z \nearrow z_0} \frac{p'}{p}(z) = -\infty$, and $\lim_{z \searrow z_0} \frac{p'}{p}(z) = +\infty$, so the outgoing arc of $L_0\left(\frac{p'}{p}\right)$ is a line segment on the real axis to the right of z_0. This shows, as with the infinite point, that $\operatorname{Arg}(z - z_0)$ approaches $-c$ as $z \to z_0$ on the outgoing segment of $L_c\left(\frac{p'}{p}\right)$, so this segment cannot lie in H_+.

Finally, let z_0 be a real critical zero of p of order 1 and multiplicity m. From Part 4. of Proposition 2.23, we infer that z_0, being an m-fold zero of $\frac{p'}{p}$, has m incoming segments of $L_c\left(\frac{p'}{p}\right)$, at angles $2\pi/m$ of each other.

- If m is even, then $m/2$ segments lie on either side of the real axis, so z_0 has m incoming segments of $L_c\left(\frac{p'}{p}\right) \cap H_+$.

- If m is odd. we have to distinguish between two cases:
  - If $p(z_0)p^{(m+1)}(z_0)$ (which cannot be zero) is positive, then $\frac{p'}{p}$ is increasing on the real axis around z_0, and thus positive to the right of z_0. By Part 6. of Proposition 2.23, the m incoming segments of $L_c\left(\frac{p'}{p}\right)$ are directed at angles $\frac{c}{m}, \frac{2\pi+c}{m}, \ldots, \frac{2(\frac{m-1}{2})\pi+c}{m}, \ldots, \frac{2(m-1)\pi+c}{m}$ from the positive real direction, so the first $\frac{m+1}{2}$ of these segments lie in H_+.
  - If $p(z_0)p^{(m+1)}(z_0)$ is negative, then $\frac{p'}{p}$ is decreasing on the real axis around z_0, and thus positive to the left of z_0. The incoming segments of $L_c\left(\frac{p'}{p}\right)$ have angles $\frac{c}{m}, \frac{2\pi+c}{m}, \ldots, \frac{2(\frac{m-1}{2})\pi+c}{m}, \ldots, \frac{2(m-1)\pi+c}{m}$ from the negative real direction; so the first $\frac{m+1}{2}$ of them lie in H_-. This leaves $\frac{m-1}{2}$ segments in H_+.

In each of these three cases, the number of incoming segments at z_0 in H_+ is equal to the critical multiplicity of z_0, as in Definition 2.8. This completes the proof.

Note that points of critical multiplicity 0 are included in the last case. □

The next lemma has similarities to Theorem 2.17. In fact, this will provide the first "iteration" of the proof.

Lemma 2.34

1. *There exists an algorithm which, given any value $c \in (0, \pi)$ and any polynomial $p \in \mathcal{P}$, determines a mapping $f^{(0)}_{p,c}$ from a partition of $Z(p)$ to subsets of $C_1(p)$. No class of this partition is mapped to the empty set, and every point in $F(p)$ is assumed at least once.*

2. *For a fixed polynomial p, all but a finite number of values $c \in (0, \pi)$ in fact define an $f^{(0)}_{p,c}$ that maps one-element sets to one-element sets, and $f^{(0)}_{p,c}$ can be identified with a surjective function $f^{(0)}_{p,c} : Z(p) \to C_1(p)$.*

3. *If in addition all points in $C_1(p)$ are of (critical) multiplicity 1, then $f^{(0)}_{p,c}$ maps a partition of $Z(p)$ to a partition of $C_1(p)$; furthermore, all but a finite number of values $c \in (0, \pi)$ define an $f^{(0)}_{p,c}$ that maps one-element sets 1:1 to one-element sets, and $f^{(0)}_{p,c}$ can be identified with a bijective function $Z(p) \to C_1(p)$.*

Proof: As before, we restrict our attention to the closed upper half plane, in order to deal with points instead of conjugate point pairs.

1. We define both the partition of $Z(p)$ and the mapping $f_{p,c}^{(0)}$ by means of $S_c(p)$. The classes of $Z(p)$ are formed by the sets of zeros which are connected with each other through arcs of $S_c(p)$; by Lemma 2.33, all these zeros are nonreal. For such a class C, we define $f_{p,c}^{(0)}(C)$ as the set of all points in $C_1(p)$ which are also connected to these points through arcs of $S_c(p)$. It follows from Corollary 2.24 that $f_{p,c}^{(0)}$ is well-defined, every point in $C_1(p)$ is assumed at least once, and $f_{p,c}^{(0)}$ maps every class to at least one critical zero, which by Lemma 2.33 lies in $C_1(p)$.

2. By Corollary 2.24, two points in $Z(p)$ can only be connected (thus forming a class of more than one element) if the arc between these points contains a w-point of $\frac{p'}{p}$ with $\operatorname{Arg} w = c$. But $\frac{p'}{p}$ has only finitely many w-points. Therefore, all but finitely many loci $S_c(p)$ are free of w-points of $\frac{p'}{p}$. In this case, Corollary 2.24 states that unique points in $Z(p)$ are mapped to unique critical points, which, again by Lemma 2.33, lie in $C_1(p)$. Since $C_1(p)$ is covered completely, $f_{p,c}^{(0)}$, regarded as a mapping from $Z(p)$ to $C_1(p)$, is surjective.

3. Points in $C_1(p)$ of (critical) multiplicity 1 are connected through exactly one arc of $S_c(p)$. Hence, they are contained exactly once in the image of some class C. So if all points in $C_1(p)$ are single, then the images of all classes form themselves classes of a partition of $C_1(p)$. If in addition $S_c(p)$ contains no w-points of $\frac{p'}{p}$, then Corollary 2.24 implies that every point in $C_1(p)$ is connected to exactly one point in $Z(p)$, which shows that $f_{p,c}^{(0)}$ is injective. Together with the surjectivity in Part 2., we have shown that $f_{p,c}^{(0)}$ is bijective.

□

As a corollary of this theorem, we can say about the set mapped to a class C of $Z(p)$, that it must have the same cardinality as C. For classes consisting of only one element, this is trivial. For larger classes, the arcs of $S_c(p)$ connecting the zeros of this class must contain w-points of $\frac{p'}{p}$. We can, starting at the zeros in C, follow the arcs of $S_c(p)$. Whenever some k arcs join at a k-fold w-point of $\frac{p'}{p}$, there also exist k outgoing arcs on which the tour can be resumed. So throughout the journey the number of arcs remains unchanged, until we finally get to the critical zeros of p. Note however that several of these arcs could terminate at the same critical zero[6], so we have to count them with the number of incoming arcs from zeros of C.

We summarize this as follows:

Corollary 2.35 *For any class C of $Z(p)$, the sets $f_{p,c}^{(0)}(C)$ and C have the same cardinality, if multiply connected critical zeros are counted with the number of arcs through which they are connected to zeros in C.*

It is easy to show that the loci $S_c(p)$ are continuous in c, i.e. slight variations of c result only in slight displacements of $S_c(p)$. Given this, we obtain:

Corollary 2.36 *If $S_c(p)$ contains no w-point of $\frac{p'}{p}$, then slight variations of c do not affect the mapping $f_{p,c}^{(0)}$ constructed from $S_c(p)$. More precisely: There exists an open interval containing c, so that for all values c' in this interval, $f_{p,c'}^{(0)} = f_{p,c}^{(0)}$.*

Proof: The segments of $S_c(p)$ can be separated from all zeros, poles and w-points of $\frac{p'}{p}$ other than the one zero and one pole they connect. If c is slightly varied, they will still be separated from the other zeros. Consequently, they must still connect the same zero to the same pole, thus leaving $f_{p,c}^{(0)}$ unchanged. □

Using this corollary and a compactness argument on any closed subinterval of $(0, \pi)$, we see that the cases where $S_c(p)$ contains a w-point of $\frac{p'}{p}$ mark "limiting cases", in that the mapping $f_{p,c}^{(0)}$ from points of $Z(p)$ to points of $C_1(p)$ changes only at these values:

Corollary 2.37 *Let $c_1, c_2 \in (0, \pi)$, $c_1 < c_2$. If no w-point of $\frac{p'}{p}$ satisfies $\operatorname{Arg} w \in [c_1, c_2]$, then $f_{p,c_1}^{(0)} = f_{p,c_2}^{(0)}$.*

We will now prove Theorem 2.17. If p has only real zeros, then we have nothing to show[7]. Otherwise, let n be the smallest derivative such that $p^{(n)}$ has only real zeros. (Such an n always exists, and it is at most one less than the degree of p.) As before, we restrict ourselves to the closed upper half plane, and do the same considerations in the lower half

[6] Although no example of this has been found so far.
[7] To be exact, $f_{p,c}$ would be the empty mapping.

plane with $-c$ instead of c, thereby preserving symmetry. In defining the required partition on $Z(p)$, we have to take special care of multiple critical zeros. For this reason, we replace the intuitive meaning of "connectedness" with the following definitions:

Definition 2.38 Let $Z(p^{(k)})|_c$ and $f_{p,c}^{(k)}$ be the partitions and functions defined by applying Lemma 2.34 to $p^{(k)}$ (which is, by means of $S_c(p^{(k)})$). For two nonreal zeros or critical zeros $z_0, \hat{z}_0$, we write $z_0 \sim \hat{z}_0$ if they belong to the same class in $Z(p^{(k)})|_c$, for some $k = 0, \ldots, n-1$.

Definition 2.39 Let z_0 be a point in $Z_l(p)$ with multiplicity k, and z_{cr} a point in $C_{l+m}(p)$, where $1 \leq m \leq k$. We call z_0 and z_{cr} *directly connected*, if z_0 belongs to a class C in $Z(p^{l+m-1})|_c$ such that $z_{cr} \in f_{p,c}^{(l+m-1)}(C)$ holds.

By the construction of $f_{p,c}^{(l+m-1)}(C)$ in Lemma 2.34, z_0 and z_{cr} are directly connected if and only if $S_c(p^{(l+m-1)})$ has an arc that connects z_0 (as a zero of $p^{(l+m-1)}$) and z_{cr} (as a critical zero thereof). Note that the order of derivative, $l+m-1$, is constant for all z_{cr} of the same order, and independent of the order of the point z_0 they are connected to. However, the multiplicity of z_0 must be large enough to cover the difference between the order of z_0 and that of z_{cr}. In the case $m = 1$ where z_0 is simple, z_0 can only be directly connected to critical zeros of order $l+1$; whereas an m-fold zero is directly connected to at least one critical zero each of order 1 through m (This follows from Lemma 2.33).

Definition 2.40 Let z_0 be a nonreal zero of p, and z_{cr} a critical zero of order k. We call z_0 and z_{cr} *connected* (and write $z_0 \mapsto z_{cr}$) if there exists a sequence $z_1, \ldots, z_l = z_{cr}$ of critical zeros of order $k_1, \ldots, k_l = k$, respectively, such that for every $j = 1, \ldots, l$, z_{j-1} and z_j are directly connected. (We define $k_0 = 0$). For technical reasons, we also define $z_0 \mapsto z_0$ for every $z_0 \in Z(p)$.

It is clear that $0 < k_1 < \ldots < k_l$, and that $z_1, \ldots, z_{l-1}$ all have to be nonreal. If all zeros $z_0, \ldots, z_{l-1}$ are of multiplicity 1, then we just have $k_j = j$, and $l = k$. The next lemma is a kind of generalization of Corollary 2.24:

Lemma 2.41

1. *Every point $z_0 \in Z(p)$ is connected to at least one point $z_{cr} \in F(p)$.*

2. *Every point $z_{cr} \in C(p)$ is connected to at least one point $z_0 \in Z(p)$.*

3. *If all points in $Z(p)$ and $Z_+(p)$ are of multiplicity 1, and no locus $S_c(p^{(k)})$ contains a w-point of $\frac{p^{(k+1)}}{p^{(k)}}$, then every point $z_0 \in Z(p)$ is connected to exactly one point $z_{cr} \in F(p)$, and every point $z_{cr} \in Z_+(p)$ is connected to exactly one point $z_0 \in Z(p)$.*

Proof:

1. As stated, one can find at least one critical zero z_1 of some order k_1 that is directly connected to z_0. If $z_1 \in F(p)$, we are done. Otherwise, z_1 is a nonreal zero of $p^{(k_1)}$, and we can find another critical zero z_2 of order $k_2 > k_1$ that is directly connected to z_1. Continuing this, we obtain a sequence of critical zeros of strictly increasing order, bounded by n. Hence, after finitely many steps we must reach a real critical zero of p, which is thus connected to z_0.

2. We prove this by induction on k. For $k = 1$, the proof simply follows from Part 1. of Lemma 2.34. Assume that the statement holds for every point in $C_1(p), \ldots, C_k(p)$, and let $z_{cr} \in C_{k+1}(p)$. By Lemma 2.34, z_{cr} must be linked to a point $z_1 \in Z(p^{(k)})$ through an arc of $S_c(p^{(k)})$. If z_1 is a zero of p (of order at least $k+1$), then z_{cr} is directly connected to z_1, and we are done. Otherwise, we must have $z_1 \in Z_{k_1}(p)$, where $1 \leq k_1 \leq k$, and z_1 has multiplicity at least $k - k_1 + 1$. z_1 (as a critical zero of order k_1), and z_{cr} are thus directly connected. By the induction hypothesis, there exists a point z_0 with $z_0 \mapsto z_1$. This implies $z_0 \mapsto z_{cr}$.

   The proof follows by induction on k.

3. Let $z_{cr}, \hat{z}_{cr} \in F(p)$ be two Fourier critical zeros satisfying $z_0 \mapsto z_{cr}$ and $z_0 \mapsto \hat{z}_{cr}$, and $z_1, \ldots, z_l = z_{cr}$ and $\hat{z}_1, \ldots, \hat{z}_m = \hat{z}_{cr}$ the corresponding sequences of critical zeros in Definition 2.40. Without loss, we may assume $l \leq m$. We already observed that in the absence of multiple zeros, the zeros z_j and $\hat{z}_j$ must be of order j. Now suppose that $z_j = \hat{z}_j$ (which is at least true for $j = 0$). Then we infer from Part 2. of Lemma 2.34 that $f_{p,c}^{(j)}$ maps z_j to a ***unique*** point in $C_{j+1}(p)$. Hence, z_{j+1} and $\hat{z}_{j+1}$ must also be equal, and by induction on j we get $z_{cr} = z_l = \hat{z}_l$. Since, as stipulated, z_{cr} is in $F(p)$ and hence real, Lemma 2.33 prohibits z_{cr} to be directly connected to any critical zero of order greater than l. So we must have $l = m$, and z_{cr} and $\hat{z}_{cr}$ must be equal. The

second part of 3. follows from a similar induction proof, keeping in mind that Part 3. of Lemma 2.34 guarantees uniqueness at every step.

□

Definition 2.42 We call two points $z_0, \hat{z}_0 \in Z(p)$ *connected* (written as $z_0 \simeq \hat{z}_0$), if one of the following conditions is satisfied:

1. There exist two points $z_{cr}, \hat{z}_{cr} \in Z_*(p)$ such that $z_0 \mapsto z_{cr}$, $\hat{z}_0 \mapsto \hat{z}_{cr}$, and $z_{cr} \sim \hat{z}_{cr}$.
2. There exists a $z_0' \in Z(p)$ such that $z_0 \simeq z_0'$ and $z_0' \simeq \hat{z}_0$.

It is important to note that in Condition 1., the points z_0, $\hat{z}_0$, z_{cr} and $\hat{z}_{cr}$ need not be different. As a special case, we have $z_0 \simeq z_0$. Condition 2 ensures the transitivity of $\simeq$, and the symmetry is obvious. Hence, connectedness is an equivalence relation on $Z(p)$. Note that $Z_*(p)$ does not include real critical zeros, so two zeros are not necessarily connected if they are just connected to a common (multiple) Fourier critical zero of p.

Now we partition $Z(p)$ into the equivalence classes defined by $\simeq$. For each class $C \in Z(p)|_{\simeq}$, we define

$$f_{p,c}(C) = \{z_{cr} \in F(p) : \text{There exists } z_0 \in C \text{ with } z_0 \mapsto z_{cr}\} \tag{2.13}$$

With this definition, Part 1. of Theorem 2.17 reduces to saying that every $z_0 \in Z(p)$ is connected to some $z_{cr} \in F(p)$ (which prohibits empty sets as images), and vice versa (which shows that $F(p)$ is completely covered). Both these statements were shown in Lemma 2.41.

If neither $Z(p)$ nor $Z_+(p)$ have multiple zeros, then we have the equality $Z_k(p) = Z(p^{(k)})$. Now recall that every logarithmic derivative $\frac{p^{(k+1)}}{p^{(k)}}$ has only finitely many w-points, and there are only finitely many derivatives $p^{(k)}$, $k = 0, \dots, n-1$. Hence for all but finitely many values of c, $S_c(p^{(k)})$ contains no w-point of $\frac{p^{(k+1)}}{p^{(k)}}$. Then it follows from Lemma 2.34, Part 2., that all partitions $Z(p^{(k)})|_c$ consist only of one-element classes. Hence, $z \sim \hat{z}$ implies $z = \hat{z}$ for all $z \in Z_*(p)$. Also, from the proof of Lemma 2.41, Part 3., we infer that for each $z_0 \in Z(p)$, we get a unique Fourier critical zero z_{cr} in some $F_l(p)$ with $z_0 \mapsto z_{cr}$. Furthermore, in every set $Z_j(p)$, $j = 1, \dots, l-1$, there is a unique point z_j so that $z_0 \mapsto z_j$.

Conversely, for each of these z_j, z_0 is the only zero of p with $z_0 \mapsto z_j$. Hence, no zero of p except z_0 itself satisfies Condition 1. in Definition 2.42, so $Z(p)|_{\simeq}$ consists of one-element classes only. These classes, as shown, are mapped to single elements of $F(p)$. If thus $f_{p,c}$ is identified with a function mapping $Z(p)$ to $F(p)$, then Part 1. of the theorem shows that $f_{p,c}$ is surjective. This completes the proof of Part 2.

Now let z_{cr} be a single Fourier critical zero of some order k. Assume that $z_0 \mapsto z_{cr}$ and $\hat{z}_0 \mapsto z_{cr}$, for $z_0, \hat{z}_0 \in Z(p)$. Then we find points $z_1, \hat{z}_1 \in Z_*(p)$ (possibly of different orders or order 0) with $z_0 \mapsto z_1$, $\hat{z}_0 \mapsto \hat{z}_1$, and both z_1 and z_{cr} as well as $\hat{z}_1$ and z_{cr} are directly connected. By Lemma 2.34, Part 3., z_{cr} is in the image $f^{(k-1)}(C)$ of exactly one class C of $Z(p^{(k-1)})|_c$. In other words, if both z_1 and $\hat{z}_1$ are directly connected to z_{cr}, then $z_1 \sim \hat{z}_1$. Now according to Condition 1. in Definition 2.42, we have $z_0 \simeq \hat{z}_0$. This proves Part 3. of the theorem.

Part 4. of the theorem follows readily from Parts 2. and 3.: If by Part 2. $f_{p,c}$ is identified with a surjective function mapping $Z(p)$ to $F(p)$, then Part 3. shows that $f_{p,c}$ is injective.□

2.9 Problems and fixes

The proof of Theorem 2.17 looks very intricate and circumlocutionary, to say the least. It obscures the underlying, simple idea that can be written down in just one sentence: "From the nonreal zeros of p, follow the arc of constant argument of the logarithmic derivative of p and its derivatives, until you finally reach the Fourier critical points." The problems arise from two possible exceptions:

1. Zeros or critical points of p may be multiple.
2. On $S_c(p^{(k)})$ we may encounter w-points of $\frac{p^{(k+1)}}{p^{(k)}}$.

If we have neither of these exceptions for a given p and c, the proof of Theorem 2.17 indeed reduces to a matter of following analytic arcs from single zeros to single zeros of the next-higher derivative, as we have experienced in Example 2.30.

The second exception is so important that we will investigate it more closely in the next chapter, although it can be easily avoided by choosing c suitably. However, it is easy to find a polynomial for which a given $c \in (0, \pi)$ is exceptional in the sense of 2., so we cannot hope to find a "universal" c that avoids 2. for *every* polynomial in $\mathcal{P}$.

The exception 1. is inherent to a polynomial, and no choice of c can fix the problem. Strictly speaking, we must say that, whenever 1. applies, our algorithm does not provide a "definite, natural relationship" as sought in Conjecture 2.16. But on the other hand, it seems very unlikely that we can expect more. If nonreal zeros of p or Fourier critical zeros of p are multiple, it is impossible to find a bijection between them. And since being a real critical zero of some order k is a property that depends on $p^{(k-1)}$ rather than p, one would suggest that multiple nonreal zeros of $p^{(k+1)}$ (and of any other derivative of p) likewise prohibit a non-arbitrary bijection. Multiple zeros indicate a "limiting case" in which zeros "exchange their roles" (see Corollary 2.37).

Yet there are ways to circumvent these problems. They all have in common that they are arbitrary in some way. Here is a possible solution:

- Assign unique indices to every nonreal zero or critical point z of p, in the form $(z, k_1), (z, k_1+1), \ldots, (z, k_1+m_1-1), (z, k_2), \ldots, (z, k_l+m_l-1)$ where the k_j are the orders of the (critical) zeros, and the m_j their respective multiplicities at z. (We have the inequality $0 \leq k_1 < k_1+m_1 < k_2 < k_2+m_2 < \ldots < k_l+m_l \leq \deg p$.) Hence, the indices above list all derivatives of p that vanish at z.

- Index every multiple Fourier critical point z in the same way, except that the m_j are now the critical multiplicities of the critical zeros at z. (Here the indices do not list all derivatives of p that vanish at z.) Define $F(p)$ as the set of all these indexed points.

- Let $Z(p)$ consist of all points $(z, k_1), \ldots, (z, k_1+m_1-1)$, where z is a zero of p, i.e. $k_1 = 0$.

We are now going to construct arcs from *unique* points in $Z(p)$ to unique points in $F(p)$ as follows:

- The unique outgoing arc from an indexed zero point (z, j) is the arc of $S_c(p^{(j)})$ outgoing from z.

- If (z, k_j+h) is a nonreal critical zero (i.e. $k_j \geq 1$), then the unique incoming arc into this point is the arc of $S_c(p^{(k_j-1)})$ incoming into z between the angles $\frac{2h\pi}{m_j}$ and $\frac{2(h+1)\pi}{m_j}$ of the positive real direction.

- If $(z, k_j + h)$ is a *real* critical zero ($k_j \geq 1$), then the unique incoming arc into this point is the arc of $S_c(p^{(k_j-1)})$ incoming into z between the angles $\frac{h\pi}{m_j}$ and $\frac{(h+1)\pi}{m_j}$ of the positive real direction.
- An incoming arc of $S_c(p^{(j)})$ into a w-point z of $\frac{p^{(j+1)}}{p^{(j)}}$ of multiplicity, say, k, will be continued on the arc at the angle $\frac{+\pi}{k+1}$ of the incoming arc. (This is just the "next arc to the right", and it is an outgoing arc, as we have seen in Section 2.5.)

As a final answer to Gauss' question, we conclude with a statement that summarizes what we have found in this chapter:

> Except for certain "limiting cases", one can algorithmically construct a family of bijective mappings from the pairs of nonreal zeros to the Fourier critical zeros of any given polynomial.

In Chapter 6, we will spend a few words on the practical use of this result. To end this chapter, let us now consider a few geometrical results that are found in the literature.

2.10 Geometric Relationships

Most of the geometric results presented in the sequel make use of one of the Formulae (2.8) and (2.10). We list them here for easier reference:

$$\frac{p'}{p}(z) = \sum_{j=1}^{n} \frac{k_j}{z - \alpha_j}$$

$$\frac{p'}{p}(z) = \sum_{\alpha_j \text{ real}} \frac{k_j}{z - \alpha_j} + 2 \sum_{\substack{\alpha_j = x_j + y_j i \\ y_j > 0}} \frac{k_j(z - x_j)}{(z - x_j)^2 + y_j^2}$$

The "classical" and most general result is Lucas's Theorem [15], commonly viewed as a generalization of Rolle's Theorem to complex-valued polynomials. Before stating it, we give a preliminary lemma:

Lemma 2.43 *Let p be a complex, nonconstant*[8] *polynomial. If all zeros of p lie in the closed upper half plane $\overline{H_+}$, then so do all zeros of p'. If additionally not all zeros of p are real, then all zeros of p', except multiple real zeros of p, lie in H_+.*

[8] In all other theorems, this is understood as well.

Proof: Let $z \in H_-$. Then for every term in (2.8), $\Im(z - \alpha_j) < 0$, so $\Im(\frac{k_j}{z-\alpha_j}) > 0$. The same must then hold for the sum of all terms, so $\frac{p'}{p}(z)$ cannot be 0. Since $p(z) \neq 0$ (as no zero of p lies in H_-, we conclude that $p'(z) \neq 0$. Therefore, all zeros of p' must be in the closed upper half plane.

In the second part, if z is real, and at least one zero α_{j_0} of p lies in H_+, then we have $\Im(\frac{k_{j0}}{z-\alpha_{j0}}) > 0$, and at least $\Im(\frac{k_j}{z-\alpha_j}) \geq 0$ for all other terms. Hence we also get $\frac{p'}{p}(z) \neq 0$. If $p(z) \neq 0$ as well, we conclude as above that $p'(z) \neq 0$. (Note that simple zeros of $p(z)$ can never be zeros of $\frac{p'}{p}$.) This shows the second part. □

Another result needed for Lucas's Theorem shows that the location of the zeros of p' relative to the zeros of p is invariant under shifting and scaling:

Proposition 2.44 *Let $\hat{p}(z) = p(c+z)$ and $\tilde{p}(z) = p(\alpha z)$ be the polynomials obtained from p by shifting and scaling with complex, nonzero constants c and α, respectively. Then*

1. $\hat{p}'(z) = p'(z+c)$.
2. $\tilde{p}'(z) = \alpha p(\alpha z)$.

Proof: Both parts follow from applying the chain rule of differentiation. □

We are now ready to state and prove Lucas's theorem itself:

Theorem 2.45 *(Lucas's Theorem) Let $K = \operatorname{conv}\{\alpha_1, \ldots, \alpha_n\}$ denote the convex hull of all zeros of a complex polynomial p. Then the zeros of p' also lie in K. If not all zeros $\alpha_1, \ldots, \alpha_n$ lie on a straight line, then all zeros of p', except possibly multiple zeros of p, lie in the interior of K.*

Proof: K as the convex hull of a finite set is a polygon with some (not necessarily all) of the α_j as vertices. Like every polygon, K can be written as the intersection of the closed half-planes in $\mathbb{C}$ determined by its sides and containing K. By means of Proposition 2.44, and by choosing suitable values α and c, each of these half-planes can be translated to $\overline{H_+}$. Then Lemma 2.43 shows that the zeros of p' also lie in each of these half-planes, and thus in K. This shows the first part of the theorem. The second part follows from the second part of Lemma 2.43 by the same argument. □

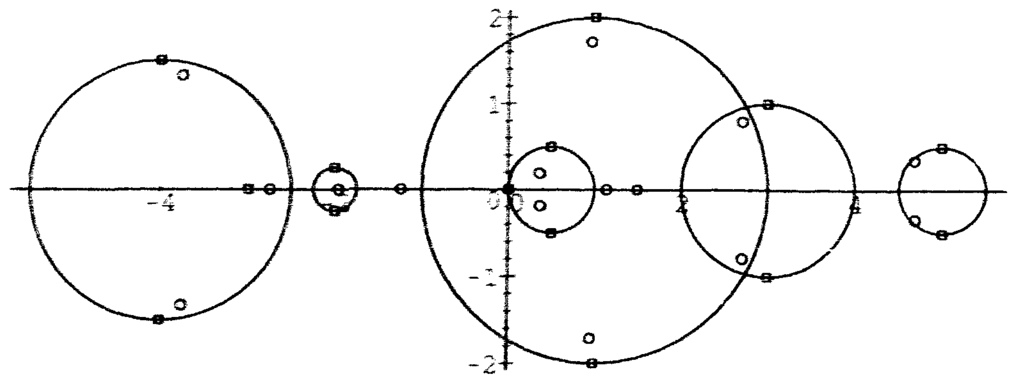

Figure 2.7: *The Jensen circles of a real polynomial p. Displayed are also the zeros(boxes) and critical zeros (small circles) of p. Note that the critical zero outside the Jensen circles is not a Fourier critical zero.*

In general, this result is also best possible in the sense that, given any finite set S of points in the interior of K, one can construct a polynomial with all zeros in K that takes all points in S as zeros of its derivative. Further restrictions on the locations of the zeros of p however can lead to improved results.

We will now restrict ourselves to the class $\mathcal{P}$ of polynomials we are most interested in.

Definition 2.46 For a pair of conjugate-complex values $\alpha, \overline{\alpha} = x_0 \pm y_0 i$ (which will frequently be zeros of p), we define the *Jensen circle of α* to be the circle with centre x_0 and radius y_0, this is, the set

$$\{z : |z - x_0| = y_0\}. \tag{2.14}$$

Similarly we define the (open or closed) *Jensen disk* of α, replacing "=" by "<" and "≤" in (2.14).

We will commonly use the term *Jensen circles of p* for the Jensen circles of all pairs of nonreal zeros of p. Figure 2.7 shows an example of the Jensen circles of a real polynomial.

Theorem 2.47 *(Jensen's Theorem) Every zero of p' lies either on the real axis or on one of the Jensen disks of p.*

This theorem was stated by Jensen [14] and proved later by Walsh [26] and Nagy [19]. The proof given here is based on the one in [17], p.26:

Proof: Take $z = x + iy$, and consider $\Im(\frac{p'}{p}(z))$. For each of the terms in the first sum of (2.10), we get

$$\Im\left(\frac{k_j}{z-\alpha_j}\right) = \frac{-k_j y}{(x-\alpha_j)^2 + y^2} \tag{2.15}$$

and for each term in the second sum, after re-arranging and factorizing the terms:

$$\Im\left(\frac{k_j(z-x_j)}{(z-x_j)^2+y_j^2}\right) = \frac{-2k_j y((x-x_j)^2+y^2-y_j^2)}{((x-x_j)^2+(y-y_j)^2)((x-x_j)^2+(y+y_j)^2)} \tag{2.16}$$

If a critical zero $z_0 = x_0 + y_0 i$ lies outside the Jensen circles of p, which is,

$$(x_0 - x_j)^2 + y_0^2 > y_j^2 \quad \text{for all } j, \tag{2.17}$$

then all terms in (2.15) and (2.16) have the same sign, namely $-\operatorname{sgn} y_0$. Hence, we have $\operatorname{sgn}\Im(\frac{p'}{p}(z_0)) = -\operatorname{sgn}\Im(z_0)$, and *a fortiori*, $\frac{p'}{p}(z_0)$ can only vanish if z_0 is real. □

A special corollary of this proof provides useful information about the loci we constructed in the proof of Theorem 2.17:

Corollary 2.48 *For any $c \in (0, \pi)$, the symmetric locus $S_c(p)$ lies completely inside the Jensen circles of p.*

Proof: In the proof of Theorem 2.47, we observed that for any $z \in H_+$ outside the Jensen circles of p, $\Im(\frac{p'}{p}(z))$ is negative, and thus $\operatorname{Arg}(\frac{p'}{p}(z)) \in (-\pi, 0)$. By a continuity argument, we get $\operatorname{Arg}(\frac{p'}{p}(z)) \in [-\pi, 0]$[9] on the Jensen semicircles of p in H_+. This shows the statement on the upper half plane. The rest of the proof follows by the already familiar symmetry argument. □

This shows that the mapping found in the proof of Lemma 2.34 is "in line" with these geometrical results: Nonreal zeros of p can only be mapped to (nonreal or real) critical zeros if they lie in the same connected set of Jensen disks of p. Moreover, if the Jensen disks of k pairs of nonreal zeros of p (counting multiplicities) are disjoint (and hence separated)

[9] Here $-\pi$ is understood to be the principal value of $\operatorname{Arg} z$ for negative real z. Normally, this value would be π.

from all other Jensen disks of p, then by Corollary 2.35 they are connected through arcs of $S_c(p)$ to k real or pairs of nonreal critical zeros of order 1, counting *critical* multiplicities. Hence we immediately obtain the following two corollaries:

Corollary 2.49 *Every Fourier critical zero of p lies on one of the closed Jensen disks of p.*

Corollary 2.50 *Let $I = [x_1, x_2]$ be an interval of the real axis such that neither x_1 nor x_2 is contained in any closed Jensen disk of p, and J the union of all closed Jensen disks intersecting I. If J contains k pairs of nonreal zeros of p, counting multiplicities, then J also contains k critical zeros of p, counting critical multiplicities.*

This corollary can be generalized to include *all* zeros and critical zeros of a polynomial in a certain interval:

Theorem 2.51 *[25] Let $I = [x_1, x_2]$ be an interval of the real axis such that neither x_1 nor x_2 is a zero of p, or is contained in any closed Jensen disk of p. Let further J be the union of I and all closed Jensen disks intersecting I. If k and k' are the number of zeros of p and p' in J, respectively, then the inequality $k - 1 \leq k' \leq k + 1$ holds.*

Conventional proofs of this theorem (see e.g. [17], p. 27) are very complicated, and they use higher-level tools such as the Argument principle. Using the previous results of this chapter, our proof reduces to a matter of careful counting:

Proof: We define the following numbers:

m: the number of real zeros of p in I, counting multiplicities,

d: the number of pairs of nonreal zeros of p in $J \setminus I$, counting multiplicities,

c: the number of real (Fourier) critical zeros in I, counting *critical* multiplicities,

m': the number of real zeros of p' in I, counting multiplicities,

d': the number of pairs of nonreal critical zeros in $J \setminus I$, counting multiplicities.

Then by definition we have $k = m + 2d$ and $k' = m' + 2d'$. By Corollary 2.50, we also have $d = c + d'$, and in Theorem 2.13, we showed $m + 2c - 1 \leq m' \leq m + 2c + 1$. On putting these relations together, we obtain the above inequality. □

Theorem 2.47 has many more corollaries, and it is impossible to present all of them here. We will pick the most interesting ones. For a further study of the topic, see [17, 27].

Next, we give an "iterative" version of Theorem 2.47 and Corollary 2.49. For a pair of conjugate nonreal values $\alpha, \overline{\alpha} = x_0 \pm y_0 i$, we define the *Jensen ellipse of order* k *of* α to be the ellipse with the segment from $\overline{\alpha}$ to α as minor axis, and the real interval $[x_0 - \sqrt{k}y_0, x_0 + \sqrt{k}y_0]$ as major axis. We use the expressions *Jensen ellipse of* p and *elliptical Jensen disk* in the same fashion as their equivalents in Definition 2.46. Note that the Jensen circles are just the Jensen ellipses of order $k = 1$.

Theorem 2.52 *[14] For any* k *smaller than the degree of* p*, all nonreal zeros of* $p^{(k)}$*, and all Fourier critical zeros of order* k *lie in the closed elliptical Jensen disks of order* k *of* p*.*

Proof: See e.g. [27],p. 84. The proof uses induction on k. It is based on the (easy-to-prove) fact that the elliptical Jensen disk of order $k+1$ of some point α is just the union of the (circular) Jensen disks of all points on the elliptical Jensen disk of order k of α. □

A converse to Theorem 2.47 is the following

Corollary 2.53 *Let* $z_0, \overline{z_0} = x_0 \pm y_0 i$ *be a pair of conjugate nonreal critical zeros of* p*. Then the equilateral hyperbola centered at* x_0 *with vertices* z_0 *and* $\overline{z_0}$ *contains at least one pair of nonreal zeros of* p*.*

Proof: In the proof of Theorem 2.47, the necessary condition for z_0 being a critical zero is that (2.17) is false for at least one pair $\alpha, \overline{\alpha} = x \pm yi$. Rewriting the negation of (2.17) as

$$y_0^2 \leq y^2 - (x_0 - x)^2, \tag{2.18}$$

we see that α and $\overline{\alpha}$ must be inside or on the hyperbola described above. □

Of course, this converse has a corollary for Fourier critical zeros, similar to Corollary 2.49. In this case, $y_0 = 0$, so the hyperbola above degenerates to the equilateral angular region centered at x_0. Also, a similar extension of Theorem 2.52 is possible (See e.g. [20]).

The last result we present in this chapter applies to polynomials the zeros of which can be grouped in two circles:

Theorem 2.54 *[25](Two-Circle Theorem)*

Let p be a complex polynomial of degree $m_1 + m_2$ that has m_1 and m_2 zeros in the disks $C_1 = \{z : |z - \alpha_1| \leq r_1\}$ and $C_2 = \{z : |z - \alpha_2| \leq r_2\}$, respectively ($\alpha_1, \alpha_2 \in \mathbb{C}, r_1, r_2 \geq 0$). Let C be the disk

$$\left\{z : \left|z - \frac{m_2\alpha_1 + m_1\alpha_2}{m_1 + m_2}\right| \leq \frac{m_2 r_1 + m_1 r_2}{m_1 + m_2}\right\}.$$

Then the zeros of p' lie in C_1, C_2, and C. If C_1, C_2, and C are pairwise disjoint, then they contain $m_1 - 1$, $m_2 - 1$ and 1 zeros of p', respectively.

Proof: See e.g. [27], pp. 13-17. $\square$

Chapter 3

The Wronskian of a polynomial

3.1 Definition

Let p be a (real or complex) polynomial. We define the *Wronskian*[1] of p (written as Wp) by

$$Wp(z) = \begin{vmatrix} p(z) & p'(z) \\ p'(z) & p''(z) \end{vmatrix} = p(z)p''(z) - (p'(z))^2 \tag{3.1}$$

Comparing Equations (3.1) and (2.7), one observes that Wp is closely related to the second logarithmic derivative:

Corollary 3.1 $\left(\frac{p'}{p}\right)'(z) = \frac{Wp(z)}{p^2(z)}$

In particular, $\left(\frac{p'}{p}\right)'$ and Wp have the same zeros, except possibly for zeros of p. Furthermore, if p is a real polynomial, they also have the same sign along the real axis. From Equation (2.9), we further get

$$Wp(z) = -p^2(z) \sum_{j=1}^{m} \frac{k_j}{(z - \alpha_j)^2} \tag{3.2}$$

[1] The term "Wronskian" is taken from a wider class of differential operators defined in a similar, but more general way than (3.1).

where, as before, α_j are the zeros of p and k_j their respective multiplicities ($j = 1, \ldots, m$). This will prove to be a helpful formula.

Wp is difficult to deal with, because it is a nonlinear operator. However, we have the following useful scaling and shifting properties (compare Lemma 2.44):

Lemma 3.2 *Let $\hat{p}(z) = p(c + z)$ and $\tilde{p}(z) = p(\alpha z)$ be the polynomials obtained from p by shifting and scaling with complex, nonzero constants c and α, respectively. Then*

1. $W\hat{p}(z) = Wp(z + c)$
2. $W\tilde{p}(z) = \alpha^2 Wp(\alpha z)$.

Proof: Both parts follow from applying the chain rule of differentiation. □

The following arithmetic properties are also useful:

Lemma 3.3 *[4] Let p, p_1, p_2 be complex polynomials, $\alpha \in \mathbb{C}$, and $n \in \mathbb{N}$. Then*

1. $W(p_1 p_2) = p_1^2 Wp_2 + p_2^2 Wp_1$.
2. $W(p^n) = np^{2n-2} Wp$.
3. $W(z - \alpha) = -1$.

Proof: The first statement can be easily obtained from (3.2), and the third follows immediately from the definition of Wp, whereas Statement 2. follows from recursive application of 1. □

3.2 General properties

As we have seen in the previous chapter, the zeros of Wp provide a lot of information about the behaviour of the loci of $\frac{p'}{p}$. Therefore, we wish to have some results on their possible locations. First, we cite some results for general complex polynomials:

Proposition 3.4 *If z_0 is a zero of p with multiplicity $k \in \mathbb{N}$, then z_0 is also a zero of Wp with multiplicity $2k - 2$.*

Proof: This can be seen from (3.2). If $p^2(z)$ is multiplied with each of the sum terms, then the factor $(z - z_0)$ appears $2n - 2$ times in the numerator of exactly one term, and $2n$ times in the numerator of all other terms. □

We call zeros of Wp *trivial* if they arise from multiple zeros of p. Hence we can conclude:

Corollary 3.5 *A zero z of Wp is a zero of $\left(\frac{p'}{p}\right)'$ if and only if it is non-trivial. In that case, z has the same multiplicity both in Wp and $\left(\frac{p'}{p}\right)'$.*

When can zeros of p, p' and Wp coincide? We have already discussed multiple zeros of p. As a special case of Proposition 3.4, a single zero of p can never be a zero of Wp. Finally, we get from Corollary 3.5 that a k-fold critical zero of order 1, being a k-fold zero of $\frac{p'}{p}$, is a $(k-1)$-fold zero of Wp. These zeros we will not call trivial, because they will not be exceptional in the theorems to come.

Although trivial zeros themselves are not interesting to investigate, the multiplicity of zeros of p does have an influence on the location of the nontrivial zeros of Wp. However, if all zeros of p are multiple, one may divide by their common denominator:

Proposition 3.6 *For any n, Wp and $W(p^n)$ have the same set of nontrivial zeros.*

Proof: This follows immediately from Part 2. of Lemma 3.3. □

This result also allows one to safely "double" all zeros, so one can handle a double real zero as a limiting case of a pair of nonreal zeros.

Now we will turn our attention to nontrivial zeros of Wp:

Proposition 3.7 *[4] If all zeros of p lie on a straight line in $\mathbb{C}$, then no nontrivial zero of Wp lies on that line. In particular, if p has only real zeros, then Wp has no real nontrivial zeros at all.*

Proof: By means of Lemma 3.2, we can translate the line to the real axis. But if all zeros of p are real, then every term in the sum in (3.2) is positive, and hence $Wp(z) < 0$, except at (multiple) zeros of p. □

On the other hand, we can show that the zeros of Wp cannot be too far away from the zeros of p either. For this, we cite a result by Marden ([17], p. 30f.) for the special case needed here:

Lemma 3.8 *If K is a convex region containing all zeros of p, then $Wp(z) \neq 0$ at any point z where K subtends an angle less than $\pi/2$.*

Proof: Apply [17, Thm. 8.1] to (3.2). □

Applying this to the unit circle and to the interval $[-1, 1]$, we get the following special results:

Theorem 3.9 *[4] If all zeros of p lie inside or on the unit circle, then all zeros of Wp lie inside or on the circle of radius $\sqrt{2}$ around the origin.*

Proof: Let z be a point at which the unit circle subtends an angle of $\pi/2$. The tangents to the unit circle through z, and the normals from the tangent points to the origin form a square of unit length, the diagonal of which is the vector from z to the origin. Hence we get $|z| = \sqrt{2}$, and at every point at a distance greater than $\sqrt{2}$, the unit circle subtends an angle less than $\pi/2$. The theorem follows from Lemma 3.8. □

Theorem 3.10 *[4] If all zeros of p lie in the real interval $[-1, 1]$ then all zeros of Wp lie inside or on the unit circle.*

Proof: This follows in a similar way from Lemma 3.8 and a simple geometric argument: The unit circle is the locus of all points at which its diameter (the interval $[-1, 1]$) subtends an angle of $\pi/2$. □

Note that these theorems generalize to arbitrary circles and line segments in $\boldsymbol{C}$, by applying Lemma 3.2. All three results are sharp, as is shown in [17] for Lemma 3.8, and in [4] for Theorems 3.9 and 3.10. In order to get better results, one must impose further restrictions on the locations of the zeros of p. [4] contains a few more results for polynomials with only real zeros. We will not present them here. Instead, we will now come back to the wider class of real polynomials.

3.3 The Wronskian of real polynomials

For the rest of this chapter, let $p \in \mathcal{P}$. For this class of polynomials, Craven, Csordas and Smith conjectured the following:

Conjecture 3.11 *[3] Let p have exactly $2d$ nonreal zeros. Then Wp does not have more than $2d$ real zeros.*

They attribute this conjecture to Gauss, referring to him indirectly through the papers of Pólya [24] and Nagy [20]. However, the references given in those papers refer to [7] and the letters [10, 11, 12], given and translated in Appendix A of this work. These letters do not mention the Wronskian, nor even the logarithmic derivative, with any word. It is likely that Gauss thought of using the logarithmic derivative in order to prove his question; and in the course of his investigations, he might have discovered zeros of the Wronskian as a complication, perhaps even the one he could not overcome. But this is merely speculative, and it would be entirely inappropriate to give him the credit for Conjecture 3.11.

We already proved the case $d = 0$, which is just Proposition 3.7. In [3], the conjecture is shown for all p such that $\frac{p'}{p}(z) + \gamma$ has only real zeros for some real constant γ. We will not prove the conjecture. In Chapter 5, we will suggest a similar approach as we used in Chapter 2. Here we will only give geometrical results.

First we note that clearly $Wp \in \mathcal{P}$ as well. In particular, $Wp(z)$ takes only real values on the real axis. We can even gain more information from degree considerations, from the properties of $\frac{p'}{p}$ we found in the previous chapter, and from Rolle's Theorem:

Proposition 3.12 *(Behaviour of Wp on the real axis)*

1. *Wp is negative in some neighbourhood of each zero of p, and outside a certain interval $[x_1, x_n]$.*
2. *Wp has an even number of real zeros, counting multiplicities.*
3. *In the closed interval between two adjacent real zeros of p', there is either a zero of p, or an odd number of zeros of Wp. Between two zeros of p, or between a zero of p and a zero of p', there is an even number (possibly 0) of zeros of Wp.*
4. *If the infinite point is regarded as a zero of p', then 3. generalizes to infinite real intervals.*
5. *If p has no real zeros at all, then Wp has at least two real zeros.*

Proof: These statements all follow from previous results, using elementary methods of real calculus. For example, to prove 5., we recall that $\frac{p'}{p}(z)$ tends to 0 as $z \to \infty$. Since $\frac{p'}{p}$ does not have any poles, it is continuous and bounded on the whole real line. By Corollary 2.21, we see that $\frac{p'}{p}$ takes both positive and negative values. Therefore, it must attain both its maximum and minimum at some points. At those points, we have $(\frac{p'}{p})'(z) = 0$. □

We will now establish another criterion for Wp to be negative. For this, we construct a formula similar to (2.10) for $(\frac{p'}{p})'$, either by differentiating (2.10), or by grouping conjugate terms together in (3.2):

$$Wp(z) = -p^2(z)\left(\sum_{\alpha_j \text{ real}} \frac{k_j}{(z-\alpha_j)^2} + 2\sum_{\substack{\alpha_j = x_j + y_j i \\ y_j > 0}} \frac{k_j((z-x_j)^2 - y_j^2)}{((z-x_j)^2 + y_j^2)^2}\right) \tag{3.3}$$

This leads us to the following theorem, apparently due to Nagy:

Theorem 3.13 *[20] Wp is negative for all real points outside the union of all closed Jensen disks of the nonreal zeros of p, except for trivial zeros of Wp.*

Proof: Let $\alpha_j = x_j + iy_j$, $y_j > 0$, be any nonreal zero of p. The Jensen disk of α_j and $\overline{\alpha_j}$ consists of all points z for which $(z - x_j)^2 \leq y_j^2$ holds. If z is outside this disk, then the sum term in (3.3) corresponding to α_j is positive. Therefore, if a point z with $p(z) \neq 0$ lies outside all Jensen disks, then all sum terms are positive, and consequently $Wp(z)$ is negative. □

The following corollary is immediate:

Corollary 3.14 *Every nontrivial real zero of Wp lies in the closed Jensen disk of a pair of nonreal zeros of p.*

Theorem 3.13 has many other corollaries, refinements and generalizations. Many of them are analogues to the geometrical results stated in Section 2.10, e.g. the following generalization to higher derivatives:

Theorem 3.15 *[20] The intervals in which $W(p^{(k-1)})$ is nonnegative ($1 \leq k < \deg p$) are contained in the union of the closed elliptical Jensen disks of order k of p, except possibly for multiple zeros of $p^{(k-1)}$.*

Proof: By Theorem 2.51, we locate the nonreal zeros of $p^{(k-1)}$ on the closed elliptical Jensen disks of order $k-1$. Then we apply Theorem 3.14 to $p^{(k-1)}$ to locate the zeros of $W(p^{(k-1)})$ on the (circular) Jensen disks of the nonreal zeros of $p^{(k-1)}$. Now we use the same geometrical argument as in the proof of Theorem 2.51 to obtain the theorem. □

A converse of Theorem 3.13 describes the nonreal zeros of p in terms of the zeros of Wp:

Corollary 3.16 *If x_0 is a real point with $Wp(x_0) \geq 0$, then the equilateral angular region*

$$\{z = x + yi : |x - x_0| \leq |y|\} \tag{3.4}$$

contains at least one pair of nonreal zeros of p.

Proof: Analogous to the proof of Corollary 2.53. Note that (3.4) is just the degenerate hyperbola (2.18), for $y_0 = 0$. □

Corollary 3.16 gives a fairly good (and in fact sharp) result, if p has no real zeros at all. However, if p has real zeros, then the nonreal zeros in the angular region of x_0 cannot be too far away from x_0:

Theorem 3.17 *[5][2] Let $\alpha_1, \ldots, \alpha_m$ be the real zeros of p, and x_0 be a real zero of Wp. Then p must have at least one pair of nonreal zeros in the intersection of the angular region (3.4) and the closed disk centered at x_0 with radius $\sqrt{2h}R(x_0)$, where h is the number of closed Jensen disks of p containing x_0, and $R(x) = \left(\sum_{j=1}^{m} \frac{1}{(x-\alpha_j)^2}\right)^{-1/2}$.*

As a converse to this theorem, one can say that a real zero α "extinguishes" real zeros of Wp in the Jensen disk of a pair $z, \bar{z}$ of nonreal zeros, if α is too close to $\Re(z)$, or z is too far away from the real axis. The following theorem describes this quantitatively:

Theorem 3.18 *Let $z, \bar{z} = x \pm yi$ be a pair of nonreal zeros of p. If p has a real zero α with $\frac{|x-\alpha|}{y} < c$, where*

$$c = \sqrt{3/2}\left(\sqrt[3]{16\sqrt{2}+13} - \sqrt[3]{16\sqrt{2}-13}\right) \cong 0.4947092,$$

then the points of the closed Jensen disk of z which don't lie in any other Jensen disk of p contain no real zeros of Wp.

[2]The remaining theorems of this chapter are all taken from [5] and cited without proof.

In [5], this theorem, though looking similar to Theorem 3.17, is stated and proved independently. Furthermore, criteria have been derived for Wp to have no zeros in a given interval, or no zeros at all. We will not cite them here.

The next two theorems give two different sufficient criteria for Wp to *have* real zeros in certain intervals:

Theorem 3.19 *Let $\alpha, \overline{\alpha} = x \pm yi$ be a pair of simple nonreal zeros, and $J(\alpha)$ the Jensen disk of α. If $J(\alpha)$ has no point in common with any other Jensen disk of p, and no other zero of p lies in the strip*

$$x - y\sqrt{2} \leq \Re(z) \leq x + y\sqrt{2}$$

then $J(\alpha)$ contains exactly two zeros of Wp.

Theorem 3.20 *(Two-Circle Theorem for Wp)*

Let p have exactly $2n$ zeros, n each in the closed disks centered at $\alpha = x + yi$ $(y > 0)$, and $\overline{\alpha}$, respectively, with radius $r < y/2$. Then Wp has exactly two real zeros, located in the intervals $[x - y - r\sqrt{2}, x - y + r\sqrt{2}]$ and $[x + y - r\sqrt{2}, x + y + r\sqrt{2}]$, respectively.

This theorem can be generalized to further cases where the zeros of p split into several small groups sufficiently separated from each other and from the real axis.

The last result describes the possible locations of the *nonreal* zeros of Wp:

Theorem 3.21 *For all (real or pairs of nonreal) zeros $\alpha_j, \overline{\alpha_j} = x_j \pm y_j i$ of p, let*

$$S_j = \{z = x + yi : (x - x_j)^2 + y_j^2 \leq y^2\}$$

be the region bounded by the equilateral hyperbola with vertices at α_j and $\overline{\alpha_j}$. (For $y_j = 0$, S_j degenerates to a double angular region.) Then every zero of Wp lies either on one of the Jensen disks of p, or in one of the S_j.

One can even show that the nontrivial zeros of Wp outside the Jensen circles don't lie too close to the real axis:

Theorem 3.22 *With the definitions in Theorem 3.21, let d be the minimum distance between two distinct real parts x_j and x_k of two zeros α_j, α_k. Then every nontrivial zero of Wp lies either in a Jensen disk of p, or outside the strip*

$$|\Im(z)| \geq \frac{\sqrt{3/2}}{2\pi} d. \tag{3.5}$$

It is conjectured that if p has m pairs of nonreal zeros and no real zeros, then no less than $2m - 2$ zeros of Wp satisfy (3.5). If one could show this, it would prove Conjecture 3.11, at least in the case that p has no real zeros.

Chapter 4

Using the xzero package

This chapter describes the facilities that come with the *xzero* package, from a user's point of view. We assume that *xzero* has already been installed, and is ready to use. We will neither deal with the system requirements of *xzero*, nor with modifications of the computation package or of application resources. For details about these topics, we refer to Appendix B, and to the manual pages and help files included in the package.

For a thorough knowledge of *xzero*, however, its computing strategy needs to be understood, especially in the event of unexpected behaviour or errors. We will introduce the basics of it later in this chapter, before we describe possible problems and errors in the package. Also, the data file format used in *xzero* will be explained later on, so that one can utilize *xzero* data files in one's own applications.

4.1 The concept of xzero

The purpose of *xzero* is to design polynomials interactively, and to use them in various computations. Therefore, *xzero* consists of two major components: The graphical user interface and the computation engine.

The way polynomials are designed in *xzero* is through their zeros. From the Fundamental Theorem of Algebra, one knows that a polynomial is determined, up to a multiplicative factor, by its roots. Therefore, at least in theory, arbitrary polynomials can be generated[1].

[1] There do exist problems with this approach, though. Polynomials created randomly by their coefficients

Zeros are displayed and modified in a window called the *drawing area*. The user can do most of the polynomial editing just by using the mouse and a few keyboard keys. In this. xzero resembles other graphical tools and packages. such as *Geometer*, *Geometer's Sketchpad*. and *xfig*. Those who have some experience with these or similar packages will find it straightforward to work with xzero. Some familiarity with basic features of graphical windows systems. such as clicking. selecting. dragging. will be assumed throughout this description.

Computations of xzero will be executed only upon request by the user. and only on the polynomial that has been designed in the first step. No other parameters need to be specified[2]. This gives the input data for computations a unique format. namely the set of zeros that specifies the polynomial.

The output format however is variable. depending on the type of computation. Possible results of computations are:

- The equation of the polynomial. and of related structures.
- Numerical values. e.g. winding numbers.
- Graphs of the polynomial on various curves in the complex plane.
- A boolean value. stating if a given boolean expression involving the current polynomial is true or not.
- Another set of complex values, e.g. the zeros of the derivative or of the Wronskian.

While all these types of output are supported, the last one deserves special treatment. since it is the same type in which the polynomial itself is defined. Therefore, a set of result values can also be displayed just as. and together with the values that determine the polynomial. Of course. one wants to distinguish between input and output. Therefore, xzero maintains and displays them as two different sets. or better: lists, of complex points. The former will henceforth be called ***root list***. the latter ***result list***.

---

tend to have a larger percentage of real zeros than polynomials created by their zeros.

[2]However. the user may specify additional fixed parameters in a text file that is read at the beginning of each xzero session.

The points in the root list can be freely edited and manipulated by the user, in order to generate arbitrary polynomials. It is in the nature of the result list to be dependent only on the input list, and the type of computation that generates them. Therefore, result values are not modifiable. Both roots and results can be saved as files. In order to keep the above-mentioned distinction between input and output, roots and results have to be saved separately. (This is only at first glance a disadvantage: Zero lists that exist in separate files can be loaded separately into other programs or back into *xzero* itself. Also, one often wants to save only the root list without results, or one wants to keep several result lists (from different computations) along with a root list.)

The other kinds of result types are incompatible to the *xzero* input type. They can be displayed through devices outside the graphical area, which may be other graphical or text windows. It is left up to these devices how the result data may be permanently stored. In practical use, text output can be redirected to a file, and plotting windows usually have options that let one save or print a graph.

xzero supports installing additional computation procedures that may be desirable beyond those that are already realized.

4.1.1 Symmetry concept

The concept of symmetry is used in *xzero* to generate polynomials guaranteed to have real coefficients. To achieve this, *xzero* has a *symmetric mode*, in which roots are forced to lie symmetric to the real axis. Each creation, deletion, movement and even selection of a zero will affect the conjugate zero in exactly the same way, but with inverse sign in imaginary direction, thereby preserving symmetry. Details for each of these operations will be given when the operation itself is described.

The symmetry concept also extends to the computations. Computations in symmetric mode are performed as real operations which ensures that the result is a real polynomial, without possible rounding errors leading to non-zero imaginary parts.

At startup, *xzero* is in symmetric mode. This mode can be switched on or off by clicking the `Symmetric` button in the `Options` menu. When turning symmetry on, *xzero* will automatically add the conjugate complex zero of each existing zero to the list, except for real zeros. *xzero* doesn't recognize already existing pairs of conjugate-complex zeros. Instead, it simply doubles these pairs.

When turning symmetry off, the conjugate zeros will disappear again, as if they were never created. It is important to remember that the conjugates are only "virtually" existent.

Symmetric roots can also be created manually, without switching to the symmetric mode. But in non-symmetric mode, neither the editing operations nor the computations are performed in a way that preserves manually created symmetry. Besides, *xzero* has no way of "recognizing" a manually created, symmetric list of zeros.

4.2 Starting xzero

xzero is invoked by typing

`xzero &`

on the UNIX command prompt. (The ampersand after the command is recommended in order to launch the application in the background. This allows one to type other commands in the shell window, while *xzero* is running.)

Alternatively, one can type

`xzero` <*filename*> `&`

to work on a file previously created by *xzero*. The result is the same as starting *xzero* without a file name, and then loading the corresponding file from the menu (Version 1.1 only).

4.3 The graphical interface of xzero

At startup, *xzero* opens the main application window that looks as in Figure 4.1. This appearance, as well as mouse and keyboard specifications of *xzero*, can be widely customized. How to do that will be described in Appendix B.4. Here we will only refer to the default specifications.

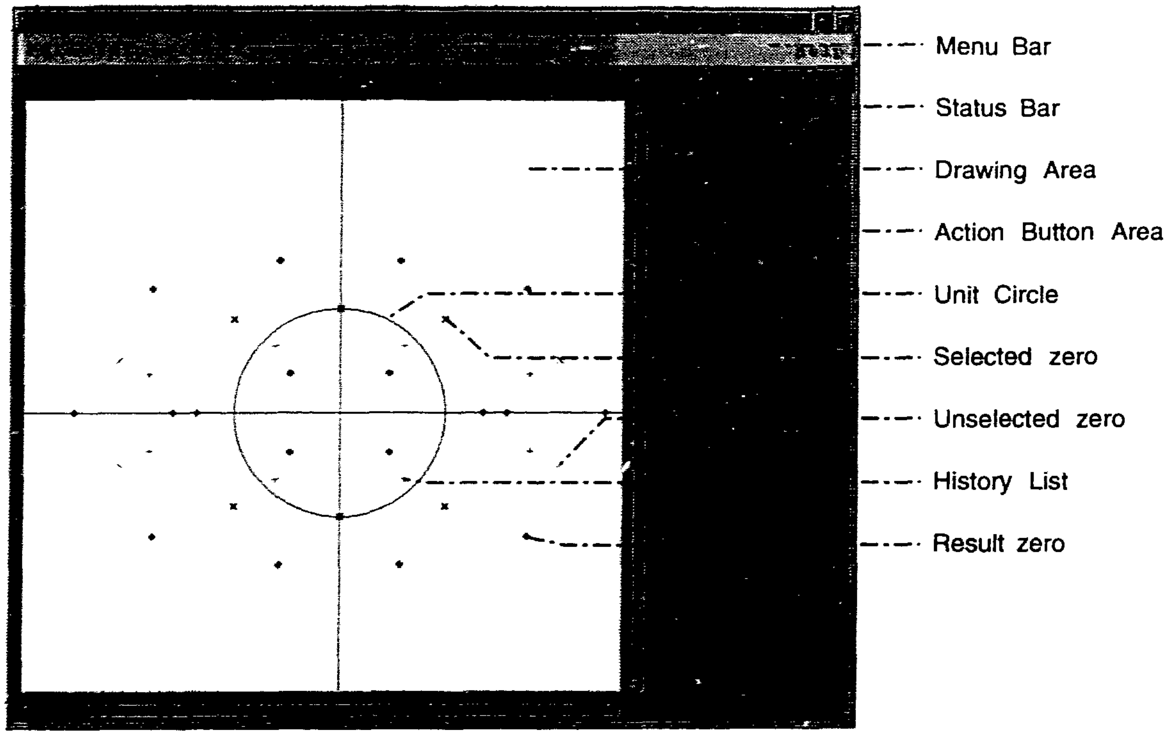

Figure 4.1: xzero *in a typical application.*

One can distinguish between the following 4 areas:

- The drawing area (white window), in which the zeros can be drawn, modified, or deleted.

- Action buttons (next to the drawing area) which let the user execute various functions (Version 1.1 only).

- The status bar (above the drawing area) in which point coordinates and status information are displayed.

- The menu bar (above the status bar), containing menus for all possible operations, as well as an online help menu.

All these areas, and how to use them, will now be described in detail.

4.4 The elements of the drawing area

The white drawing area contains a part of the complex plane, defined by the coordinate translation specifications. The following objects can (but do not always or at the same time) appear in this window:

- unselected, modifiable zeros, represented by red, '×'-shaped images,
- selected zeros, represented by thicker '×'es,
- zeros that arise as results of a computation, as green '+'es. They cannot be modified,
- zeros taken from previous result, as thin green '+'es, the so-called *result history* (see Section 4.6.5): cannot be modified,
- coordinate axes, the unit circle etc., shown in black,
- a selector frame (see "Selecting points in a rectangular area" in Section 4.4.1)
- the cursor, appearing in various shapes, depending on the current operation

The visible drawing area displays only a part of the complex plane. In fact, the area where points can be set is virtually unlimited (although physical limits are given by the numerical ranges and precisions of the hardware). To access any part of the virtual drawing area, the visible drawing area can be scrolled both in real and imaginary direction (see Section 4.4.3). In addition, there are several functions in the `View` menu (see Section 4.6.3) that zoom or move the visible part of the drawing area (Version 1.1 only). It is also possible to modify the default coordinate translation specifications.

4.4.1 Operating modes

xzero distinguishes between two different mouse input modes. The *Add* mode lets the user add new zeros, whereas the *Select* mode is used for various selection and "drag and drop" routines. At startup, *xzero* is in Select mode. The current mode can be recognized by the cursor shape: In Select mode it is a pointer arrow; in Add mode it is a dot shape. The cursor will also change its appearance in the course of some of the operations described below.

There are two ways to toggle between the two mouse modes. Clicking the right mouse button permanently changes the mouse mode, while pressing one of the `Ctrl` keys changes it for the time this key is held down. Several subsequent operations can be done while `Ctrl` is pressed. However, the mouse mode cannot be changed for an operation that has already been initiated (e.g. a mouse move), since the change takes effect *after* the operation.

Operations in Add mode

A new point is added to the current list of zeros by clicking the left mouse button. The value represented by this new point is the complex value corresponding to the current pointer position, as defined by the coordinate translation parameters (see Section 4.4.2). This value is shown in the status bar. As long as the mouse button is held down, the point can be moved around with the pointer, allowing for precise placement according to the coordinates shown in the status bar. (During this, the cursor will show as a downward arrow.) When the left mouse button is released, the point is "dropped" to its final position. It will appear as a selected point, replacing the previous selection. If, however, the `Shift` key is held down when pressing the mouse button, the point will be added to the current selection.

The pointer can be clicked at the same position for several times, thereby creating multiple zeros (see the remarks about multiple zeros below).

Select mode

Not only the drag-and-drop operations, but also many of the menu functions do not operate on the whole list of points, but only on the current *selection* of zeros. Selected zeros are visually distinguished from normal points by their thicker point shape. There is a large variety of operations to select or unselect single or sets of zeros. All of them (except `Select All` which is issued from the `Edit` menu) are performed by mouse operations in *Select* mode.

In symmetric mode, every selection of a non-real zero affects the conjugate zero in the same way.

Selecting a single point

A zero can be selected by clicking the left mouse button on or close to its image. *Close* means that the distance (in the maximum norm) between the arrow tip and the centre of the point image does not exceed 5 pixels. (Similarly, by *far* we mean that this condition is not satisfied.) If there are several points within that distance from the pointer, the one with minimum distance will be selected. Among several points at the same distance, priority is given to already selected points, and then to the first zero in the order of the root list. The complex coordinates of the selected point are displayed in the status bar.

This selection replaces the previous one, i.e. whichever zeros were previously selected, are automatially unselected.

Adding a point to a selection

If the `Shift` key is held down while clicking on or near an *unselected* point, this point will be selected in addition to the previous selection. This allows for selecting several zeros arbitrarily. (Other ways to select several points at once are explained below.) The same mechanisms as above are used to determine the closest point.

Unselecting a point

If the `Shift` key is held down while clicking on or near a *selected* point, this point will be removed from the selection and reappear as an ordinary point, without affecting the rest of the selection.

Selecting all points

To select all points, choose `Select all` from the `Edit` menu. If the `Shift` button is pressed while choosing `Select all`, the points are *toggled* rather than selected. (Every selected point is unselected and vice versa.)

Unselecting all points

If the left mouse button is clicked far away from all zeros, the complete selection is unselected. Additionally, the complex coordinates of the pointer position are displayed in the status bar.

Clicking on a result point

Result points can never be selected like roots. Therefore, clicking on or near a result point causes the same operation as if there was no zero at all. However, *xzero* will display the exact value of this result point (not only the approximate value according to the pointer position) in the status bar.

Selecting all points in a rectangular area

Pressing the left mouse button far away from all zeros also initiates another operation: If the pointer is moved with the mouse button held down, a rectangular frame opens up. One corner of this rectangle is determined by the cursor position when the button was pressed, whereas the diagonally opposite corner follows the cursor movements. (During the movement, the coordinates of these two points are displayed in the status bar.) When the left button is released, all points inside or on the frame are selected. The number of points thus selected is shown in the status bar.

In symmetric mode, this operation actually opens two frames on opposite sides of the real axis. When the pointer crosses the real axis while opening the frame, the two frames merge into one.

Together with the `Shift` key, the frame can be used to unselect or toggle points. (Every selected point inside the frame is unselected and vice versa.)

Selecting multiple zeros

xzero is capable of handling multiple zeros. However, zeros with the same value are simply drawn on top of each other, so they cannot be distinguished. The selection mechanism works in the following way for multiple zeros, say, with multiplicity k.

- Clicking on or near the position affects only one of the zeros.
- Selecting with a frame selects all k zeros.

When clicking on a multiple zero consisting of both selected and unselected zeros, priority is given to the selected zeros. Thus *xzero* allows the selected zero to be moved or unselected, rather than selecting a second zero at the same position. This is because it will hardly ever be necessary to select between 2 and $k-1$ zeros at the same position. Doing so requires the following trick:

- Select one of the zeros (if not already done), and remove it with `Cut` from the `Edit` menu.
- Select another zero at the same position.

- Hold the `Shift` key, and issue `Paste` from the `Edit` menu.

This restores the deleted zero, leaving both itself and the other selected point selected. By repeating these steps, one can select more than two zeros.

Moving points around

If the pointer is moved after the left mouse button was pressed on or close to a point, the entire selection of points will be dragged around in the window. The offset by which the selection is moved, as well as the new pointer position, are shown in the status bar. The points are finally placed at their new position by releasing the mouse button.

In symmetric mode, points are moved in an unusual way, in order to preserve symmetry: All selected zeros follow the direction of the pointer with respect to the real axis. This is, when the pointer is moved away from the real axis, so do all zeros. Those with positive imaginary part move up, while those with negative imaginary part move down, and conversely, if the mouse is moved towards the real axis. All points however will keep their direction relative to the pointer, even if they cross the real axis. Real points split into a pair of conjugate-complex points and move away to both sides of the real axis. Point pairs that come to lie on the real axis after the movement will turn into *single* real zeros. This description sounds rather complicated, but it will feel straightforward when using it.

Unlike in a normal mouse-click, a point movement does not unselect the current selection. This operation is also indifferent to whether or not the `Shift` key is pressed.

Deleting a selection

Selecting `Clear` from the menu, or pressing the `Backspace` key deletes all selected zeros.

Special remarks on Select Mode operations

The *Select* mode operations are designed to compromise optimal handling with robustness against abusive usage. Yet, the behaviour of *xzero* under some extreme conditions might look weird. These are some of the known special cases:

- Points and frames can be dragged out of the visible drawing area. The operations will be executed on a virtual window the size of the screen. For example, dragging a zero out of the visible area and releasing it at some point outside the window will place it at the position corresponding to that point, and it would appear that the zero just vanished. However, the point still exists, and a coordinate change (e.g. zooming out) would make it visible again. One should avoid such movements, though, since a point dragged out of the window would remain selected, and it could be modified or deleted inadvertently in subsequent operations.

- If a selected point is clicked upon without the `Shift` key, the current selection remains unaffected, just as it would do if the points were moved.
- If a selected point is clicked upon together with the `Shift` key, the point will be deselected only if the pointer is not moved. If the pointer is moved, it is assumed that the intention of this operation was a point movement, not a deselection. Hence the point remains selected.
- The complex coordinates of a pixel are shown with any click of the left mouse button, even if there is no point near the cursor position. This lets one determine the complex coordinates of any position in the window.
- To determine the coordinates of a pixel *without* unselecting the current selection, one needs to hold the `Shift` key while clicking.
- If a zero or a result point is found near the cursor position, the cursor "jumps" to this zero. This is necessary in order to display the exact coordinates of the point in the status bar, and to correctly move points relative to their origin. Since this cursor jump is within the sensitivity range (at most 5 pixels in each direction by default), it should not confuse the user.

4.4.2 Coordinate translations

We were previously speaking of the "complex value corresponding to the current pointer position". This should be restated more precisely: The set of pixels in the drawing area — which are pairs of (possibly negative) integer values — must be mapped in some linear way into the set of complex numbers. This is done simply by specifying two complex numbers: `centre` and `offset`. `centre` is the image of the centre point of the drawing area (the point with the coordinates (*windowwidth* `div 2`, *windowheight* `div 2`)). `offset` is the image of the vector `(1,-1)`, i.e. it specifies in its real (imaginary) part the difference of the real (imaginary) parts of the values of two pixels adjacent in `x` (`y`) direction. Note that, as defined in the X protocol, the y coordinates increase from the top to the bottom of the screen, which is contrary to the usual way of displaying complex values in the Gaussian plane. The negative sign is used to compensate for this. The centre point of the drawing area is used as reference point rather than the top-left point `(0,0)`, because it keeps the central part of the drawing invariant to resizing. Also, zooming into the central part of the drawing (which is most often desired) can be accomplished more easily. Although simple coordinate translations become slightly more complicated.

The values `centre` and `offset` are specified as resources and can be modified. When zooming, `offset` is divided by the zoom factor. In the `Goto` operation, and upon using the scrollbar (see below), the coordinates of `centre` are affected. Finally, resizing the drawing area makes *xzero* redetermine the pixel coordinates of the centre points; the values of `centre` and `offset` remain unaffected.

4.4.3 Scrolling the drawing area

Up to Version 1.1 of *xzero*, the scrollbars at the edges of the drawing area serve the customary purpose of manœuvring on a virtual area that is larger than the visible area, but finite. However, this is inappropriate to model a virtually unlimited area like the complex plane. The user must be allowed to scroll infinitely in any direction. On the other hand, the scroll bars should still indicate the current position relative to some finite area, namely the smallest rectangle that encloses all zeros.

In Version 1.2, the drawing area and the scrollbars will be managed differently to meet these needs. In particular,

- The drawing area will be redrawn after scrolling.
- The sliders can no longer be dragged to the end of the scrollbar.
- The sliders can always be dragged in both directions.
- Both size and position of the sliders will each depend on both the position and size of the displayed part relative to the rectangle containing all zeros.
- If the drawing area contains no zeros, the sliders will be positioned in a central position, independent of shifting and scaling.

- Otherwise, the sliders will behave closely to the normal behaviour, to keep their use simple.

Details on the so modified scrollbars will be given in the manual pages of the upcoming Version 1.2.

4.5 The status bar

The status bar above the drawing area displays information about completed operations. At startup, it contains the version number, and a copyright message.

- For most operations, xzero counts the number of zeros affected by the operation, and displays it on the status bar, when the operation is completed. In symmetric mode, not the number of zeros, but the number of "point pairs" is referred to. A *point pair* is defined as either a single real zero, or a pair of conjugate-complex, nonreal zeros.

- Also, the status bar displays coordinates in ongoing *select* and *move* operations. See the respective operations in Section 4.4.1 on the meaning of the coordinates. All coordinates are shown as complex values in the format `a` $[+|-|\pm]$ `b I`. ($\pm$ is shown when referring to a pair of conjugate points in symmetric mode.)

- For some computations that produce one line of output, the status line serves as output line. The result can be any kind of text, depending on the operation chosen. It may contain values or just ordinary text.

4.6 Menu functions

The following menu functions are described in the order in which they appear in the menu bar. Note that there exist keyboard shortcuts, listed in the menus themselves, and action buttons for many of these functions, to accelerate their use.

4.6.1 File menu

Most of the functions in this menu are self-explaining, standard file operations. For completeness, they are listed in full:

`New` Empties all lists, and the drawing area, except for the points in the history list. (In order to remove them, one has to select `Clear History` from the `Results` menu.)

`Open` Loads a previously generated *xzero* file into the memory, replacing the old root list, and removing the old results.

`Merge` also loads an *xzero* file, but adds its contents to the zeros already in the list.

`Save` saves the current root list.

`Quit` ends the program.

Note that unlike some other applications, *xzero* has no `Close` item, since this function would be identical to `New`.

There exist similar file functions to those above for the result and history lists, to be found in the `Results` menu.

All functions that load or save files show a convenient file selection dialogue, known from many other *X* applications, in which directories can be easily listed and files selected, just by a few mouse clicks.

Whenever unsaved work or results are in danger to be destroyed, as in the `New`, Load and `Quit` operation, the user will be notified and given the opportunity to save the data, or abort the file operation. All file operations can be cancelled before they are physically executed, thus keeping the previous state of *xzero* intact.

The zeros of a loaded or merged list automatically appear as selected points.

4.6.2 Edit menu

The upper half of the `Edit` menu contains the `Cut/Paste/Clear`-operations that are commonly found in window applications:

`Undo` Reverts the last modification done by the user. Only one operation can be reverted, and some functions are not reversible. (Version 1.2 only.)

`Copy` Copies the selected zeros into the *cut buffer*. This clipboard-like buffer maintained by *X* allows exchange of data within and between applications. From *xzero*. zeros can be exported to the same *xzero* session. other *xzero* sessions running on the same terminal. or other applications with a `Paste` function.

`Cut` Same as subsequent execution of `Copy` and `Clear`.

`Paste` Adds the zeros contained in the cut buffer to the existing zeros. The data in the cut buffer may originate from the same or from different *xzero* sessions. or from another application that can specify zeros in a compatible format.

Pasted zeros appear as selected points. replacing the previous selection. Instead. the zeros can also be added to the old selection. by pressing `Shift` when issuing `Paste`.

`Clear` Deletes the selected zeros from the list.

`Select all` Selects the complete root list. including zeros that are outside the visible area. When issued together with `Shift`. the list will be toggled.

The lower half of the menu contains some *xzero*-specific zero-editing functions. They all operate on the current selection of zeros:

`Move/Set...` Version 1.1 only. An alternative to the "drag and drop" way of moving zeros around. Zeros can be more precisely positioned using this method. because the new values can be typed in with the precision supported by the hardware.

`Make Real` Sets the imaginary part of the zeros to 0. This operation is recommended to ensure that zeros are real. Placing them manually close to the real axis might not put them precisely *onto* the real axis. so that they would be considered non-real.

`Make Unit` Divides the coordinates of the zeros by their respective absolute value. thus normalizing them to points on the unit circle.

`Reflect` Replaces the selected zeros by their conjugates. In connection with `Copy` and `Paste`. this provides a way to create symmetric roots. However. xzero does not recognize and treat the polynomial as symmetric. If this is desired. one should use symmetric mode. *instead of* reflecting nonreal zeros.

4.6.3 View menu

(Version 1.1 only)

`Go To...` A window will appear, asking to specify a pair of coordinates. When done, one can either click the `Go To` button to move *to* the specified coordinates (making them the new centre point of the drawing area); or one can move the centre point *by* these coordinates, by clicking the `Offset` button. Either the x or the y coordinate can be left unspecified, in which case only the specified coordinate will be changed.

`Zoom...` A similar window will appear, asking for a pair of zoom factors, which must be positive. If two factors are specified, then the x and y coordinates are zoomed separately. If only one factor is specified, both coordinates are zoomed by the same factor. A factor greater than 1 means that an area around the centre is enlarged (*zoom in*), and a factor greater than 1 means that the area is shrunk towards the centre (*zoom out*).

`Reset Drawing` Scrolls the drawing area so that the centre point of the current root list becomes the centre of the drawing area. After a `Go To` into an area without zeros, this is the easiest way to return to an area where zeros are most likely to be found.

4.6.4 Execute menu

This menu contains the following predefined computations (p is the polynomial defined by the root list):

`Polynomial Plot` Plots p. Ranges can be specified by the user (Version 1.1 only).

`Derivative Zeros` Computes the zeros of p' and returns them as result.

`Wronskian Zeros` Computes the zeros of Wp and returns them as result.

`Play`

`User Defined` Test functions that by default do nothing. However, the user may insert code that performs special computations.

Some of these operations return a new result list that replaces the previous list. If this is the case, and the computation is completed successfully, the opportunity will be given to save the old list to a file. If the history function is active (see the `Results` menu), the old result list will also be copied into the history list.

This menu can be easily extended by new procedures (see Appendix C). For more details about computations in general, see Section 4.9.

4.6.5 Results menu

The first half of the results menu provides some functions operating on the result list, namely `Load`. `Save`. `Copy`. `Paste`. `Clear`. They are equivalents to those for the root list, and need no further explanation. Namely, these are Note that, since root and result files cannot be distinguished from each other, saved (or copied) results can be loaded (or pasted) as roots, and vice versa.

The second half of the menu concerns the result history list. The two radio buttons allow the user to select either one of the history modes, or turn off the history function completely:

- `Display Previous`: Each time a new result is computed or loaded, the old results are transferred into the history list, allowing for comparison to the current (new) results.
- `Display History`: The history list is made accumulative, i.e. each time new results are computed or loaded, the previous results are *appended* to the history list. It will appear that the zeros in the history list (if computed with the same command on slightly modified versions of the root list) describe "tracks" or trajectories on the complex plane.
- Turning off the history option (by clicking on the currently active radio button) removes the result history from the display. Yet it does *not* clear the list. If the history option is turned on again, the old history list will reappear. This way, the accumulation of the history list can be interrupted (e.g. for intermediary computations) and resumed later on.
- `Clear History`: Clears the result history list physically. It works in all history modes. It is recommended that the history list be cleared if it is no longer needed, for displaying long lists is a time-consuming operation, especially on slow terminals.

- `Save History`: Save the current history list.

4.6.6 Options menu

`Restart Maple` Resets the computation engine of *xzero*. For the use of this function, see Section 4.9.1. *Problems with the computation.*

`Symmetric` Toggles between normal and symmetric mode. The symmetric mode is described thoroughly in Section 4.1.1.

4.6.7 Help menu

This menu brings up a list of functions *xzero* provides, with short help texts on how to use them.

4.7 The action buttons

(Version 1.1 only)

These icon buttons to the right of the drawing area provide an easier way to use some of the functions in the menu bar. These functions have already been described with the menu they belong to.

4.8 xzero file and data structures

We will now shed some light on the structure of files and lists in *xzero*, as far as they are of concern to users.

4.8.1 File format

Zero lists are stored externally in a consistent way, regardless of their nature and of the storage medium, which can be a file or a memory area called the *cut buffer* (see the `Copy` function). For simplicity, we call any stored list of zeros a *file*.

Zero files are stored in the following format:

- They are ASCII text files.
- Each zero occupies a separate line of the file.
- Its complex value is represented as two floating-point numbers, separated by white space.

This relatively simple format makes it easy to edit zero lists manually with text editors, and to load them into mathematical packages. Conversely, one can *create* or modify data files intended for use with *xzero*. Note however that the following conventions are to be adhered to, if a file is to be successfully loaded into *xzero*:

- Real values must be displayed as two numbers, with the imaginary part set to zero.
- The numbers can be in any numeric format: integer, fixed point or floating point.
- The line length is limited to 80 characters.
- No text is allowed in any line. However, text at the end of a line containing two valid floating point numbers will be ignored without an error message.
- When two numbers in a line are successfully read, the rest of the line is ignored. Therefore, comments at the end of a line are allowed, but no more than two numbers, specifying one complex value, should be stored in one line.
- Empty lines are allowed and will be ignored.

Files that don't match these specifications will cause error messages when loading them into *xzero*. (see Section 4.10.1).

Due to the consistent file format, result lists can be re-loaded into *xzero* as root lists. This lets one do multi-step or recursive computations, e.g. for obtaining higher derivatives.

4.8.2 Internal list structure

Internally, roots and results are stored in two independent linked lists. Without stepping into the details of linked lists, we remark that this way of storing defines a total ordering on the list, equivalent to the order in which they were created. The order is preserved when zeros are stored in a file, and when they are loaded back into a list. This may be relevant to computations on zeros in which the order of the list plays an essential role.

4.9 Computation Strategy

So far it has not been necessary to understand how *xzero* accomplishes the computations. However, after using *xzero* extensively, one will almost certainly encounter problems with the computation. This will usually happen in the following situations:

- A computation has been initiated, that, with a large number of zeros in the root list, takes too long, and should be interrupted.
- After incorrect parameter specifications, an error occurs within the computation.
- One wants to write one's own computing routines, or modify existing routines.

xzero does not have its own computation engine. It relies on a powerful symbolic computation package called *Maple*. *Maple* is called by *xzero* at the beginning of the program as an independent process, and run in batch mode. This is, *Maple* receives its input — the commands that issue the computation procedures and contain the data — from *xzero*, and executes these commands in the background, invisible to the user[3].

This approach was chosen for a good reason: Coding one's own subroutines for determining zeros of polynomials without the aid of a mathematical package is quite troublesome: a good algorithm has to be selected, and care to be taken to avoid programming mistakes and control numerical errors. On the other hand, there exist powerful symbolic computation

[3] Of course one can, as with any running process, get information about *Maple*'s status, using the top command in a UNIX shell. This might sometimes be useful, e.g. to find out if *Maple* has unexpectedly quit.

packages such as *Maple* and *Mathematica*[4] in which these algorithms are already implemented. Therefore, it is a lot easier to write and modify computation routines with these packages. The only disadvantages of having a symbolic package do the mathematics are:

- These packages are universal, and not optimized to a special task, so in general they are comparatively slow.
- *Maple* provides only a primitive interface to other UNIX applications. This makes the communication between *Maple* and *xzero* difficult, and sometimes unreliable, as we will show in the next section.

During the computation, *xzero* is waiting for the results to be returned by *Maple*. A window will appear, notifying the user of the ongoing computation. This window contains a `cancel` button that lets the user abort the computation at any time. If this button is pressed, or when all results have been received, *xzero* will return to normal operation. Note that, even if *Maple* does not return any results, *xzero* will wait for *Maple* to finish all its computations.

4.9.1 Problems with the computation

Maple, as an independent UNIX process, gives *xzero* only two ways of controlling its behaviour, namely feeding it with input, and killing it. Interrupting a computation in particular would be useful, but it is only possible for the cost of killing *Maple* completely, and restarting it before the next computation.

As a well-known fact in computing, it cannot be predicted if a certain operation will yield its result after an acceptable amount of time, or even if it will terminate at all. Only the user can decide what computing time will be acceptable for which type of computation, given a certain number of zeros to be computed. If one decides that the computation gets too lengthy, one can press the cancel button. This will terminate *xzero*'s waiting for results, and then kill and restart *Maple*.

In open systems like UNIX, virtually every process running under a person's account is allowed to kill every other process under the same account, and can also be killed by the

[4] Forthcoming versions of *xzero* will have a modified interface that will allow one to use *Mathematica* instead of *Maple*.

person itself. This kind of bloodshed doesn't normally happen. But there are known cases when *Maple* dies accidentally, the most prominent being a recursive symbolic reference in the code. *xzero* is capable of recognizing if *Maple* is no longer available, and restarts it if necessary. And finally, one can restart *Maple* manually, if one considers it necessary[5].

Another problem arises from the way in which *Maple*'s results are transmitted back into the program. A correct transmission is accomplished through control messages that are sent before and after the result list. If *Maple* terminates because of an error, *xzero* will not receive the terminating control message. *xzero* cannot distinguish between this and an ongoing computation. However, the user will most likely have been warned in advance by a *Maple* warning or error message. When in doubt, one should check with a UNIX tool like `top`, if the *Maple* process is still computing (that is typically when it uses a large percentage of CPU activity). If *Maple* is no longer computing, or doesn't show up in the process list at all, one should interrupt the computation, restart Maple, and re-issue the same computation, if desired.

4.10 Errors, Warnings, and Bugs

4.10.1 Error messages

Two different types of error messages may be displayed when running *xzero*: *system errors* or *user errors*.

System errors

System errors occur if system limitations or configurations prohibit *xzero* to function normally, especially during startup, when xzero tries to allocate the resources it needs. *xzero* will display an error message on `stderr` and then terminate. In general, this means that xzero will not run on this system or terminal, or that settings have to be reconfigured. The program documentation of xzero contains a list of error messages, along with ways to avoid the respective error. Other types of system errors, not listed in the documentation, originate

[5] A useful application of this is garbage collection: In the course of its computations, *Maple* allocates memory. Through excessive computations, the amount of memory allocated may exceed the available main memory, and UNIX will start swapping to the disk, which drastically slows down computations. Restarting *Maple* frees the memory previously allocated and helps avoid this problem.

from the *X* protocol and may indicate incompatibilities of *xzero* with the specific version of *X* on the system used. Some help in case of such an error may be found in any good *X* window user's manual. e.g. [21].

When *xzero* terminates due to a system error. one should type

```
destroy
```

on the next UNIX prompt. This frees up internal structures which at this point may still be allocated.

User errors

These error messages are somewhat expected. They occur in case of bad user input. or operations that cannot be issued in certain situations. The error messages will appear in a separate message box on the screen. and they never cause xzero to terminate. Depending on the severeness of the error. one is given some or all of the following options:

- cancel the current operation.
- re-issue the same operation.
- continue as if no error had occurred.

The possible choices for the respective error are indicated by the sensitive buttons in the error message box (**abort**. **retry**. **ignore**. respectively).

`Zero value not permissible.`

The user tried to input a value that would later in the program lead to a division by zero. In particular. this message occurs when the user attempts to zoom by a factor of 0. or by a very small value that evaluates to 0.

`List is empty.`

An operation is attempted that is senseless if there are no zeros defined in the list. These operations are: Saving a file. copy. and computing results.

`Bad line format.`

A piece of data is read into *xzero* that doesn't match the correct specification for complex zeros, as described in Section 4.8.1. This piece of data can either be a file, a list of zeros pasted from other applications, or a result list received from *Maple*. The erroneous line is displayed on `stderr`, so that the user can find the source of the error.

If the `Ignore` option is chosen, the line as a whole will be skipped, and the reading process resumed. Otherwise, the operation will be cancelled. It is in the nature of some read operations that repeating them will not be successful. Therefore, the `Retry` option is not always available.

`An error occurred in Maple.`

During *Maple*'s command execution, an error occurred, recognized by the typical *Maple* message `Error (in ...)`. *xzero* doesn't have control over the correctness of *Maple*'s computations , so these errors cannot be prevented. The user should know about possible sources of error in the *Maple* code (e.g. possibility of division by zero). *Maple*'s error message, which is printed out on `stderr`, may be of some help in finding the source of the error.

`Unexpected message type.`

This error hardly ever happens. In the message transmission between *Maple* and *xzero*, different types of messages are used for data, control, and error messages. (The type of message is simply an unsigned byte-long integer value.) This error indicates that messages did not occur in the right order. It is possible, though unlikely, that the message queue is used by a different UNIX application. Previous runs of *xzero* that terminate unexpectedly (e.g. without closing the message queue) may also leave some messages in the queue, which may confuse *xzero*. Retrying the computation will most likely be successful, since the message queue is cleared after an error occurred.

`Terminated by user.`

The *Maple* computations have been interrupted by the user clicking the cancel button in the waiting message. Upon `Retry`, the computations may be repeated. `Ignore` will not continue the computation, but any results that have already been obtained will be made available. (In the builtin computations, all results are transmitted after all computations have been done: the interrupt occurs most likely at a time when no results have been returned. `Ignore` will not be any different from `Abort`, then.)

`Error in` <*file*> `operation:` <...>

During file operations, many system-defined file errors are possible, e.g. `No such file or directory`, `Permission denied`, `Disk space full`. The above message specifies the *file* operation that failed, and a string that describes the type of error. This string is the same as generated in response to a UNIX command and defined in the standard **C** error library, so it can be looked up in the UNIX manual page `intro(2)`, if it is not self-explaining.

`Cut buffer full.`

An unusually large number (>200) of zeros was attempted to be cut or copied. This is actually a warning message. Therefore, no `Retry` or `Ignore` option is given. The cut buffer contains the zeros that were successfully written into it, and in the case of a `cut` operation, these zeros are deleted. (To get them back, one can simply paste them.) To avoid this error, a large list should be copied in several smaller parts.

4.10.2 Warnings

By default, *xzero* warns a user in every instance when (s)he is about to lose essential parts of their work. Namely, the following three warning messages may appear:

`Zeros have been modified.`

This message appears when the root list is about to be cleared or replaced in a `New` or `Load` operation (but not with `Clear`), and has not been saved after it was last modified. (Selections don't count as modifications.) The user may either save the list now, discard the changes, or cancel the ongoing operation, thus leaving *all* lists in their previous state.

`Current results are unsaved.`

Similar to the above message, this appears when an unsaved result list is to be replaced by a file, a new computation, or to be deleted through the `New` operation. The user is given the same choices as above.

`File` <*filename*> `exists.`

This message is shown whenever a list of zeros is written into an existing file, unless:

- the list to be saved was previously given the same file name (e.g. when it was loaded or last saved). Equality is tested by string comparison, so *xzero* doesn't recognize if two files with different names are physically identical, due to path specifications or symbolic links. This may sometimes result in a groundless warning.
- the list is newly created, and is to be saved under the default filename (`unnamed` or `unnamed.res`).
- the list is a result or history list, and it is saved under the same name as the root list, plus default extension (i.e. *filename*`.res` or *filename*`.hist`).

The user has the choice to either overwrite the old file, or cancel the `Save` operation. In the latter case, *xzero* will return to the file selection dialogue, allowing the user to save under a different name or abort the whole `Save` operation.

As a general rule, *xzero* warns too often rather than not warning when it ought to. However, some of these messages can become very bothersome, especially in "trial-and-error" runs, when results are generally not to be saved. Therefore, displaying each of the above warning messages can be turned off separately for the rest of the *xzero* session, by holding the `Ctrl` key while clicking `Discard` or (in the `File exists` warning) `Cancel`. Also, the application resources `confirmSaveEnable`, `confirmResultsEnable`, and `fileExistEnable` can be set to `False`, disabling all warning messages for the entire *xzero* session. However, the warning messages should be only disabled if they are definitely not needed. There is no way to re-enable them from within a running *xzero* session.

Another type of warning are *Maple* warnings, originating from computations. (They may also occur in the startup phase of *xzero*.) *Maple* warnings are simply written to `stderr`. The user should check for such messages in case of unexpected results.

The most prominent *Maple* warning is `Empty plot`, by which *Maple* notifies the user of insufficient variable specifications in a plot. This may be due to a mistake in user-defined functions, or to incorrect range specifications by the user.

4.10.3 Known bugs in xzero

During the testing stage, much care has been given to discovering and removing bugs and instabilities of *xzero*. Yet it is only too likely that *xzero* will contribute to the painful experience users frequently get from using *X* applications. Major *X* applications are so complex and flexible that they can never be completely debugged on all platforms, in all thinkable configurations and for all imaginable situations. Usually, in some extreme situations (where *extreme* depends on the program and how it is written) they crash without warning, leaving nothing but a coredump behind. Hence, a user should not experiment with extreme configurations of *xzero* (just as with any other *X* application). Here is some advice on how to avoid problems with *xzero*:

- Never zoom to extremely large or small intervals. This may cause numerical overflows or large rounding errors.
- Don't make the drawing window too small, or too large.

  Large windows should not cause any trouble, but they might affect the performance of xzero on slow terminals, when the entire window needs to be redrawn frequently.
- Don't hit too many keys or mouse buttons at the same time.

  In doing this, several operations may be initiated simultaneously, causing inconsistencies in updating the lists.
- Avoid extensive mouse movements when moving objects around.

  Redrawing the moved objects takes a considerable amount of time. Especially on slow terminals, it will appear as if the points are lagging behind the cursor. This will also slow down other drawings on the terminal.
- Don't configure resources unless you are sure about what you are doing.

  One cannot destroy the program with badly configured resources, but some of the *xzero* operations may be disabled, or some of the drawings may become invisible. In any case. a backup copy of the default resource file should be kept. It is not guaranteed that *xzero* runs correctly with any resource setting other than the default — although e.g. changes in colour or text specifications should prove harmless.

Chapter 5

Some experimental results

5.1 Another application

The *xzero* package has been used extensively to verify the results in Chapter 2, and to examine its various special cases. We will now show its use in another application to give some experimental evidence for Conjecture 3.11.

The central idea is the same as in Chapter 2, namely, examining the loci of $\frac{p'}{p}$. This time however, we take the union of the two loci of arguments 0 and π, joined by the zeros and poles of $\frac{p'}{p}(z)$. We write this set as $L_R\left(\frac{p'}{p}\right)$ and refer to it as the *real locus of* $\frac{p'}{p}$. So $L_R\left(\frac{p'}{p}\right)$ is the inverse image with respect to $\frac{p'}{p}$ of the extended real axis, and its points can be determined by the equation $\Im\left(\frac{p'}{p}(z)\right) = 0$. This equation can be rewritten as

$$\Im(p'(z)\overline{p(z)}) = -\Re(p'(z))\Im(p(z)) + \Re(p(z))\Im(p'(z)) = 0$$

and further, on substituting $z = x + yi$, it can be written as a polynomial in x and y, each variable occurring with degree at most $2n-1$. Hence, $L_R\left(\frac{p'}{p}\right)$ is an algebraic curve of degree at most $4n-2$. (This fact is true for all "pairs" of loci of the form $L_c\left(\frac{p'}{p}\right) \cup L_{c\pm\pi}\left(\frac{p'}{p}\right)$.)

From the results about loci of $\frac{p'}{p}$ in Chapter 2, we know (or derive easily) the following properties of $L_R\left(\frac{p'}{p}\right)$:

- $L_R\left(\frac{p'}{p}\right)$ is symmetric to the real axis.

- The real axis is part of $L_R\left(\frac{p'}{p}\right)$.
- All arcs of $L_R\left(\frac{p'}{p}\right)$ besides the real axis (henceforth called the *nonreal arcs*) are finite.
- These arcs lie completely on the closed Jensen disks of the nonreal zeros of p.
- Every nonreal zero of p lies on exactly one arc.
- Every nonreal critical zero of order 1 lies on k arcs, where k is its multiplicity.
- If z is a nonreal zero of Wp of multiplicity k, and $\frac{p'}{p}(z)$ is real, then z lies on $k+1$ arcs.
- The nonreal arcs consist of components which are either closed (we will call these *closed arcs*), and components which begin and end on the real axis (*open arcs*). If an arc passes through a multiple critical zero, or to a zero of Wp, it has to be suitably continued (by taking the adjacent arc to the left or to the right).
- The points on the real axis where open arcs originate are exactly the real zeros of Wp. At a real zero of Wp with multiplicity k, exactly $2k$ nonreal arcs originate.

The last point is the most important one, because it lets one express the number of real zeros of Wp in terms of the open arcs of $L_R\left(\frac{p'}{p}\right)$. Given this relation, Conjecture 3.11 is equivalent to saying that the number of nonreal zeros of p is greater or equal to the number of open arcs of $L_R\left(\frac{p'}{p}\right)$. One is led to suggest the following stronger versions of the conjecture, all of which will prove to be false:

- The number of nonreal zeros *on the open arcs* is greater than or equal to the number of open arcs of $L_R\left(\frac{p'}{p}\right)$.
- The total number of nonreal zeros is greater than or equal to the *total* number of arcs in H_+ and H_-.
- The number of nonreal zeros, *not* counting multiplicities, is greater or equal to the number of open arcs.

Instead of a complete case distinction, we will give a number of examples which illustrate the typical cases that occurred in our experiments. We will also state which cases satisfy

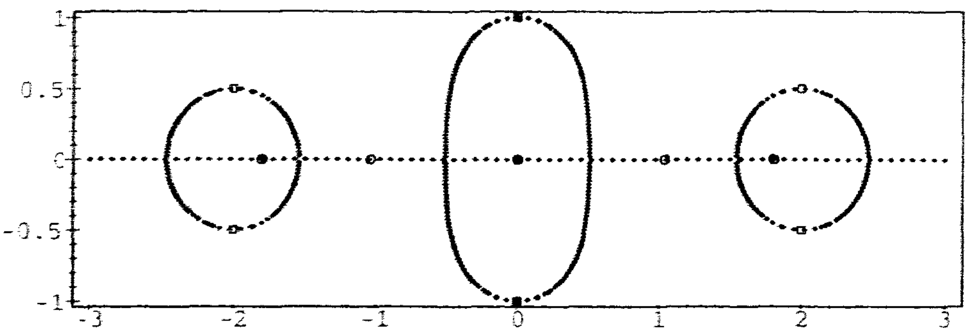

Figure 5.1: *The real locus of* $\frac{p'}{p}$. *for* $p(z) = (z^2+1)(z^2+4z+4.25)(z^2-4z+4.25)$

the above three conjectures. and relate these cases to theorems presented in Chapter 3. whenever possible.

Case 1: (see Figure 5.1) All nonreal arcs are open. They connect pairs of adjacent real zeros of Wp. and contain one nonreal zero each.

One can show that this happens exactly if all critical points of p are real. In particular. this includes all cases in which Theorem 3.19 holds for all nonreal zeros of p.

Case 2: (see Figure 5.2) All nonreal arcs are open. They connect pairs of adjacent real zeros. and they contain one nonreal zero and one nonreal critical zero each.

This happens if p has no Fourier critical zero of order 1. and the real parts of the nonreal zeros are sufficiently well-spaced.

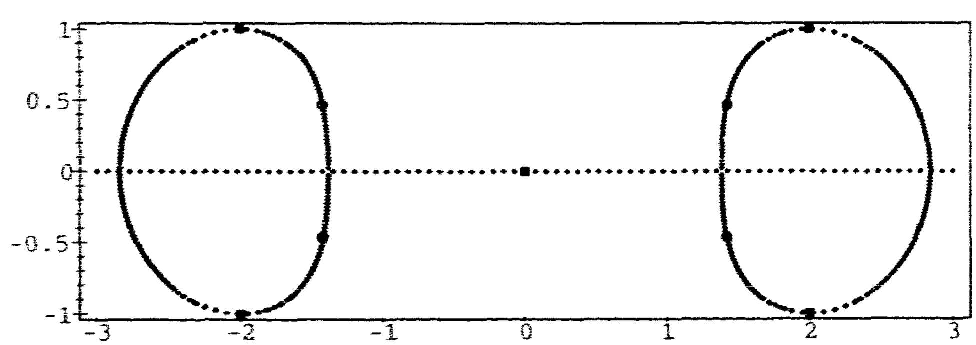

Figure 5.2: *The real locus of* $\frac{p'}{p}$. *for* $p(z) = z(z^2+4z+5)(z^2-4z+5)$

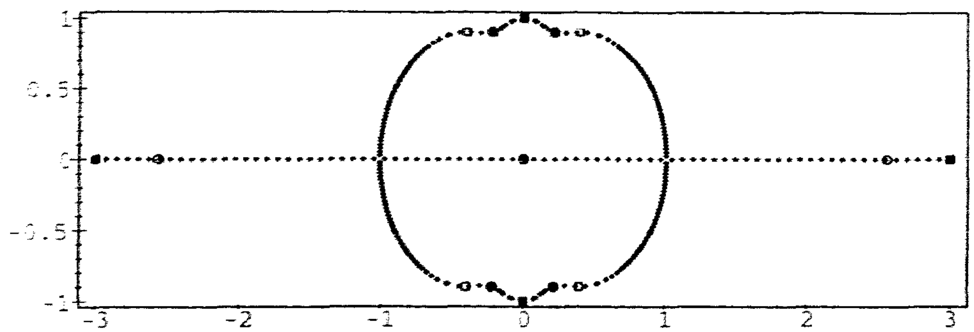

Figure 5.3: *The real locus of* $\frac{p'}{p}$*, for* $p(z) = (z-3)(z+3)(z^2+1)(z^2+0.8z+0.97)(z^2-0.8z+0.97)$

Case 3: (see Figure 5.3) There exists only one pair of arcs, which are open and contain all nonreal zeros and critical zeros of p.

This includes some, but not all cases of Theorem 3.20. (In the general setting of this theorem, $L_R\left(\frac{p'}{p}\right)$ may also have closed arcs.)

Case 4: (see Figure 5.4) All arcs are closed; they contain at least one real and one nonreal zero each.

This case applies if and only if Wp has no real zeros at all; in particular, this includes all cases in which Theorem 3.18 holds for all nonreal zeros of p. It is obvious that every closed arc contains at least one zero and one critical zero of p; otherwise, the maximum principle, applied either to $\frac{p'}{p}$ or to $\frac{p}{p'}$, would imply that $\frac{p'}{p}$ is constant in the interior of the arc.

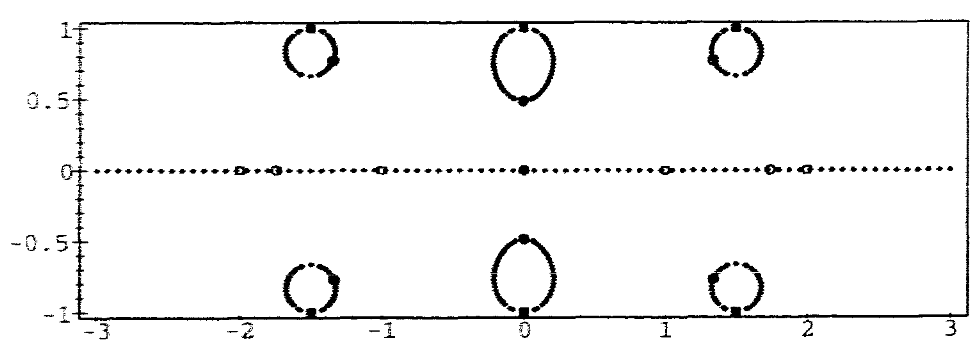

Figure 5.4: *The real locus of* $\frac{p'}{p}$*, for* $p(z) = (z^2-1)(z^2-4)(z^2+1)(z^2+0.75z+3.25)(z^2-0.75z+3.25)$

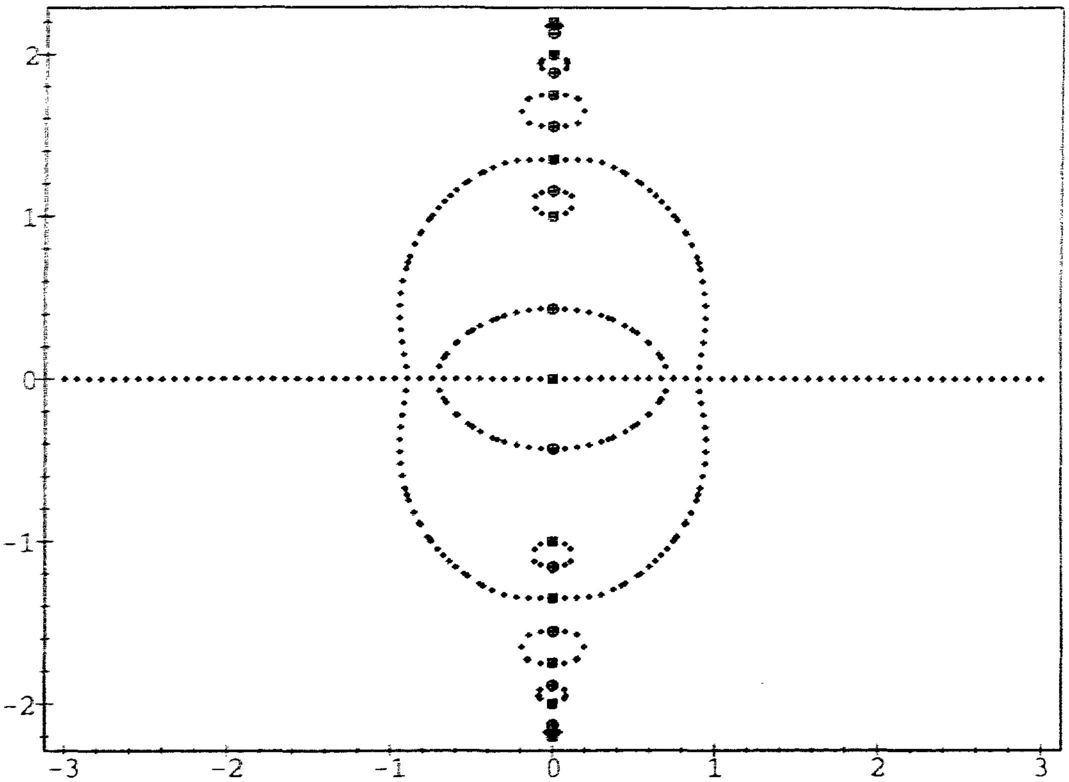

Figure 5.5: *The real locus of $\frac{p'}{p}$, for* $p(z) = z(z^2+1)(z^2+1.8225)(z^2+3.0625)(z^2+4)(z^2+4.84)$

Case 5: (see Figure 5.5) There exists an open arc which contains a nonreal critical zero, but no nonreal zero of p.

One can show that this arc must be contained in the interior of another arc of $L_R\left(\frac{p'}{p}\right)$. This example shows five pairs of nonreal zeros, but six pairs of nonreal arcs; only one pair of zeros lies on an open arc, while there are two such arcs. This shows that the first two conjectures above are false.

A counterexample for the third conjecture can be obtained similarly, by grouping all five pairs of nonreal zeros together at, say, $\pm i$.

Case 6: There exists an open arc that contains no nonreal zeros or critical zeros of p at all.

No example has been found for this case. However, Figure 5.6 shows that this case is possible for the locus $L_R(r)$ of an arbitrary rational function. We conjecture that a similar example can be found for a logarithmic derivative as well, but involving a

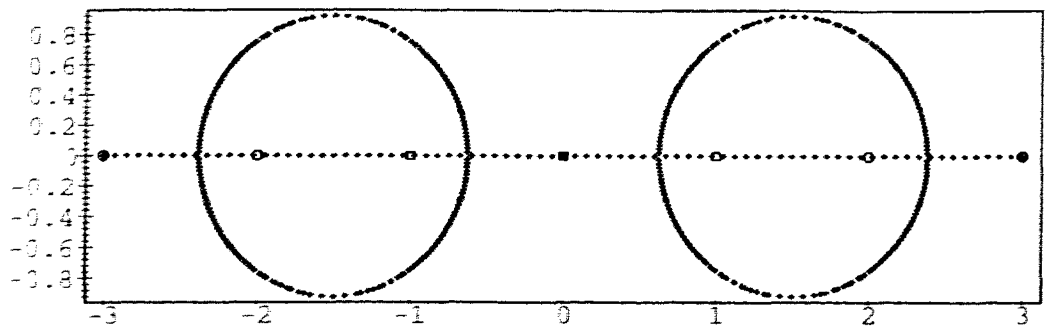

Figure 5.6: *The real locus of* $r(z) = \frac{z(z^2-1)}{(z^2-1)(z^2-9)}$

large number of nonreal critical zeros. If such an example is found, it can most likely be modified to give another counterexample to one or all of the three conjectures above.

We note without further consideration that mixtures of the cases above are possible. Also, there exist limiting cases between the above cases. These limiting cases always have one of the following properties:

- $L_R\left(\frac{p'}{p}\right)$ contains a nonreal zero of Wp.
- A real zero of Wp is multiple.

The Cases 5 and 6 (if it exists) of "zero-free open arcs" are undoubtedly the most intriguing ones, for they are the only cases from which counterexamples to Conjecture 3.11 could arise. They need to be further classified and studied. It would be helpful to find a lower bound for the number, and mutual distance, of nonreal zeros necessary to procure zero-free arcs, for it seems that far more nonreal zeros are placed on closed arcs, or as extra zeros on existing open arcs, than arcs become zero-free.

5.2 Numerical errors

Rounding errors are a crucial and often underestimated factor in numerical packages. This affects *xzero* as well, since *xzero* uses a numerical algorithm for finding the zeros of a given polynomial. And in fact, numerical errors in intolerable dimensions have been found in some instances, while working with *xzero*. For instance, consider the polynomial with zeros at $-0.8 \pm i, -0.6 \pm i, \ldots, 0.8 \pm i$. Compute the zeros of Wp using a precision of 15 digits throughout the computation. The result is shown (as circles) in Figure 5.7 along with the zeros of p (boxes).

Now shift the zeros uniformly by $+1.5$ in real direction. By Lemma 3.2, the zeros of Wp should be shifted by the same amount, but otherwise unchanged. However, the actual result computed by *xzero* differs considerably, as shown in Figure 5.8.

Upon further investigations, it was observed that polynomials with multiple zeros, or with clusters of zeros at small distances from each other, relative to their absolute values, are the most ill-conditioned and yield the largest numerical errors. The loss of digits of precision in the zero finding algorithm turned out to be proportional to the degree of the polynomial to be solved, at a rate of up to 1 digit per degree increment.

Theoretically, these numerical errors could be prevented by computing to a sufficiently high number of digits. However, this slows down the computation drastically. Specifically in ill-conditioned examples, the computation time seemed to increase at least with the square of the number of digits of precision. In fact, in some instances the algorithm didn't seem to terminate at all, if the precision was chosen too large[1]. In an interactive application such as *xzero*, large response times are generally unacceptable. Therefore, the numerical precision should be just high enough to get reliable results in the range of the drawing resolution, but low enough to give a fast response in well-conditioned cases. It would be desirable to have an algorithm that "recognizes" the necessary precision to compute with, in order to get results within a given "output precision"

Another observation concerns the coefficients of a polynomial constructed from its roots. If many roots have absolute values far smaller, or far larger than 1, then the coefficients are of widely different decimal ranges, which results in loss of precision in subsequent steps of the computation. This might in fact be one of the reasons for the numerical errors observed in Figure 5.8. In this light it seems questionable if the presently used method of computing the coefficients of the polynomial, namely, differentiating, and determining the zeros of the derivative or Wronskian is a suitable approach. It seems worth considering algorithms which make use of the Formulae 2.8 and 2.9, without bringing the numerator into standard form.

[1] Or it terminated with an error message saying that the polynomial to solve was too ill-conditioned.

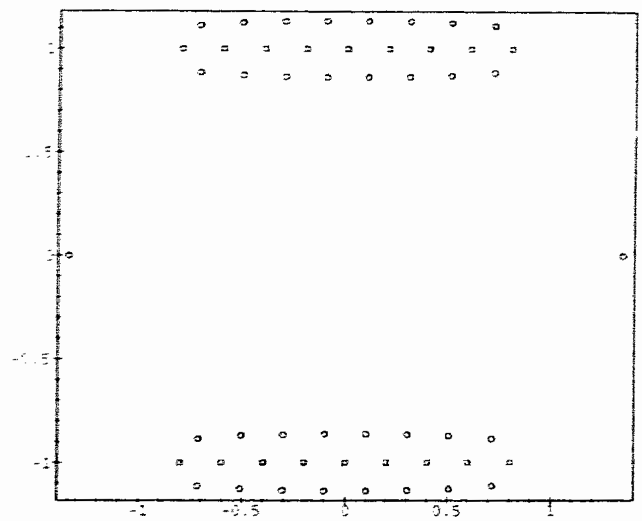

Figure 5.7: *The zeros of* p *and* Wp, *where* $p(z)$ *is a polynomial with zeros at* $0.2k \pm i, k = -4, \dots, 4$

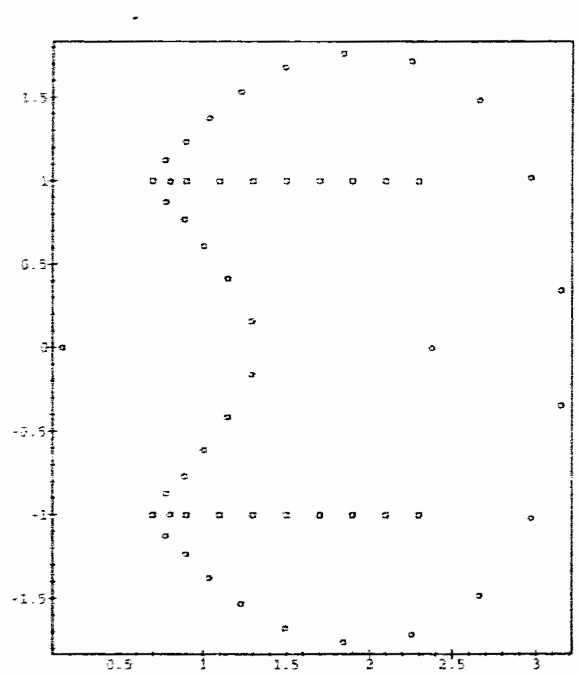

Figure 5.8: *The zeros of* q *and* Wq, *where* $q(z) = p(z - 1.5)$

Together with good initial guesses[2] which ensure that all zeros are found, such an algorithm would have the advantage of faster convergence and greater robustness against numerical instabilities.

The graphs presented in this chapter and in Chapter 2 have been computed to a precision of 30 digits. None of these examples involved a polynomial of degree greater than 11, so the results shown are guaranteed to be exact within the graphing precision.

[2]The results in Chapters 2 and 3 should prove very helpful here, because they provide geometrical constraints for the location of the zeros of p' and Wp in terms of the zeros of p.

Chapter 6

Conclusions and Outlook

The proof of Theorem 2.17 leaves many open questions for further investigations. We will mention some of them briefly:

- Can the theorem be generalized to a larger class of analytic functions than polynomials?

- For many polynomials without multiple zeros or critical points. Theorem 2.17 does not give a single relationship between the d pairs of nonreal zeros and Fourier critical zeros. but rather a family of finitely many bijections. This sounds trivial. given that the possible number of bijections between finite sets — here $d!$ — is always finite. However. the number of different bijections arising from Theorem 2.17 is usually much smaller than this. (One obtains from Corollary 2.37 that it is at most one greater than the number of different values of $c \in (0, \pi)$ so that $S_c(p)$ contains a zero of Wp or $W(p^{(k)})$ for some positive integer k.) Is it possible to determine. just from the location of the zeros and Fourier critical zeros of p. which bijections are possible and which ones are not?

- If this in general cannot be determined. is it at least possible to determine the possible bijections between $Z(p)$ and $C_1(p)$ which arise from Theorem 2.33. just from the location of all zeros. critical zeros of order 1. and perhaps the zeros of Wp?

- Can more. or sharper. geometrical results in the flavour of those in Section 2.10 be obtained from the proof of Theorem 2.17?

- The arcs of $S_c(p)$ turned out to be very short, especially those for $c = \pi/2$, which were examined the most. Is it possible to determine a maximum distance between a nonreal zero and the critical zero it is connected to, depending on the value of this zero and the surrounding zeros?

The following conjecture could be an answer to a particular case of the last question, if it could be proved:

Conjecture 6.1 *(Ilieff-Sendov-Conjecture) Let p be a (complex) polynomial all roots of which lie inside or on the unit circle. Then inside or on a circle of radius 1 around each of the roots of p, there lies at least one zero of p'.*

Of course this result is sharp. The extreme case is attained in the polynomials $c(z^n - 1)$, where n is a positive integer, and c a complex constant with $|c| = 1$. There are many partial results towards this conjecture, and it is shown for circles around roots which lie *on* the unit circle. (See [18] for a thorough treatment of this conjecture.)

We also observed that the zeros of Wp contain much information about the appearance of the loci of $\frac{p'}{p}$. For this reason, they should be studied further. As outlined in Chapter 5, there are some unresolved cases in a possible proof of Conjecture 3.11 which should be considered. From the experiments done in this context, it seems possible that this conjecture can be further generalized:

Let d be the number of pairs of nonreal zeros of p, and q be the number of points $z = x + yi$, where $Wp(z) = 0$ and $\operatorname{sgn}\left(\Im\left(\frac{p'}{p}(z)\right)\right) = \operatorname{sgn} y$ (these are the zeros of Wp, which must lie inside the Jensen circles of p. Is there an upper bound, other than the trivial one, for the value of q in terms of d? Do we always have $q \leq 2d$? Is there also a lower bound for q in terms of d?

The next discussion of possible generalizations concerns the *xzero* package. This package in its basic design is general enough to allow experiments with polynomials other than studying the derivative, the logarithmic derivative, and the Wronskian. For instance, the Ilieff-Sendov-Conjecture could be examined experimentally with *xzero*. Basically, one just needs to write suitable *Maple* code that verifies the conjecture for a given polynomial.

Numerical problems and the need for better algorithms to resolve them have already been discussed in Section 5.2.

It need not be mentioned that improvements to the user interface, and versions of *xzero* running on different platforms and supporting different computation packages are necessary, if the package is to be more widely distributed.

A desirable extension of *xzero* would be to maintain several polynomials at once. This would, with a few tricks, open the field of rational functions and polynomial arithmetic to *xzero*, with countlessly many other applications.

Appendix A

Gauss' original letters

This chapter contains translations of five texts. the original (German) versions of which can be found in [7. 10. 11. 12. 13]. The first piece is an announcement and summary of [6] by Gauss. published in the "*Göttingische Gelehrte Anzeigen*". the University of Göttingen newsletter. Texts 2. through 4. are letters out of the correspondence between Gauss and his friend Schumacher. dealing. though superficially. with the problem he mentioned in the announcement. Texts No. 5. is a comment by Alfred Loewy [13]. revealing mistakes Gauss made in his announcement. and giving a more precise specification of the problem.

1. **Announcement. Göttingische Gelehrte Anzeigen. 1833. February 25**

   Analyse des équations déterminées par Fourier. Première partie. Paris 1831. Chez Firmin Didot frères.

   According to the author's sketches. this work was proposed to consist of two volumes. The first one was to contain. besides a general survey of the whole. the first two chapters. whereas the second one was to contain the remaining five chapters. The author passed away from the sciences when printing the work had just begun. Meanwhile. the whole first section. as we obtain it here through Mr. Navier. had been worked out almost completely. while the rest was still far away from completion. This is the more regrettable. as it appears from the general survey that many interesting investigations were meant to be included in it. Here. we must not speculate what we could have expected if the author had had a longer life. and shall restrict ourselves to an announcement of what we have actually obtained.

The aim of the work is to turn the method of solving the definite algebraic equations[1] into a most reliable and easy one: namely, determining the number of real and imaginary roots with full accuracy, bounding each of them into fixed limits and describing an algorithm for a step-by-step approximation to their values with arbitrary exactness. The fundamental basis for all of this is a theorem due to the author himself. It can be regarded as a successful generalization of Descartes' theorem (usually named after Harriot): in principle, it consists of the following: Let X be an ordered[2], entire, algebraic function in x (with only real coefficients); further, let a, a' be two arbitrary different real values such that $a' - a$ is positive. By substituting $x = y + a$, $x = y + a'$, X turn into the (likewise ordered) functions Y, Y', in which the coefficients have g, g' sequences of signs[3], respectively: the exceptional case in which one or more coefficients vanish will be disregarded here for brevity. Given this, the equation $X = 0$ cannot have more than $g' - g$ real roots between $x = a$ and $x = a'$; more exactly, if the number of real roots between those two bounding values is λ, then $g' - g - \lambda$ must be either zero or even and positive. Hence, if $g' - g$ (which cannot be negative) is zero, there exists no real root at all; if $g' - g = 1$, then the bounds include exactly one real root in between; finally, if $g' - g = 2$, then as of now it is not clear whether there are two roots between the limits or none. Instead of two bounds a, a', we can handle a greater number of bounds similarly; we can choose them in such a way that, first, the smallest [bound] corresponds to a function Y having only changes of signs, and the greatest to a function only having sequences of signs; therefore, if the numbers corresponding to the sequences are called g, g', g'', g''' and so on, in this order, the first of these numbers will be 0 and the last one be equal to the degree of the equation; second, that every single difference $g' - g$, $g'' - g'$, $g''' - g''$ etc. will be equal to either 1 or 2. In this manner, all real roots will be enclosed in bounds, so that in every interval there may be either one or two roots. How in the last case it can be methodically determined (by further narrowing down the limits, if necessary) if the two roots indeed exist, or if they are missing, cannot be worked out here more precisely, for lack of space. We just note that every time the latter occurs, there must be an intermediary value between the bounds where one coefficient is missing[4] in Y before

[1] Factorizing a polynomial.
[2] By the degree of their terms.
[3] Occurrences of the same sign in two subsequent coefficients.
[4] Zero.

the last one, while the preceding and the following [coefficient] must have the same sign. Fourier calls these points *critical*. Hence, each critical point leads to two missing real roots. However, we can not agree with Fourier saying that every two missing real roots become imaginary, because this could give rise to a misconception. It is indeed true that the equation $X = 0$ *altogether* has as many pairs of imaginary roots as gaps or critical points. The values of all imaginary roots are themselves well-defined, just as the real ones are; the other expression could be understood as if *each* gap belonged to a *specific* pair of imaginary roots, a fact which is not only unconfirmed by Fourier, but must stay in doubt, until further investigations have enlightened this interesting point. Yet this should not mean that Fourier himself meant his statement this way. We would rather assume the opposite and suppose he had been in doubt about the existence or non-existence of such a definite relationship, and purposely avoided a manifestation of this ambiguous expression. Moreover, Fourier did not deal with the matter of imaginary roots and their computation in his work; so there is still a large field to work on.

Once having such a nice theorem — and according to the notes Mr. Navier shared with us, he has maintained it for a long time — it couldn't have been hard for a skilful researcher like Fourier to base the technique of the numerical solution of equations on it; and this development has been given with completeness and in great detail. More practised readers would probably prefer a narrower description that cuts down on repeating itself; for the less practised ones, the numerous, well-selected, and detailed examples will be welcome. Anyway, even in this part of the Theory of Values[5], this work assures the name of Fourier an honourable position, a position he long since asserts in other parts.

2. **Gauss to Schumacher. Göttingen, 1833, April 2**
Correspondence between Gauss and Schumacher, II. Altona 1860, p. 328

Many thanks to you, my dear friend, for letting me know about the ruled paper. I find it useful, especially for drawings of all kinds based on rectangular coordinates, as not the greatest accuracy is required. This is never the case with my drawings anyways, since I never use them to derive anything of them definitely by measuring; namely

[5] Algebra.

[they are useful] for drawing geodetical points, even for small astronomical charts, illustrating the pace of the barometer, variations of the magnetic [compass] needle etc., as well as for drawings referring to purely mathematical topics; for example those concerning the imaginary roots of the [algebraic] equations; with topics of the latter kind in particular I have dealt recently. Now as you'll have read my announcement about Fourier, you might be interested if I mention that, with regard to "but must stay in doubt ..." I have expressed myself moderately on purpose; this is not because I have been uncertain about the existence or non-existence of such a relationship; but rather because the G.G.A.[6] were not the place to express myself in a more definite way. I believe I can prove most clearly that such a relationship does not exist; this however will not happen before I have taken the chance to work out my investigations on the roots of the equations and publish them. You know, I write slowly; this is because I am only pleased if there is as much in a small space as possible. Besides, writing in short takes much more time than writing in length. If I explained my investigations — which, once developed, ought to take only a small number of sheets [7] — as broadly as Fourier's book is written, it would take me just a quarter of the time and several huge volumes.

3. **Gauss to Schumacher. Göttingen, 1836, June 20**

Correspondence between Gauss and Schumacher, III. Altona 1861, p. 68/69

On the occasion of the lecture I am giving these days, I was given to coming back to the theory of equations. I have gained an entirely new view of them. In light of this view, I still find it quite probable to find a connection between Fourier's critical points and definite pairs of imaginary roots. Three years ago, I mentioned an opposite opinion in my letter to you; I admit I had not developed all the details about that topic, which is necessary to be sure about a negative theorem one is going to prove. All the common *apperçus*[8] working in 999 cases may finally lead to a *cul de sac* in the 1000th case. Please correct my former, perhaps too positive assertions. But at the same time, note that I have not without reason made *pauca sed matura*[9] my motto for everything I publish. Those common *apperçus* sprang from one hour's work. To make

[6] Göttingische Gelehrte Anzeigen, see introduction at the beginning of this chapter.
[7] A *sheet* consists of 16 pages.
[8] Ideas.
[9] Few but ripe.

something mature from that, it often takes years of detailed work; one can anticipate one could do that, if only one complied with doing it; still, it will be necessary to do some similar second- or third-class work to come in order. *Procreare iucundum, at parturire molestum.*[10] As to my present standpoint, it would first of all require many wearisome details in order to follow all branches; I don't believe, now that so much is ..... *[the rest of the letter is missing]*

4. **Gauss to Schumacher. Göttingen, 1836, June 24**
Correspondence between Gauss and Schumacher, III. Altona 1861, p. 72

During the last days, I have pursued my ideas on the equations. The result is rather the opposite, and it tends to confirm my former view that there is no common, natural, non-arbitrary relationship between the single critical points and the pairs of imaginary roots at all. I myself have the impression that in my last letter I wronged the view which I had expressed in a previous letter, now that, after a long interruption, this view was not present to me in the same freshness in which I had had it in those days. It still remains true that, with negative theorems such as this, converting a personal conviction into an objective one (which others can share) would require a deterringly detailed work. To visualize the whole variety of cases, one would have to display a large number of equations by curves; each curve would have to be drawn by its points, and determining a single point alone requires lengthy computations. You will not see from Fig. 4 in my first publication of 1799[11], how much effort was required for a proper drawing of that curve; nonetheless, that is a very simple case, compared with the many ones we have to regard here.

5. **Annotations**
by Alfred Loewy

Gauss's notes in his letters to Schumacher as well as in his earlier letters to Drobisch, referring to [6], require that we deal with Gauss's announcement of this work, for some statements in this announcement have to be corrected. The statement "It is indeed true that the equation $X = 0$ *altogether* has as many pairs of imaginary roots as gaps

[10] Bring forth the pleasant, but work on giving birth to the tedious.
[11] The proof of the Fundamental Theorem, his Doctor's thesis. See [9].

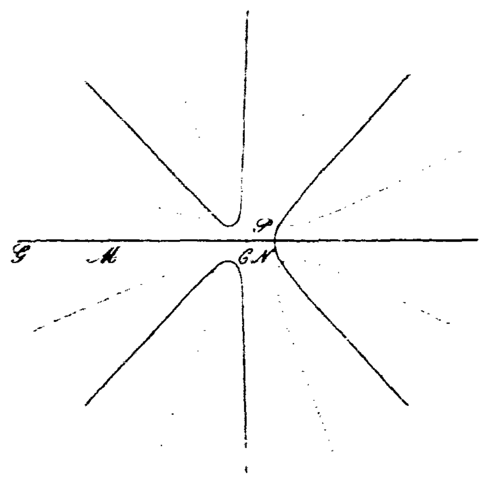

Figure A.1: *The* Figure 4 *of [9], cited by Gauss in his letter [12] to Schumacher.*

or critical points" is not true with regard to the definition given in Text 1., which for instance is easily shown by the equation

$$(x-b)^n - c = 0$$

where n is even and ≥ 4, b is real, and c positive. Despite its $n-2$ distinct imaginary roots, this equation wouldn't have any critical points according to Gauss. We rather define (compare with [2]) a critical point in the following way: Let $f(x) = 0$ be an equation of degree n with real coefficients, $f'(x)$, $f''(x)$, ... be the derivatives of $f(x)$ and α a real number; in the sequence $f'(\alpha), \ldots, f^{(n)}(\alpha)$ be e the number of vanishing intermediary[†] functions, c and d the number of changes and sequences of signs interrupted by an odd number of missing terms, and $\sigma = e - c + d$. If $\sigma \geq 2$, then α is called a ***critical point*** of $f(x) = 0$[††]. Using this definition, the following (more precise) Fourier theorem holds:

Let a, a' be two real numbers, $a < a'$; if Z_a and $Z_{a'}$ are the numbers of changes of signs in the sequences

[†]If, say, $f(\alpha) = f'(\alpha) = \ldots = f^{l-1}(\alpha) = 0$, then these first l vanishing terms are not to be regarded in determining e.

[††]In the above example, b is a critical point with values $e = n-1$, $c = 1$, $d = 0$, and therefore $\sigma = n-2$.

$$f(a), f'(a), f''(a), \ldots, f^{(n)}(a), \tag{A.1}$$

$$f(a'), f'(a'), f''(a'), \ldots, f^{(n)}(a'), \tag{A.2}$$

and if further 2λ is the sum of all numbers σ (which are always even, according to their definition), corresponding to all critical points of $f(x) = 0$ between a and a', then

$$Z_a - Z_{a'} - 2\lambda \geq 0 \tag{A.3}$$

and the number defined on the left-hand side of (A.3) is the *exact* number of roots of $f(x) = 0$ between the bounds a and a', counting multiplicities. In the case that the bounds a and a' are critical points or roots of $f(x) = 0$, a is to be excluded, and a' to be included[‡] in computing 2λ and counting the roots. For $a = -\infty$, $a' = \infty$, we get the following theorem that replaces the statements by Gauss: The number of imaginary roots of $f(x) = 0$ equals the sum of all σ corresponding to all critical points of $f(x) = 0$ between $-\infty$ and ∞. From the inequality (A.3) it follows that, provided there is a critical point between a and a' with $\sigma > 2$, $Z_a - Z_{a'}$ must be > 2. Therefore, Gauss's statement "that every single difference $g' - g, g'' - g', g''' - g''$ etc. will be equal to either 1 or 2" if the bounds are chosen sufficiently narrow, is not correct. Fourier knew that $g' - g$ cannot always be made ≤ 2, which can be seen from his "rule of the double sign" [6, p. 103]

The fact that it is not possible to find a connection between Fourier's critical points and definite pairs of imaginary roots, follows from the example $(x-b)^n + c = 0$ (n even, b, c real, $c > 0$), with b as the only critical point and n distinct imaginary roots. The following question however remains open: Let $f(x) = 0$ have m critical points where $m \leq n/2$ and $\sigma = 2$ for each of them[‡‡]; is there a "common, natural, non-arbitrary relationship between" the m single critical points and the pairs of imaginary roots of $f(x) = 0$?

[‡]The "exceptional case" , set aside by Gauss in Text 1., stipulates that no term in (A.1,A.2) be 0. It is thus a slightly stronger restriction than requiring a and a' neither to be roots nor critical points.

[‡‡]The case $\sigma = 2$ arises not only from Gauss's announcement which allows only $d = 1$, but it occurs in general for an arbitrary integer e, when $c = e - 1$, $d = 1$ or $c = e - 2$, $d = 0$. This follows from $\sigma = 2$ and from the relation $e \geq c + d$ which is always valid, due to the definition of c, d, e.

Appendix B

How to get xzero

B.1 Obtaining the package

The *xzero* package is available through the Internet. It is stored on the `ftp` server of the Centre for Experimental and Constructive Mathematics. To download the package, one needs to follow these steps:

1. On the UNIX prompt, type

   ```
   ftp ftp.cecm.sfu.ca
   ```

   The `ftp` server will connect, asking for a user name:

   ```
   Name (ftp.cecm.sfu.ca:happe):
   ```

2. Type `anonymous`.

3. Enter your full e-mail address as password.

4. Change the directory:

   ```
   cd pub/Math/Software
   ```

5. You may also want to change your local directory. The `xzero` directory will be created as a subdirectory of the current local directory.

6. Type

   ```
   get xzero.tar.z
   ```

7. End the `ftp` session with

   ```
   bye
   ```

8. Uncompress and unarchive the files in *xzero* with the two commands

   ```
   unzip xzero.tar.z tar xvf xzero.tar
   ```

Users who possess a World Wide Web browser such as *Mosaic* may prefer to use it to download *xzero*. The correct URL for *xzero* is

`ftp://ftp.cecm.sfu.ca/pub/Math/Software/xzero.tar.z`

Most Web browsers will automatically uncompress `xzero.tar.z`, so it just needs to be unarchived.

B.2 The files in the package

The *xzero* package consists of the following files:

`README`	Information about the version of *xzero*, known bugs, and instructions how to compile *xzero*.
Source code	All files ending with `.c` (source files) or `.h` (header files). The code is modularized into several source files which carry names indicating their functionality.
`xzero`	The executable *xzero* program.
`makefile`	The compilation rules for *xzero* and related tools.
`beamer`	A program called by *xzero*. It runs in the background, handling the transmission of data from *Maple* to *xzero*.

`destroy` A tool that clears up message structures. Normally, it need not be used, since *xzero* clears them up itself. But it is recommended that it be used (by typing `destroy` on the UNIX prompt) after an abnormal termination of *xzero*.

`go.txt`

`polynom.txt` A collection of *Maple* routines, which build the computing functionality of *xzero*. These files, and how they can be modified to install new functions, are described in Appendix C.

`go.m` The compiled routines of both `go.txt` and `polynom.txt`, for use in the background *Maple* session issued by *xzero*.

`polynom.m` A reduced version of `go.m`, to be used as a package for interactive *Maple* sessions.

`functions.reg` A text file in which all *Maple* functions that are to be issued from *xzero* need to be registered. See Appendix C.

`XZero` A file containing important application resources for *xzero*. For more information, see Section B.4.

Bitmaps All files ending with `.bmp`. These files contain images of various shapes in which points are displayed in the *xzero* drawing area. The user can select their own preferred set of point shapes, since they are specified as resources (see Section B.4).

B.3 Installing xzero

The *xzero* package is written on Silicon Graphics Workstations under UNIX, using the X Window System. Its graphical appearance is realized using the X library (X11R4) and the Motif toolkit (Version 1.1). These are also the requirements on a system on which *xzero* is to be compiled successfully. In order to *run* successfully, a few more requirements have to be satisfied. *Maple* (V5R2 or higher) must be installed and accessible, and *xzero* must be run on a server with a colour display. With future versions, some of these restrictions may become obsolete, and the installation procedure may also change. Any information in the file `README` that differs from the steps given below overrides them.

1. Change the working directory to `xzero`.

2. Type the command

   ```
   make
   ```

   You will see the source files of *xzero* being compiled. If `make` stops with an error message, this probably means that some of the above system requirements are not satisfied. Note that the executable code may take more than 2 Megabytes of disk space. This may lead to problems on systems with small disk capacity or disk quotas.

3. Copy the file `XZero` (mind the capitals!) from the current directory into the system directory `/usr/lib/app-defaults`. If you don't have write permission to this directory, you can ask your system administrator to do this. Or do the following steps:

   - Go to your home directory, and create a subdirectory, say, `.app_defa`. (You can give it any name, and in fact any location. However, a subdirectory of your home directory is the most convenient location.)
   - Copy the file `XZero` into this directory.
   - In your home directory, edit the file `.login` and add the line

     ```
     setenv XAPPLRESDIR $HOME/.app_defa
     ```

   - Before running *xzero*, logout of the system, to make the changes take effect. (On most systems, it actually suffices to open another shell window.)

   You can use your new directory not only for *xzero*, but for any X application you wish to configure with your own resource files. More information about application resource files and their installation can be found in [21].

In order to run *xzero* properly, you have to leave it in a directory together with its essential tools, namely the `.m` files, the `.bmp` files, and `beamer`. This may change with future versions.

B.4 xzero resources

Like every other X application, *xzero* uses resources that make it configurable, up to a certain level, by the user. A workable standard configuration is contained in the resource file `XZero` that comes with the package. This configuration can be modified or extended by further specifications. to the user's own taste. One should remember. however. that some resources affect the functionality of *xzero*. Therefore. it is advisable to keep a copy of the original file `XZero` before modifying it. No responsibility can be taken for abnormal behaviour of *xzero*, when resources are modified.

B.4.1 Application resources

These resources were specifically introduced for xzero, and they control some features a user may want to modify. They are listed completely in the documentation to *xzero*. For brevity, we only list the types of configurable resources, not the names themselves:

- point shapes and colours for all list types,
- cursor shapes for all mouse modes,
- coordinate translations,
- Boolean variables to disable certain warning messages,
- some system parameters that might improve *xzero*'s reliability on slow or non-standard machines,
- digits of numerical precision,
- Initialization commands for *Maple*,
- default filenames and extensions,
- message strings.

For each of the resources one wants to modify, one has to add a line to the file `XZero` like this:

`XZero*`*resourcename*`:` *value*

If a resource is already specified, then the old line has to be replaced rather than adding a new line at the end. Otherwise, the new specification will not take effect.

B.4.2 Widget resources

These resources obey the standards of the X and Motif toolkits. They will not be described here. For reference about specific resources, their meaning and usage, one should consult [21] or better [23]. Some of the features that can be modified are:

- foreground, background and border colours of all windows, except the drawing area
- window sizes and locations
- text and label fonts and sizes
- bitmaps shown on the action buttons
- most of the menu and help texts (which makes it possible to "translate" *xzero* into other languages)
- keyboard shortcuts for menu and action buttons of *xzero*
- mapping of mouse keys to mouse functions
- default settings of the toggle and radio menu buttons.

Specifying these resources in the file `XZero` is more complicated. The user may specify some resources only for specific windows, for specific types of windows, or for all types of windows. The file `XZero` contains many examples which may be of help. Beyond that, more information regarding the syntax of resource specifications can be found in [22].

Appendix C

Writing your own Maple code

The computational part of *xzero* is most flexible. Not only is it accessible to be read and modified: also, an interface is provided to incorporate newly created computation procedures into *xzero*. We will both describe the existing *Maple* code, and give guidelines for using the *xzero* interface correctly.

C.1 Existing procedures

The builtin functions are constructed in a modular fashion. The code consists of several procedures which may also be used by newly added functions. Hardly any new routine will need to be written from scratch: most often, one can use some of the builtin procedures. It is therefore advisable to know them. Also, the *Maple* code can be used in interactive sessions, independent from *xzero*. In fact, some procedures, mainly for drawing, are designed solely for interactive *Maple* sessions, and not called from any computation by *xzero*. All these procedures are contained in the file `polynom.m`. They are automatically read into the *Maple* session called by xzero. In an *interactive Maple* session, one should make them available with

```
read(`polynom.m`);
```

Complex numbers naturally play an essential role throughout the package. However, it became apparent that computations with complex numbers in *Maple* are not only slow; but also. *Maple* (in Release 3) is unable to handle the real and imaginary part of a complex number as real numbers. at least in symbolical computations[1]. Therefore. most of the code also accepts complex numbers in *list form*. as pairs (lists) of 2 real numbers. In the code description. by using the term *zero* we imply that the value may be represented in either form. To enhance both performance and robustness against rounding errors. *xzero* specifies all its zeros in list form.

`_symmetric`

Global variable that affects some of the procedures. equivalent to the symmetric mode in *xzero*. In essence. for each of the terms a procedure creates on each nonreal zero. it creates the same term on the conjugate of this zero. For example. `make_poly_from_roots` creates the factor `(x-a-b*I)` from a given zero `a+b*I` when `_symmetric` is not set to `true`. Whereas. when `_symmetric=true`. it creates `(x-a-b*I)(x-a+b*I)` and simplifies it to a real value.

`make_roots_from_file` (*filename*)

Reads zeros from a file specified by *filename*. The filename must be enclosed in backquotes. if it contains any special characters. The file format is the same as that used in *xzero*. i.e. roots or results files created by *xzero* can be read with this procedure. `make_roots_from_file` returns a list of all zeros read.

`make_poly_from_file` (*filename*)

Reads coefficients from a file and creates a polynomial with these coefficients. beginning with the lowest order coefficient. File name and format specifications are the same as in `make_roots_from_file`. `make_poly_from_file` returns the polynomial created.

`make_poly_from_roots` (*list*)

Takes a list of zeros and constructs a polynomial from them. assuming the highest-order coefficient to be 1. When `_symmetric` is set to `true`. it implicitly takes the conjugate of each given zero as well. and ensures. by doing real computations. that the resulting polynomial is real.

[1]One can. of course. explicitly define the real and imaginary parts of a complex number to be real. using the assume function. But first. this is not done automatically. and second. the property handling with the assume function is rather slow.

`xfactor` (*val*)

Takes a zero and creates one of the factors for `make_poly_from_roots`.

`make_wrons_from_roots`(*list*)

Creates the second logarithmic derivative of p directly from the zeros given in *list*, and takes the numerator as Wronskian. When `_symmetric` is set to `true`, it implicitly takes the conjugate of each given zero as well, and also ensures that the resulting polynomial is real.

This procedure is obsolete since this way of computing the Wronskian has been found to be slower than the two-step process of constructing the polynomial and its derivatives. Also, the trivial zeros of Wp cancel out when taking the numerator of the second logarithmic derivative. This procedure has mainly been left in the package for reference.

`make_contrib_list` (*list*)

Applies the procedure `wterm` to each of the values in *list*; for `make_wrons_from_roots` and `plot_components`.

`wterm` (*val*)

Computes a single term of the second logarithmic derivative, corresponding to the zero given in *val*; for `make_wrons_from_roots`.

`make_wrons_from_poly` (*p*)

Creates the Wronskian of the (real or complex) polynomial p in standard form.

`lccords` (*val*)

Turns a complex value *val* into list format.

`ccoords` (*val*)

Turns a complex value given as a list into complex (numeric) format.

`cexpand` (*val*)

Creates a sequence of two complex values. *val* and $\overline{val}$. The format of the result values depends on the format of the parameter *val*.

`make_zdlist_from_poly` (p)

Creates a list of two lists of zeros, the first being the list of roots of p; the second being the list of roots of p'.

`make_zdlist_from_roots` (*list*)

Creates the same list as in `make_zdlist_from_poly`, but p is specified by a list of roots.

`make_zwlist_from_poly` (p)
`make_zwlist_from_roots` (*list*)

Similar to `make_zdlist_from_poly` and `make_zdlist_from_roots`, except that the second list is the list of roots of Wp here.

`pcomponents` (p, *val*)

Determines the real and imaginary part of $p(val)$. Unlike the `Re` and `Im` functions in *Maple*, `pcomponents` accepts and correctly handles symbolic values of *val*. *val* can be specified as either a complex value or a list. `pcomponents` always returns $p(val)$ in list format.

`rcomponents` (r, *val*)

Similar to `pcomponents`; but `rcomponents` takes a rational function r in the form p/q, where p and q may be real or complex polynomials. `rcomponents` converts $r(val)$ into the standard form $\frac{c_1+c_2 i}{c_3}$, where c_1, c_2 and c_3 are real, and returns `[`c_1`,`c_2`,`c_3`]`.

`fchecksolve` (*expr*, *var*, *range*)

Similar to *Maple*'s `fsolve` command; but before solving *expr*, it is verified if *expr* contains an indeterminate. If not, `NULL` is returned, otherwise the sequence of solutions.

`my_implicit_plot` (*expr*, *xrange*, *yrange* [, <*options*>])

An improved routine for implicit plots. *expr* must be an expression in two variables, and *xrange* and *yrange* must be range specifications of these two variables, in the form *x*=*min*..*max*. Any of the usual `plot` options can be specified, but the `style` and `grid` options will have no effect. Instead, one can specify `gridnum=`*n* to adjust the graphing resolution.

The result of `my_implicit_plot` is a plot containing a set of points.

`less`

Defines an ordering on the set of complex numbers. This ordering is currently not used.

Plotting routines

Some of the plotting routines, e.g. the animation routines, are for interactive use only. Others can also be called by xzero. All plotting routines return a `PLOT` structure.

`cplot` (*list*, *colour*)

Convenience procedure to plot a list of zeros in a certain colour.

`accumulate_zlist` (*args*)

Plots a sequence of pairs of zero lists, specified in *args*. These lists are usually the results of `make_zdlist_from_poly` or `make_zwlist_from_poly` or `..._from_roots`. The zeros of the polynomials (first list) are displayed as red, and the zeros of the derivatives/Wronskians (second list) as green circles. This procedure can be used to display just a single list, or to display several related lists together, in order to show trajectories of zeros.

`animate_zlist` (*args*)

Similar to `accumulate_zlist`. This plot however is an animation of the zero lists, where every list given in the argument sequence is displayed in its own frame.

`plot_one` (*listlist*)

Takes one of the arguments given to one of the procedures `accumulate_zlist` and `animate_zlist`. and builds it into a plot, which will be a component of the accumulated plot created in these procedures. *listlist* is a pair of zero lists.

`plot_one_from_roots` (*list*,*xrange*,*yrange*)

Plots p and Wp. where p is given by its roots in *list*. *xrange* and *yrange* must be valid range specifications for *Maple*'s `plot` command.

`animate_polys_from_roots` (*listlist*,*xrange*,*yrange*)

Creates an animated sequence of polynomials, each frame of this sequence being a graph as in `plot_one_from_roots`. The polynomials are described by the root lists contained in *listlist*.

`plot_components` (*list*,*xrange*,*yrange*)

Plots the second logarithmic derivative of p (black), and all sum terms that constitute it in Formula 3.3 (red). *list* is the list of zeros that specifies p. *xrange* and *yrange* must be valid range specifications for *Maple*'s `plot` command. This procedure works only when `_symmetric` is `true`.

`plot_loci_from_poly` (*p*,*c*,*xrange*,*yrange*,*mode*)

Plots the loci of p of constant argument c. The plotting ranges must be specified as *min*..*max*. *mode* can be

- `single` for the locus $L_c(\frac{p'}{p})$.
- `symmetric` for the locus $S_c(p)$.
- `double` for the union of $L_c(\frac{p'}{p})$ and $L_{c\pm\pi}(\frac{p'}{p})$. In particular, $c=0$ yields $L_R(\frac{p'}{p})$.

C.2 The interface to xzero

This section is mandatory for programmers who wish to write their own routines for use in *xzero*. A genuine knowledge in the (not too complex) input/output protocol between *xzero* and *Maple* is necessary, in order to write code that works under *xzero*.

C.2.1 Input from xzero

The input *Maple* receives from *xzero* is a sequence of *Maple* commands. The most important command herein is `go`, which calls a *Maple* procedure defined in `go.txt`. The syntax of `go` is

`go ( <proc>, [<zero>,...] [, <options>] [,symmetric= <string>]);`

The first parameter of `go` is the procedure name to be executed, as defined in the file `functions.reg`. Following the procedure name are the zeros of the polynomial to compute with. They are specified by *xzero* in list form. The remaining parameters are optional and, except for `symmetric=` *<string>*, must be recognized by the procedure to be called. *xzero* places the options specified for *proc* in `functions.reg` here. The option `symmetric=` *<string>* is evaluated by `go` itself. *string* should be either `true` or `false` (other values are considered as `false`). `go` sets the value of the global *Maple* variable `_symmetric` accordingly. Other *Maple* procedures that distinguish between a symmetric and an unsymmetric case can then query the variable `_symmetric`.

Other than this, ordinary commands can be issued through `Maple Command` in the `Option` menu of *xzero* (Version 1.1 only). These commands are executed directly by *Maple*. The user should refrain from commands that change *Maple*'s status or set global variables in a way that confuses definitions in `go.m`. In particular, one should never issue `Quit` to *Maple*, since this only quits *Maple*, but doesn't reset the interface. To quit *Maple* (and start a new session), one should use `restart Maple` from the `Options` menu.

In the startup phase of *xzero*, a few other *Maple* commands are issued by *xzero*. Only two of them are of relevance to the user. (The others accomplish a correct setup of the communication interface and will not be described.) The first one,

```
read ('go.m'):
```

reads the predefined computation routines into the *Maple* session. `go.m` is the compiled version of `go.txt`, `polynom.txt`, and some other files with *Maple* code (see next Section). The other command,

```
plotsetup(x11):
```

changes *Maple*'s plot interface, so that it displays plots in separate windows. These two commands are specified as *xzero* resources, so they can be modified or supplemented by other startup commands.

C.2.2 Output from Maple

The various computation routines yield different types of output. Some plot graphs, some generate a list of zeros, some only produce text output, while others might not produce any output at all. *Maple* itself only distinguishes between two types of output: Plots, which are displayed on the screen, and text output. The latter includes all error and warning messages that might occur. We will not explain the actual mechanism that sends and filters *Maple*'s output to *xzero*[2]. We will rather describe the concept in which *Maple*'s output is interpreted, and how to use it properly.

Prior to the results of a computation, the computation routine should print a control string. A control string consists of a `'#'` and a digit that specifies how *xzero* is to interpret the subsequent output lines:

#0 Output, if any, is printed to `stderr`. This is the most flexible way of handling all kinds of *Maple* output, but it requires that `stderr` be either printed to a shell window, or logged to a file, to be accessible to the user.

#1 Output is written into the status line of the *xzero* main window. This requires that only one output line be printed. Although each line transmitted will be printed to the status line as soon as it is received, only the last output line will be visible, unless the *Maple* computation is very slow. (This option is only available in Version 1.1.)

#2 Output is treated as a list of zeros, in the usual file specification. The output lines are expected to match this specification, otherwise an error message will appear, and

[2]This is the job of the **beamer** program and the message queue, which are cited in some error messages. See the documentation of *xzero* for details.

the erroneous line will be printed to `stderr` for debugging purposes. The zeros will be printed as result points in the *xzero* drawing area.

It is possible to send different parts of output in different types, if each part is preceded by the corresponding control string. However, some of the output types cannot be mixed. For instance, a zero list causes a status message to be written at the end of the transmission. Therefore, any status line written under the `#1` option will be replaced by this message.

Note the following peculiarities for special kinds of output:

- *Maple* plots, as mentioned above, are independent from text output. Therefore, they always show up on the display, regardless of the output type.
- *Maple* warnings are always printed to `stderr`.
- *Maple* errors interrupt the transmission, and the user will be asked to retry or abort the computation. For user information, the *Maple* error message is printed to `stderr`.
- Empty lines are not handled consistently. One should avoid them, if possible.

C.3 Placing and compiling one's own code

Now that all requirements on additional *Maple* code have been described, we will explain where to place and how to compile it. Only the compiled code is used during the *xzero* session, in the form of the file `go.m`. The source code, namely the files `polynom.txt` and `go.txt`, is provided only for modifications and extensions by the user. Consequently, in order for modifications to take effect, these files have to be re-compiled to `go.m`.

To relieve the programmer from the drudgery of manually rebuilding `go.m`, the compilation has been automated with a tool that has its established position in compiling **C** programs: `make`. This makes the translation of the (error free) *Maple* source as easy as typing

```
make go.m
```

We assume that the concept of makefiles is known. The `make` rules for `go.m` are contained in the file `makefile` which also contains the rules for compiling the *xzero* package.

First, `polynom.txt` is compiled into the file `polynom.m`. These lines of `makefile` accomplish the translation:

```
POLYSRC = polynom.txt
polynom.m: $(POLYSRC)
        cat $(POLYSRC) |maple -q -f
```

This rule takes the list of files defined as `POLYSRC` (more files can be added to the list), and pipes them to *Maple*. The files in `POLYSRC` must therefore consist of valid *Maple* definitions. They may also contain output commands which will be written to `stdout`. If any of the files produces a *Maple* error, the computation will stop.

The last line of `polynom.txt` reads:

```
save `polynom.m`;
```

which causes the operation status of *Maple* (i.e. the set of all definitions given so far) to be written to `polynom.m`. Since all definitions from all files are to be compiled, `polynom.txt` must always be the *last* file in `POLYSRC`.

`go.m` is created in a similar way as `polynom.m` is. The corresponding `make` rule reads:

```
GOSRC = pborwein.txt go.txt
go.m: $(GOSRC) polynom.m
        cat $(GOSRC) |maple -q -f
```

Note that `polynom.m` is specified as a dependency of `go.m`. This ensures that upon a change in `polynom.txt`, both `polynom.m` and `go.m` will be rewritten. The definitions in `polynom.m` are read into *Maple* in the first line of `go.txt`:

```
read `polynom.m`;
```

Exactly as in `polynom.txt`, the last line of `go.txt` contains a command that writes the *Maple* definitions into `go.m`. Therefore, `go.txt` must be the last file in the list specified under the variable `GOSRC`.

There are two ways to place one's own procedures in the *Maple* code:

First, all additions can be written into the files `polynom.txt` and `go.txt`. Thus, all definitions are kept in a single file, easy to modify or to refer to.

Alternatively, code additions may be separated from the builtin procedures, by keeping them in different files. These files need to be added to the lists in `POLYSRC` or `GOSRC`. This approach is preferable if many users are writing extensions, since it supports modularization. Every user could have their own makefiles, in which one or several common files (among them `polynom.m` and `go.m`) and individual, private extensions could be specified in the user's private `makefile`. Note that this does not require multiple copies of *xzero* on the system.

It is entirely up to the *Maple* programmer where to put their own extensions. A distinction between source code for `polynom.m` and code for `go.m` is not at all enforced. It is recommended, however, that extensions to the interface (e.g. procedures that accept input or create output suitable for *xzero*) be added to `go.txt` or to another file specified in `GOSRC`, whereas procedures that can be independently used in an interactive *Maple* session be put into `polynom.txt` or another file in `POLYSRC`. Then only the file `polynom.m` needs to be read into an interactive session.

C.4 How to register your functions in xzero

In the `Execute` menu, *xzero* provides room for a user-defined *Maple* procedure called `user`. This is intended for temporary experiments with new code. In order to make one's own *Maple* functions permanently available in *xzero*, one should assign an individual name to them, and then register them as a new menu entry.

For every new registered function, a push button is created in the `Execute` menu. On pressing this button, *Maple* will be called with the corresponding procedure name as parameter. All information *xzero* needs for this comes from the file `functions.reg`. This is a text file whose specifications are easy to read and modify. Every function — except `user` — is registered through this file. Each function consists of three lines with the following meanings:

- The first line contains the string that is to appear on the push button.
- The second line is the name of the *Maple* procedure.

- The third line contains additional parameters given to the *Maple* procedure when it is called.

The third line can be left empty, if the function needs no options. This is the case with all predefined functions. Lines beginning with a '`*`' are ignored. Thus one can insert comments into `functions.reg` and more clearly separate function definitions from one another.

The file `functions.reg` is read at the beginning of each *xzero* session. Therefore, changes in `functions.reg` take place the next time *xzero* is started. It is not necessary to re-compile the code of *xzero* when registering new *Maple* functions.

Bibliography

[1] L.V. Ahlfors. *Complex Analysis*. 3rd ed., McGraw-Hill, NY, 1979.

[2] *Archiv der Mathematik und Physik* (3) 16, 1910, p.157.

[3] T. Craven, G. Csordas, W. Smith, *The zeros of derivatives of entire functions and the Pólya-Wiman Conjecture*. Ann. of Math 125(1987), 405-431.

[4] K. Dilcher, K.B. Stolarsky, *Zeros of the Wronskian of a polynomial*, J. Math. Anal. Appl. 162(1991), No.2, 430-451.

[5] K. Dilcher, *Real Wronskian zeros of polynomials with nonreal zeros*, J. Math. Anal. Appl. 154(1991), 164-183.

[6] J.B.J. Fourier, *Analyse des équations déterminées*, Paris 1831.

[7] *Gauss Werke Bd. III*, 2nd ed., Hildesheim, New York 1981, 119-121.

[8] *ibid.*, p. 112.

[9] *ibid.*, p. 30.

[10] *Gauss Werke Bd. VIII*, p. 130.

[11] *ibid.*, p. 131.

[12] *ibid.*, p. 132-133.

[13] *ibid.*, p. 133-134.

[14] J. L. W. V. Jensen, *Recherches sur la théorie des équations*. Acta Math. 36(1913), 181-195.

[15] F. Lucas, *Propriétés géométriques des fractions rationelles*, Paris Comptes Rendus 78(1874), 271-274.

[16] F. Lucas, *Géométrie des polynomes*, Jour. de l'École poly., vol. 28(1879), cah. 46, 1-33.

[17] M. Marden, *Geometry of Polynomials*, 2nd ed., Amer. Math. Soc., Providence, RI, 1966.

[18] G. V. Milovanović, D. S. Mitrinović, Th. M. Rassias, *Topics in Polynomials: Extremal Problems, Inequalities, Zeros*, World Scientific, Singapore, 1994

[19] J. v. Sz. Nagy, *Zur Theorie der algebraischen Gleichungen*, Jber. Deutsch Math. Verein 31(1922), 238-251.

[20] J. v. Sz. Nagy, *Über die Lage der nichtreellen Nullstellen von reellen Polynomen und von gewissen reellen ganzen Funktionen*, J. Reine Angew. Math. 170(1934), 133-147.

[21] V. Quercia, T. O'Reilly, *X Window System User's Guide*, Vol. 3, O'Reilly&Associates Inc., 1993.

[22] A. Nye, T. O'Reilly, *X Window System User's Guide*, Vol. 4, O'Reilly&Associates Inc., 1992.

[23] P. M. Ferguson, *X Window System User's Guide*, Vol. 6B, O'Reilly&Associates Inc., 1993.

[24] G. Pólya, *Some problems connected with Fourier's work on transcendental equations*, Quart. J. Math., Oxford Ser. 1(1930), 21-34.

[25] J. L. Walsh, *On the location of the roots of the derivative of a polynomial*, C.R. du congrès international des Math., Strasbourg(Toulouse) 1920.

[26] J. L. Walsh, *On the location of the roots of the derivative of a polynomial*, Ann. of Math. 22(1920), 128-144.

[27] J. L. Walsh, *The location of critical points of analytic and harmonic functions*, Amer. Math. Soc. Colloq. Publ. Vol. 34, Amer. Math. Soc., Providence, RI, 1950.