Chromatic Zeros of Boolean Functions

by

Benjamin Porter

Thesis

Submitted by Benjamin Porter

in partial fulfillment of the Requirements for the Degree of

Bachelor of Computer Science with Honours (1608)

Supervisor: Graham Farr

Clayton School of Information Technology

Monash University

November, 2006

© Copyright

by

Benjamin Porter

2006

Contents

List of Tables

List of Figures

Chromatic Zeros of Boolean Functions

Benjamin Porter
benjamin.porter@gmail.com
Monash University, 2006

Supervisor: Graham Farr

Abstract

Graph colouring is one of the largest fields of graph theory. Counting colourings of graphs, or merely determining whether they exist, is an important and complex problem. The chromatic polynomial is a useful tool in the investigation of this problem. The roots of this polynomial (known as chromatic roots) have interesting properties and a lot of research has been devoted to their study; for example, finding regions free of chromatic roots.

Generalising the notion of the chromatic polynomial to other structures is one possible strategy in the investigation of graph colouring. This has been done, for example, for matroids, arbitrary relations, and Boolean functions. The latter is the focus of this project.

The aim of this project is to investigate the distribution of chromatic roots (technically zeros) of Boolean functions. The project's scope is quite broad, as it is an unexplored field of mathematics. However, this research was intended to give an overall view of this new mathematical landscape.

This thesis presents a qualitative analysis of the distribution of chromatic zeros of Boolean functions and some new theoretical results. It presents observations of chromatic zeros of various Boolean function classes, such as Boolean 2–CNF functions, quadratic polynomials, and other, novel, Boolean function classes. A new graph–like combinatorial object, isomorphic to the class of 2–CNF functions, is introduced. An interesting phenomenon of zero "trails" is observed and some theoretical material supporting this is presented.

The project provides initial insight into the field of chromatic zeros of Boolean functions.

Chromatic Zeros of Boolean Functions

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Benjamin Porter
November 6, 2006

Acknowledgments

I thank my supervisor, Graham Farr, for suggesting this interesting, and deceptively complex, project. It united several interesting fields such as combinatorics, numerical methods, and complex analysis, and was a great experience. Graham supported me immensely through weekly meetings, where his endless questions and observations steered the project down many different paths. I thank him also for his help with my academic writing style and am inspired by his attention to detail.

I also would like to thank my reader, David Albrecht, for his helpful feedback throughout the year.

Benjamin Porter

Monash University
November 2006

Chapter 1

Introduction

The theory of graphs is a diverse and widely applicable field of mathematics. Graphs are objects that are composed of elements (called vertices) and connections (called edges) between these elements. Graph theory is very broad and can be used, for example, to find optimal routes, to determine the robustness of a computer network, or in resource allocation. This latter is an abstraction of *graph colouring*.

A colouring of a graph is simply an assignment of colours to the vertices, such that no two adjacent vertices are assigned the same colour. A q–colouring is merely a colouring of a graph using q colours or less. The theory developed out of consideration of the *Four Colour Theorem*, which states that four colours are the minimum number required to colour any map such that neighbouring territories get coloured differently.

Graph colouring has numerous real-world applications such as timetabling and register allocation[1]. However, the theory of graph colouring is deceptively complex and many problems are still unsolved. These problems include: efficiently finding colourings, determining the minimum number of colours required to colour some graph, and counting colourings.

We can investigate the theory of counting colourings using the *chromatic polynomial*. The chromatic polynomial of a graph takes a parameter q and gives the number of q–colourings. It provides an analytic and algebraic tool for the study of graph colouring and a basis for generalising the notion of colouring to other structures.

This project is an investigation of a recent such generalisation to arbitrary Boolean functions [14], with a specific focus on the zeros of the chromatic polynomial (known as chromatic zeros). The investigation is empirically driven, using zero-finding software and qualitative analysis of the zero distribution. Attention is restricted to specific classes of Boolean functions.

The aim of the project is to make qualitative observations and perform an investigation of the theory of chromatic zeros of Boolean functions. The project consisted of a software development component and an analytical component. Zero finding algorithms were researched and zero–locating software was developed to support the project's primary aim. The output of the project included, among other results: observations on the chromatic zeros of several interesting Boolean function classes, some theoretical results on a zero bounding region, and the introduction of a novel combinatorial structure used to reason about 2–CNF Boolean function colouring. This work contributes to the greater research area of graph colouring by providing some initial observations and conjectures on the theory of Boolean function colouring.

[1] Some applications are covered in §3.

This thesis is structured as follows. Some basic concepts are first introduced (§2), followed by some history and motivation behind graph colouring (§3). The history of the chromatic polynomial is then reviewed and some generalisations presented (§4). Some motivation and theory of chromatic roots and zeros is then presented along with a detailed review of zero–locating methods (§5). The subsequent chapters present the output of the research, including: an overview of the software developed (§6), the main results and discussions (§7), and the conclusion (§8).

Chapter 2

Basic Concepts and Definitions

The fundamental concepts used throughout this thesis are defined here. More complex concepts are discussed when required and a glossary (Appendix A) is provided for uncommon terms.

2.1 General Notation

A finite set A has size $|A|$. The set of all subsets of A is denoted 2^A, $\bar{A}$ denotes the set of all elements *not* in A, and $\emptyset$ denotes the empty set. A^n denotes the n-dimensional vector space on the field A.

A *Boolean function* defined on a ground set S is a function $f : 2^S \to \{0,1\}$. The *support* of this function is defined as $\operatorname{supp} f = \{X \subseteq S : f(X) = 1\}$. An *indicator function* of a set $T \subseteq 2^S$ is a Boolean function $f : 2^S \to \{0,1\}$, where

$$f(X) = \begin{cases} 1, & \text{if } X \in T; \\ 0, & \text{otherwise.} \end{cases}$$

A *Boolean function specification* is a bitstring which provides a shorthand method for describing a Boolean function. Given the truth table representation of a function, the specification is the string formed by reading the evaluated function downwards. Table 2.1 demonstrates this.

Example: The bitstring 1110 represents the function f with $\operatorname{supp} f = \{\emptyset, \{a\}, \{b\}\}$. The bitstring 10101010 represents the function g with $\operatorname{supp} g = \{\emptyset, \{a\}, \{b\}, \{a,b\}\}$.

a	b	c	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Table 2.1: This Boolean function has specification 10101010.

2.2 Graph Theory

A *graph* $G = (V, E)$ consists of a set of vertices $V = V(G)$ and a multiset of edges $E = E(G)$. Each edge is an unordered pair of vertices (e.g., an edge e connecting vertices a and b is $e = \{a, b\}$.) Let us write $e = ab$ as shorthand for $e = \{a, b\}$. Note that $ab = ba$ and hence the edges do not have *direction* (in graph colouring we are usually only concerned with undirected graphs). A vertex a is *adjacent* to another vertex b if and only if there exists an $e \in E$ such that $e = ab$. A *loop* is an edge of the form $e = aa = \{a\}$.

A *subgraph* G' of G is a graph with $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$. A subset $E' \subseteq E(G)$ *induces* a subgraph $G\langle E'\rangle = G' = (V(G), E')$. A *component* of G is a maximal connected subgraph of G. The number of components of G is denoted by $k(G)$. A *cutset* $E' \subseteq E(G)$ is a minimal set of edges such that $k(G\backslash E') > k(G)$. The *cutset space* of a graph, G, denoted $CS(G)$, is the minimal set that includes all cutsets of G and is closed under disjoint union.

Two common operations on a graph are the *contraction* and *deletion* of edges. The deletion of an edge simply removes the edge from the graph. The contraction of an edge removes the edge from the graph and merges its two endpoints into a single vertex.

The rank $\rho(G)$ of a graph G is defined as the rank of its incidence matrix (see e.g., [19, §13] and the glossary). We also have the simple relation $\rho(G) = n - k(G)$. Let us write $\rho(E')$ to mean $\rho(G\langle E'\rangle)$ where $E' \subseteq E(G)$.

A *proper k-colouring* (from now on referred to simply as a k-colouring) is a mapping $\sigma : V(G) \rightarrow \{1, \ldots, k\}$ such that $\forall \{a, b\} \in E : \ \sigma(a) \neq \sigma(b)$.

Chapter 3

The Origin and Applications of Graph Colouring

Graph colouring is one of the largest subfields of graph theory and there are numerous introductions available (see e.g., [12], [19], and [4]). The problem of *colouring* the vertices so that no two adjacent vertices are coloured the same is a popular and complex problem. Its history lies in the related problem of map colouring.

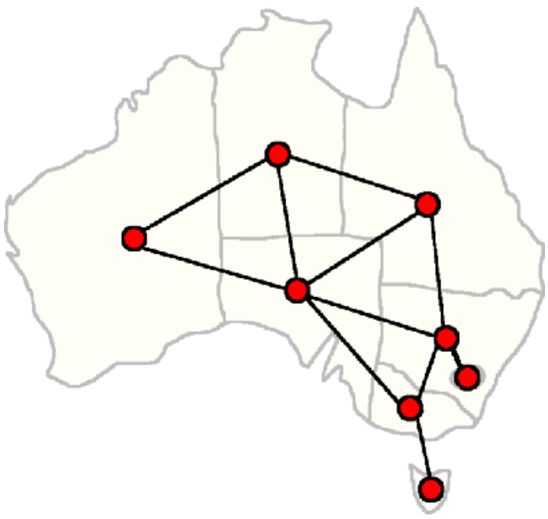


Figure 3.1: Australia's state graph.

When colouring a map, it is visually helpful to assign adjacent countries (or states) different colours. Adjacency is usually taken in the literal physical sense: if two states share a border then they are adjacent. In 1852, Francis Guthrie [12, p121] raised a seemingly simple question: *What is the minimum number of colours needed to colour any map?* A variant of this question, known as the Four Colour Conjecture, became one of the most popular problems in graph theory and was unsolved until the 1970s. It is now known that four colours is the minimum number required to colour all planar[1] graphs. A brief history of this problem is presented by Mitchem [28].

The abstraction of this problem to graph colouring is trivial and an example is shown in Figure 3.1. Here we represent states as vertices and have edges between adjacent state's corresponding vertices (with the assumption that Tasmania and Victoria are adjacent).

[1] A graph is *planar* if and only if it can be drawn on a plane such that no edges cross each other.

Graph colouring has numerous applications, primarily within problems of resource allocation. Welsh and Powell [48] originally identified the relationship between graph colouring and scheduling or timetabling. Consider an exam timetabling problem where a set of students have some exams to sit on one of q days. Let the vertices of the graph represent exams and let an edge connect two exams if a student has to sit both of them. A proper colouring of this graph using the days as colours represents an exam timetable that guarantees a student will not have two exams in one day. These ideas have been refined and developed into complete systems for timetabling (see e.g., [7]).

Chaitin *et al.* [10] demonstrated a method of register allocation using graph colouring. In the method the vertices represent variables or virtual machine registers and edges indicate *interference*[2] between these. If the graph has a k-colouring then the program can be compiled using k registers.

We now present the chromatic polynomial, a useful object in the study of graph colouring and the basis of many generalisations.

[2]Two variables interfere if they are in use at the same time (i.e., in the same code block or module).

Chapter 4

The Chromatic Polynomial and Generalisations

4.1 Chromatic Polynomial

The chromatic polynomial $P(G, q) : \mathbb{N} \to \mathbb{N} \cup \{0\}$ counts the number of q-colourings of the graph G.[1] More formally, the chromatic polynomial of a graph G is defined as,

$$P(G; q) = |\{\sigma : \sigma \text{ is a proper } q\text{-colouring}\}|. \tag{4.1}$$

The proof that this function is a polynomial is trivial, and can be found in any graph theory textbook. It is useful to extend the domain and codomain of this function to $P(G, z) : \mathbb{C} \to \mathbb{C}$ for analytical simplicity.

P can be calculated for any graph using the *contraction–deletion* relationship. Consider any edge $e \in E(G)$. Then we have the simple relation $P(G, \lambda) = P(G \backslash \{e\}, \lambda) - P(G/\{e\}, \lambda)$. This provides us with a simple (yet time consuming) way of calculating P for *any* graph.

The chromatic polynomial was introduced in 1912 by Birkhoff [5], possibly as a way to attack the, then unsolved, four colour problem. He also presented the first proof that it is a *polynomial*. Later, Hassler Whitney, a student of Birkhoff, derived the *subgraph expansion* [50] of the chromatic polynomial:

$$P(G; z) = \sum_{i,j} (-1)^{i+j} m_{ij} z^{n-i}, \tag{4.2}$$

where n is the number of vertices in G and m_{ij} counts the number of subgraphs of G with rank i and nullity j (the *nullity* is equal to the number of edges minus the rank). After a trivial expansion this can be expressed in the following, modern, form[2]:

$$P(G; z) = z^{k(G)} \sum_{X \subseteq E(G)} (-1)^{|X|} z^{\rho(G) - \rho(X)}, \tag{4.3}$$

where $k(G)$ is the number of connected components of G and $\rho(X)$ is the rank of $G\langle X \rangle$. This expression provides a starting point for the generalisation to arbitrary Boolean functions discussed later on.

[1] For a good introduction see the paper by Read [33], it presents a simple introduction to chromatic polynomials, is well written, and is accessible to a wide audience.

[2] Appendix B presents a derivation of the subgraph expansion proposed by Whitney.

4.2 Functional Generalisations

Generalisation is a useful and common strategy in mathematics. Some generalisations of the chromatic polynomial are now discussed. We make the distinction between *functional* generalisations, where the focus is on the polynomial itself, and *structural* generalisations, where the graph itself is generalised.

4.2.1 Whitney Rank Generating Function and Tutte Polynomial

Two extremely useful generalisations of the chromatic polynomial are the *Whitney rank generating function*, $R(G;x,y)$, and the *Tutte polynomial*, $T(G;x,y)$. These two functions contain a large amount of information about a graph and are, surprisingly, almost equivalent.

The Whitney rank generating function is a direct generalisation of (4.2). The coefficients (m_{ij}) of (4.2) were extensively investigated by Whitney [49]. While not specifically giving the polynomial in his paper, according to Farr, it *"seems fairly likely that Whitney thought of his numbers m_{ij} as the coefficients of a two-variable polynomial"* [16, p6]. The form of this polynomial, if present, would have been:

$$R(G;x,y) = \sum_{i,j} m_{ij} x^{n-i} y^j.$$

Using a different approach to that of Whitney, Tutte [44] introduced the *dichromate*, now known as the *Tutte polynomial*. He considered the dichromate as a generalisation of both the chromatic and flow polynomials[3] and defined it in terms of internal and external activities of spanning trees. More precisely, the dichromate is defined as:

$$\chi(G;x,y,>) = \sum_{T} x^{r(>,T)} y^{s(>,T)},$$

where the sum is over all spanning trees T of G, and $r(>,T)$ and $s(>,T)$ represent the number of edges which are internally and externally active[4] under a specific edge ordering $>$. Interestingly, Tutte showed that the actual edge ordering is insignificant, which leads to the definition of the Tutte polynomial:

$$T(G;x,y) = \chi(G;x,y,>),$$

where $>$ is an *arbitrary* ordering.

Even though the constructions of these polynomials are very different, Tutte showed later [45] that $T(G;x,y) = R(G;x-1,y-1)$. He does this indirectly by presenting the *dichromatic polynomial*, then showing the equivalence of this to both the dichromate and the Whitney rank generating function. The modern form of this polynomial is:

[3] A flow polynomial counts the number of nowhere-zero q-flows of a directed graph. A q-flow is a mapping from edges to $\{0,\ldots,q-1\}$ such that for all vertices the sum of incoming flows equals the sum of outgoing flows. For a planar graph a q-flow generates q-colourings of the dual of that graph.

[4] Assign some edge ordering $>$ to the edges of G. Consider any spanning tree T of G. Any edge e of T is *internally active* in T if all edges in $E(G)\backslash T$ forming a bridge between the two components of $E(T)\backslash\{e\}$ are *less than* e (under the ordering $>$). Any edge f of $E(G)\backslash T$ is *externally active* in T if all the edges in the unique path $P \subseteq T$, where $P \cup \{f\}$ is the unique cycle of $T \cup \{f\}$, are less than f.

$$R(G;x,y)=\sum_{X\subseteq E(G)} x^{\rho(E)-\rho(X)}y^{|X|-\rho(X)}. \tag{4.4}$$

(4.5)

The Tutte polynomial (and equivalently the Whitney rank generating function) contains a large amount of information about a graph. It can be specialised to the chromatic polynomial, the flow polynomial, the Potts model, and other useful functions. Certain evaluations of the function also reveals important properties such as the number of spanning trees, forests, and colourings of a graph. For example, the relationship between the chromatic polynomial and the Tutte polynomial is:

$$P(G;z)=(-1)^{\rho(G)}z^{k(G)}R(G;-z,-1).$$

(The reader is encouraged to verify this by examining the relationship between (4.3) and (4.4).)

The multivariate form of the Tutte polynomial was recently reviewed by Sokal [39]. This is a form of the polynomial where the second variable y is replaced by $|E|$ variables, each representing an edge weight. A very recent paper by Farr [16] reviews some history of the Tutte polynomial and Whitney rank generating function and generalisations.

4.2.2 Potts Model

A useful model in statistical mechanics is known as the Potts model, originally introduced by Potts [31]. This model is useful because it provides a mathematical method of studying atomic lattices and phenomena such as *phase transitions*, *per site free energy*, *per site magnetisation* (see e.g., [53, p237]). A tutorial by Wu [53] gives a complete introduction to the Potts model, however it requires a certain level of physics knowledge to understand.

The *Potts model partition function* is a multivariate polynomial used to study the Potts model and can be specialised to the chromatic polynomial. We will consider a special case of the Potts model by limiting the interactions to *two–site interactions* (limiting the interactions to pairs of vertices, see e.g., [53, $\S II.A$]). Sokal's introduction, "Chromatic polynomials, Potts models and all that" [37], neatly explains the relationship between the two–site interaction Potts model partition function and the chromatic polynomial. Summarised, consider a graph with weighted edges and let v_e represent the weight of edge e. The partition function is then given by:

$$Z(G;q,\{v_e\})=\sum_{\sigma:V\to\{1,\ldots,q\}}\prod_{ab\in E(G)}\left[1+v_e\delta(\sigma(a),\sigma(b))\right],$$

where σ is a (not necessarily proper) q-colouring of G, and δ is the Kronecker delta[5]. If we assign to each edge a weight of -1 we obtain the chromatic polynomial,

$$\begin{aligned} Z(G;q,\{-1,\ldots,-1\}) &= \sum_{\sigma}\prod_{ab\in E(G)}\left[1-\delta(\sigma(a),\sigma(b))\right] \\ &= \#\text{ of proper colourings on } G \\ &= P(G;q). \end{aligned}$$

[5] $\delta(x,y)=\begin{cases} 1, & \text{if } x=y; \\ 0, & \text{otherwise.}\end{cases}$

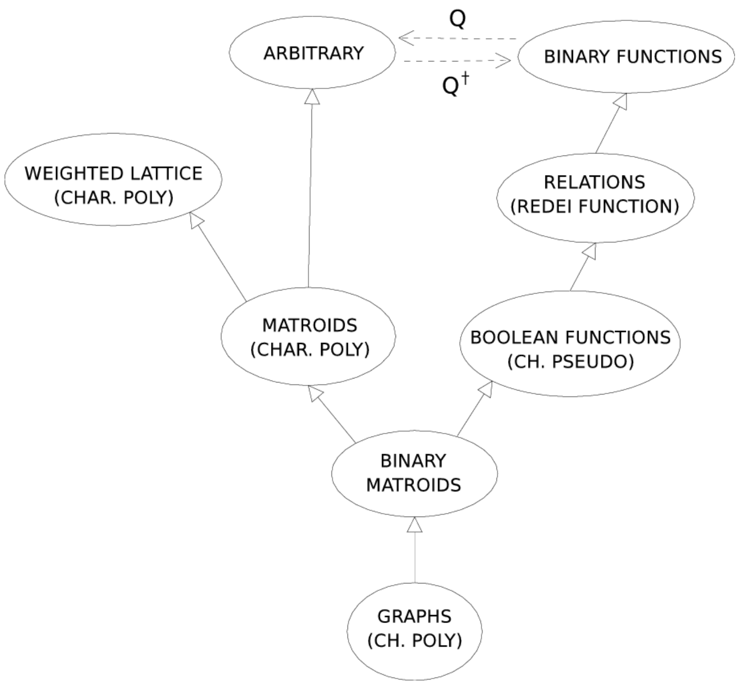


Figure 4.1: This diagram illustrates some structural generalisations of graphs and the chromatic polynomial. Boolean functions are more general than binary matroids, but less general than arbitrary relations. Binary functions, $f : 2^S \to \mathbb{R}$, are even more general, and under Farr's Q transform, shown to be equivalent to structures with arbitrary rank functions.

4.3 Structural Generalisations

The chromatic polynomial has been generalised to other structures such as matroids [11], relations (see the Rédei function below), weighted lattices [52], and Boolean functions [14, 15]. It is this latter generalisation that is the focus of this project. Figure 4.1 illustrates the relationships among these generalisations. For a review of important generalisations of the Tutte polynomial consult Farr's recent paper [16].

4.3.1 Chromatic Pseudonomial of Boolean Functions

In 1992, Farr [14, 15], extending Kung's work (presented below in §4.3.1), introduced the Whitney quasi-rank generating function which generalised the Whitney rank generating function to arbitrary Boolean functions. The key in his approach was to express the rank of a graph in terms of the indicator function of its cutset space, which may be regarded as a Boolean function [16, §6]. This rank is then generalised to a *quasi-rank* of an *arbitrary* Boolean function (Farr's Qf function [14]). Farr then defined the *Whitney quasi-rank generating function*, *Tutte function* [14, §5], and the *chromatic function* [14, §8], for Boolean functions. Farr's chromatic function is the focus of this project (and termed the chromatic pseudonomial in this review).

The chromatic pseudonomial for arbitrary Boolean functions $f : 2^S \to \{0,1\}$, with the restriction that $f(\emptyset) = 1$, is:

$$P(f; z) = \sum_{X \in S} (-1)^{|X|} z^{\log_2 \sum_{Y \subseteq \bar{X}} f(Y)}.$$

If f is the indicator function of the cutset space of a graph G then we have $P(f; z) = z^{-k(G)} P(G; z)$. Note that $P(f; z)$, in general, is a sum of irrational powers of z and therefore is not a polynomial but a *generalised polynomial* (see §5.3.3). The name *chromatic pseudonomial* was given to this class by the author to indicate their close correspondence with polynomials.

An Interpretation by Kung

The chromatic polynomial for graphs was generalised by Kung [24] (see also [25]) to arbitrary relations. Here Kung defines the *Rédei function* on a relation $R \subseteq S \times T$ as:

$$\zeta(n) = \#n\text{-tuples distinguishing S in T}.$$

An n-tuple $u = (u_1, \dots, u_n)$ of elements from T formally *distinguishes* S if $\bigcap_{i=1}^{n} u_i^{\perp} = \emptyset$, where $u_i^{\perp} = \{x \in S : (x, u_i) \notin R\}$. The Rédei Zeta function, $\zeta_{S|T}(n)$, gives the number of $n-$tuples in T distinguishing S with respect to some relationship $R \subseteq S \times T$. Consider an arbitrary graph G and its cutset space $C \subseteq 2^{E(G)}$. $\zeta_{E|C}(n)$ is then directly related to the number of 2^n colourings of G as follows.

From the definition of *distinguishing* and the relation $\in$ we have,

$$\begin{aligned}\zeta_{E|C}(n) &= |\{n-\text{tuple from } C \text{ distinguishing } E\}| \\ &= |\{u = (u_1, \dots, u_n) \in T^n \text{ s.t. } \bigcup_i u_i = E\}|.\end{aligned}$$

The close relationship between $P(f; 2^z)$ and $\zeta(z)$ is discussed in Appendix C. This relationship provides a meaningful interpretation of Boolean function colourings:

$$P(f; 2^n) = \ \# \text{ of } n-\text{tuples of elements in } \operatorname{supp} f \text{ whose union is } S.$$

This relationship also led to the development of EC–graphs discussed in Section 7.4.2.

Chapter 5

Chromatic Zeros

Zeros of functions are studied extensively in mathematics. The location of the zeros of a function often reveals intricate structure or some property that is not apparent from the syntactic form of the function. This section is concerned with *chromatic* zeros (i.e., zeros of chromatic polynomials).

Some motivation is first presented (§5.1), followed by a review of the current theory of chromatic zeros (§5.2), and finally a review of various methods available for locating these zeros.

5.1 Some Cross Discipline Motivation

Sokal's multivariate Tutte polynomial paper contains a section appropriately titled *"Complex zeros of Z_G: Why should we care?"* [39, §5] (the Z_G he is referring to is the Potts model partition function discussed in §4.2.2.) That section provides two main reasons for studying the complex zeros of this function:

- Mathematical inquiry, and
- applicability to statistical mechanics.

The first is the fundamental driver of modern mathematics and needs no explanation. The second of these reasons, however, relates back to the Potts model. The complex zeros of this model have very important implications in statistical mechanics and *phase transitions*. More specifically the *"possible points of physical phase transitions are precisely the real limit points of (the) complex zeros."* [39, p194]. The Potts model however is just a special case of the multivariate Tutte polynomial and can be specialised to the chromatic polynomial. By investigating the zeros of the chromatic polynomial some results may generalise back to the Potts model.

5.2 The Theory of Chromatic Zeros

Research into chromatic zeros is currently active and there exists a large theory concerning the locations of these zeros. We now present and classify these theorems and conjectures into four groups:

- Those that constrain the locations of *real* zeros,
- Those that constrain the location of *complex* zeros,
- Those that identify special values that zeros group around, and

- Those that are concerned with families of graphs.

5.2.1 Real Zeros

The study of real zeros was driven by the Four Colour problem. From this it has developed into an interesting theory, some recent and not so recent results include:

- Real chromatic roots of graphs are either integral or irrational (due to the Rational Root Theorem, see e.g., [17]).
- Real chromatic roots of graphs are dense everywhere in $[\frac{32}{27}, \infty)$ [21, Thm. 4(a)].
- Real chromatic roots of planar graphs are dense everywhere in $[\frac{32}{27}, 3]$ [21, Thm. 11].
- The maximal zero-free intervals for chromatic roots are $(-\infty, 0)$, $(0, 1)$ and $(1, 32/27]$. [42]

5.2.2 Complex Zeros

A recent paper by Jackson [21] presented a survey of modern theorems and conjectures regarding the distribution of zeros of chromatic polynomials and flow polynomials. This work is extensive and details overs 23 theorems and conjectures about chromatic polynomials.

Generalised theta graphs are graphs with two *end points* connected via paths of varying lengths. Recently Sokal *et al.* analytically proved that all the chromatic roots of these graphs lie within a certain disc [6]. Sokal also recently proved [38] that there exists a constant $C(r) < \infty$ such that for all loopless graphs of maximum degree less than or equal to r the chromatic roots z all lie within the disc $|z| \leq C(r)$, furthermore, he showed $C(r) \leq 7.963907r$. This proves a weaker version of a conjecture made by Read and Royle [34, p1027] that all roots of cubic graphs lie in the disc $|z| \leq 3$.

A summary of these theorems are:

- The cumulative complex chromatic roots of all graphs are dense everywhere in the complex plane [21, Thm. 4(b)].
- All chromatic roots of generalised theta graphs lie in the disc $|z - 1| \leq (1 + o(1))k/\log k$ [6].
- For all graphs with maximum degree, r, and zero z we have $|z| \leq C(r) \leq 7.963907r$ for some constant $C(r)$ [38].

Sokal also conjectured:

- Compared to other Theta graphs, chromatic roots z of Theta graphs with path lengths all equal to 2 maximise $|z - 1|$ [6].

5.2.3 Popular Roots

In 1969, Berman and Tutte [2] wrote a small paper that presented some interesting results and posed some intriguing questions. They found that a large class of planar graphs had real chromatic roots close to the golden ratio. Following this up, Tutte [46] showed that for the class of *2-sphere triangulations* (also known as *wheels*) M we have

$$|P(M; 1 + \phi)| \leq \phi^{5-k},$$

where $\phi = 1/2(1 + \sqrt{5})$ is the golden ratio and k is the number of vertices. This explained the empirical observations made in the previous paper. Similar results by Beraha *et al.* [1] extended this work and led the way to theory of limits of complex zeros of recursive families reviewed later.

Farrell [17] presented some interesting theory based on empirical observations. He made numerous conjectures, some of which have since proven to be false (e.g., [17, Conj. 1] which conjectures that all roots have positive real part is false as chromatic roots are dense in the complex plane) and observed some interesting phenomena. He observed that some zeros are *"more popular"* than others, more specifically the sequence: $\{3/2 \pm i\sqrt{3}/2, 2 \pm i, 5/2 \pm i\sqrt{3}/2, \ldots\}$.

5.2.4 Families of Graphs

Recently, progress has been made in the theory of zeros of recursive families[1]. Chromatic zeros of recursive families of graphs obey a limiting process demonstrated by Read and Royle [34], building on new theory developed by Beraha *et al.* [1]. In this they explain observations made by Biggs *et al.* [3] by showing that the zeros of ladders (see e.g., Figure 5.2.4) all converge to one of three curves. They also noted [34, §6] that extra behaviour is not explained by the new theory, specifically the behaviour of the curves that the zeros seem to follow while converging to the limit curves. Further observations in the same paper suggest that this theory *may* extend to non-recursive families of graphs.

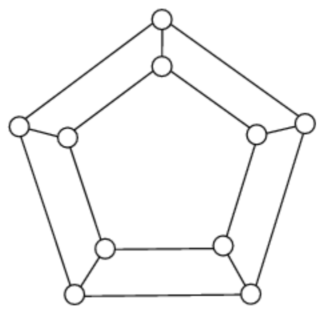


Figure 5.1: A ladder of degree 5.

Shrock and Tsai [36] prove that a certain family of graphs have zeros lying on circles and relate the results to a certain Potts model. This paper provides an interesting link between chromatic zeros and the Potts model on a certain class of graphs.

This research, while providing a handy theory of chromatic zeros to possibly generalise, is important because it shows that a lot of research into chromatic zeros follows a common process. First, empirical data is collected by, for example, enumerating a class of graphs and then collecting, and possibly plotting, the chromatic zeros. Conjectures are then made by analysing the location of the zeros and proofs are attempted. This general process will be followed during the course of this project.

5.3 Locating Zeros

A major aspect of this project is to efficiently, and to some degree of accuracy, locate the real and complex zeros of a chromatic pseudonomial $p(z)$. Some general root finding techniques for both polynomials and arbitrary functions are discussed in Section 5.3.1. As $p(z)$ is defined on the

[1]A sequence of graphs $\{G_1, \ldots, G_n\}$ is a recursive family if $P(G_n; n) = f(P(G_{n-1}), \ldots, P(G_1))$, where f is some simple linear function.

complex plane we can transform the zero finding problem into one of local minima search — this approach is discussed and various methods are reviewed in Section 5.3.2. Many techniques have been developed and refined for locating zeros of a polynomial and hence a plausible method of locating zeros of p would be to approximate it with some polynomial q. The zeros of q could then provide information about the zeros of p. Two methods illustrating this approach are discussed in Section 5.3.3. The section also discusses some recent results regarding generalised polynomials.

5.3.1 General Root Finding

Many methods exist for determining the zeros of a polynomial. Henrici [20, §6] presents several of these methods. They vary from algorithms that find one zero at a time, to algorithms that locate all zeros simultaneously [20, §6.11]. These methods are specifically designed for polynomials and it is not likely that these approaches are suitable for this project.

In the same chapter he presents some common iteration algorithms [20, §6.12] such as Newton's method, Schröder's iteration functions, and Laguerre iteration. These iterating methods work by creating a sequence (z_i) from an initial and user supplied z_0, such that $(p(z_i))$ converges to zero. The differences between the methods are in the speed at which they converge and the complexity of implementation. The benefit of these methods is that they work for *any* analytic function and thus can be immediately applied to find zeros of our chromatic pseudonomials.

Another general root–finding method uses the of homotopic continuation (see e.g., [40]). This method was implemented as the primary zero–finding program and is discussed later (§6.4).

5.3.2 Function Minimisation Techniques

Due to the *Minimum Modulus Theorem* (see e.g., [41, Thm. 10.15]) we have the useful property that $g(z) = |p(z)|$ is 0 wherever a local minimum of $g(z)$ occurs. Therefore we can use any function *minimisation* algorithm as a *zero finding* algorithm.

Methods of descent find local minima by constructing a sequence (z_i), such that each z_{i+1} lies *lower* than z_i (i.e. $g(z_{i+1}) < g(z_i)$). Henrici [20, §6.14] formalises the method of descent as follows:

Given an analytic function $p(z)$ and a *descent function* $f(z)$ with the properties:

$$\begin{aligned} f(T) &\subset T, \\ p(z) \neq 0 &\Rightarrow |p(f(z))| < |p(z)|, \\ p(z) = 0 &\Rightarrow f(z) = z, \end{aligned}$$

where T is a closed set, then for any $z_0 \in T$, the sequence $\{z_n\}$, where $z_{n+1} = f(z_n)$, converges to a zero of p. The proof [20, Thm6.14c] of this is specific to polynomials, however it seems trivial to extend this result to arbitrary analytic functions.

The primary problem with this method is the determination of a suitable descent function. Using the *steepest descent*[2] will not always result in a descent function (note that this is equivalent to Newton's method [20, p544]). This approach is also known as the *conjugate gradient method* [32, §10.6] and *Cauchy's method of steepest descent* [23, p991]. This method is sufficiently general to apply to pseudonomials and was implemented as part of the supporting zero–finding software developed (§6.4).

[2]Using the direction of steepest gradient (i.e., $f(z_{n+1}) = f(z_n) - t\nabla(|p|)$, for some appropriate t).

Some other common descent functions are Ward's method, Nickel's method, and Kellenberger's method [20, pp547–549]. Henrici notes that Ward's method is not really suitable as it does not possess the descent property at certain points, however it can be used in the non-polynomial case and thus may be considered as a viable "quick and dirty" method. Nickel's and Kellenberger's approaches, however, rely on specific properties of polynomials, and it is not clear whether these methods will generalise to chromatic pseudonomials.

Some other iterative techniques of function minimisation can be found in [32, §10]. These techniques are applicable to any continuous function of arbitrary dimension. Two notable algorithms are Nelder and Mead's *Downhill Simplex Method* and *Powell's method*. The downhill simplex method (see e.g., [32, §10.4]), in the two dimensional case, constructs a triangle (known as a *simplex*) of points and at each iteration manipulates (using a series of reflections, scalings and other transformations) the triangle down the slopes until it reaches a local minimum. Powell's method (see e.g., [32, §10.5]) is based on constructing lines to evaluate the function along and applying single dimensional minimisation techniques on this line. Both these methods show the broad range of techniques available for function minimisation and seem easy to implement and apply to chromatic pseudonomials. Due to the focus of this project these methods were not implemented, as the already developed programs were sufficiently fast and accurate.

It was considered that plotting the modulus of a complex function f would provide a simple method of visualising the function and (visually) locating the roots. We can plot the function $|f(z)|$ as a surface in three dimensions. This approach was first done computationally in [27]. Due to the minimum modulus principal any *dip* in the surface reveals the location of a zero.

5.3.3 Generalised Polynomials

Polynomials can be generalised by allowing exponents to be non-integral. These functions are called *generalised polynomials* and have not received much attention in the mathematics community. Chromatic pseudonomials are a subclass of generalised polynomials and thus we look at some results for generalised polynomials, hoping that they will present both applicable theorems *and* techniques for extending theorems about polynomials.

Recently [43], Tossava presented a simple proof that all functions $f : \mathbb{R} \to \mathbb{R}$ of the form

$$f(t) = \sum_{j=0}^{n} \alpha_j p_j^t \tag{5.1}$$

have at most n real zeros. Consider a generalised polynomial $g : \mathbb{R} \to \mathbb{R}$,

$$g(t) = \sum_{j=1}^{n} \alpha_j t^{p_j}. \tag{5.2}$$

Restricting the domain of g to the positive reals and using the transform $t = e^x$ and $b_j = e^{p_j}$ we have

$$G(x) = g(e^x) = \sum_{j=1}^{n} \alpha_j b_j^x$$

which is of the same form as required in (5.1). The number of zeros of G is then bounded above by $n - 1$, which is also the bound on the number of *positive* zeros of g. Jameson [22] reviews an alternative approach (due to Laguerre) and bounds the number of positive real zeros of (5.2) by $n - 1$. These results, while handy, do not provide methods for locating the zeros and are restricted to the positive reals. However, they do support the theoretical basis and support some proofs presented later on.

Van De Lune and Te Riele [47] used a different, iterative, approach to locate the complex zeros of a finite version of the Riemann zeta function (of the same form as (5.1)). Lapidus and van Frankenhuysen demonstrate a technique in [26] for finding the complex zeros of Dirichlet polynomials[3]. These polynomials are easily transformed into generalised polynomials. The technique approximates the equation using polynomials of higher and higher degree by finding rational approximations of the irrational exponents with common denominators. These methods have a major drawback — they require a polynomial of extremely large degree to obtain suitable results, which require an extremely long computation time when locating their roots. This method does not seem to be suitable for chromatic pseudonomials because of this.

Castensen and Petkovic [8] present a novel method for simultaneously determining the real zeros of algebraic (i.e., normal), exponential, and trigonometric polynomials. Carstensen and Reinders [9] (extending Frommer [18]) give a method for the simultaneous determination of all zeros of generalised polynomials. These results are not directly applicable to finding complex zeros of pseudonomials (as they do not fall directly into the classes under consideration), however, their method does implicitly use the same idea of homotopic continuation which was used in this project.

[3]Dirichlet polynomials are functions of the form $f(z) = \sum_{i=1}^{n} a_i e^{p_j z}$. Note that $f(\ln z)$ is a pseudonomial if the p_j are equal to $\log k_j$, for integral k_j.

Chapter 6

Algorithms and Software

The specific zero–finding methods reviewed in Section 5.3.3 had various disadvantages or were overly complex for the requirements of this project. The key features required in the zero–finding algorithm were that it was fast, and not overly complex. A tradeoff between speed, simplicity, and numerical accuracy was made, with the implemented methods favouring the first two. The programs developed are reasonably fast and were used to get a qualitative view of the chromatic zero distributions of various classes of functions. They will not necessarily locate *all* the chromatic zeros of a given function, and may have a small amount of error involved, however, as mentioned above this was part of the tradeoff.

No existing software is available (at the time of writing) that locates zeros of generalised polynomials. All the existing zero–finding packages primarily found roots of polynomial, or found zeros of functions by using large Taylor–series approximations. The nature of the pseudonomials dictates that a Taylor–series approximation will always be a bad approximation.

The software developed to support this investigation consists of a set of programs written in C++ and tested under the Linux operating system. It was designed with usability and extensibility in mind, supporting these qualities primarily through modularisation, object–oriented principals, a useful command–line help system, and a progress bar. The programs implemented will now be summarised and the core algorithmic ideas presented. Consult Appendix D for more information regarding the availability and use of this software.

6.1 `chromatic`: Chromatic Polynomial Generator

This program takes a Boolean function specification (see Appendix D) and outputs the chromatic pseudonomial of that function. The output is available in latex or ascii format. `chromatic` can also enumerate a range of functions. Table 6.1 presents some sample output. The implementation is in the function `Function::chromatic()` in `Fun.cpp`.

11101000	$-1 + 3\lambda - 3\lambda^{\log_2 3} + \lambda^2$
10011000	$1 - 2\lambda + \lambda^{\log_2 3}$
11011000	$\lambda - 2\lambda^{\log_2 3} + \lambda^2$
10111000	$\lambda - 2\lambda^{\log_2 3} + \lambda^2$
11111000	$-1 + 3\lambda - 2\lambda^{\log_2 3} - \lambda^2 + \lambda^{\log_2 5}$
10100100	$1 - 2\lambda + \lambda^{\log_2 3}$
11100100	$\lambda - 2\lambda^{\log_2 3} + \lambda^2$

Table 6.1: Sample output from `chromatic`

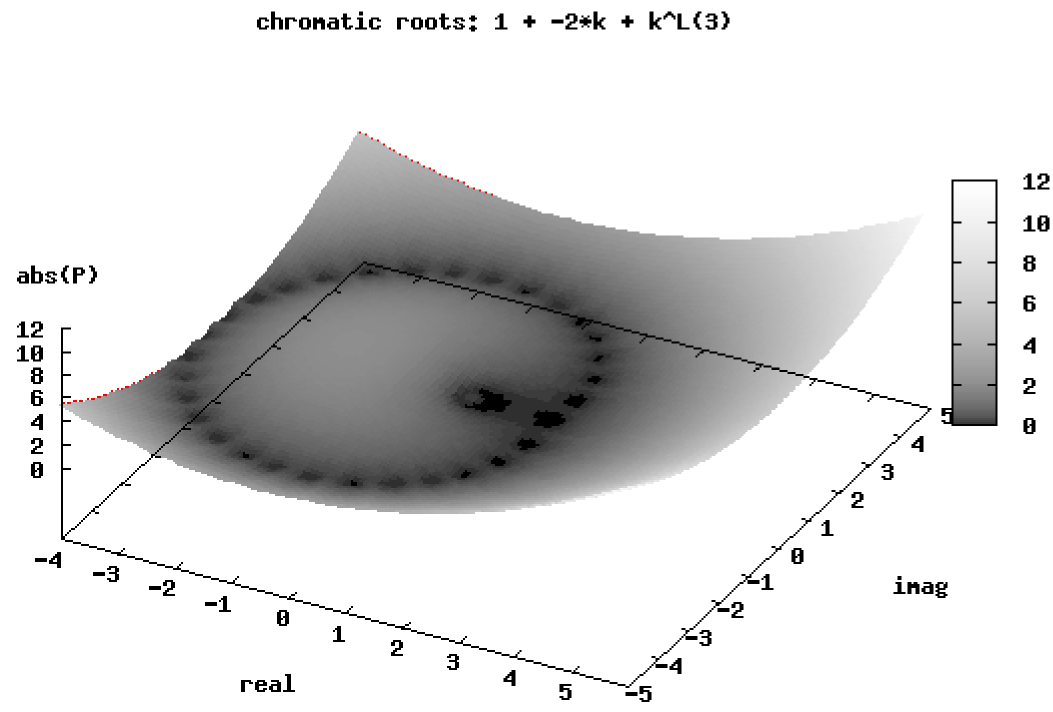


Figure 6.1: Surface plot of $\min_{b \in \text{branches}} |P_b(AT_2; z)|$.

6.2 `sample`: Surface Visualiser

This utility takes a function, f, as a parameter and plots

$$\min_{b \in \text{branches}} |P_b(f; z)|,$$

over some specified region (see Figure 6.1). This was used in the preliminary stages of the investigation to get a general overview of the nature of pseudonomials.

6.3 `real`: Real Zero Finder

`real` can either plot a chromatic pseudonomial over a positive real interval, or it can locate the principal chromatic zeros in a positive real interval. The zero–locating functionality is implemented using a combination of the bisection method and golden section search. We can use this to verify that certain intervals are zero–free, for example, Figure 6.2 demonstrates that there are no chromatic zeros in the interval $(0, 1)$ for the specific function considered.

6.4 `iterative`: Steepest Descent Iterator

`iterative` implements Newton's method to locate complex zeros of pseudonomials (the use of function minimisation as zero location was discussed in Section 5.3.2). The starting points given to the minimisation algorithm are sampled uniformly over a user–specified region. The user can

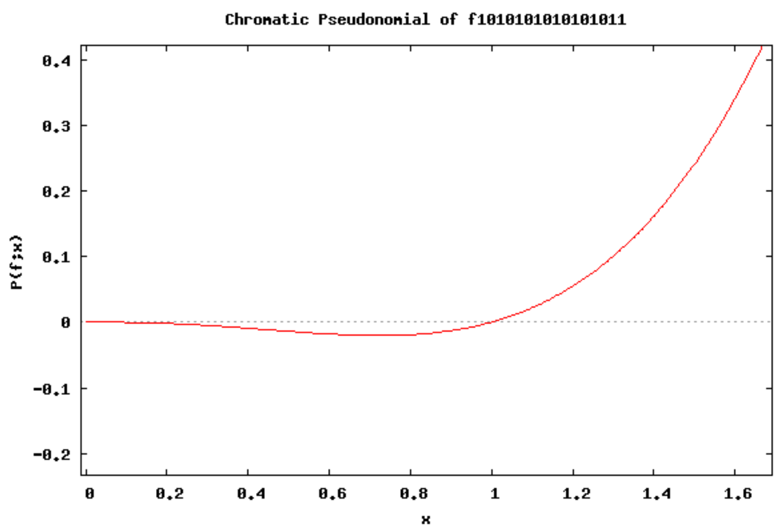


Figure 6.2: Output of `real` (plotted with Gnuplot).

control various parameters of the algorithm via command–line arguments. The possible parameters are:

- Dimensions and location of rectangular sample region;
- Maximum number of iterations;
- Accuracy;
- Number of sample points;
- Step size;
- Number of branches.

6.5 `pc`: Predictor–Corrector

`pc` is the primary zero finding program used to produce most of the plots in this thesis. The program implements a predictor–corrector and is based on the principal of homotopic continuation (specifically the exact algorithm given in [40, p23,§2]). The core implementation is in the functions `predictor_corrector` and `iterative1` in `PC.cpp`. The code is also included in Appendix F for reference. Various user–definable parameters are available, including number of iterations, step size and precision. The basic algorithm is described in Appendix E.

6.5.1 Discussion

The predictor–corrector is still a naïve algorithm, however it was found to be sufficient for the requirements of this project. The main two problems with this method are:

- Accuracy: Newton's iteration has trouble when the local region is "flat". This is evident in Figure 6.3 where some false zeros have occurred. These are not zeros of the function, and the method fails to converge to the zero at 0.

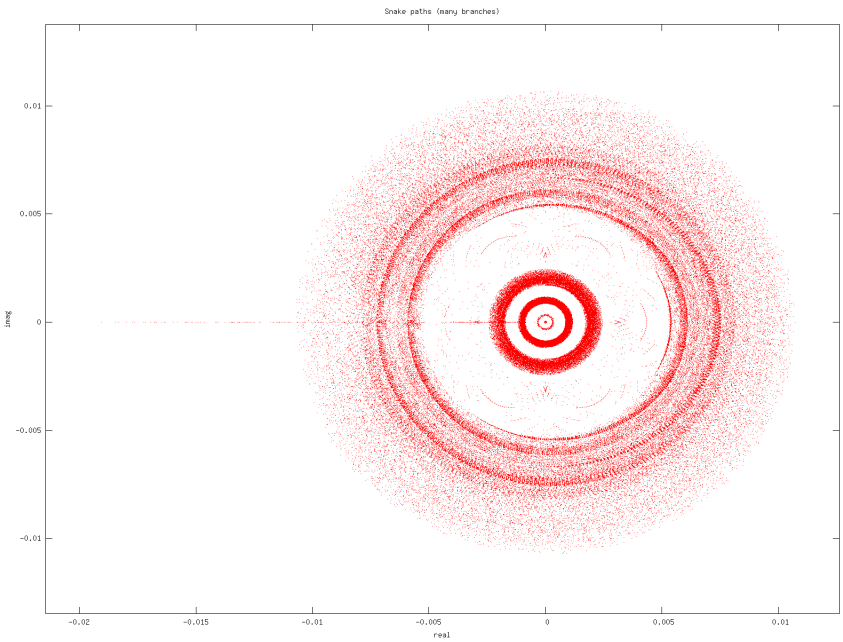


Figure 6.3: Some false zeros cause by numerical error of Newton iteration along "flat" regions. The only true zero here is at the origin.

- Number of Zeros: Not all zeros on a single branch are always found. This happens when several unique starting points of the homotopy all converge to the same zero of the pseudonomial in question.

6.6 `tracer`: Zero Trail Tracer

The theory of zero trails is discussed later (§7.1.4). This program locates a principal zero and then proceeds to "trace" out the zero trail that it lies on. The functionality of `tracer` is limited and the algorithm is naïve. Future work in this area is definitely required.

Chapter 7

Results and Discussion

Using the zero–finding software discussed in the previous section, chromatic zeros of classes of Boolean functions were studied. The qualitative and theoretical results are now presented. We first present some basic theoretical work (§7.1). We then present specific classes (§7.3–7.6). An overall discussion on the project is then presented (§7.1.4).

7.1 General Theory

We now consider some general theoretical work done in this project. We first review the multiform nature of pseudonomials and the specific solutions to deal with it, and then present some new theoretical results.

7.1.1 Multiformity

The chromatic pseudonomial is difficult to work with, primarily because it is not polynomial (see Appendix M for a list of different ones). Consider a simple example of a function with a chromatic pseudonomial with a single irrational exponent. Let $S = \{a, b, \}$ and f be the function such that $\mathrm{supp}\, f = \{\emptyset, \{a\}, \{b\}\}$, and hence $P(f; z) = 1 - 2z + z^{\log 3}$. Taking $z \in \mathbb{C}$ results in

$$P(f; z) = 1 - 2z + |z|^{\log 3} e^{i.\log 3.(\arg z)},$$

which is multivalued (as $\arg z$ is multivalued), for example,

$$\begin{aligned} P(f; 1) &= \{-1 + e^{i.\log 3.2k\pi} : k \in \mathbb{Z}\} \\ &= \{0, -1 + e^{i.2\pi.\log 3}, -1 + e^{i.4\pi.\log 3}, \ldots\}. \end{aligned}$$

In fact, more surprisingly, $P(f; z)$ evaluates to a countably infinite set of values. [1]

To deal with this technicality we expand the complex plane to that of a Riemann surface. Practically this just results in the addition to extra parameters to the pseudonomial, one for each irrational exponent. We denote these, integral, parameters using a subscript notation. For example,

[1] We can prove this as follows. Consider P evaluated on some branch n, first we expand the exponential term in P:

$$e^{i.\log 3.(Arg(z)+2n\pi)} = e^{i.\log 3.Arg(z)} e^{i.\log 3.2n\pi}.$$

We now show that for $n \neq m$ we have $e^{i.\log 3.2n\pi} \neq e^{i.\log 3.2m\pi}$. This can only occur if $\log 3.2m\pi - 2k\pi = \log 3.2n\pi$, for some $k \in \mathbb{Z}$. Rearranging the terms gives $\log 3(m - n) = k$, which is a contradiction as $\log 3$ is irrational. Therefore P evaluates to countably infinite unique values.

the above pseudonomial is transformed into:

$$\begin{aligned}P_m(f;z) &= P(f;z) \text{ evaluated on the } m\text{'th branch of the Riemann surface}\\ &= 1 - 2z + |z|^{\log 3} e^{i.\log 3.(Arg(z)+2m\pi)}.\end{aligned}$$

Note that $P_m(f;z)$ is now single–valued. The more general case of a pseudonomial with k irrational exponents is written as $P_{m_1,\dots,m_k}(f;z)$, and the *principal branch* is the branch with $m_i = 0$ for all i. Zeros in the principal branch are called *principal zeros*.

From the result of Laguerre (presented in Jameson [22]) we can bound the number of real zeros in any single branch by the number of sign changes in the pseudonomial (where the terms are ordered from maximum exponent to minimum exponent). Section 5.3.3 reviewed some work regarding zeros of this class of functions, however, this general class is very difficult to work with. Only recently has some progress been made in this area, and hence, a large amount of effort was put into this project trying to locate and formulate necessary theoretical results.

7.1.2 Zero Pseudonomials and Loops

The Boolean function generalisation has its origin in graph theory. We use the graph analogy to present a useful result.

A loop in a graph is an edge that has identical endpoints. The effect of the existence of a loop is two–fold:

- $P(G;z) = 0$ (i.e., the graph is uncolourable);
- The loop is not part of any member of the cutset space.

These two events are logically equivalent, and it is natural to explore if this notion is generalisable to Boolean function colouring.

Definition: Loop
A Boolean function, f, has a *loop* if, and only if, there exists a variable in the ground set that does not appear in any member of $\operatorname{supp} f$.

Theorem 1. *Let f be a valid Boolean function on ground set S. Then f has a loop if and only if $P(f;z) = 0$.*

Proof. We prove this in both directions.

($\Rightarrow$) Let v be a loop of f. We then have

$$\begin{aligned}P(f;z) &= \sum_{X\subseteq S} (-1)^{|\bar{X}|} z^{\log F(X)}, \text{ where } F(X) = \sum_{Y\subseteq X} f(Y)\\ &= \underbrace{\sum_{X\subseteq S\backslash\{v\}} (-1)^{|\bar{X}|} z^{\log F(X)}}_{\text{A}} + \underbrace{\sum_{X\subseteq S\backslash\{v\}} (-1)^{|\overline{X\cup\{v\}}|} z^{\log F(X\cup\{v\})}}_{\text{B}}.\end{aligned}$$

But

$$\sum_{Y\subseteq X\cup\{v\}} f(Y) = \sum_{Y\subseteq X} f(Y) + \underbrace{\sum_{Y\subseteq X} f(Y\cup\{v\})}_{=0},$$

since v is a loop, and hence $F(X) = F(X\cup\{v\})$. Therefore $B = -A$ and $P(f;z) = 0$, as required.

($\Leftarrow$)

$$P(f;z) = 0 \iff \sum_{X \subseteq S} (-1)^{|\bar{X}|} z^{\log F(X)} = 0,$$

with the same $F(X)$ as defined above. This is only true if there exists a set of pairs of subsets of S, $\{(U_i, V_i)\}$, such that for all i, z,

$$(-1)^{|\bar{U}_i|} z^{\log F(U_i)} = -(-1)^{|\bar{V}_i|} z^{\log F(V_i)}$$

and every subset of S appears exactly once. This can be seen to be true because each term of $P(f;z)$ in the sum given above has a coefficient of 1 or -1, and hence we must be able to cancel out by pairs.

We now have a set of pairs $\{(U_i, V_i)\}$. Furthermore without loss of generality we can assume that for all i, $|U_i| = j + |V_i|$ (for some odd j) and $F(U_i) = F(V_i)$ (due to the equality given above).

Now consider what happens when we let $U_k = S$ for some k. Let $V_k = S \backslash T$, for some odd sized $T \subseteq S$. In order for $F(U_k) = F(V_k)$ we must have that:

$$\begin{aligned}
&\sum_{X \subseteq S} f(X) = \sum_{X \subseteq S\backslash T} f(X) \\
\iff &\sum_{X \subseteq S\backslash T} \sum_{Y \subseteq T} f(X \cup Y) = \sum_{X \subseteq S\backslash T} f(X) \\
\iff &\sum_{X \subseteq S\backslash T} f(X) + \sum_{X \subseteq S\backslash T} \sum_{\substack{Y \subseteq T \\ Y \neq \emptyset}} f(X \cup Y) = \sum_{X \subseteq S\backslash T} f(X) \\
\iff &\sum_{X \subseteq S\backslash T} \sum_{\substack{Y \subseteq T \\ Y \neq 0}} f(X \cup Y) = 0 \\
\iff &\forall X \subseteq S\backslash T . \forall Y \subseteq T . Y \neq \emptyset \wedge f(X \cup Y) = 0. \\
\iff &\forall X \subseteq S . X \cap T \neq \emptyset \Rightarrow f(X) = 0.
\end{aligned}$$

Specifically there exists an element in T that is not in any member of $\operatorname{supp} f$. This element is a loop of f. □

Example: Example of f with a loop that is not a graph.

Let $S = \{a, b, c\}$ and $\operatorname{supp} f = \{\emptyset, \{a\}, \{a, b\}\}$. Then, by definition, c is a loop of f. $P(f;z)$ is calculated below, and each individual term of the summation is annotated with the corresponding subset X of S for clarity.

$$P(f;z) = \underbrace{-1}_{\emptyset} + \underbrace{z}_{\{a\}} + \underbrace{1}_{\{b\}} + \underbrace{1}_{\{c\}} - \underbrace{z^{\log 3}}_{\{a,b\}} - \underbrace{z}_{\{a,c\}} - \underbrace{1}_{\{b,c\}} + \underbrace{z^{\log 3}}_{\{a,b,c\}} = 0.$$

We note that the cancelling pairs in this summation have a difference of $\{c\}$.

We can also generalise the component theorem of graph colouring: If H and I are separate components of G, then we have $P(G;z) = P(H;z)P(I;z)$. This property generalises to arbitrary Boolean functions and was shown to hold for $P(f;z)$ by Farr [14, p276] in his original paper. [2]

[2]Where he shows that it holds for the Whitney rank generating function, which subsumes the chromatic pseudonomial case.

7.1.3 A Zero Bound

We now present a new result on the bound of chromatic zeros of Boolean functions.

Theorem 2. *Consider an arbitrary Boolean function f, with $|S| = m$, $k = |\operatorname{supp}(f)|$ and with $k > 2^{m-1}$. Its corresponding chromatic polynomial $P(f; z)$ then has all its zeros lying within the region $|z| < 2^s$, where*

$$s = \frac{m}{1 + \log_2 k - m}.$$

Proof. Let

$$P(f; z) = \underbrace{z^{\log_2 k}}_{A} + \underbrace{\sum_{\substack{X \subseteq S \\ X \neq \emptyset}} (-1)^{|X|} z^{\log_2 \sum_{Y \subseteq \bar{X}} f(Y)}}_{B}.$$

By the triangle inequality we have $||A| - |B|| \leq |P(f; z)|$, and so $|P(f; z)| > 0$ if $|A| > |B|$. We can bound $|B|$ above as follows:

$$\begin{aligned} |B| &= |\sum_{\substack{X \subseteq S \\ X \neq \emptyset}} (-1)^{|X|} z^{\log_2 \sum_{Y \subseteq \bar{X}} f(Y)}| \\ &\overset{\triangle\text{-ineq}}{\leq} \sum_{\substack{X \subseteq S \\ X \neq \emptyset}} |(-1)^{|X|}||z^{\log_2 \sum_{Y \subseteq \bar{X}} f(Y)}| \\ &= \sum_{\substack{X \subseteq S \\ X \neq \emptyset}} |z|^{\log_2 \sum_{Y \subseteq \bar{X}} f(Y)}. \end{aligned}$$

But

$$|z| > 1 \Rightarrow |z|^{\log_2 \sum_{Y \subseteq \bar{X}} f(Y)} \leq |z|^{\log_2 2^{m-1}} = |z|^{m-1},$$

and hence $|B| \leq (2^m - 1)|z|^{m-1} \leq 2^m |z|^{m-1}$.

$$|A| > |B| \text{ if } |z|^{\log_2 k} > 2^m |z|^{m-1},$$

which occurs only if

$$|z| > 2^{\frac{m}{\log_2 k - m + 1}}.$$

□

Unfortunately this bound is not very applicable in practice due to the exponential term. For example, consider the function with support $\operatorname{supp} f = \{\emptyset, \{a\}, \{b\}\}$ on ground set $S = \{a, b\}$. Applying the bound to this gives an outer bound of ≈ 10.7, however it is trivial to show that the chromatic zeros of f are bounded within a radius of 4 from the origin. This bound increases exponentially as the number of elements in the ground set increases.

7.1.4 Zero Trails

Plotting many branches of some functions reveals an interesting property: the zeros seem to follow curving paths. Consider Figure 7.1 for example. The function is the *almost–tree* function (see §7.6) on 3 variables; intriguingly, the plot reveals two nice curves which the zeros appear to follow.

These curves (called *zero trails* from now on) seem to give sufficient information about the locations of the zeros and a parametric (or similar, e.g., level set) representation of them would be ideal. Other considerations regarding these curves include the number of trails and the actual

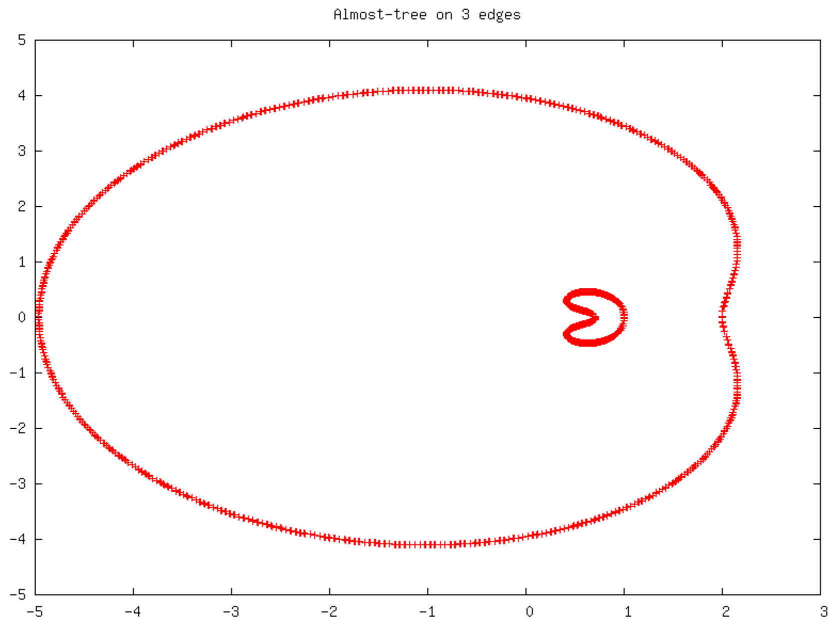


Figure 7.1: Chromatic zeros of the almost–tree on 3 edges. $P(f;\lambda) = -1 + 3\lambda - 3\lambda^2 + \lambda^{\log 7}$

distribution of zeros on these trails. It would also be beneficial to be able to automatically trace these trails.

Zero Trail Examples

The following plots (Figures 7.1, 7.2 and 7.3) demonstrate how the qualitative nature of the chromatic zero distribution differs between pseudonomials with different forms. Figures 7.1 and 7.2 demonstrate that zero trails can follow nicely defined curves. Figure 7.3 however, does not contain any obvious zero trails. We now analyse these observations in more detail.

Number of Zero Trails

More plotting reveals that zero trails are not evident when the number of *irrational* exponents in the pseudonomial becomes two or more, demonstrated by Figure 7.3. It seems likely that these zeros result from the overlaying of a large (or infinite) number of zero trails.

If we restrict our attention to pseudonomials with only a single irrational exponent then the empirical evidence suggests that they have a finite number of zero trails. Consider Figure 7.1: the pseudonomial has two real zeros (at 1 and 2 – supported by Figure 7.4) and hence it is natural to conjecture that the number of zero trails a function has is equal to the number of real zeros it has. This however, is false as the almost–tree on 3 variables (review Figure 7.1 again) has 3 real zeros (all principal) and only 2 zero trails; however, it leads us to our final conjecture on this matter:

Conjecture 3. *The number of zero trails of a pseudonomial with a single irrational exponent is bounded above by the number of distinct zeros in a single branch.*

A Parametric Form of the Zero Trails

The zero trails seem to have a nice and "smooth" nature and it seems highly likely that they have a simple parametric form.

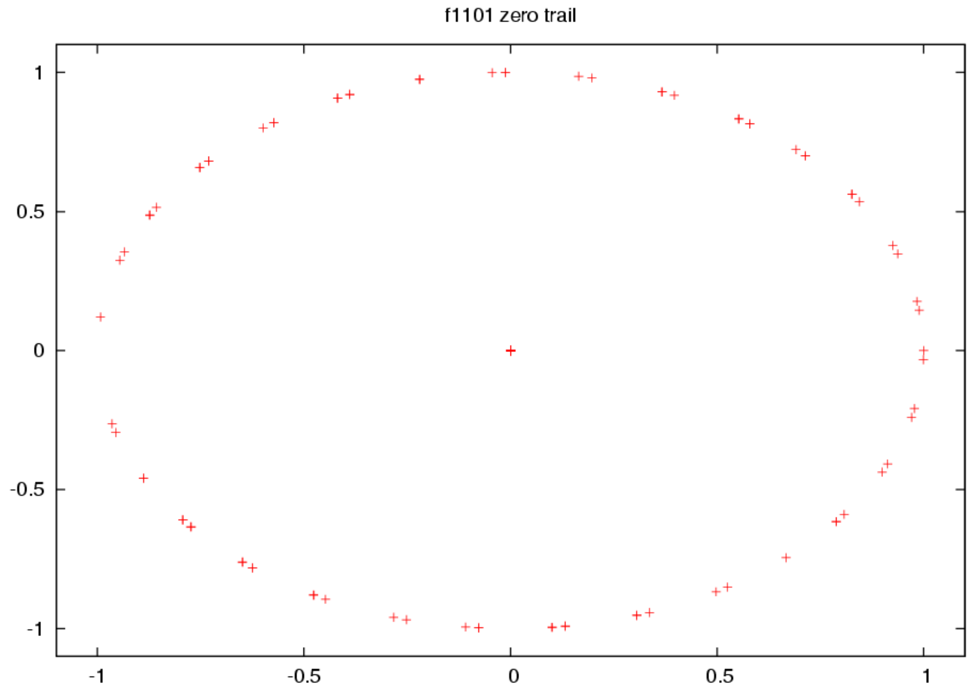


Figure 7.2: Chromatic zeros of the f1101 function. $P(f;\lambda) = -\lambda + \lambda^{\log 3}$

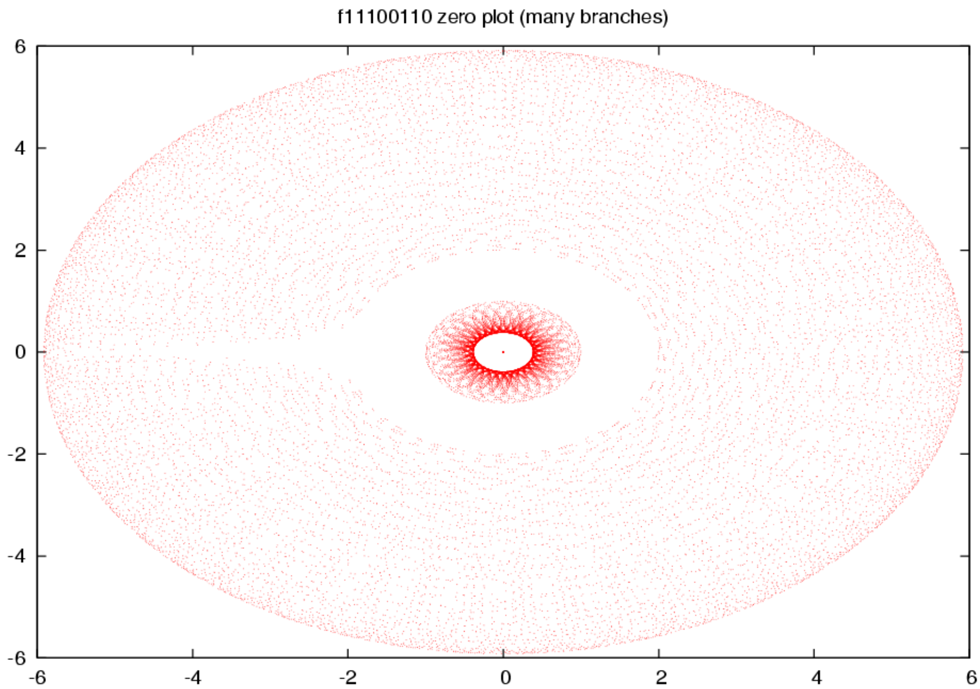


Figure 7.3: Chromatic zeros of the f11100110 function. $P(f;\lambda) = 2\lambda - 3\lambda^{\log 3} + \lambda^{\log 5}$

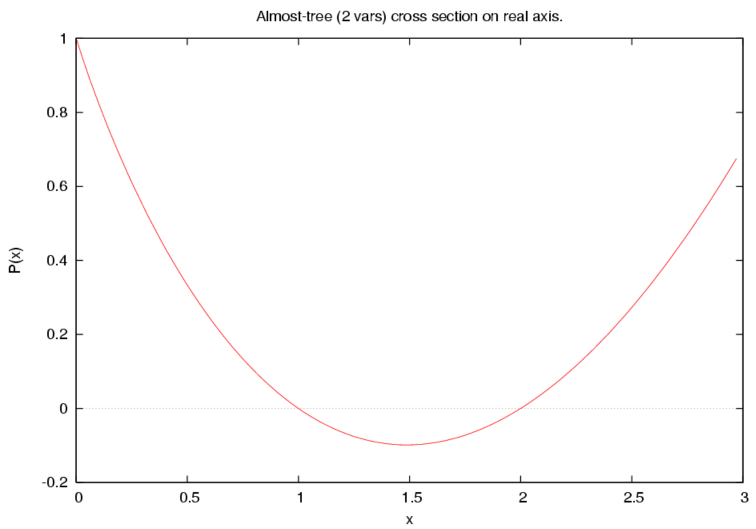


Figure 7.4: The almost–tree on 2 variables has only two real chromatic roots.

Example: The Boolean function $f1101$ has chromatic pseudonomial $P(z) = -z + z^{\log 3}$ with zeros shown in Figure 7.2. We can easily retrieve the parametric form of the zero trail from this pseudonomial. Let $\gamma(t) = e^{it}$, then all the non–trivial chromatic zeros of f lie on the image of γ.

Increasing complexity, we now consider the almost–tree function f1110. Qualitatively the inner trail (Figure 7.1) looks like an ellipse and the outer curve looks like a "dented" ellipse. The chromatic pseudonomial ($P(z) = 1 - 2z + z^{\log 3}$) is more unwieldy than that of the simpler function considered above, and it is not obvious how to go about "extracting" these curves.

We can, however, take a more "engineered" approach by identifying critical points and attempting to fit "generic" curves to the plot. The example included in Appendix G demonstrates that even the simple looking zero trails of a trivial function ($f1110$) are more complex than they appear. Further work in this area is definitely required, and could reveal very important properties of chromatic zeros.

Density of Zeros on the Trails

If we are to be dependant on the trails to describe the location of chromatic zeros it is reasonable to inquire into the distribution of the zeros along these trails. We sketch the outline of a proof in Appendix H that proves for any zero on any branch we can always find another zero on another branch that is arbitrarily close to it. We also conjecture a more general result. These results apply only to pseudonomials with a single irrational exponent, however these results no doubt generalise to the arbitrary pseudonomial case. Time constraints on this project led to this second conjecture being left unproven, however, it is likely to be not too difficult to prove.

Theorem 4. *Let $P_n(z) = g(z) + az^b$, where $g(z)$ is some polynomial and b is irrational. Then for all n, z, and $\delta > 0$,*

$$P_n(z) = 0 \Rightarrow \exists m \neq n, \tilde{z} \neq z \text{ such that } |z - \tilde{z}| < \delta \text{ and } P_m(\tilde{z}) = 0.$$

Conjecture 5. *Let $\gamma(t) : [0, 1) \to \mathbb{C}$ describe a zero trail of some chromatic pseudonomial P. Given some a, b where $0 \leq a < b < 1$ then there exists an m and $\tilde{z} \in \{\gamma(z) : a < z < b\}$ such that $P_m(\tilde{z}) = 0$.*

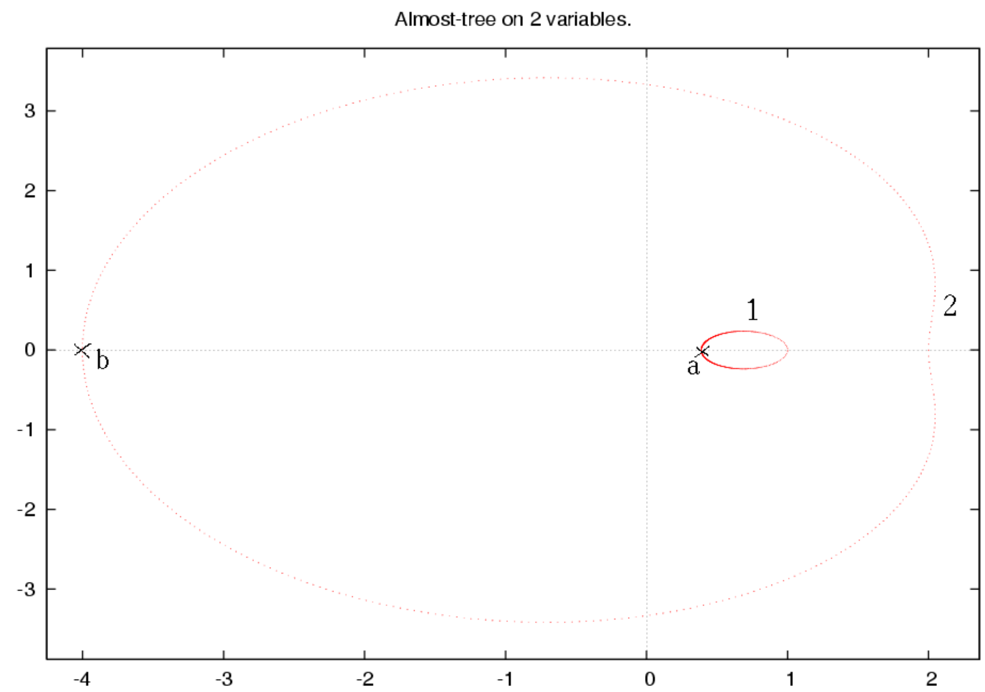


Figure 7.5: Almost–tree on 2 variables.

Tracing the Zero Trails

The results presented before indicates that the parametric forms of these zero trails are difficult to obtain. However, we can still trace out these trails using the method presented in this section. This section also presents the final algorithm developed that traces the zero trails.

Given a chromatic pseudonomial $P_n(z)$ with a zero z_0 that lies on a branch n_0 we would like to find some other zero, z_1, that is close to z_0. We have

$$P_n(z) = g(z) + a|z|^{\log b} e^{i \log b (Arg(z) + 2n\pi)} \tag{7.1}$$

$$= g(z) + \underbrace{az^{\log b}}_{\text{principal value}} .e^{i 2\pi \epsilon_n}, \tag{7.2}$$

where $n \log b = n' + \epsilon_n$, $n' \in \mathbb{Z}$ and $\epsilon_n \in [0, 1)$. Given the continuous nature of $g(z)$ and $az^{\log b}$ it is reasonable to assume that if we find an n_1 such that $\epsilon_{n_1} \approx \epsilon_{n_0}$ then z_0 will be very close to a zero on the branch n_1. This is the basic idea behind the zero trail following algorithm implemented in the program `tracer`.

Without loss of generality, let us assume that $n_0 = 0$ (i.e., we have a principal zero). Since $\epsilon_0 = 0$ we then need to find an n_1 such that $|n_1 \log b - n_1'| < \delta$ for some small user–supplied δ. Once we have found an appropriate branch we then use a steepest descent iteration to locate the nearby zero, that is most likely on the same trail. All that is left is to locate an appropriate n_1.

Given $\log b$ and some small δ we intend to find an m and m' such that $|m \log b - m'| < \delta$. We do this by constructing a convergent series $(p_j/q_j) \to \log b$ (using Shanks' algorithm [35]) until some k where $1/q_k < \delta$. Using a simple property of convergent sequences, namely that for any convergent sequence $(h_n/k_n) \to \zeta$ we have $|\zeta - g_n/h_n| < 1/h_n^2$, we have $|q_k \log b - p_k| < 1/q_k < \delta$. Thus $n_1 = q_k$ is a suitable branch.

The program `tracer` implements this idea and provides the functionality of "tracing" zero trails. The implementation can be found in `tracer.cpp` and `rationalLog2Approx` in `Utils.cpp`.

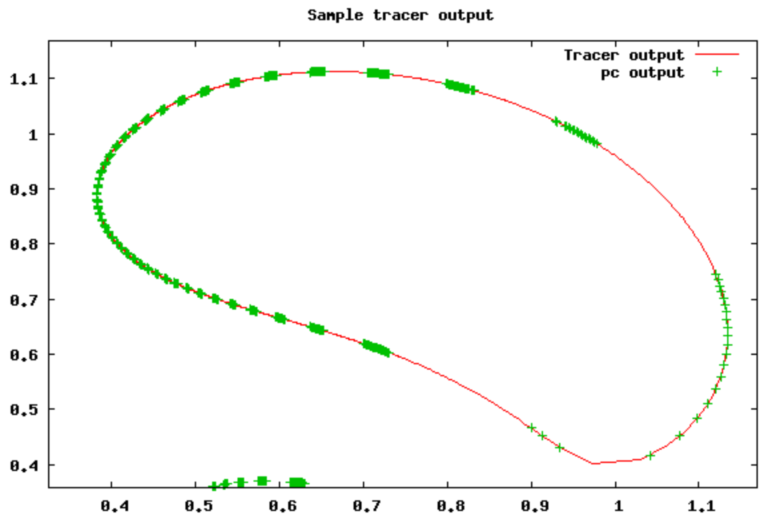


Figure 7.6: Tracer output (interpolated with lines) versus output from predictor–corrector.

The basic algorithm is as follows:

Algorithm: Tracer

- *Input:*
  - f: function.
  - c: start point.
  - i: degree.
  - ϵ: accuracy.
  - N: number of steps.
- Calculate $P(f; z)$.
- Locate a principal zero, z, near c.
- Let $log(k)$ be the i'th irrational exponent.
- Calculate $|p/q - \log k| < \epsilon$.
- Jump forward p branches and locate nearby zero. Repeat this N times.

Figure 7.6 shows a plot generated using zeros found by `tracer` versus zeros found by `pc`. The predictor–corrector samples branches in sequence and, as a side–effect of this, the zeros along the zero trails are disjoint and "clumped". The output from the tracer provides a sequence of zeros along the trail which can then be interpolated to produce a nice qualitative view of the zero trails.

Discussion

Zero trails are an interesting phenomenon of chromatic zeros of Boolean functions, however, the scope of the investigation meant that limited time was spent exploring this area. The theoretical work of zero tracing is still in its early stages and uses naïve approximations. This tended to cause problems such as that demonstrated in Figure 7.7.

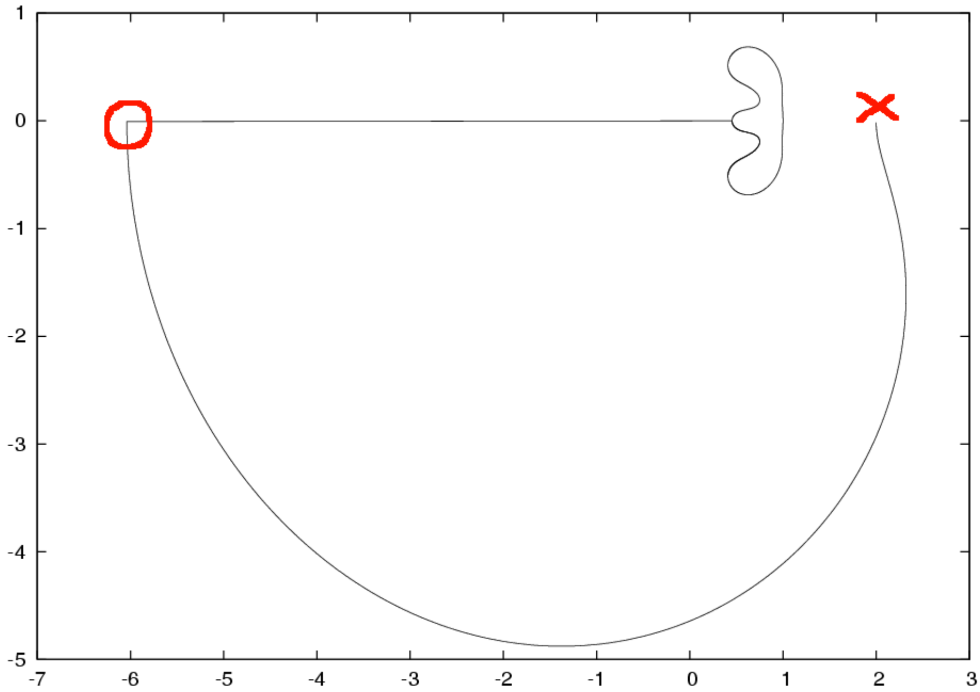


Figure 7.7: This plot demonstrates a flaw with the zero tracing algorithm. The tracer starts at the cross and the continues around on the outer zero trail. When it approaches the real axis again it fails to find a nearby zero and hence iterates all the way to the inner curve.

7.2 Boolean Function Classes

The space of Boolean functions can become extremely large as n (the number of variables in the ground set) becomes large. The computation of most, or even some, chromatic zeros of all Boolean functions on n variables will become infeasible when $n > m$ for some small m (even just generating the chromatic polynomials for all Boolean functions on 5 variables was extremely slow and tedious). We then must limit our investigation to certain classes of functions. For small classes of functions we can enumerate all the functions on n variables in the class. For larger classes however, we must resort to a random sampling of members.

The Boolean function subclasses that were examined are:

- Boolean Polynomials
  - Affine (e.g., $f(\langle x_1, \cdots, x_n \rangle) = 1 + x_1 + x_2$.)
  - Quadratic (e.g., $f(\langle x_1, \cdots, x_n \rangle) = 1 + x_1 x_2 + \cdots + x_n$.)
- Standard classes of Boolean functions
  - 2-CNF (e.g., $f(\langle x_1, \cdots, x_n \rangle) = (x_1 \vee x_2) \wedge (x_3 \vee x_1)$.)
  - 3-CNF (e.g., $f(\langle x_1, \cdots, x_n \rangle) = (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_3 \vee x_4)$.)
- Other classes
  - Almost–trees
  - Cross–trees
  - All functions on 3 variables

We now consider specific results regarding chromatic zeros of Boolean function classes.

7.3 Boolean Polynomials

Boolean polynomials are polynomials restricted to Boolean values. They are a proper subclass of Boolean functions and were considered due to their simplicity. We write ab as the multiplication (*and*) of a and b and $a+b$ as $(\mod 2)$ addition (*exclusive or*) of a and b.

We classify polynomials further by restricting the multiplication operation. *Affine* polynomials do not have any multiplications. *Quadratic* polynomials can have an arbitrary number of multiplication of up to two variables. We now consider these classes.

7.3.1 Affine

Considering valid affine Boolean polynomials restricts the class of functions to:

$$\mathrm{Aff}_T(X) = 1 + \sum_{x\in T} \begin{cases} 1, & \text{if } x \in X; \\ 0, & \text{otherwise;} \end{cases}$$

where $T \subseteq S$.

Example

Let $S = \{a, b, c\}$ and $T = \{a, b\}$. Then $\mathrm{Aff}_T(X) = 1 + a + b$ and for example $\mathrm{Aff}_T(\{a, c\}) = 0$. We have $\operatorname{supp}\mathrm{Aff}_T = \{\emptyset, \{c\}, \{a, b\}, \{a, b, c\}\}$.

Properties of the Affine class

Theorem 6.

$$\operatorname{supp} \mathit{Aff}_T = 2^{S\setminus T} \times \{X : X \subseteq T, |X| \text{ even}\},$$

where $\times$ represents the element–wise union.

Proof. We have

$$\mathrm{Aff}_T(X) = \begin{cases} 1, & \text{if } X \cap T \text{ is even;} \\ 0, & \text{otherwise.} \end{cases}$$

Hence, only those subsets of T of even size are in the support. In addition any of the variables from $S\setminus T$ can be included in any support member as they do not affect the value of f. □

Theorem 7. *Given an arbitrary affine function Aff we can construct a unicyclic G graph such that $Aff = f_G$. Likewise given an arbitrary graph H, f_H is expressible as an affine function.*

Proof. Consider an affine function Aff_T with ground set S. Construct the graph G as follows:

1. Add a cycle with $|T|$ edges and label them with the variables in T;

2. For all variables in $S\setminus T$ add an edge to G such that it does not increase the number of cycles.

It is not too difficult to see that $CS(G) = supp\{Aff_T\}$. The reverse transformation is done in a similar manner. □

This result demonstrates that the class of affine Boolean polynomials is a subclass of graphic functions, and hence there is nothing really new here. By transforming any affine polynomial into its corresponding graph we see that the chromatic pseudonomial has the following form:

Corollary 8. $P(Aff_T; z) = ((z-1)^{|T|} + (-1)^{|T|}(z-1))(z-1)^{|S\setminus T|}$.

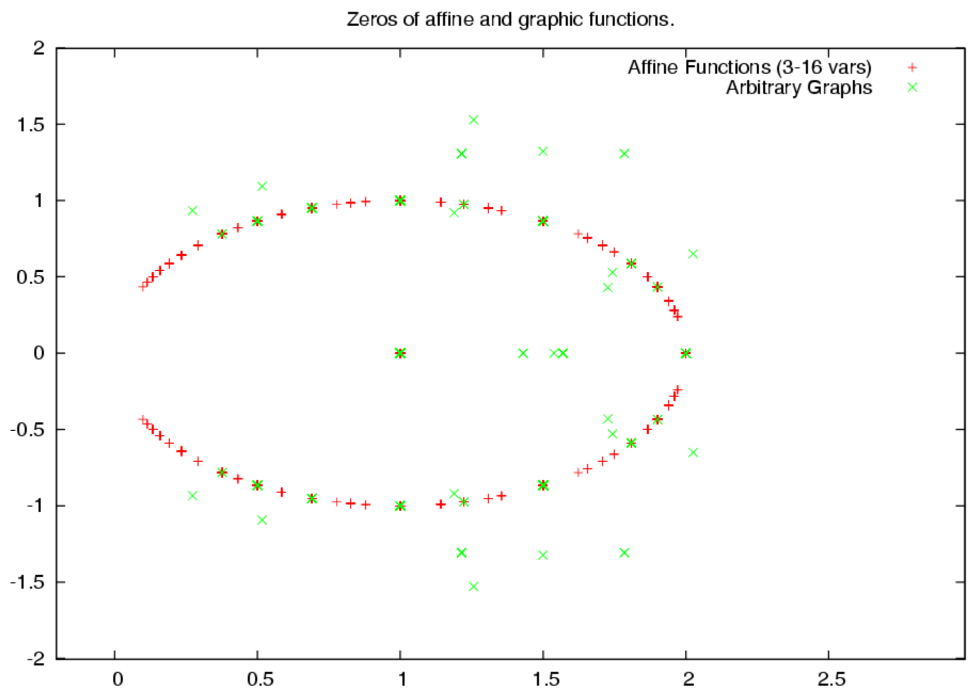


Figure 7.8: Chromatic zeros of a sample of affine polynomials.

Function	Support	Chromatic Pseudonomial
$1 + a + bc$	$\{\emptyset, \{b\}, \{c\}, \{a, b, c\}\}$	$z^2 - z^{\log 3}$
$1 + a^2 + ab$	$\{\emptyset, \{b\}, \{a, b\}, \{b, c\}, \{a, b, c\}\}$	$z^{\log 5} - 2z^{\log 3} + z$
$1 + ac + bc$	$\{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b, c\}\}$	$z^{\log 5} - 3z^{\log 3} + 3z - 1$
$1 + ba + bc + ac$	$\{\emptyset, \{a\}, \{b\}, \{c\}\}$	$z^{\log 7} - z^3 + (z - 1)^3$

Table 7.1: Some example quadratic polynomials and their corresponding chromatic pseudonomials. They all have ground set $S = \{a, b, c\}$.

Moreover, by plotting some chromatic zeros (Figure 7.8) and using the above corollary, we can show:

Theorem 9. *Aff_T has $|T|$ distinct zeros, at $z = 1$, and at $z = 1 + e^{2\pi ik/(|T|-1)}$, for $k = 0, \ldots, |T| - 2$.*

Theorem 10. $P(Aff_T; z) = 0 \Rightarrow z = 1$ *or* $|z - 1| = 1$

The proofs of which follow directly from the form of $P(Aff_T; z)$ given above.

7.3.2 Quadratic

Quadratic polynomials are more interesting than affine, as they are not a subclass of graphic functions. Some examples of quadratic polynomials and their corresponding chromatic pseudonomials are given in Table 7.1.

For a quadratic polynomial to be a valid Boolean function it must contain the constant 1. Some quadratic polynomials can be factored into two affine polynomials, and hence we expect that a subclass of quadratic polynomials have chromatic zeros on the same circle as affine polynomials.

A sample of principal chromatic zeros of quadratic polynomials is presented in Figure 7.9. The zeros at a radius of 1 from $(1, 0)$ are evident, however, there are numerous other zeros, some of which correspond to graphic function zeros. Other important observations are that it seems the zeros lie on simple curves, and it is apparent that there are some real zeros in the interval

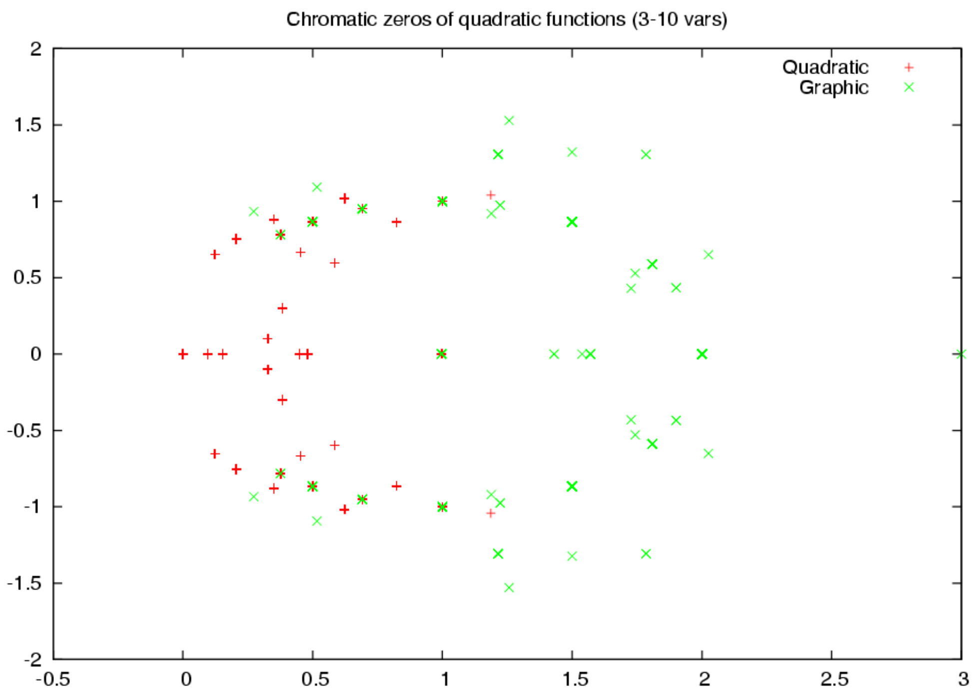


Figure 7.9: Principal zeros of a random sample of quadratic polynomials.

$(0, 1)$. Further work in this area could focus on the distribution of principal zeros of quadratic polynomials.

A sample of many branches of zeros is presented in Figure 7.10. This plot reveals a distinct zero free region at the origin (excluding the trivial zero at $(0, 0)$). They also reveal a dense moon-shaped distribution, however this is most likely due to the uniform method of picking branches. Randomly selecting branches would most likely reveal a more symmetric annulus–type distribution.

Some general theoretical work was begun on quadratic polynomial chromatic zeros, however the time constraints and the scope of the project resulted in the theory being incomplete.

7.4 2–CNF

We now restrict our attention to functions with a 2–CNF representation. Consider a valid Boolean function expressible in 2–CNF, i.e.,

$$f(\vec{x}) = \bigwedge_i (a_i \vee b_i), \text{ where } a_i, b_i \text{ are literals.}$$

The validity requirement ($f(0) = 1$) implies that each clause contains at least one negative literal, which means that each clause can be categorised as one of two types:

1. $\bar{x}_i \vee x_j$
2. $\bar{x}_i \vee \bar{x}_j$.

Type 1 can be rewritten as $x_i \rightarrow x_j$ and Type 2 as $\overline{x_i \wedge x_j}$. We now consider the semantics of these types with respect to their effect on $\operatorname{supp} f$.

- $x_i \rightarrow x_j$: This constrains $\operatorname{supp} f$ in the following way: if x_i is in some member of $\operatorname{supp} f$, then x_j is included in that same member.

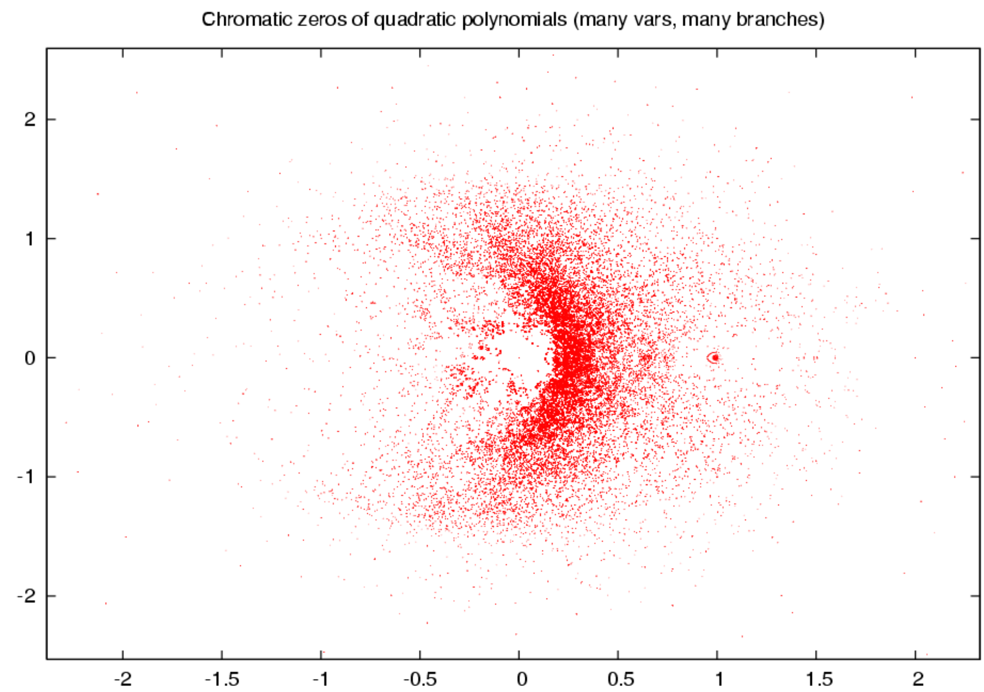


Figure 7.10: Zeros of a random sample of quadratic polynomials.

- $\overline{x_i \wedge x_j}$: This constrains $\operatorname{supp} f$ such that x_i and x_j can *never* be in the same member in $\operatorname{supp} f$.

Example: Let $f(\vec{x}) = (\bar{x_1} \vee x_2) \wedge (\bar{x_3} \vee x_2)$, then

$$\operatorname{supp} f = \{\emptyset, \{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}, \{x_1, x_2, x_3\}\}.$$

Note that f cannot be the indicator function of the cutset space of *any* graph. Calculating the chromatic pseudonomial as usual, gives:

$$P(f; z) = z^{\log 5} - 2z^{\log 3} + z.$$

Using Kung's interpretation (§4.3.1) we check $P(f; 4) = 11$, by counting all 2-tuples in $\operatorname{supp} f$ whose union is S.

$$\begin{aligned}
\#\text{2-tuples} = \#\{&(\emptyset, x_1x_2x_3), (x_1x_2x_3, \emptyset), (x_1x_2x_3, x_2), \\
&(x_2, x_1x_2x_3), (x_1x_2x_3, x_2x_3), (x_2x_3, x_1x_2x_3), (x_1x_2x_3, x_1x_2), \\
&(x_1x_2, x_1x_2x_3), (x_1x_2, x_2x_3), (x_2x_3, x_1x_2), (x_1x_2x_3, x_1x_2x_3)\} \\
= 11&,
\end{aligned}$$

where $x_a x_b x_c$ is shorthand for $\{x_a, x_b, x_c\}$.

We now consider the relationship between graphs and 2–CNF formulas.

7.4.1 Graphs and 2–CNF formulas

We now present some theoretical work that shows that the correspondence between 2-CNF formulas and graphs is not strong.

Consider an arbitrary tree with n edges. It has a cutset space which is equal to 2^E and hence has a tautologous indicator function. Note that $f = a \vee \bar{a} = \top$ is a tautologous 2–CNF formula, and hence all trees have Boolean functions expressible in 2–CNF.

Consider now a tree, T, with a single multi–edge (i.e. two edges a and b that are incident on the same pair of vertices). The cutset space of this tree is then $\{X : X \subseteq S \wedge (a \in X \iff b \in X)\}$, and the corresponding indicator function can be expressed as $f_T = (a \to b) \wedge (b \to a)$.

Theorem 11. *A multi–edged tree has a cutset space indicator function with a 2–CNF representation.*

But what happens when we consider arbitrary graphs? It seems highly likely that a graph with a cycle does not have a 2–CNF representation. We support this claim with a proof (in Appendix I) that even the simple 3–cycle has no 2–CNF representation.

Theorem 12. *The cutset space indicator function of the 3–cycle has no 2–CNF representation.*

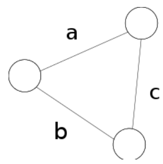


Figure 7.11: The 3–cycle.

This result reveals that the dependency between elements in the cutset space induced by a cycle cannot be expressed in 2–CNF form. It is assumed that the proof technique used generalises to the arbitrary case of a graph with many cycles, however the proof was not found during the limited time constraints of this project.

Conjecture 13. *Any graph with a cycle has a cutset space indicator function inexpressible as a 2–CNF formula.*

7.4.2 EC-graphs

If we consider what kind of combinatorial "graph–like" object corresponds to Boolean functions in 2–CNF (in the same way that graphs correspond to graphic functions) we are lead directly to the concept of an *EC-graph*.

Definition: EC–Graph
An edge–constrained graph (or EC–graph) consists of a simple graph labelled with *edge constraints*. There are two types of constraints – both of which act on pairs of edges.
Arrow: The arrow is a one–way constraint depicted diagrammatically as a dashed arrow drawn between two edges. Formally we write the constraint as $e_1 \to e_2$ if an arrow exists from e_1 to e_2.

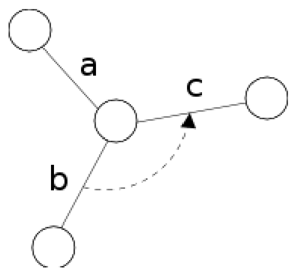


Figure 7.12: Example of an EC–graph with an arrow constraint.

Cross: The cross is a symmetric constraint depicted as a dashed line between two edges annotated with a cross through it (as shown in Figure 7.13). Formally we write $e_1 \times e_2$ to indicate that e_1 and e_2 are bound by a cross.

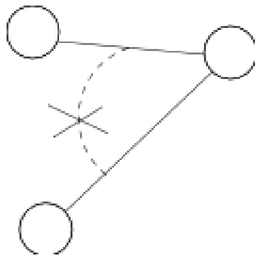


Figure 7.13: Example of an EC–graph with a cross constraint.

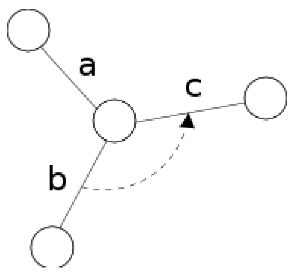
A 2^k*–colouring* of an EC–graph is an assignment $\sigma : E \to \{0,1\}^k$ such that the following conditions hold:

For all edges e and e',

1. $\sigma(e) \neq 0$;
2. $e \to e' \Rightarrow \sigma(e) \subseteq \sigma(e')$;
3. $e \times e' \Rightarrow \sigma(e) \cap \sigma(e') = \emptyset$;

Note that a 2^k–colouring of an EC–graph corresponds exactly to a tension (see Appendix C) of the underlying simple graph (and hence to a proper colouring).

Example: How many 2^k–colourings does the following EC–graph have?



We can calculate this combinatorially as follows. We can assign any of the valid $2^k - 1$ colours to edge a, the number of colours we can assign to b and c is equal to:

$$\sum_{\substack{X \in \{0,1\}^k \\ X \neq 0}} \sum_{\substack{Y \in \{0,1\}^k \\ X \leq Y}} 1,$$

which, by the binomial theorem and some trivial manipulation, can be found to be equal to $3^k - 2^k$; Hence, $P(H; 2^k) = (2^k - 1)(3^k - 2^k)$.

2–CNF is Isomorphic to EC–Tree

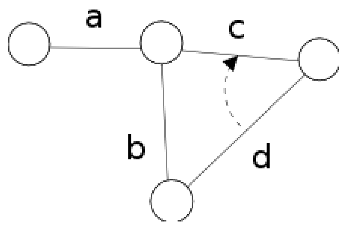
The semantics of the arrow and cross constraints purposefully affect the cutset space of the EC–graph. The cutset space of an EC–graph is the cutset space of the underlying simple graph with the additional constraints defined as follows:

$e_1 \to e_2$: An arrow from e_1 to e_2 constrains the cutset space of the corresponding graph such that if e_1 is in a member of the cutset space then e_2 is also included in that member.
$e_1 \times e_2$: A cross between e_1 and e_2 restricts the cutset space by removing all members of the cutset space that contain both e_1 and e_2.

Example:

The following EC–graph has cutset space equal to

$$\{\emptyset, \{a\}, \{b\}, \{c\}, \{a, c\}, \{b, c\}, \{c, d\}, \{a, c, d\}, \{b, c, d\}\}.$$



Note that any member that contains d also contains c, otherwise the cutset space is the same as the underlying simple graph.

By using the same transformations from graphs to functions we can transform EC–graphs to Boolean functions.

Theorem 14. *The class of EC–trees is isomorphic to the class of valid 2–CNF functions.*

Proof. We prove this result by demonstrating the transformation in both directions.

EC–trees to 2–CNF Let H be an arbitrary EC–tree on n edges. Let f be a 2–CNF formula with ground set $S = E$ and no clauses to begin with. For each edge constraint do the following:

- $e_1 \to e_2$: Add the clause $\bar{e_1} \vee e_2$ to f.
- $e_1 \times e_2$: Add the clause $\bar{e}_1 \vee \bar{e}_2$ to f.

We then have that the cutset space of H is exactly equal to the *support* of f by definition of the constraints.

2–CNF to EC–trees This transformation is just the reverse of the process above. For a 2–CNF formula f with ground set S, draw an arbitrary tree with edge set $E = S$. Then for each clause in f add the corresponding edge constraint as detailed above. (Note that a valid 2–CNF has clauses only of the two types above.) □

Corollary 15. *A 2–CNF colouring with 2^k colours can be interpreted as a 2^k–colouring of its corresponding EC–tree.*

7.4.3 EC–trees

A more detailed investigation of 2–CNF colourings can now proceed by looking directly at EC–tree colourings. We now present a theorem that exposes the highly constraining notion of the *cross* constraint.

Theorem 16. *For all k there exists an EC–tree that has no 2^k–colouring.*

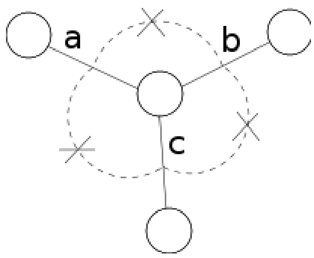
Proof. Construct an EC–tree with $k + 1$ edges and cross constraints between *all* pairs of edges. Colouring any edge then reduces the number of remaining colours assignable to *all other edges* by half. Colouring k edges causes the last edge to be uncolourable, and hence this EC–tree is not 2^k–colourable. □

This observation led to the following subclass of EC–trees.

7.4.4 Cross–Trees

Definition: Cross–Tree
A *cross–tree* on n edges, denoted X_n, is an EC–tree constructed as in the proof of Theorem 16.

Example:



The cross–tree above has no 2^2–colouring (by the above theorem). We can easily verify this by attempting to colour it with 2^2 colours (i.e., bit strings of length two). Colouring a with 01 constrains the possible colours of b to 10 which, in turn, exhausts all colours assignable to c.

Theory

Some simple theoretical results are now presented.

Theorem 17. $CS(X_n) = \{\emptyset\} \cup \{\{e\} : e \in E\}$.

Proof. Follows from the semantics of the cross constraint. □

This simple cutset space leads to cross–trees having an easily expressible chromatic pseudonomial.

Theorem 18. $P(X_n; z) = \sum_{k=0}^{n} \binom{n}{k} (-1)^{n+k} z^{log(k+1)}$

Proof. Follows from the definition of the chromatic pseudonomial, and the fact that $\sum_{Y \subseteq X} f_{X_n}(Y) = 1 + |X|$. □

Zeros of Cross–Trees

The zero distributions of cross–trees are quite extra–ordinary. The plots in Figures K.1 through K.5 (Appendix K) show the zeros of cross–trees up to 6 edges. X_2 and X_3 do not exhibit anything unusual (having a single irrational exponent in their chromatic pseudonomials), however X_4 through X_6 demonstrate a very high level of complexity.

Figure 7.14 (reproduced larger in Appendix K) reveals a few interesting features about the distribution of zeros of X_4. Firstly, it seems that there is a small set of curves which bound *most* of the zeros. Secondly, there are distinctive zero–free regions that also seem bounded by a set of simple curves. These observations hold for more complex plots too (see Figures K.4 and K.5), however, due to the limited number of branches sampled the outer bounds are less defined.

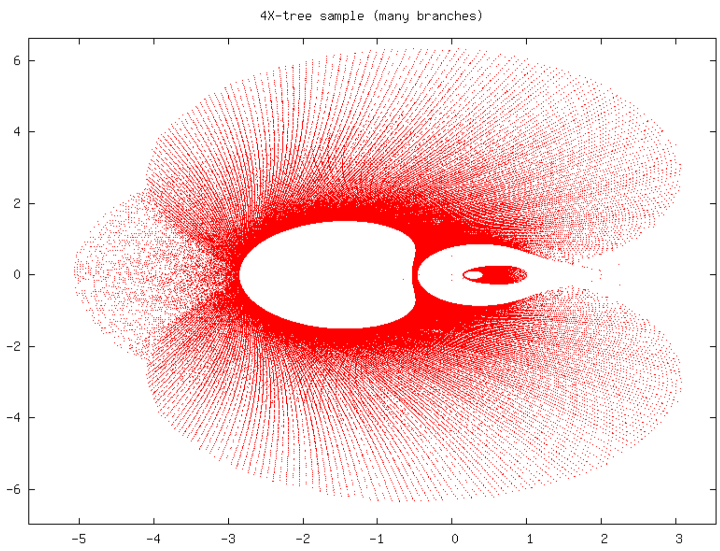


Figure 7.14: Chromatic zeros of X_4.

7.4.5 Snake Paths

Other interesting families of EC–trees can be constructed by identifying simple repetitive structures. This demonstrates the usefulness of EC–trees as a tool for identifying new subclasses of 2–CNF functions.

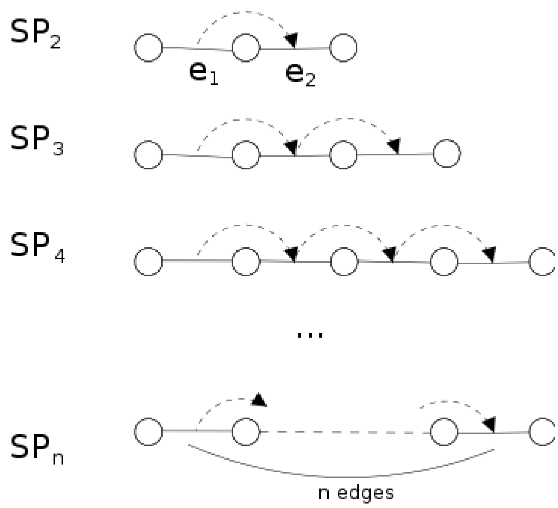


Figure 7.15: Snake path sequence.

Definition: Snake Path
A *snake path* on n edges is an EC–tree constructed as in Figure 7.15.

Theorem 19. $P(SP_n; 2^k) = (n+1)^k - n^k$.

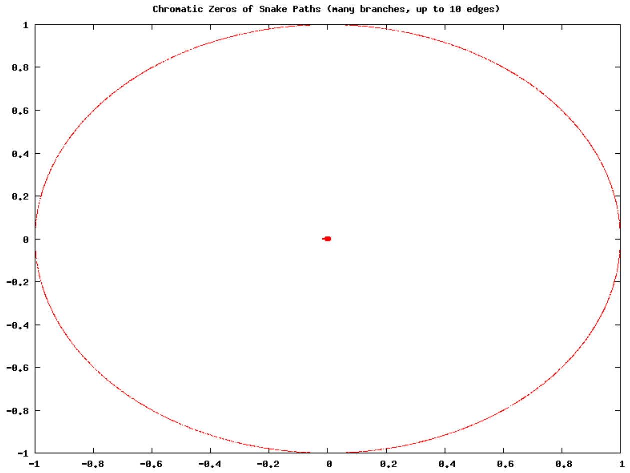


Figure 7.16: Chromatic zeros of snake paths.

Proof. Combinatorially we have:

$$\begin{aligned}
P(SP_n;2^k) &= \underbrace{\sum_{\substack{X_1\in\{0,1\}^k\\ X_1\neq 0}}}_{\text{colouring of } e_1} \underbrace{\sum_{\substack{X_2\in\{0,1\}^k\\ X_1\subseteq X_2}}}_{\text{colouring of } e_2} \cdots \sum_{\substack{X_{n-1}\in\{0,1\}^k\\ X_{n-2}\subseteq X_{n-1}}} \underbrace{\sum_{\substack{X_n\in\{0,1\}^k\\ X_{n-1}\subseteq X_n}}}_{\text{colouring of } e_n} 1\\
&= \sum_{\substack{X_1\in\{0,1\}^k\\ X_1\neq 0}} \sum_{\substack{X_2\in\{0,1\}^k\\ X_1\subseteq X_2}} \cdots \sum_{\substack{X_{n-1}\in\{0,1\}^k\\ X_{n-2}\subseteq X_{n-1}}} 2^{k-|X_{n-1}|}\\
&= \sum_{\substack{X_1\in\{0,1\}^k\\ X_1\neq 0}} \sum_{\substack{X_2\in\{0,1\}^k\\ X_1\subseteq X_2}} \cdots 3^{k-|X_{n-2}|}\\
&= \cdots\\
&= (n+1)^k - n^k,
\end{aligned}$$

by the repeated application of the binomial theorem.

□

Plotting the zeros (see Figure 7.16) illustrates something not immediately obvious – chromatic zeros of snake paths occur only when $|2^k| = 1$ or 0. The plot shows a slight numerical error as there appears to be a small "clump" of zeros at the origin, these are false zeros and only appear due to the limited iterations of the predictor–corrector. Note also that the zeros plotted are $2^k = x + iy$ and *not* values of k. This plot led to the following result.

Theorem 20. *$P(SP_n;2^z) = 0 \iff z = \frac{2\pi k i}{\ln(n/(n+1))}$, for some $k \in \mathbb{Z}$.*

Proof. Let $z = x + iy$, we then have:

$$\begin{aligned}
& P(SP_n;2^z) = (n+1)^z - n^z = 0\\
\iff & (n+1)^z = n^z\\
\iff & n^x = (n+1)^x \text{ and } n^{iy} = (n+1)^{iy}\\
\iff & x = 0 \text{ and } \underbrace{e^{iy\ln n} = e^{iy\ln(n+1)}}_{(1)}.
\end{aligned}$$

So x is now fixed at 0, and we need to find the valid values for y. Condition (1) is satisfied if and only if $y \ln n - 2\pi k = y \ln(n+1)$ for some $k \in \mathbb{Z}$. Rearranging the terms gives us

$$y = \frac{2\pi k}{\ln(n/(n+1))},$$

as required. □

7.4.6 Random 2–CNF

When undertaking an investigation of the chromatic zeros of *any* class of functions, it is beneficial to locate the zeros of a random sample and plot them together. The following plot (Figure 7.17) shows the *principal* chromatic zeros of a random sample of 2–CNF functions. [3]

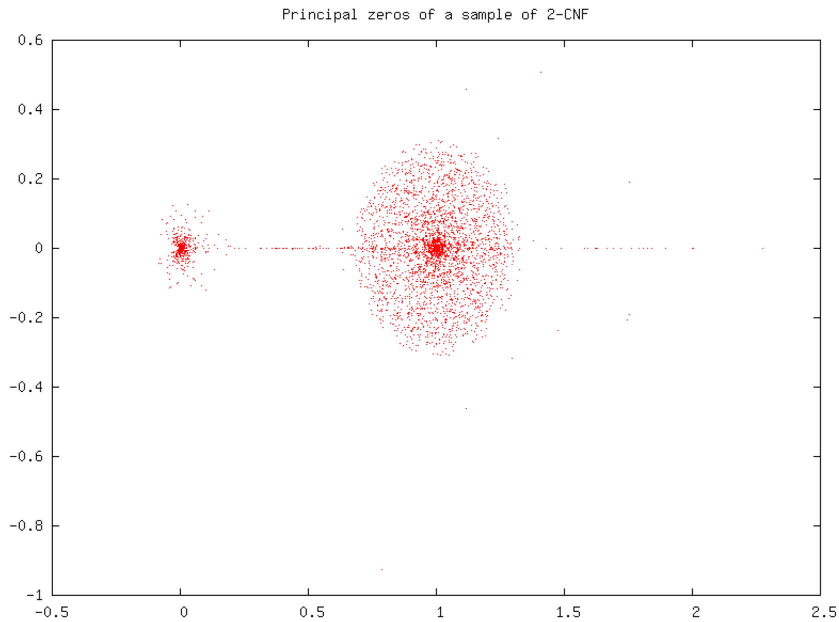


Figure 7.17: Chromatic zeros of a random sample of 2–CNF functions (2200 functions overall, between 3 to 14 variables each).

This plot raises some interesting conjectures about the chromatic zeros of the 2–CNF class of functions:

Conjecture 21. *2–CNF functions have no negative real chromatic zeros.*

Conjecture 22. *We can find a sequence of 2–CNF formulas with chromatic zeros converging to the real value 1. More formally: Given any ϵ, we can find a 2–CNF function with a zero, z, such that $0 < |z - 1| < \epsilon$.*

Conjecture 23. *We can find a sequence of 2–CNF formulas with chromatic zeros converging to the real value 0. More formally: Given any ϵ, we can find a 2–CNF function with a zero, z, such that $0 < |z| < \epsilon$.*

Furthermore, if we look at a histogram of a random sample of 2–CNF real chromatic zeros versus a random sample of 2–CNF graph chromatic roots (Figure 7.18) we see that the 2–CNF generalisation allows a much broader distribution of real chromatic zeros.

[3] Random 2–CNF formulae were constructed by simply generating a random number of random clauses (bounded by $\binom{n}{2}$). See `random2cnf.cpp` for the implementation details.

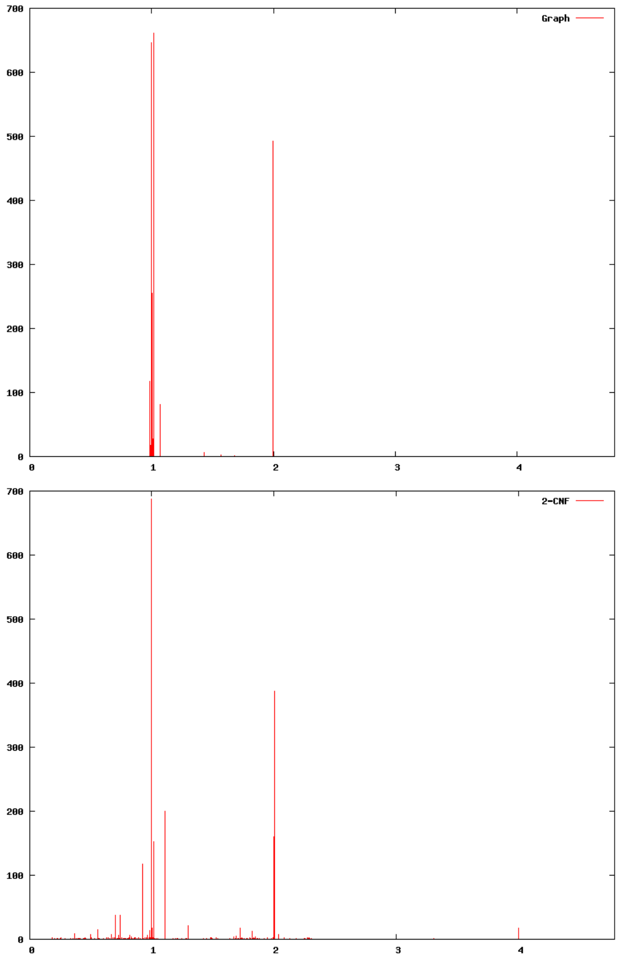


Figure 7.18: Histogram of real graph chromatic roots (top), versus a histogram of real 2–cnf chromatic zeros (bottom). This illustrates that the class of 2–CNF (equivalently EC–Trees) has a broader distribution of real zeros, and the zero free regions for graphs $(0, 1), (1, 32/27)$ are no longer zero free for 2–CNFs.

7.4.7 Discussion

The class of 2–CNF functions is interesting and many problems regarding their chromatic zero distribution still remain to be unsolved. Figure 7.19 demonstrates the correlation among the 2–CNF domain and the graph–like objects domain. We see that 2–CNF is sufficiently general to include non–graphic objects, however it is not so general that it includes graphic functions as a subclass. This is important as it provides us with a class that is not *too* general, but still contains sufficiently *different* objects that graphs.

Conjecture 22 presents an interesting aspect of the 2–CNF domain, and it would be interesting to see if a sequence of related 2–CNF formulas satisfied this conjecture. EC–graphs and EC–trees in particular provide us with a new technique of calculating the chromatic pseudonomial for 2–CNF functions (a technique that does not necessarily require $O(2^{|}S|)$–time), as well as a mechanism for visualising colourings of 2–CNF formulas. EC–trees were beneficial to this investigation in the identification of novel subclasses of 2–CNF formula (i.e., cross–trees and snake–paths). These subclasses can be generated easily using diagrammatic relationships and symmetry — something that is not possibly when working solely with 2–CNF formulas.

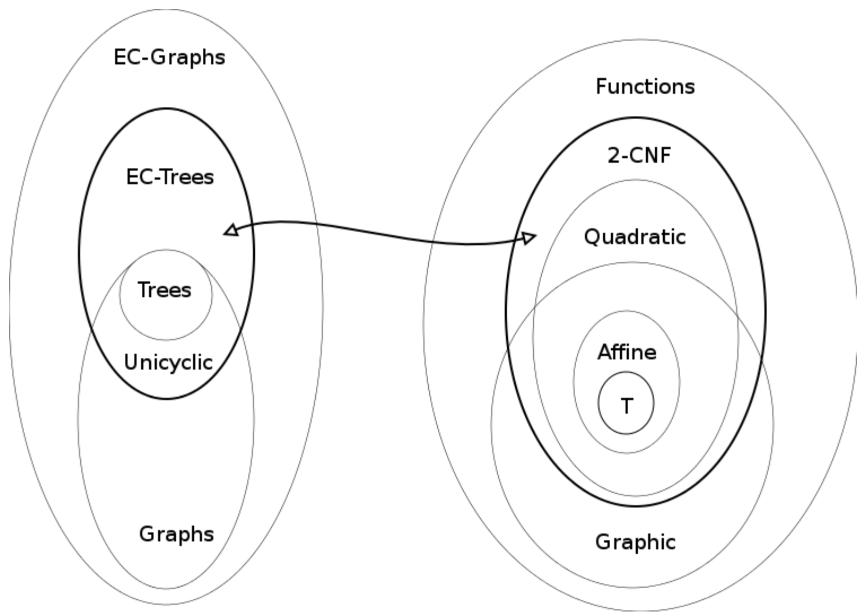


Figure 7.19: Correlation between graph–like and function–like objects.

7.5 3–CNF

3–CNF functions were briefly considered. The plot in Figure 7.20 shows the principal chromatic zero distribution of a random sample of 3–CNF functions. Comparing this to Figure 7.17 shows more “clumps” of zeros are appearing. It would be interesting to follow this sequence up to arbitrary k–CNF to see if a pattern occurs.

7.6 Almost–Trees

Definition: Almost–Tree
An almost–tree, AT_n, on n variables is a Boolean function with support $2^S \backslash \{S\}$.

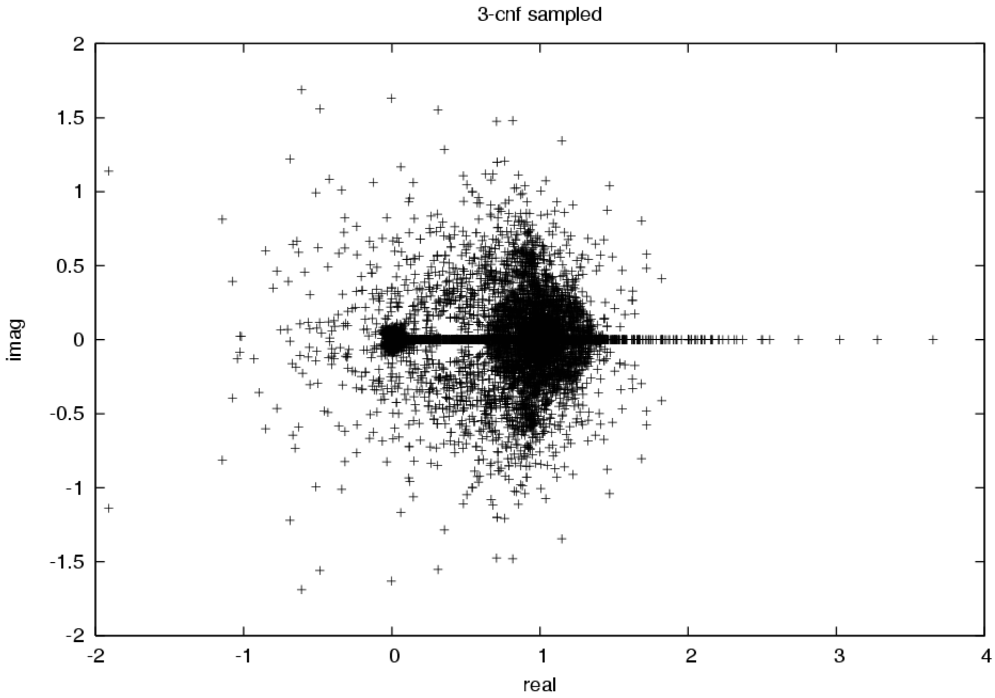


Figure 7.20: Principal chromatic zeros of a random sample of 3–CNF functions.

An almost-tree has a support that is isomorphic to the cutset space of a tree with the largest member removed. An empirical study of the chromatic zeros reveals that the region $(0, 1)$ is *not* zero free, which is converse to the situation of chromatic roots of graphs. Almost–trees also have an easily expressible chromatic pseudonomial, as illustrated by the following theorem.

Theorem 24. $P(AT_n; z) = z^{\log(2^n-1)} - z^n + (z-1)^n$.

Proof.

$$P(AT_n; z) = \sum_X (-1)^{|\bar{X}|} z^{\log F(X)}, \text{ where } F(X) = \sum_{Y \subseteq X} f(Y).$$

But $F(X) = 2^{|X|}$ for all X *except* $X = S$, where $F(S) = 2^{|S|} - 1$. Hence

$$\begin{aligned} P(AT_n; z) &= \sum_X (-1)^{|\bar{X}|} z^{|X|} + (z^{\log(2^{|S|}-1)} - z^{|S|}) \\ &= (z-1)^n + (z^{\log(2^n-1)} - z^n). \end{aligned}$$

□

We can see from Theorem 24 that all almost–trees have chromatic zeros at the real points 1 and 2. Additionally, looking at a plot of $P(AT_3; z)$ over a real interval (see Figure 7.21), for example, reveals that AT_3 has a chromatic zero in the interval $(0, 1)$. More almost–tree plots (see Appendix J) led to the following conjectures:

1. Other almost–trees have chromatic zeros in $(0, 1)$;

2. They have exactly two zero trails;

3. The zero trail crosses the negative real axis at $-n - 2$;

4. They have chromatic zeros at 1 and 2;

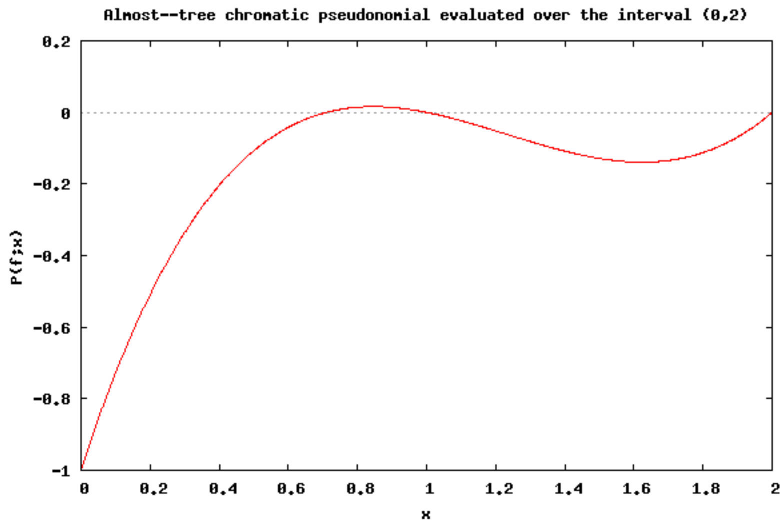


Figure 7.21: A plot indicating that AT_3 has a chromatic zero in the interval $(0, 1)$.

5. The inner curve contains $n - 1$ "bumps";

6. The outer curve has a "dent" at 2 that becomes sharper as n increases.

We now consider each of these conjectures in turn.

7.6.1 Other Almost–Trees have Chromatic Zeros in $(0, 1)$

The fundamental observation in section is the almost–trees have chromatic zeros in $(0, 1)$. In fact, we can prove a stronger result than this:

Theorem 25. *If n is odd, then there exists a $c \in (1/2, 1)$ such that $P(AT_n; c) = 0$.*

Proof. Assume m is an arbitrary odd integer.

Evaluating $P(AT_m; z)$ at $z = 1/2$ gives,

$$\begin{aligned} P(AT_m; \frac{1}{2}) &= (\frac{1}{2^m - 1} - \frac{1}{2^m}) - \frac{1}{2^m} \\ &= \frac{1}{2^m - 1} - \frac{1}{2^{m-1}} \\ &< 0, \end{aligned}$$

as $2^m - 1 > 2^{m-1}$ for all m.

Likewise we can show that $P(AT_m; z)$ at $z = 1 - \epsilon$ is positive, for small enough ϵ. Evaluating the derivative of P at $z = 1$ gives

$$\begin{aligned} \frac{dP(AT_m; z)}{dz} &= m(z-1)^{m-1} + \log(2^m - 1) z^{\log(2^m-1)^{-1}} - m z^{m-1} \\ \frac{dP(AT_m; 1)}{dz} &= 0 + \log(2^m - 1) - m \\ &< 0, \end{aligned}$$

and thus $P(AT_m; 1 - \epsilon) > 0$ for ϵ infinitesimally small, as P is continuous. The intermediate value theorem then implies that there exists a $c \in (1/2, 1)$ such that $P(AT_m; c) = 0$.

□

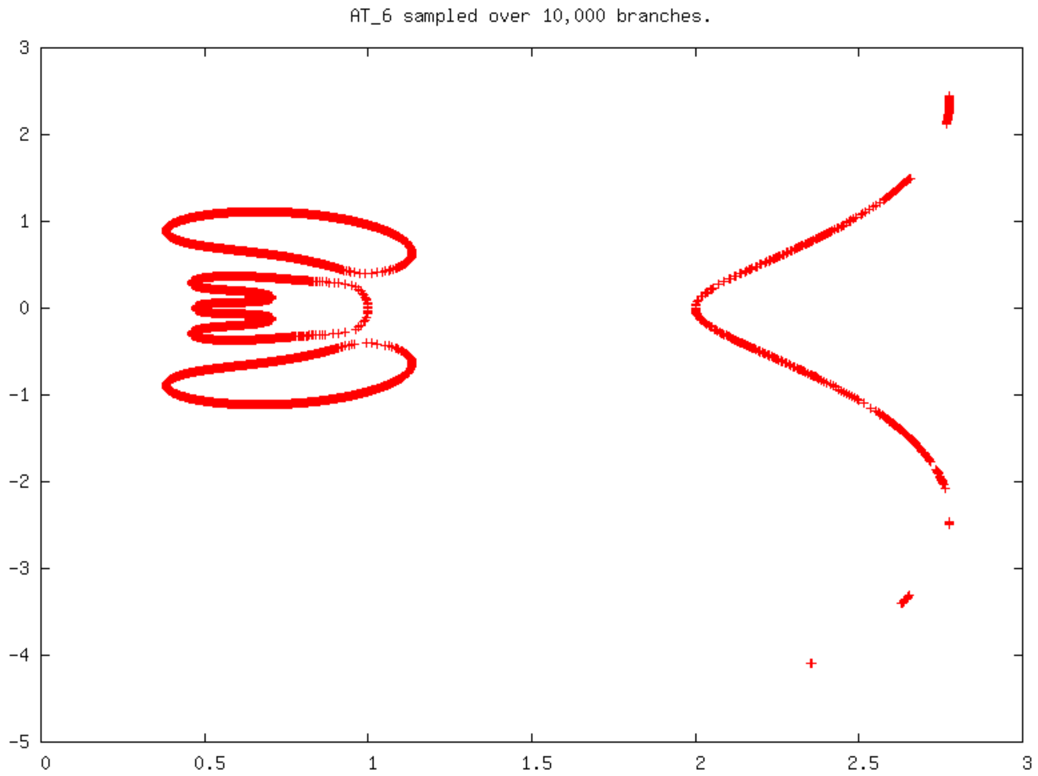


Figure 7.22: The inner zero trail of AT_6 appears as if it is composed of three separate curves.

7.6.2 Number of Zero Trails

The plots reveal that almost–trees have two zero trails. This observation occurs for low n but empirical evidence at $n = 6$ suggests that it is incorrect. Consider Figure 7.22, in this plot of $10,000$ branches of chromatic zeros of AT_6 we see that there is a distinct set of 4 curves. This poses an extremely interesting question and direction for future work:

How many distinct zero trails does a given pseudonomial have?

7.6.3 Negative Real Axis Intercept of the Zero Trails

The plots reveal that the zero trail seems to cross the negative real axis at $-n-2$. We now present some theory which invalidates this conjecture.

Consider the m'th branch of P evaluated at some negative real number, $-x$,

$$P_m(AT_n; -x) = (-x-1)^n + |x|^{\log(2^n-1)} \underbrace{e^{i.\log(2^n-1).(\pi+2\pi m)}}_{(*)} - (-x)^m.$$

This can only equal 0 when (*) $= 1$ or -1, and we consider these cases separately:

1. *(*) = 1*

Letting (*) $= 1$ implies that

$$\begin{aligned} P(AT_n; -x) &= (-x-1)^n + |x|^{\log(2^n-1)} - (-x)^n \\ &= (-1)^n((x+1)^n - x^n) + |x|^{\log(2^n-1)}, \end{aligned}$$

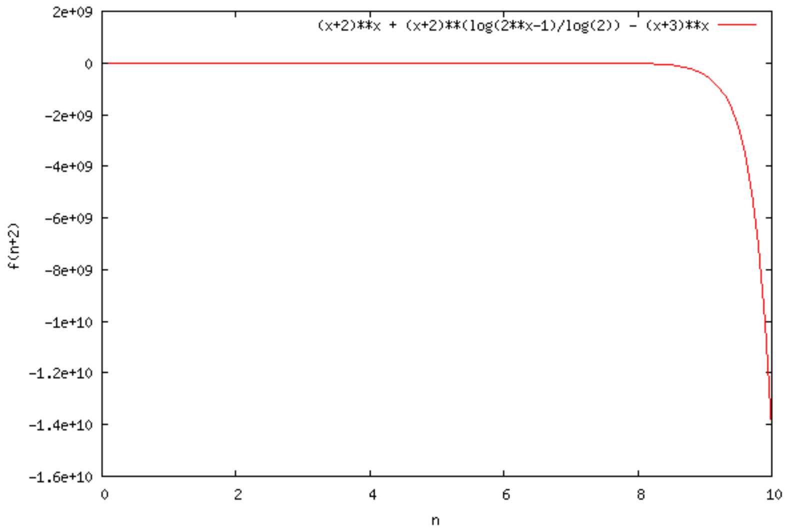


Figure 7.23: A plot of $f(n+2)$ for increasing n. At $n < 8$ the function is reasonable small, indicating a zero occurs in the vicinity of $c = n + 2$. At $n \geq 8$ the function suddenly decreases exponentially, illustrating without doubt that $c \neq n + 2$.

which can only be zero when n is odd. Letting n be odd then implies that the zero trail crosses the negative axis at the point $-c$, where c is the positive real solution to $f(c) = c^n + c^{\log(2^n-1)} - (c+1)^n = 0$.

2. *(*) = -1*

Letting (*) $= -1$ implies that

$$\begin{aligned} P(AT_n; -x) &= (-x-1)^n - |x|^{\log(2^n-1)} - (-x)^n \\ &= (-1)^n((x+1)^n - x^n) - |x|^{\log(2^n-1)}, \end{aligned}$$

which can only be zero when n is even. Letting n be even implies that the zero trail crosses the negative axis at the point $-c$, where c is the positive real solution to $g(c) = -c^n - c^{\log(2^n-1)} + (c+1)^n = 0$.

Note that the two cases above specify the same point $-c$ to be the negative real axis intercept of the zero trail (i.e., $f(c) = 0 \iff g(c) = 0$. We also know that $f(c)$ has at most one real solution. [22, Theorem 3.1]

The driving observation of this section conjectures that $c = n + 2$. However plotting $f(n + 2)$ for increasing n (see Figure 7.23) demonstrates that this conjecture is wrong. It also indicates why our observations might have led to the conjecture.

7.6.4 Almost–Trees have Chromatic Zeros at 1 and 2

All trees are 2–colourable, but intriguingly, no almost–tree is 2–colourable. This result follows directly from the definition of $P(AT_n; z)$.

Theorem 26. *No almost–tree is* 1 *or* 2*–colourable.*

Proof. We have $P(AT_n; z) = z^{\log(2^n-1)} - z^n + (z-1)^n$, and hence $P(AT_n; 1) = 1 - 1 = 0$ and $P(AT_n; 2) = -2^n + 2^n - 1 + 1 = 0$. □

7.6.5 Other Observations

We observed that the inner curve contains $n - 1$ "bumps" and the outer curve has a "dent" at 2 that becomes sharper as n increases. These observations are of a more qualitative nature, and it is not clear how to go about formalising them.

7.6.6 Discussion

The family of almost–trees is an example of a simple set of functions that are easily defined, but which produce myriad complex problems. We presented a proof that $(0, 1)$ is not zero–free showing that there are a countable number of almost–trees with zeros in the region $(1/2, 1)$. Without much more effort this result could be strengthened to a larger interval. Future work could look at the limit of the zero in $(0, 1)$ for the sequence of almost–trees, or extend the family of almost–trees to almost–graphs, for example.

7.7 Miscellaneous

We now consider some miscellaneous investigations. These include plots of: a random sample of functions on a ground set with 3 variables; the principal chromatic zeros of random functions; and, the chromatic zeros of a complex pseudonomial.

7.7.1 Random 3 Variable Sample

Consider Figure 7.24. This is a plot of the chromatic zeros (over 100 branches) of a random sample of Boolean functions over a ground set of 3 variables. There is not much structure evident in this random sample, however, three important features do stand out:

- The zeros seem more dense on concentric bands;
- There is a distinct zero–free region at the origin, with radius approximately 0.3;
- All the zeros seem to occur within a circle of radius 6 centered at the origin.

It is unclear how to go about developing theoretical results to support the first observation. We now consider the third of these observations.

A new zero bound was presented in section 7.1.3. For arbitrary Boolean functions on n variables this bound is largest when $k = 2^{n-1} + 1$, i.e., for the tautologous function. The bound then states that all chromatic zeros of all functions on n variables is 2^s, where,

$$s = \frac{n}{1 + \log(2^{n-1} + 1) - n}.$$

Applying this bound to $n = 3$ gives an outer radius of approximately 638, which demonstrates the impracticality of the bound.

Lapidus and Frankenhuysen [26, Theorem 2.5] give a bound for *fractal strings*[4] that we can apply here. After applying the required transformations, their result gives a larger bound than that above, and hence was discarded.

Obviously these bounds are not helpful, and new theoretical work needs to be developed. This could be yet another future direction for work in this area.

[4] Fractal strings are essentially pseudonomials under a trivial transformation.

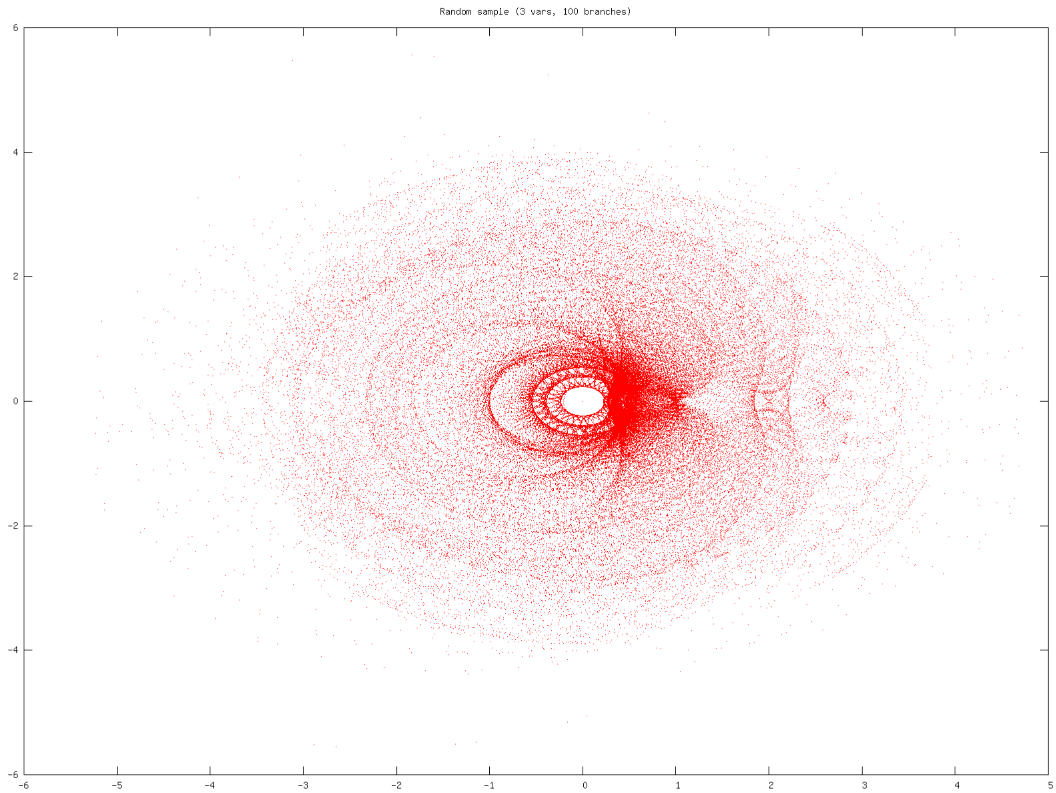


Figure 7.24: Chromatic zeros of a random sample of functions bounded to 3 variables only.

7.7.2 Random Function Samples

Some principal chromatic zeros of a random sample of functions, limited by number of variables, are presented in Appendix L. Limiting the sample to functions of three variables (Figure L.2) produces a simple zero distribution. In the plot we see distinct zeros along the real axis, but also "fingers" of zeros that seem to originate from the point $(1, 0)$. Increasing the limit to four variables (Figure L.2) reveals a "spray" of zeros emanating from a point near 1. Increasing the limit further (Figures L.3 through L.5) demonstrates three interesting features:

- The real axis becomes denser with zeros;
- More "sprays" of zeros appear;
- A clump of zeros forms around the real value -0.3;

Further work investigating these phenomenon could produce some interesting theoretical results.

7.7.3 A Monster

The final plot in this section demonstrates the overwhelming complexity of the distribution of chromatic zeros for more complex functions. Figure 7.25 shows the complex chromatic zeros (sampled over a large number of branches) of the function

$$f = 11111111100001001000000000000000,$$

which has chromatic pseudonomial:

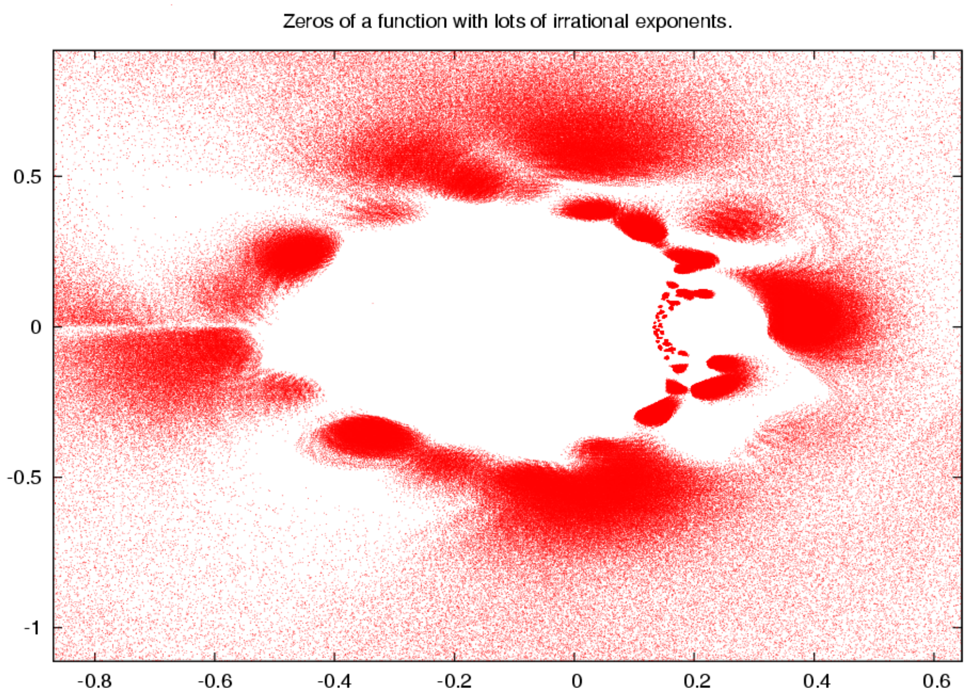


Figure 7.25: Chromatic zeros of a single Boolean function. This demonstrates the limitless complexity of chromatic zero distribution.

$$P(f;z) = -1 + 5z - 7z^{\log 3} + 5z^{\log 5} - z^{\log 6} - z^{\log 7} + z^3 - z^{\log 9} - z^{\log 10} + z^{\log 11}.$$

The complexity of chromatic zeros increases with every irrational exponent in the pseudonomial. This example demonstrates that with 7 irrational exponents the complexity is extraordinary. It is important to note that the plot is highly sensitive to which branches are actually sampled. An important extension of this work would be to sample the branches irregularly and see if any common structures appear.

7.8 Discussion

The results of this research were presented in this chapter. Discussions of the separate classes and theoretical results were presented in their respective sections, and this section will just present a general analysis of the project.

It was intended that a lot of the theorems concerning graphs could be generalised to the space of Boolean functions. In general, this was not achievable, as most of the theoretical results concerning graphs use deletion–contraction proofs, or properties of the chromatic polynomial form. Deletion–contraction proofs work by deleting and contracting edges of a graph, and is essentially a form of induction. Deletion–contraction of Boolean functions has been formalised (see Farr's original work [14]), however the space of Boolean functions is not closed under these operations, and was not considered in this project. Other graph theoretic results use properties of polynomials as a main component of the proof, and hence could not be generalised to the non–polynomial form of chromatic pseudonomials.

The new theoretical results presented are based on simple algebraic properties of pseudonomials (Loop Theorem) or specifically on properties of the function itself (Zero Bound). The zero bound (§7.1.3), while very loose, demonstrates a function–centric way of reasoning about chromatic zero properties, as it is solely based on the number of 1's in the truth table representation of the function. It is this direction which future work could go in — sufficiently abstracting away from the notions of edges and cutset spaces, to focus on function–specific properties.

Zero trails are an interesting property of generalised polynomials. The search for parametric forms was unsuccessful and much research in this area is required. Zero trails provide a geometrical pathway into the study of chromatic zeros.

Affine polynomials were shown to be equivalent to uni–cyclic graphs, with chromatic zeros lying on a circle of radius 1 with center 1. This is, however, not a new result in the graph theoretic sense. Quadratic polynomials were shown to not be in the class of graphic functions, and some brief observations were made.

EC–graphs were introduced to help visualisation of function colourings. EC–trees were shown to be isomorphic to 2–CNF functions, and hence could provide combinatorial ways of reasoning about 2–CNF colourings. The edge constraining mechanisms were shown in Theorem 16 to be highly constraining, demonstrated by the highly uncolourable family of cross–trees. Snake paths, a family of EC–trees, were shown to always be 2–colourable, with exactly one 2–colourings. Generally, 2–CNF functions appeared to have chromatic zeros converging to both 0 and 1, and this observation was formalised in Conjectures 22 and 23. 3–CNF functions were observed to be more complex than 2–CNF, and future work in this area could produce some interesting results.

Almost–trees were introduced and some interesting results obtained. Specifically it was shown that almost–trees with an odd number of edges have chromatic zeros in the region $(1/2, 1)$. The chromatic zeros also produced some interesting zero trails.

The restricted random function samples (§7.7.1,§7.7.2) presented some interesting phenomenon, described qualitatively as *concentric bands* and *zero sprays*. Interesting questions arised about zero–free regions, density, and outer bounds of these samples, however, there weren't any significant results generated in the restricted time available.

The zero–finding software doesn't usually locate all the zeros in a single branch. This could essentially mean that many outliers are being left out from the plots. Further improvements to the software would greatly support future exploration in this area. The software also has a minor flaw, in that it currently samples branches in order. This may obfuscate the structure of the plot, or introduce patterns that are not produced by the distribution of the zeros (but rather the sampling procedure itself). For example, Figure 7.25 demonstrates non–symmetry about the real axis, this is a side–effect of the branch sampling procedure.

Overall, the theorems proven, conjectures posed and observations made demonstrate that this is an interesting new field of experimental mathematics, and worthy of further investigation.

Chapter 8

Conclusion

The aim of this project was to investigate the chromatic pseudonomial of Boolean functions through an empirical study of its zeros. Some interesting zero distributions were observed and theoretical results discovered, which demonstrate that this new field is vast and complex. The developed software supported this investigation and was an integral component.

The limitations of the research were primarily theoretical. The field of generalised polynomials is still immature, and hence a lot of research time was spent trying to locate useful theorems or attempting to develop them from scratch. The other theoretical constraint was due to the relatively unstructured nature of the generalisation, causing many graph theoretic proofs to be ungeneralisable (as discussed in Section 7.8).

Boolean functions are simple to understand; however, the space of Boolean functions is extremely large[1], and hence there is an enormous variety of subclasses, including ones corresponding to graphs, matroids, and EC–graphs. This is an extreme case of generality and it is unlikely that there will ever be a Grand–Unifying–Theory of chromatic zeros of Boolean functions. A benefit of using this generalisation is that it reveals a whole plethora of interesting subclasses, of which, graphic functions are but a small subset. The particular subclasses explored in this investigation provides a wide variety of interesting behaviour, and it is evident that more subclasses need to be explored.

The 2–CNF class of functions was particularly interesting. This seemingly simple class of functions generate complex zero distributions and contain many families of functions, such as cross–trees and snake paths. It is also interesting because its size is similar to the class of graphic functions, and while there is a small amount of overlap, it generally contains non–graphic functions. It would be extremely interesting to see if graph theoretic results of chromatic roots held for 2–CNF functions, such as density in the entire complex plane.

This research contributed to the theory of graph colouring by providing a first look into Boolean function colouring. Overall, the scope of the project was immense, but the research hopefully shed some light on this new field of study. There is an enormous amount of work to still be completed, suggestions for which are now proposed.

8.1 Future Work

The scope of this project was extremely large, and there is much work to be completed in this area. We now present some suggestions.

[1]There are 2^{2^n} Boolean functions on n variables alone.

The chromatic zero–free intervals of graphs are precisely $(-\infty, 0), (0, 1),$ and $(1, 32/37]$ (see e.g., [42]). The chromatic zero–free intervals for matroids [13] are $(-\infty, 0)$ and $(1, 32/37]$. It was proven that the interval $(1/2, 1)$ contains countably many zeros in the class of almost–trees; however, two other intervals $(-\infty, 0)$, and $(1, 32/27]$ were not considered during this investigation. An interesting future project could focus on finding zero–free intervals for a range of Boolean function classes, like almost–trees, 2–CNF, or quadratic polynomials. Other unsolved problems regarding real intervals include the exact distribution of zeros in $(0, 1)$, and whether or not $(0, 1/2)$ is zero–free.

Many classes of Boolean functions were not examined in this project. Future work could look at classes such as 2–DNF, 3–DNF, k–DNF, k–CNF, other families of EC–graphs, and other polynomials. The list is limitless and it is unclear which classes will be more fruitful. This research has revealed that 2–CNF and quadratic polynomials have an immense amount of complexity within them, and as such, future work could extend the theory in these areas.

EC–graphs and EC–trees present interesting new problems in combinatorics and graph colouring; however, the theory needs a lot of work. It is also unclear whether these objects are useful for anything else except reasoning about colourings of Boolean functions. Ideally, we would like a combinatoric object that is graph–like and isomorphic to the *entire class* of Boolean functions. This would provide a mechanism for investigating Boolean functions, but could also provide a *natural* domain to study graphs and graph colourings.

The theory of zero trails and chromatic pseudonomials is still immature and could be explored much further. This is primarily a problem in complex analysis and may suit an analyst rather than a combinatorialist. Sufficient development in this area could expose important properties of graph colouring that are not currently visible.

This is a just a small sample of the possible directions for future research.

References

[1] S. Beraha, J. Kahane, and N. J. Weiss. Limits of chromatic zeros of some families of maps. *Journal of Combinatorial Theory, Series B*, 28:52–65, 1980.

[2] G. Berman and W. T. Tutte. The golden root of a chromatic polynomial. *Journal of Combinatorial Theory*, 6:301–302, 1969.

[3] N. L. Biggs, R. M. Damerell, and D. A. Sands. Recursive families of graphs. *Journal of Combinatorial Theory Series B*, 12:123–132, 1972.

[4] Norman Biggs. *Algebraic Graph Theory*. Cambridge University Press, London, 1974.

[5] George D. Birkhoff. A determinant formula for the number of ways of coloring a map. *Annals of Mathematics, Series 2*, 14(1):42–46, 1912.

[6] Jason Brown, Carl Hickman, Alan D. Sokal, and David G. Wagner. On the chromatic roots of generalized theta graphs. *Journal of Combinatorial Theory Series B*, 83:272–297, 2001.

[7] E. K. Burke, D. G. Elliman, and R. F. Weare. A university timetabling system based on graph colouring and constraint manipulation. *Journal of Research on Computing in Education*, 27 (1):1–18, 1994.

[8] C. Carstensen, Niš, and M. S. Petković. On some interval methods for algebraic, exponential and trigonometric polynomials. *Computing*, 51:313–326, 1993.

[9] Carsten Carstensen and Martin Reinders. On a class of higher order methods for simultaneous rootfinding of generalized polynomials. *Numerische Mathematik*, 64:69–84, 1993.

[10] Gregory J. Chaitin, Marc A. Auslander, Ashok K. Chandra, John Cocke, Martin E. Hopkins, and Peter W. Markstein. Register allocation via coloring. *Computer Languages*, 6:47–57, 1981.

[11] Henry H. Crapo. The Tutte polynomial. *Aequationes Mathematica*, 3:211–229, 1969.

[12] R. Diestel. *Graph Theory*. Springer-Verlag, New York, electronic edition, 2005. URL `http://www.math.uni-hamburg.de/home/diestel/books/graph.theory/download.html`.

[13] Hugh Edwards, Robert Hierons, and Bill Jackson. The zero-free intervals for characteristic polynomials of matroids. *Combinatorics, Probability and Computing*, 7:153–165, 1998.

[14] Graham E. Farr. A generalization of the Whitney rank generating function. *Mathematical Proceedings of the Cambridge Philosophical Society*, 113:267–280, 1993.

[15] Graham E. Farr. Some results on generalised Whitney functions. *Advances in Applied Mathematics*, 32:239–262, 2004.

[16] Graham E. Farr. Tutte-Whitney polynomials: some history and generalizations. In G. R. Grimmett and C. J. H. McDiarmid, editors, *Combinatorics, Complexity and Chance: A Tribute to Dominic Welsh*. Oxford University Press, to be published in Jan 2007.

[17] E. J. Farrell. Chromatic roots — some observations and conjectures. *Discrete Mathematics*, 29:161–167, 1980.

[18] Andreas Frommer. A unified approach to methods for the simultaneous computation of all zeros of generalized polynomials. *Numerische Mathematik*, 54:105–116, 1988.

[19] Frank Harary. *Graph Theory*. Addison-Wesley, California, 1972.

[20] Peter Henrici. *Applied and Computational Complex Analysis. Vol 1. Power series — Integration – Conformal Mapping – Location of Zeros*. Wiley, Toronto, 1974.

[21] Bill Jackson. Zeros of chromatic and flow polynomials of graphs. *Journal of Geometry*, 76: 95–109, June 2003.

[22] G.J.O. Jameson. Counting zeros of generalized polynomials: Descartes' rule of signs and Laguerre's extensions. *Mathematical Gazette, to appear*, 2006. URL `http://www.maths.lancs.ac.uk/~jameson/`. Electronic doc retrieved on 15.05.2006.

[23] Erwin Kreyszig. *Advanced Engineering Mathematics*. Wiley, New York, NY, 8th edition, 1999.

[24] Joseph P. S. Kung. The Rédei function of a relation. *Journal of Combinatorial Theory, Series A*, 29:287–296, 1980.

[25] Joseph P. S. Kung, M. Ram Murty, and Gian-Carlo Rota. On the Rédei zeta function. *Journal of Number Theory*, 12:421–436, 1980.

[26] Michel L. Lapidus and Machiel von Frankenhuysen. Complex dimensions of self-similar fractal strings and Diophantine approximation. *Experimental Mathematics*, 12:41–69, 2001.

[27] Cliff Long and Thomas Hern. Graphing the complex zeros of polynomials using modulus surfaces. *College Mathematics Journal*, 20(2):98–105, March 1989.

[28] John Mitchem. On the history and solution of the four-color map problem. *The Two-Year College Mathematics Journal*, 12(2):108–116, March 1981.

[29] James Oxley. What is a matroid?, December 2004. URL `http://www.math.lsu.edu/~oxley/`. Revision of the article of the same name in *Cubo*, (5):179–218, (2003).

[30] Pari/GP. PARI/GP Development Headquarters, April 2006. URL `http://pari.math.u-bordeaux.fr/`. accessed April 2006.

[31] R. B. Potts. Some generalized order-disorder transformations. In *Proceedings of the Cambridge Philosophic Society*, volume 48, pages 106–109, 1953.

[32] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 2nd edition, 2003.

[33] Ronald C. Read. An introduction to chromatic polynomials. *Journal of Combinatorial Theory*, 4:52–71, 1968.

[34] Ronald C. Read and Gordon F. Royle. Chromatic roots of families of graphs. In Y. Alavi, G. Chartrand, O. R. Oellermann, and A. J. Schwenk, editors, *Graph Theory, Combinatorics, and Applications*, volume 2, pages 1009–1029. Wiley, New York, 1991.

[35] Daniel Shanks. A logarithm algorithm. *Mathematical Tables and Other Aids to Computation*, 8(46):60–64, April 1954.

[36] Robert Shrock and Shan-Ho Tsai. Families of graphs with chromatic zeros lying on circles. *Physical Review E*, 56(2):1342–1345, August 1997.

[37] Alan D. Sokal. Chromatic polynomials, Potts models and all that. *Physica A*, 279:324–332, 2000.

[38] Alan D. Sokal. Bounds on the complex zeros of (di)chromatic polynomials and Potts-model partition functions. *Combinatorics, Probability, and Computing*, 10:41–47, 2001.

[39] Alan D. Sokal. The multivariate Tutte polynomial (alias Potts model) for graphs and matroids. In Bridget S. Webb, editor, *Surveys in Combinatorics, 2005*, volume 327 of *London Mathematical Society Lecture Note Series*, pages 173–226. Cambridge University Press, 2005.

[40] Andrew J. Sommese and II Charles W. Wampler. *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific, Singapore, 2005.

[41] Ian Stewart and David Tall. *Complex Analysis*. Cambridge University Press, Cambridge, 1983.

[42] Carsten Thomassen. The zero-free intervals for chromatic polynomials of graphs. *Combinatorics, Probability and Computing*, 6:497–506, 1997.

[43] Timo Tossavainen. On the zeros of finite sums of exponential functions. *Australian Mathematical Society Gazette*, 33(1):47–50, March 2006.

[44] W. T. Tutte. A contribution to theory of chromatic polynomials. *Canadian Journal of Mathematics*, 6:80–91, 1954.

[45] W. T. Tutte. On dichromatic polynomials. *Journal of Combinatorial Theory*, 2:301–320, 1967.

[46] W. T. Tutte. On chromatic polynomials and the golden ratio. *Journal of Combinatorial Theory*, 9:289–296, 1970.

[47] J. van de Lune and H.J.J. te Riele. Numerical computation of special zeros of partial sums of Riemann's zeta function. In H.W. Lenstra, Jr. and R. Tijdeman, editor, *Computational Methods in Number Theory – Part II*, pages 371–387, Mathematical Centre Tracts 155, Mathematisch Centrum, Amsterdam, 1982.

[48] D. J. A. Welsh and M. B. Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *Computer Journal*, 10:85–86, 1967.

[49] Hassler Whitney. The coloring of graphs. *Annals of Mathematics, Series 2*, 33(4):688–718, October 1932.

[50] Hassler Whitney. A logical expansion in mathematics. *Bulletin of the American Mathematical Society*, 38:572–579, 1932.

[51] Hassler Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57(3):509–533, July 1935.

[52] Geoff Whittle. Characteristic polynomials of weighted lattices. *Advances in Mathematics*, 99: 121–151, 1993.

[53] F.Y. Wu. The Potts model. *Reviews of Modern Physics*, 54(1):235–266, January 1982.

Appendix A

Glossary

Conjunctive Normal Form (CNF) A Boolean function is in conjunctive normal form (CNF) if it is a conjunction of clauses. A clause is a disjunction of literals. Literals are (possibly negated) variables. A function is in k–CNF if it is in CNF and each clause has at most k literals.

Incidence Matrix The incidence matrix of a graph G has $|V(G)|$ rows, each one corresponding to a particular vertex of the graph, and $|E(G)|$ columns, with each edge assigned to a unique column. The element e_{ij} in the matrix equals 1 if vertex i is incident on edge j, or 0 otherwise.

Ground Set The ground set of a Boolean function is a set that indexes the variables of the function. It is the analogue of the edge–set of a graph. It is also used in the terminology of matroids (discussed below).

Matroid Matroids are structures that arise from abstracting properties of linear dependence from other objects, for example graphs or vector spaces. Matroids were introduced by Hassler Whitney [51] in 1935. An accessible introduction to the topic of matroids can be found in James Oxley's "What is a Matroid?" [29]. There are many, equivalent, definitions of matroids, one of which is:

Definition: Matroid
A matroid consists of a ground set S, and a rank function $r : 2^S \to \mathbb{Z}$, which satisfies the following three axioms:

1. $\forall A : r(A) \leq |A|$
2. $\forall A, B : A \subseteq B \to r(A) \leq r(B)$
3. $\forall A, B : r(A \cup B) + r(A \cap B) \leq r(A) + r(B)$

Riemann Surface A Riemann surface (named after Bernhard Riemann) is a surface that locally looks like part of the complex plane. It provides a setting for which the study of multivalued functions (like the chromatic pseudonomial) becomes simpler. The Riemann surfaces considered for the purposes of this project are quite simple and can be thought of as infinite copies of the complex plane stacked upon each other and joined by cuts along the negative real axis.

Intermediate Value Theorem The intermediate value theorem states that for any continuous function f over an interval $[a, b]$ the following holds: For all x in–between $f(a)$ and $f(b)$ we can find an $x' \in [a, b]$ such that $f(x') = x$.

Appendix B

Subgraph Expansion

Whitney [50] elegantly presented a subgraph expansion of the chromatic polynomial as follows. It uses the inclusion–exclusion principle presented below.

B.1 Inclusion-Exclusion Principle

Consider a set of n items and a set of *properties* A_i that each item either has or does not have. Then the number of items without all the properties is given by the formula:

$$\begin{aligned} n(\bar{A}_1 \cdots \bar{A}_m) =& n - \sum_i n(A_i) + \sum_{i,j>i} n(A_i A_j) + \cdots \\ &+ (-1)^l \sum_{i,i_1>i,i_2>i_1,\ldots,i_l>i_{l-1}} n(A_i A_{i_1} \cdots A_{i_l}) \\ &+ \cdots + (-1)^m n(A_1 \cdots A_m). \end{aligned}$$

B.2 Subgraph Expansion

Consider a graph and let the set of items be all arbitrary colourings of that graph. For a specific item let A_e be the property that e has identical colourings on its endpoints. We then have

$$\begin{aligned} P(G;\lambda) &= n(\bar{A}_{e_1} \cdots \bar{A}_{e_m}) \\ &= \sum_{X \subseteq E} (-1)^{|X|} (\text{\# of colourings where } X \text{ is contracted and } E \backslash X \text{ is removed.}) \\ &= \sum_{X \subseteq E} (-1)^{|X|} \lambda^{(\text{\# of components on } G\langle X \rangle \text{ including isolated vertices})} \\ &= \sum_{X \subseteq E} (-1)^{|X|} \lambda^{n-\rho(X)} \\ &= \lambda^{k(G)} \sum_{X \subseteq E} (-1)^{|X|} \lambda^{\rho(G)-\rho(X)}, \end{aligned}$$

as required.

Appendix C

Rédei Function and Chromatic Polynomial

We consider, in detail, how a distinguishing $n-$tuple, u, generates (many) 2^n colourings. Consider the field $F = \mathbb{Z}_2^n$ (i.e. bitstrings of length n). To each edge e of G assign a value $(e_1, \ldots, e_n)$ from F where

$$e_i = \begin{cases} 1, & \text{if } e \in u_i \\ 0, & \text{otherwise.} \end{cases}$$

(Note that because u distinguishes E no edge is assigned a zero value.)

With an arbitrary direction imposed on the edges (which becomes irrelevant as we are working *mod* 2) these edge assignments comprise what Tutte [44] called a *colour–coboundary*, but has a modern name of a *tension*. If we fix a colour (from F) at a specific vertex then the tension provides the differences in colours between the edges, and hence $P(G;n) = n^{k(G)}\theta(G;n)$ (as noted by Tutte in [44]), where $\theta(G;n)$ is the number of $n-$tensions on G.

What remains is to show that u generates a valid tension, this just requires a proof that the sum of tensions around every cycle is 0 (and hence once we fix a colour at a single vertex in each component the generated colouring is uniquely defined). The proof is trivial in F, as for all cycles $Y \subseteq E$ we can show that $\forall i. Y \cap u_i \neq \emptyset \Rightarrow |Y \cap u_i|$ is even, and hence $\forall i. \sum_{y \in Y} y_i = 0 \pmod 2$. For any other choice of field (of order 2^n) the property, remarkably, also holds – though seems somewhat more difficult to prove.

So, to summarise, we have

$$\zeta_{E|C}(n) = \text{ \# of } \mathbb{Z}_2^n \text{ tensions on } G,$$

and moreover

$$P(G; 2^n) = 2^{nk}\zeta_{E|C}(n),$$

and hence

$$P(f; 2^n) = \text{ \# of } n\text{−tuples of elements in } \operatorname{supp} f \text{ whose union is } S.$$

Appendix D

Software and Technical Issues

The software will be available online at `www.csse.monash.edu.au/~bspor1/`. All the code is documented and a README is provided to help navigate the code.

The software was developed using C++, and compiled under Debian Linux using the Gnu C++ compiler (version 3.3.5). The plots were all produced using GnuPlot (version 4.0). The version of Pari used was 2.1.7.

D.1 Programs

The programs provided need to be compiled before use. Instructions for building the software are presented in the next section. How to use the software is explained in the section following that. The various programs available are described in Section 6, however, a brief summary is presented here.

- `chromatic`: Generates the chromatic pseudonomial from a functions specification;
- `real`: Samples a chromatic pseudonomial over a real interval, and optionally, locates the zeros;
- `sample`: Samples a chromatic pseudonomial over a complex region;
- `pc`: Locates complex chromatic zeros of a Boolean function (requires Pari);
- `iterative`: Locates complex chromatic zeros of a Boolean function (does not require Pari);
- `tracer`: Traces out a zero trail of a chromatic pseudonomial;
- `vecfield`: Samples the steepest descent of chromatic pseudonomial over a complex region.

D.2 Building the Software

A makefile is provided which assists in building the software. To build `pc` you must have the *Pari* library [30] with relevant links to the library updated in the Makefile. If you do not have access to Pari the program `iterative` also finds complex chromatic zeros using a simple steepest–descent search. Typing `make all` will generate all the available software.

D.3 Using the Software

Each program takes command–line arguments to specify various parameters. The argument `help` will provide a list of all the command–line parameters available, as well as the defaults used. An example of the output is shown below.

```
$./pc --help
Zero search via homotopic continuation.
Usage: ./pc [OPTIONS]
        -b 3, --branches 3
                Number of branches to sample.
        -c 5, --correcth 5
                Decrease this parameter for a "rougher" search.
        -e 1e-08, --epsilon 1e-08
                Accuracy. (-'ve for max accuracy)
        -m 100, --euleriters 100
                Maximum number of iterations (euler) in zero search.
        -n 50, --newtoniters 50
                Maximum number of iterations (newton) in zero search.
        -z 0.04, --newtonstep 0.04
                Iterative step size (newton).
        -o test, --out test
                Output file prefix. Generates x.dat output file.
        -p, --noprogress (def: false)
                Disable the progress output.
        -r 1, --random 1
                Amount of stochastic perturbation to homotopy.
        -s 0.01, --eulerstep 0.01
                Iterative step size (euler).
        -f 1110, --function 1110
                The Boolean function
By Benjamin Porter <benjamin.porter@gmail.com> [260706]
```

Appendix E

Homotopic Predictor–Corrector

Algorithm: Homotopic Predictor–Corrector
Input:

$$P(z) = \sum_i a_i z^{\log i}$$

Let $Q(z) = \sum_i a_i z^i$ approximate[1] P. Let $H(z,t) = tQ(z) + (1-t)P(z)$ interpolate between Q and P. We now solve for $Q(z) = 0$ using the external polynomial root–finding package Pari [30].

Consider a root, z, of Q. This is also a solution to $H(z,1) = 0$, due to the construction of H. The main idea behind homotopic continuation is that we can incrementally reduce t until $t = 0$, at each stage keeping $H(z_i, t_i) = 0$. When $t = 0$, then $H(z,0) = 0$ and hence $P(z) = 0$. We now demonstrate the prediction–correction method of tracking the zero path from $H(z_0, 1) = 0$ to $H(z_n, 0) = 0$.

The Taylor–series approximation of H is

$$H(z + \Delta z, t + \Delta t) \approx H(z,t) + \Delta z H_z(z,t) + \Delta t H_t(z,t),$$

where $H_z(z,t) = \partial H(z,t)/\partial z$, $H_t(z,t) = \partial H(z,t)/\partial t$, and Δt and Δz are sufficiently small. We use this approximation in the following prediction and correction steps.

Let $H(z_i, t_i) = 0$. We want Δz and Δt such that $H(z_i + \Delta z, t_i + \Delta t) = 0$. From the Taylor series approximation above we have:

$$\Delta z = -\Delta t \frac{H_t(z_i, t_i)}{H_z(z_i, t_i)},$$

for small enough Δz and Δt. Choose some small step size h, and let $\Delta t = -h$. Then we have

$$\Delta z = h \frac{H_t(z_i, t_i)}{H_z(z_i, t_i)}.$$

This is the *prediction* step (specifically a "tangent predictor"), and usually involves some amount of error. To correct this we hold $t_i + \Delta t$ constant, and use a *correction* step, which in this implementation is just Newton's method. Using the Taylor series approximation again we get the required correction Δc to be:

$$\Delta c = \frac{-H(z_i + \Delta z, t_i - h)}{H_z(z_i + \Delta z, t_i - h)}.$$

[1] A naïve, but practical, approximation.

We may need to repeat the correction step a few times until it converges to the solution. Once this has been done we set

$$t_{i+1} = t - h,$$

and

$$z_{i+1} = z + \Delta z + \textit{(correction)}.$$

We repeat this until we have $t = 0$, then perform a final Newton iteration to increase the accuracy of the solution.

A lot of minor technical detail has been excluded from the present discussion for the sake of clarity. Refer to the implementation for more detail.

Appendix F

Predictor–Corrector Implementation

The predictor–corrector can be found in `PC.h` and `PC.cpp`. Important components of the implementation can be found here.

The predictor–corrector interface is through the function `pc`. It has the following prototype:

```
std::list<complex> pc(Pseudonomial& p, const PCParams& par);
```

The function `pc` takes a pseudonomial and a set of system parameters and returns a list of chromatic zeros.The class `Pseudonomial` is an abstraction of a pseudonomial and contains a list of pairs of integers. For example, the pseudonomial $p(z) = 1 - 2z + z^{\log 3}$ is stored as the list $[(1,1),(-2,2),(1,3)]$. Pseudonomials are generated by the `Function` class (which abstracts a Boolean function). The system has many user–definable parameters, such as number of iterations, step size, and accuracy. These are passed in the struct `PCParams`.

```
struct PCParams
{
        int MAX_ITERS;
        real STEP_SIZE;
        real EPS;
        ...
};
```

However, `pc` is just a driver, the real work is done in `predictor_corrector`. Given a function f and a homotopy g (and their derivatives), `predictor_corrector` tracks the path from the zeros of g (passed as the parameter `gRoots`) to the zeros of f, using the homotopic prediction–correction method summarised in Section E.

```
std::list<complex> predictor_corrector(CFunction& f, CFunction& df,
        CFunction& g, CFunction& dg,
        std::list<complex> gRoots, const PCParams& par);
```

The class `CFunction` just abstracts an arbitrary complex–valued function, and specific subclasses are available corresponding to pseudonomials or polynomials. This makes the sofware reusable as the algorithm can be used for *any* function.

```
class CFunction // complex function
{
        public:
        virtual complex operator()(complex z) = 0;
};
```

The code for `predictor_corrector` is included here. The `iterative1` function call performs a simple Newton iteration.

```
std::list<complex> predictor_corrector(CFunction& f, CFunction& df,
        CFunction& g, CFunction& dg,
        std::list<complex> gRoots, const PCParams& par)
{
        std::list<complex> newRoots;
        std::list<complex>::iterator it = gRoots.begin();

        while (it!=gRoots.end()) // foreach root zi
        {
                complex zi = *it;
                real ti = 1;
                real dt = par.STEP_SIZE;
                int correct = 0;
                bool failed = false;

                // h = t(g(z)-f(z)) + f(z)
                // numerically solve h(z,t) = 0
                // given h(z_0,1) = 0
                // find z s.t. h(z,0) = 0

                real epsilon = par.EPS_L;
                int iteration = 0;
                while (iteration < par.MAX_ITERS)
                {
                        if (dt < std::numeric_limits<real>::epsilon())
                                { failed = true; break; }

                        // adaptive epsilon
                        if (ti < .4) epsilon = par.EPS_M;
                        if (ti < .2) epsilon = par.EPS_S;

                        // INVARIANT h(zi,ti) < EPS

                        if (ti < dt) dt = ti; // one more step to go..

                        // predict (Euler)
                        // z' = zi - dt * (dh/dt)/(dh/dz)
                        complex dhdtp = g(zi) - f(zi);
                        complex dhdzp = ti*(dg(zi)-df(zi)) + df(zi);
                        complex zp = zi + dt * dhdtp / dhdzp;
                        real tp = ti - dt;

                        // correct (Newton iteration)
                        // hold t constant
```

```
// z'' = z' - h(zp,tp) / (dh/dz)(zp,tp)
int newton_iters = 0;
complex zpp = zp; // z'' = z'
complex hzpptp;
while (newton_iters < par.MAX_N_ITERS)
{
        complex h = tp*g(zpp) + (1-tp)*f(zpp);
        real abszpp = abs(h);

        complex dhdz = tp*(dg(zpp)-df(zpp)) + df(zpp);
        complex hdivdhdz = h/dhdz;

        real step = par.NEWTON_STEP * dt/par.STEP_SIZE;
        int j = 0;
        complex zpp2;
        real abszpp2;
        // find an appropriate step in the right direction
        do
        {
                zpp2 = zpp - step*hdivdhdz;

                abszpp2 = abs(tp*g(zpp2) + (1-tp)*f(zpp2));
                step /= 2;
                j++;
        }
        while (j < par.MAX_N_ITERS && abszpp2>abszpp);

        if (j==par.MAX_N_ITERS)
        {
                newton_iters = par.MAX_N_ITERS;
                break;
        }
        else zpp = zpp2;

        if (abszpp2 < epsilon) break;
        newton_iters++;
}

// update (zi,ti) & adjust h (adaptive step size)

if (newton_iters == par.MAX_N_ITERS)
// the corrector failed
{
        correct = 0;
        dt /= 2;
}
else
```

```
                {
                        zi = zpp;
                        ti = tp;
                        correct++;
                }

                if (correct > par.CORRECT_H)
                {
                        correct = 0;
                        dt *= 2;
                }

                // terminate
                if (ti < par.EPS) break;

                iteration++;
        }

        // add zero to new zero list ...
        if (!failed)
        {
                // have zi close to root...
                // use steepest descent to converge to that root..
                std::pair<complex,real> res = iterative1(f,df,zi,par);

                if (std::abs<real>(res.second) < par.EPS_S)
                        newRoots.push_back(res.first);
        }

        it++;
    }
    return newRoots;
}
```

Appendix G

Engineering The Zero Trails

Example:

We now demonstrate that even the simple curves of Figure 7.5 seem to be more detailed than first appearances. We do this by attempting to fit an ellipse to the inner curve.

The two points highlighted in Figure 7.5 are important, as the zero trails seem symmetric about the real axis. To analytically obtain the zero intercepts we let the branch dependent form of P be $P_n(z) = 1 - 2z + |z|^{\log 3} e^{i \log 3(Arg(z)+2n\pi)}$ and consider the two separate cases of z being a positive real number and z being a negative real number.

$z \in \mathbb{R}^+$**:** Let $x > 0$ be real. Then $P_n(x) = 1 - 2x + x^{\log 3} e^{2n\pi i \log 3}$. $P_n(x)$ can only be a zero if $x^{\log 3} e^{2n\pi i \log 3}$ is real, which only occurs when $n = k/(\log 3)$ for all $k \in \mathbb{Z}$. But $n \in \mathbb{Z}$ which implies that $n = k = 0$, however if we let n take on non-integral values (namely $n = 1/\log 3$) then we can get the real axis intercepts of the zero trails.

If $n = 0$ then $P_n(x) = P_0(x) = 1 - 2x + x^{\log 3}$, which equals zero if and only if $x = 1$ or $x = 2$. However if $n = 1/\log 3$ then $P_n(x) = P_{1/\log s}(x) = 1 - 2x - x^{\log 3}$, which (by the same result given in [22]) has at most one zero. The zero of this function gives the real axis intercept of the zero trail, and was numerically found to be approximately 0.38834194.

$z \in \mathbb{R}^-$**:** Let $x < 0$ be real. Then $P_n(x) = 1 - 2x + |x|^{\log 3} e^{i\pi \log 3(1+2n)}$. Following the same reasoning as above we get "pseudo" zeros when $n = k/(2\log 3) - 1/2$ (for all $k \in \mathbb{Z}$), noting that when $n = 0$ there can never be a zero for $P_n(x < 0)$ as $Re(P_0(x)) \neq 0$ for all $x < 0$. Substituting the above n's into the pseudonomial gives

$$Q_1(x) = P_{(\text{even } k)/(2\log 3)-1/2}(x) = 1 - 2x + |x|^{\log 3}$$

and

$$Q_2(x) = P_{(\text{odd } k)/(2\log 3)-1/2}(x) = 1 - 2x - |x|^{\log 3}.$$

Letting $x = -c$, where $c > 0$, results in

$$Q_1(c) = 1 + 2c + c^{\log 3},$$

and

$$Q_2(c) = 1 + 2c - c^{\log 3}.$$

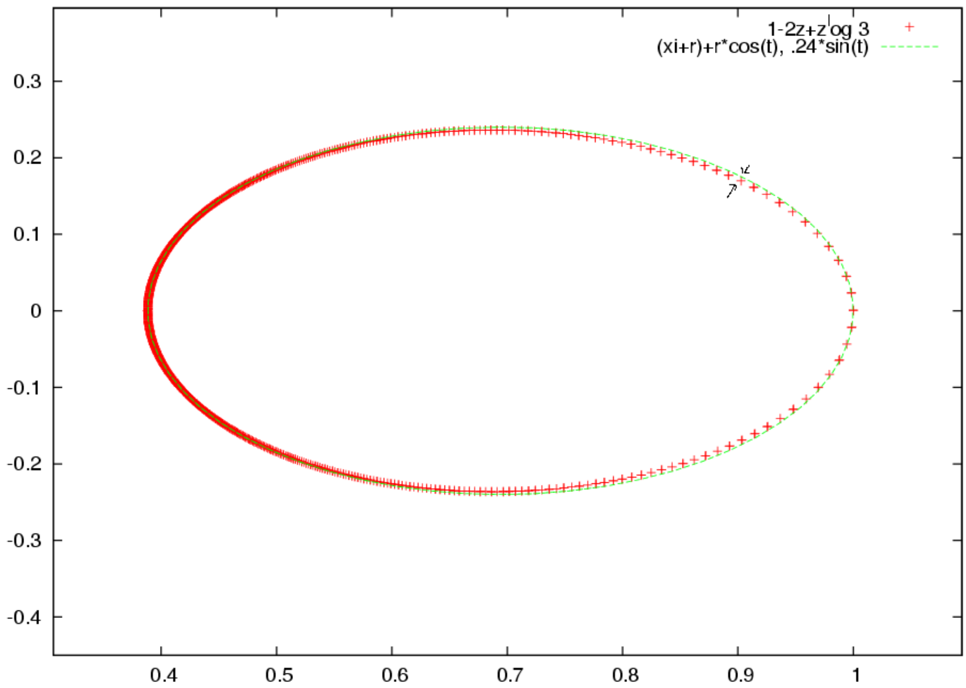


Figure G.1: Chromatic zero trail of $f1110$ and an attempt at fitting an ellipse (xi is the real axis intercept of the zero trail, and r is the horizontal radius of the proposed ellipse). Note that the ellipse fits nicely except at the region indicated by the arrows. This implies that the zero trail cannot be expressed a simple ellipse.

Q_1 obviously has no zeros, but Q_2 has a maximum of 2. Numerically we find $c = 4$ is the only zero, and hence $x = -4$ is another zero trail intercept.

These two results give the locations of the points a and b, albeit numerically. Using the location of a we can attempt to fit an ellipse to trail 1. The results of this brief attempt is given in Figure G.1, however it was a failure as the zeros deviate slightly from a perfectly elliptical path (as denoted in the Figure). This implies that the inner trail is not an ellipse but a more complicated curve.

Appendix H

Zero Trail Existence Proof

The existence of zero trails is now proven. The proof is restricted to the case of a pseudonomial with a single irrational exponent, however, the generalisation to an arbitrary number of irrational exponents follows the same argument.

Theorem 27. *Let $P_n(z) = g(z) + az^b$, where $g(z)$ is some polynomial and b is irrational. Then for all n, z, and $\delta > 0$,*

$$P_n(z) = 0 \Rightarrow \exists m \neq n, \tilde{z} \neq z \text{ such that } |z - \tilde{z}| < \delta \text{ and } P_m(\tilde{z}) = 0.$$

Proof. Pick some z and n such that $P_n(z) = 0$. We then have

$$g(z) = a|z|^b \underbrace{e^{i.\text{Arg}(z).b} e^{i.2n\pi.b}}_{h_n(z)}.$$

Pick a distance $\delta > 0$ arbitrarily. We now find an n' and z' such that $n' \neq n, z' \neq z$ such that $|z - z'| < \delta$ and $P_{n'}(z') = 0.$.

The proof structure uses the triangle inequality as follows:

$$\begin{aligned}
|P_{n'}(z')| &= |g(z') + h_{n'}(z') - g(z) - h_{n'}(z) + g(z) + h_{n'}(z)| \\
&\leq |g(z') - g(z)| + |h_{n'}(z') - h_{n'}(z)| + |g(z) + h_{n'}(z)| \\
&\leq \epsilon_1 + \epsilon_2 + \epsilon_3.
\end{aligned}$$

Each of the ϵ's can be made arbitrarily small, as now outlined.

ϵ_1 can be made arbitrarily small due to the continuity of g. ϵ_2 can also be made arbitrarily small because $h_{n'}$ is continuous. Finally ϵ_3 can be made arbitrarily small, because $P_n(z) = 0$ and we can choose a n' such that

$$\begin{aligned}
&|g(z) + h_{n'}(z)| = |P_{n'}(z)| = |-P_{n'}(z)| = \\
&|P_n(z) - P_{n'}(z)| < \epsilon_3 \\
&\iff |a|z|^b||e^{i.2n\pi b} - e^{i.2n'\pi b}| < \epsilon_3 \\
&\iff |g(z)||(e^{i.2\pi b})^n - (e^{i.2\pi b})^{n'}| < \epsilon_3,
\end{aligned}$$

for some arbitrary ϵ_3. This is possible as the second absolute term can be made as small as possible by selecting the correct n'.

$\square$

Appendix I

Proof that a 3–Cycle does not have a 2–CNF Representation

Theorem 28. *The cutset space indicator function of the 3–cycle has no 2–CNF representation.*

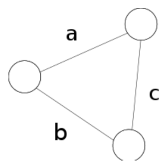


Figure I.1: The 3–cycle.

Proof. Let G be the 3–cycle (see Figure I.1). It has cutset space

$$CS(G) = \{\emptyset, \{a, b\}, \{b, c\}, \{a, c\}\}.$$

Assume f_G has a 2–CNF representation, i.e.,

$$f_G = (x_1 \vee y_1) \wedge \cdots \wedge (x_n \vee y_n), \text{ with } x_i, y_i \text{ as literals.}$$

We have:

1. $f_G(\{a, b\}) = 1$, and
2. $f_G(\{a, c\}) = 1$.

Condition 1 implies that for all i we must have

$$x_i \text{ or } y_i = a \text{ or } b \text{ or } \bar{c},$$

and condition 2 implies that for all i we must have

$$x_i \text{ or } y_i = a \text{ or } \bar{b} \text{ or } c.$$

Consider now what happens when we require that $f_G(\{b, c\}) = 1$. This implies that for all i we need

$$x_i \text{ or } y_i = \bar{a} \text{ or } b \text{ or } c.$$

Assume for some i that $x_i = \bar{a}$. Then condition 1 implies that $y_i = b$ or $\bar{c}$, and condition 2 implies $y_i = \bar{b}$ or c, hence the two conditions contradict each other. Assume then that for some i we have $x_i = b$, then condition 2 implies that $y_i = a$ or c, which cannot occur because at least one literal *must* be negative (by the validity constraint). Similary, assuming $x_i = c$ leads to a similar contradiction. The argument is similar when assuming values for y_i, and hence we conclude that f_G has no 2–CNF representation. $\square$

Appendix J

Almost–Tree Plots

Chromatic zeros of a sample of almost–trees were plotted and are presented here (see Figures J.1, J.2, J.3, J.4 and J.5).

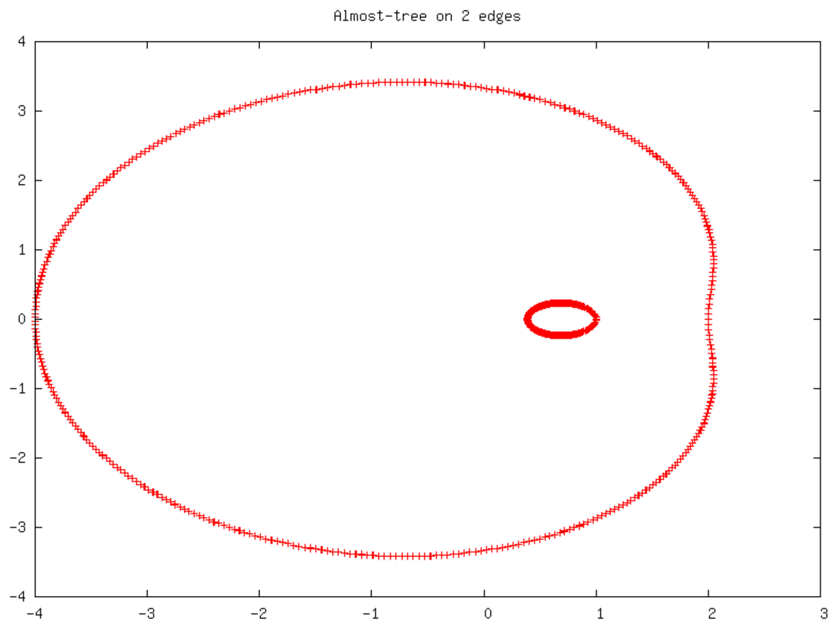


Figure J.1: Chromatic zeros of the almost–tree on 2 variables.

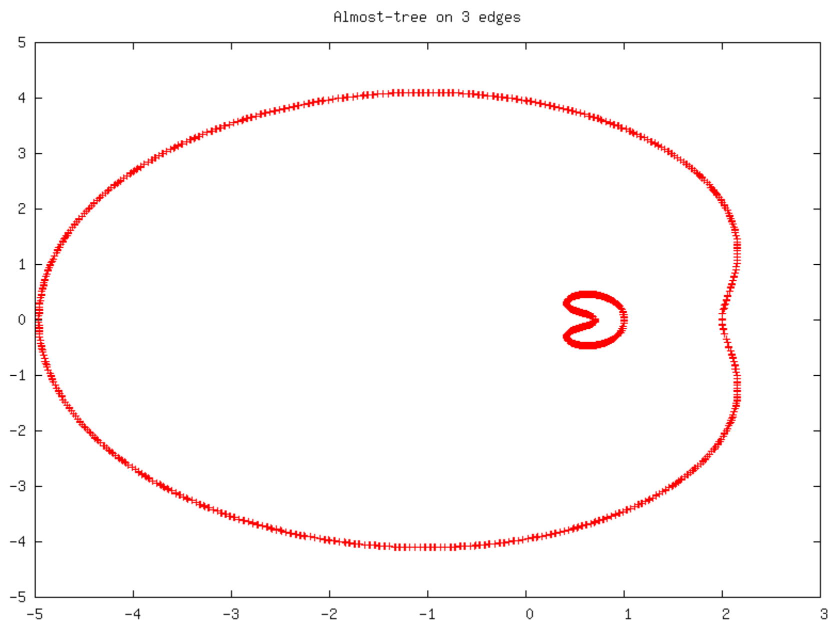


Figure J.2: Chromatic zeros of the almost–tree on 3 variables.

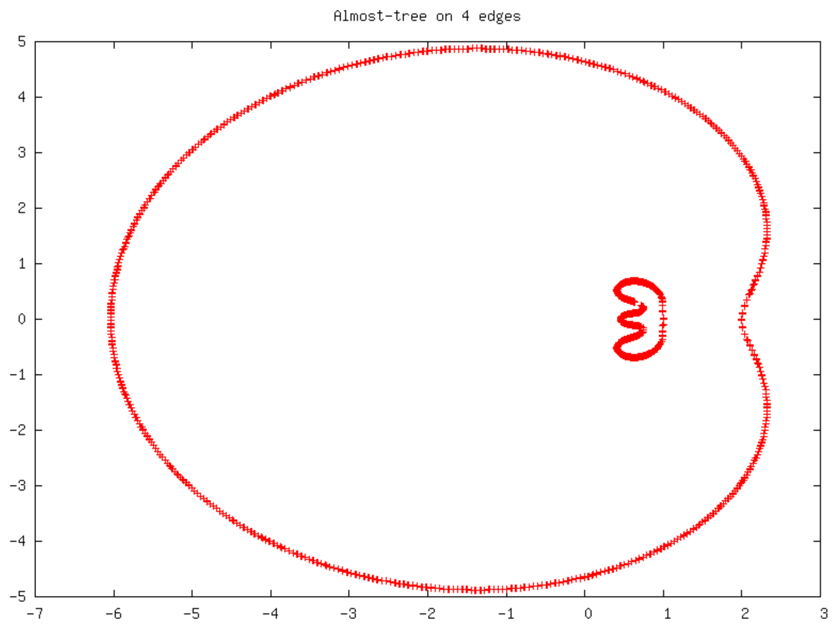


Figure J.3: Chromatic zeros of the almost–tree on 4 variables.

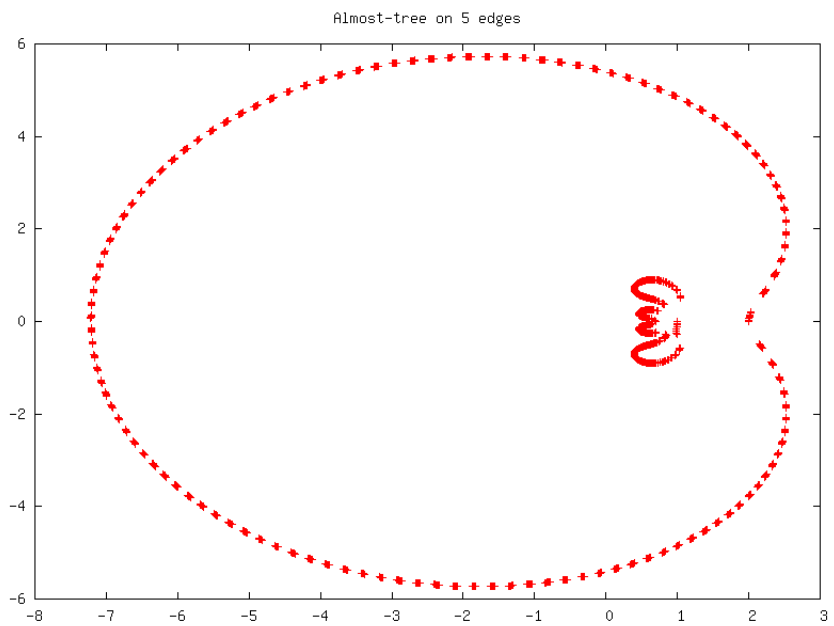


Figure J.4: Chromatic zeros of the almost–tree on 5 variables.

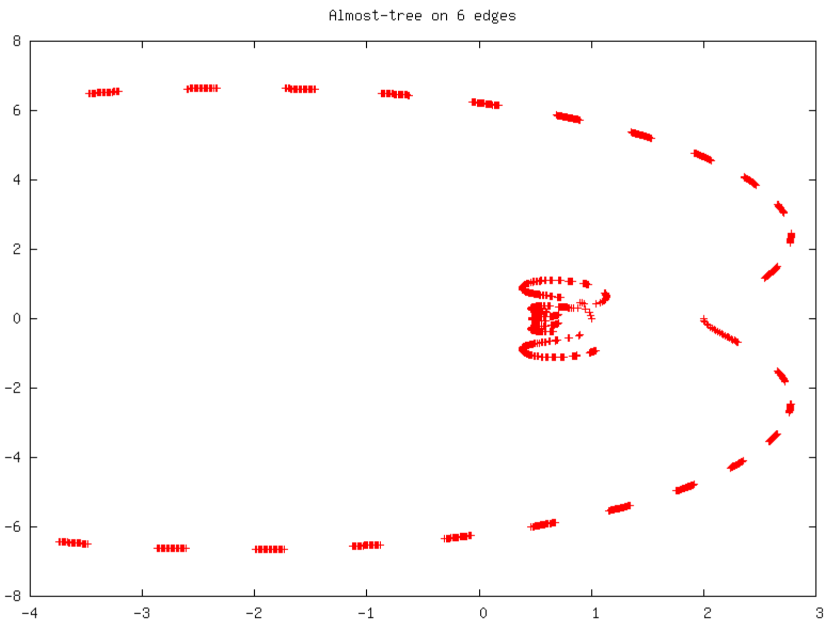


Figure J.5: Chromatic zeros of the almost–tree on 6 variables.

Appendix K

Cross–Tree Plots

Chromatic zeros of a sequence of cross–trees are presented here (Figures K.1, K.2, K.3, K.4 and K.5).

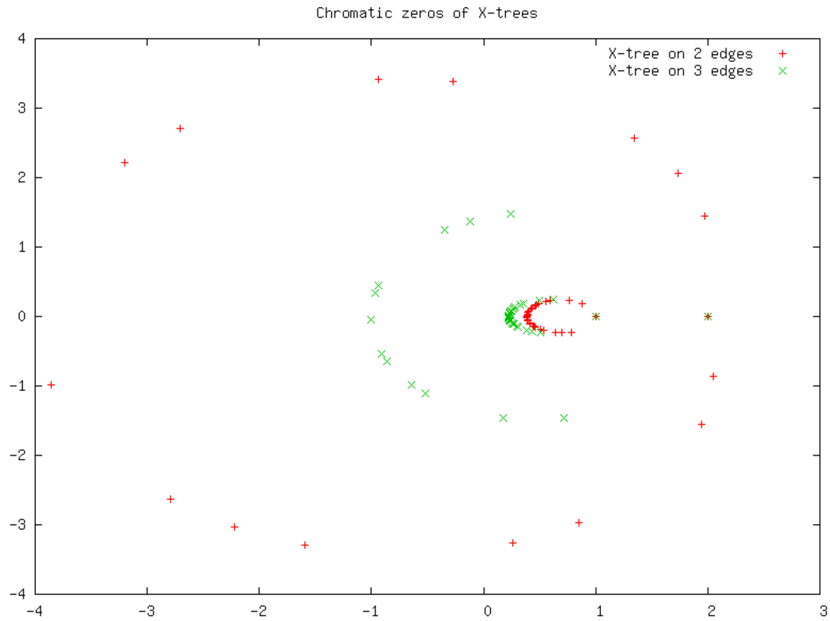


Figure K.1: Chromatic zeros of X_2 and X_3 over many (30) branches.

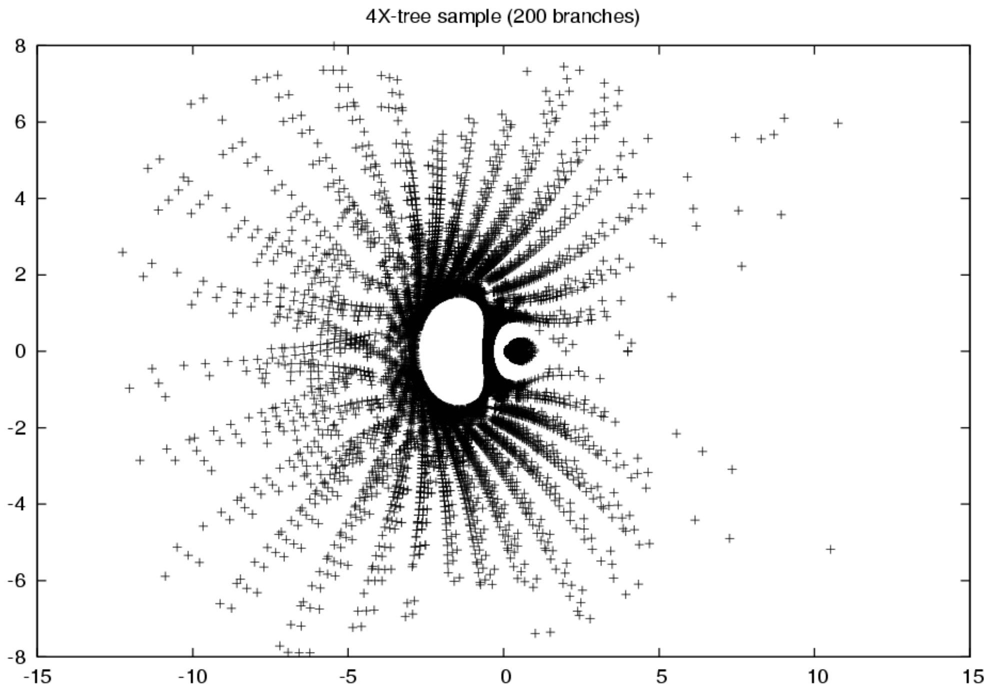


Figure K.2: Chromatic zeros of X_4.

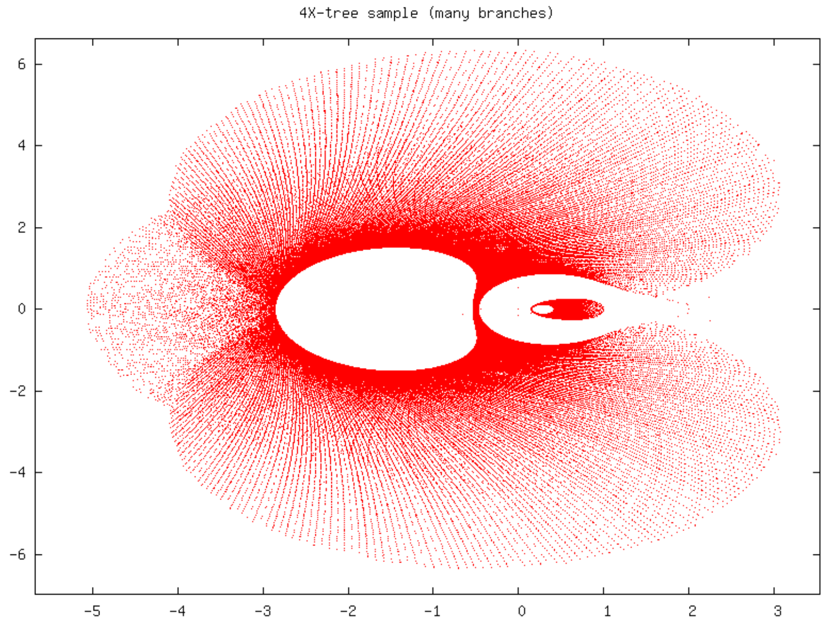


Figure K.3: More chromatic zeros of X_4.

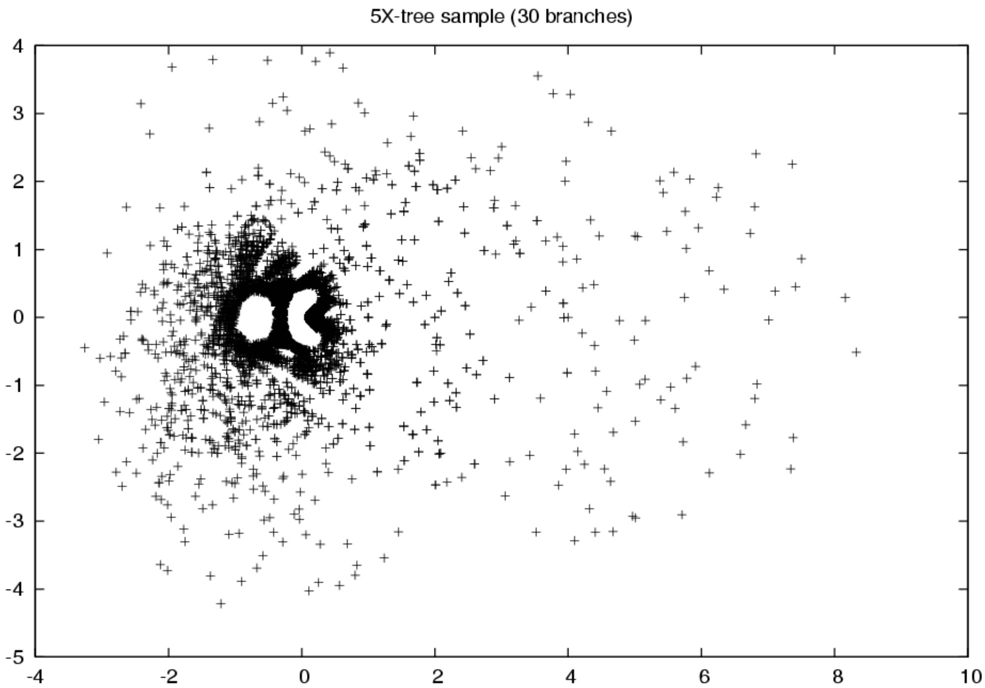


Figure K.4: Chromatic zeros of X_5.

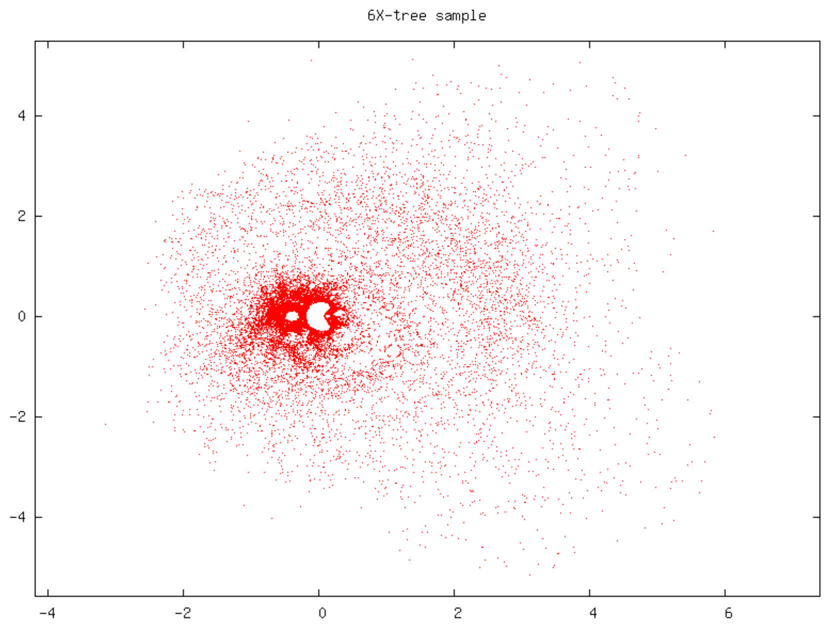


Figure K.5: Chromatic zeros of X_6.

Appendix L

Random Function Samples

The principal chromatic zeros of a random sample of functions restricted to a certain sized ground set were plotted and are presented here. Figures L.1 to L.5 demonstrate the increasing complexity of distribution of zeros.

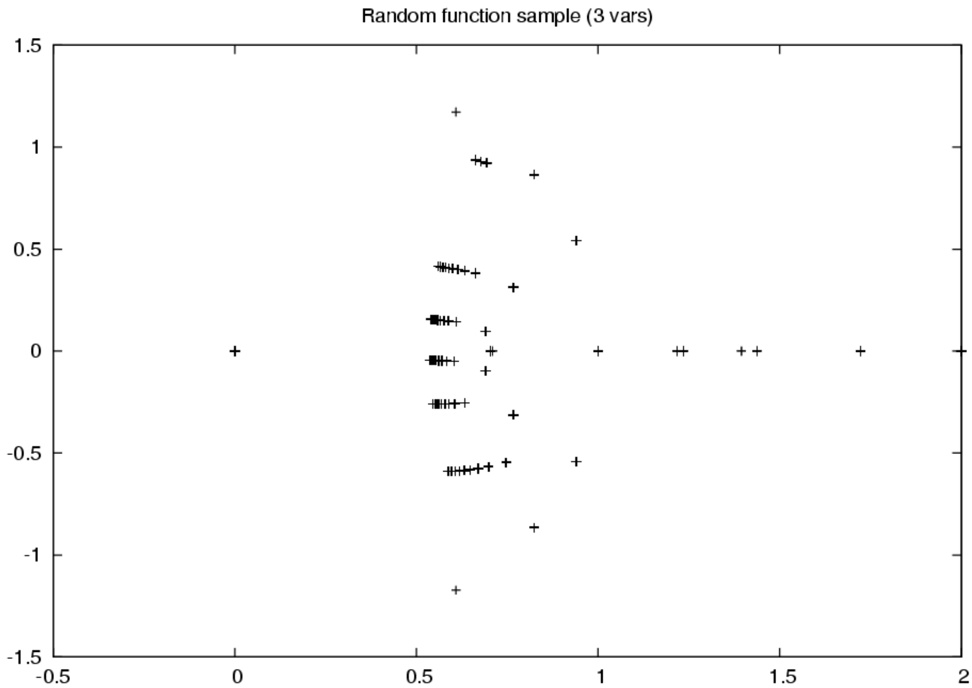


Figure L.1: Principal chromatic zeros of a random function sample on 3 variables.

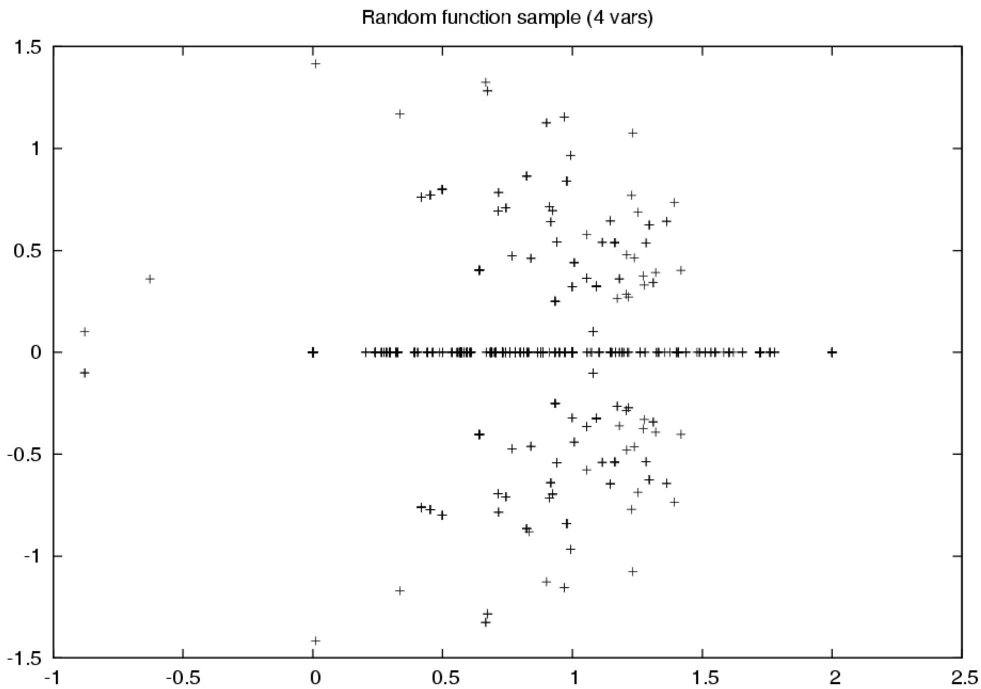


Figure L.2: Principal chromatic zeros of a random function sample on 4 variables.

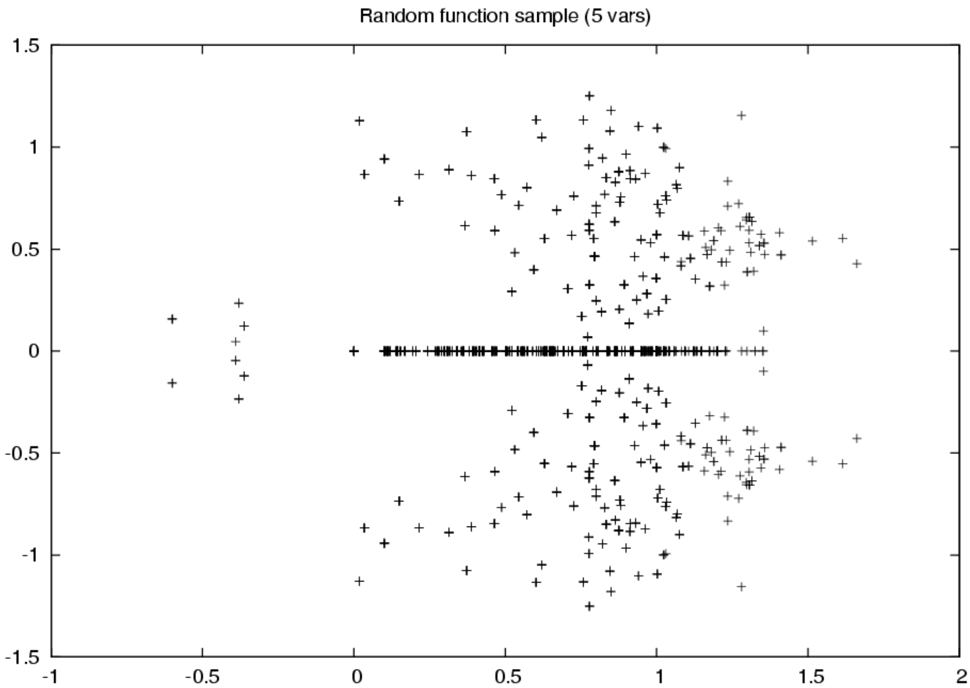


Figure L.3: Principal chromatic zeros of a random function sample on 5 variables.

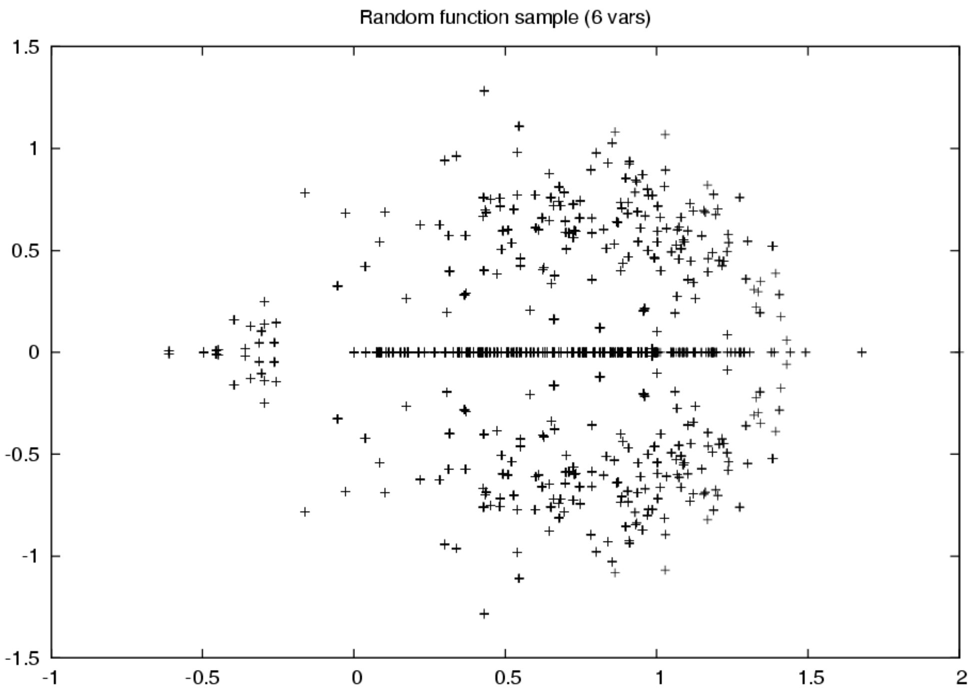


Figure L.4: Principal chromatic zeros of a random function sample on 6 variables.

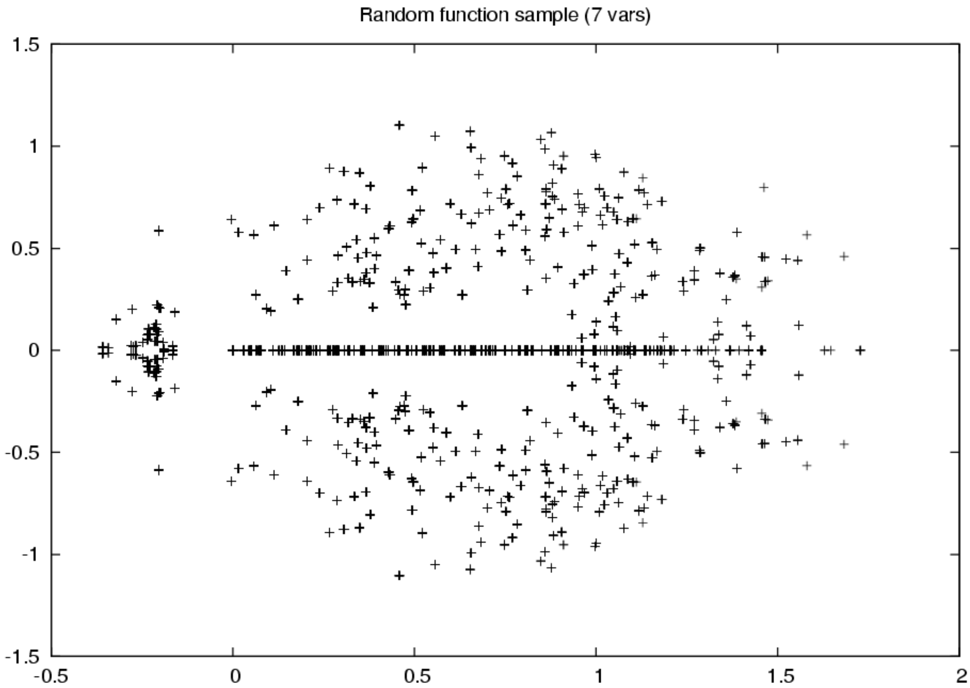


Figure L.5: Principal chromatic zeros of a random function sample on 7 variables.

Appendix M

Sample Chromatic Pseudonomials

Included here is a collection of chromatic pseudonomials for a range of Boolean functions from 2 to 4 variables.

f	$P(f;\lambda)$
1110	$1-2\lambda+\lambda^{\log 3}$
1001	$-1+\lambda$
1101	$-\lambda+\lambda^{\log 3}$
1011	$-\lambda+\lambda^{\log 3}$
1111	$1-2\lambda+\lambda^{2}$
11101000	$-1+3\lambda-3\lambda^{\log 3}+\lambda^{2}$
10011000	$1-2\lambda+\lambda^{\log 3}$
11011000	$\lambda-2\lambda^{\log 3}+\lambda^{2}$
10111000	$\lambda-2\lambda^{\log 3}+\lambda^{2}$
11111000	$-1+3\lambda-2\lambda^{\log 3}-\lambda^{2}+\lambda^{\log 5}$
10100100	$1-2\lambda+\lambda^{\log 3}$
11100100	$\lambda-2\lambda^{\log 3}+\lambda^{2}$
10010100	$1-2\lambda+\lambda^{\log 3}$
11010100	$\lambda-2\lambda^{\log 3}+\lambda^{2}$
10110100	$1-\lambda-\lambda^{\log 3}+\lambda^{2}$
11110100	$\lambda-\lambda^{\log 3}-\lambda^{2}+\lambda^{\log 5}$
10101100	$\lambda-2\lambda^{\log 3}+\lambda^{2}$
11101100	$-1+3\lambda-2\lambda^{\log 3}-\lambda^{2}+\lambda^{\log 5}$
10011100	$1-\lambda-\lambda^{\log 3}+\lambda^{2}$
11011100	$\lambda-\lambda^{\log 3}-\lambda^{2}+\lambda^{\log 5}$
10111100	$2\lambda-3\lambda^{\log 3}+\lambda^{\log 5}$
11111100	$-1+3\lambda-\lambda^{\log 3}-2\lambda^{2}+\lambda^{\log 6}$
11000010	$1-2\lambda+\lambda^{\log 3}$
11100010	$\lambda-2\lambda^{\log 3}+\lambda^{2}$
10010010	$1-2\lambda+\lambda^{\log 3}$
11010010	$1-\lambda-\lambda^{\log 3}+\lambda^{2}$
10110010	$\lambda-2\lambda^{\log 3}+\lambda^{2}$
11110010	$\lambda-\lambda^{\log 3}-\lambda^{2}+\lambda^{\log 5}$
11001010	$\lambda-2\lambda^{\log 3}+\lambda^{2}$
11101010	$-1+3\lambda-2\lambda^{\log 3}-\lambda^{2}+\lambda^{\log 5}$
10011010	$1-\lambda-\lambda^{\log 3}+\lambda^{2}$
11011010	$2\lambda-3\lambda^{\log 3}+\lambda^{\log 5}$
10111010	$\lambda-\lambda^{\log 3}-\lambda^{2}+\lambda^{\log 5}$
11111010	$-1+3\lambda-\lambda^{\log 3}-2\lambda^{2}+\lambda^{\log 6}$
10000110	$1-2\lambda+\lambda^{\log 3}$
11000110	$1-\lambda-\lambda^{\log 3}+\lambda^{2}$
10100110	$1-\lambda-\lambda^{\log 3}+\lambda^{2}$
11100110	$2\lambda-3\lambda^{\log 3}+\lambda^{\log 5}$
10010110	$2-3\lambda+\lambda^{2}$
11010110	$1-2\lambda^{\log 3}+\lambda^{\log 5}$
10110110	$1-2\lambda^{\log 3}+\lambda^{\log 5}$
11110110	$2\lambda-2\lambda^{\log 3}-\lambda^{2}+\lambda^{\log 6}$
10001110	$\lambda-2\lambda^{\log 3}+\lambda^{2}$
11001110	$\lambda-\lambda^{\log 3}-\lambda^{2}+\lambda^{\log 5}$
10101110	$\lambda-\lambda^{\log 3}-\lambda^{2}+\lambda^{\log 5}$
11101110	$-1+3\lambda-\lambda^{\log 3}-2\lambda^{2}+\lambda^{\log 6}$

10011110	$1-2\lambda^{\log 3}+\lambda^{\log 5}$
11011110	$2\lambda-2\lambda^{\log 3}-\lambda^2+\lambda^{\log 6}$
10111110	$2\lambda-2\lambda^{\log 3}-\lambda^2+\lambda^{\log 6}$
11111110	$-1+3\lambda-3\lambda^2+\lambda^{\log 7}$
10000001	$-1+\lambda$
11000001	$-\lambda+\lambda^{\log 3}$
10100001	$-\lambda+\lambda^{\log 3}$
11100001	$-\lambda^{\log 3}+\lambda^2$
10010001	$-\lambda+\lambda^{\log 3}$
11010001	$-\lambda^{\log 3}+\lambda^2$
10110001	$-\lambda^{\log 3}+\lambda^2$
11110001	$-\lambda^2+\lambda^{\log 5}$
10001001	$-\lambda+\lambda^{\log 3}$
11001001	$-\lambda^{\log 3}+\lambda^2$
10101001	$-\lambda^{\log 3}+\lambda^2$
11101001	$-1+3\lambda-3\lambda^{\log 3}+\lambda^{\log 5}$
10011001	$1-2\lambda+\lambda^2$
11011001	$\lambda-2\lambda^{\log 3}+\lambda^{\log 5}$
10111001	$\lambda-2\lambda^{\log 3}+\lambda^{\log 5}$
11111001	$-1+3\lambda-2\lambda^{\log 3}-\lambda^2+\lambda^{\log 6}$
10000101	$-\lambda+\lambda^{\log 3}$
11000101	$-\lambda^{\log 3}+\lambda^2$
10100101	$1-2\lambda+\lambda^2$
11100101	$\lambda-2\lambda^{\log 3}+\lambda^{\log 5}$
10010101	$1-2\lambda+\lambda^2$
11010101	$\lambda-2\lambda^{\log 3}+\lambda^{\log 5}$
10110101	$1-\lambda-\lambda^{\log 3}+\lambda^{\log 5}$
11110101	$\lambda-\lambda^{\log 3}-\lambda^2+\lambda^{\log 6}$
10001101	$-\lambda^{\log 3}+\lambda^2$
11001101	$-\lambda^2+\lambda^{\log 5}$
10101101	$\lambda-2\lambda^{\log 3}+\lambda^{\log 5}$
11101101	$-1+3\lambda-2\lambda^{\log 3}-\lambda^2+\lambda^{\log 6}$
10011101	$1-\lambda-\lambda^{\log 3}+\lambda^{\log 5}$
11011101	$\lambda-\lambda^{\log 3}-\lambda^2+\lambda^{\log 6}$
10111101	$2\lambda-3\lambda^{\log 3}+\lambda^{\log 6}$
11111101	$-1+3\lambda-\lambda^{\log 3}-2\lambda^2+\lambda^{\log 7}$
10000011	$-\lambda+\lambda^{\log 3}$
11000011	$1-2\lambda+\lambda^2$
10100011	$-\lambda^{\log 3}+\lambda^2$
11100011	$\lambda-2\lambda^{\log 3}+\lambda^{\log 5}$