Experimental Verification of the Random Character of Algebraic Irrationals

TOPPS · Semantics-based Program Analysis and Manipulation

Friday 12th June, 2009

Johan Sejr Brinch Nielsen

Email:	zerrez@diku.dk
Cpr.:	260886-2547
Supervisor:	Jakob G. Simonsen

Dept. of Computer Science,
University of Copenhagen

Abstract

I investigate the normality of algebraic irrational numbers, specifically by analyzing their χ^2, $\nabla^2\chi^2$ and Kolmogorov-Smirnov distributions.

I analyze 39 Pisot numbers, 47 Salem numbers and the roots of 15 randomly generated polynomials with integer coefficients, all computed to 5 bases with a precision of $2 \cdot 3^{18} \approx 2^{29.5}$ bits (about 233 million decimals).

I show that the frequencies of the digit sequences tested are equidistributed with a confidence value of 5%. Specifically, digit sequences of size 10 in bases 2 and 3, size 7 in base 5, size 6 in base 7 and size 5 in base 10 are equally frequent. This result is true for all 101 numbers. I show that no pattern can be found in the statistical anomalies and that only few anomalies survive a prefix size of $2 \cdot 3^{18} \approx 2^{29.5}$.

The results lend credence to Borel's conjecture that all algebraic irrationals are normal in all bases.

Furthermore, popular root finding algorithms are discussed and their performance when computing roots to arbitrary precision compared. Specifically, I give a comparison of Newton's, Laguerre's and Gupta-Mittal's methods using MPSolve as reference. The results show that Newton's and Laguerre's methods are much faster than the competition. Gupta-Mittal is shown to reduce to the classical Power Method, faster than MPsolve, but much slower than Newton's and Laguerre's methods. Gupta-Mittal appears to be unsuited, while Laguerre's method shows great performance.

The full paper, including all tables, is available at (550 pages):
`http://project-roots.s3.amazonaws.com/brinchj-2009_roots.pdf`

The source code produced during the project is available at:
`http://project-roots.s3.amazonaws.com/source-code.zip`

CONTENTS

Contents **iii**

0.1 Introduction

In 1909, Émile Borel introduced the concept of normal numbers as any number whose decimal expansion is uniformly distributed in any base [18]. In other words, the chance of seeing any sequence of digits (e.g. "123") is just as likely as seeing some other equally lengthed sequence (e.g. "321").

A simple example of a number, that is normal in base 10, is the Champernowne's constant, which is constructed from joining the natural numbers in order [16]:

$$C_{10} = 0.123456789101112131415\ldots$$

At first, this number may not seem normal, but consider its decimal expansion as it continues infinitely. It is clear that no finite decimal expansion can be normal. Nor can any rational number $\frac{p}{q}$ be normal, since its decimal expansion is periodic.

Borel believed normal numbers could be found in the irrational roots of polynomials. Specifically, he conjectured the irrational roots of polynomials with rational coefficients to be normal. Such roots are referred to as algebraic irrationals.

Algebraic irrationals include well-known constants, such as $\sqrt{2}$ and ϕ (the golden ratio), since the positive root of the polynomial $x^2 - 2$ is:

$$1.41421356237309504 88\ldots = \sqrt{2}$$

while the root of the polynomial $x^2 - x - 1$ is:

$$1.22474487139158904 91\ldots = \phi$$

Even though normality requires infinite digits, it is possible to substantiate the normality of the root by statistical examination digits in a finite prefix of the root (e.g. the first million digits) [39; 19; 10]. However, this is only possible for small block sizes which are expected to be uniformly distributed in this finite prefix. For example, counting sequences of the same length as the available prefix of the root would yield a distribution with just one possibility present (e.g. the prefix of size 5 of C_{10} ("0.12345") contains just one size 5 block ("12345")).

Borel's conjecture has been tested on several occasions using statistical methods [39; 19; 10], however the statistical results have been based on a small amount of numbers and only in a few bases.

In this paper, I test the conjecture against 100 algebraic irrationals, in 5 bases, using 3 common statistical methods. The roots are computed using new implementations of old well-known methods.

More specifically, I compute the roots of the first 39 Pisot numbers, the first 47 Salem numbers and 15 random roots to a precision of $2 \cdot 3^{18} \approx 2^{29.5}$.

0.2 Contributions

I reimplement the algorithm proposed in [21] and implemented in [36]. I show that this new implementation scales significantly better than the previous one.

I show why the method from [21] is in fact just an abstraction of the well-known Power iterations.

I give a GMP-implementation of floating point division of integers using Picarte iterations and show that this implementation scales better than the standard GMP division.

I move on to implement Newton's and Laguerre's methods and show that they both scale even better; specially on polynomials of high degree.

Using the implementation of Laguerre's Method, I compute 39 Pisot numbers, 47 Salem numbers and 15 randomly generated numbers to a precision of $2^{29.5}$ bits.

I analyze these roots using 3 common statistical methods and present the results in form of tables.

I show that the digit frequencies of all 101 numbers resembles an equidistribution with a confidence value of 5% in all 5 bases and that the statistical results do not vary visibly across Pisot, Salem and randomly generated numbers. The statistical results in general resembles that of normal normals.

0.3 Related Work

[34, Pathria, '62] computed and studied the randomness of the first 10,000 digits of π. [39, Stoneham, '65] computed and analyzed 60,000 digits of e. [19, Good and Gover, '68] investigated the binary expansion of $\sqrt{2}$ and presented the Generalized Serial Test . [13, Beyer et al., '69] examined square roots of integers from 2 to 15, in the bases 2 to 10. Each root was computed to a precision of 88,062 binary digits. [14, Beyer et al., '70] made further effort in the examination of square roots of non-square integers. [10, Bailey, '88] computed the decimal expansion of π to a precision of 29 million decimal places. [11, Bailey and Crandall, '01] investigate the random character and normality of the decimal expansion of fundamental constants. [12, Bailey and Crandall, '02] suggest the use of normal numbers in random generators. [24, Isaac, '05] investigated normality in the decimal expansions of irrational square roots in base 2. [36, Pedersen and Printzlau, '07] investigated the possibility of using Pisot numbers in random number generators. They analyzed the roots of the first 39 Pisot numbers in base 10 and compared the results to that of π. This was done with precisions lower than $10^7 \approx 2^{23.3}$.

0.4 Expectations of the Reader

The reader should posses basic knowledge in the mathematical fields of algebra and statistics. Also, basic knowledge in computer science is required. Specifically, the reader should be familiar with the terms: algorithms, database, multiprocessing, big-O notation and eigenvalues (as from taking a bachelor degree in computer science).

When using the environment set up during the project, basic knowledge on shell scripting, cronjobs and Linux systems in general is required.

Understanding the source code requires some experience in the Python programming language, as well as basic knowledge on threading, generators and other design patterns in imperative and functional programming.

Overview

In Section 1 I give a brief introduction to algebraic and irrational numbers and their relation to Pisot-Vijayaraghaven Numbers.
In Section 2 I describe three root finding methods: Newton's Method, Gupta-Mittal's proposed method and Laguerre's Method.
In Section 3 I go through the implementation process including code structure, and the technologies used (Amazon S3). I describe how the methods have been implemented and the optimizations that have been applied.
In Section 4 I introduce the statistical methods used and present the results obtained.

CHAPTER 1

MATHEMATICAL BACKGROUND

1.1 Algebraic Irrationals

In this project, I investigate the decimal expansion of Algebraic Irrationals [9]. More specifically, I test the digits in the decimal expansion for signs of abnormal behavior.

Irrational numbers

An irrational number is any number r that is not rational, hence:

$$r \neq \frac{a}{b}$$

where $a \in \mathbb{Z}$ and $b \in \mathbb{Z} \backslash 0$.
Such numbers include $\sqrt{2}$, π, e and ϕ.
An interesting property of irrational numbers is that their decimal expansion is never repeated. This is easily seen, since if it were, the number could be written as:

$$\frac{d_n d_{n-1} \ldots d_2 d_1}{99 \ldots 99} \in \mathbb{Q}$$

where d_i is the ith digit in the repeating part of the expansion.
For example, the number $0.123123123\ldots$ can be constructed as the fraction:

$$\frac{123}{999} = 0.123123123\ldots$$

Algebraic numbers

An algebraic number is a number that can be found as a root in a polynomial with rational coefficients. If α is an algebraic number, some polynomial P exists for which $P(\alpha) = 0$. Since the coefficients of P are rational, P must take the following form:

$$P(x) = c_n x^n + c_{n-1} x^{n-1} + \cdots + c_0 x^0$$

where $c_i \in \mathbb{Q}$.
Some prefer to define algebraic numbers from polynomials with integer coefficient ($c_i \in \mathbb{N}$). However, this does not change anything, since conversion from one form to the other is possible by:

$$\frac{c_n}{d}\alpha^n + \frac{c_{n-1}}{d}\alpha^{n-1} + \cdots + \frac{c_0}{d}\alpha^0 = 0 \Leftrightarrow$$

$$c_n \alpha^n + c_{n-1}\alpha^{n-1} + \cdots + c_0 \alpha^0 = 0 \cdot d = 0$$

This shows that a root of a polynomial with rational coefficient can be converted to one with integer coefficients - and vice versa. Because of this conversion, the group of numbers remains unchanged with both definitions. Popular examples of algebraic numbers include 0, $\sqrt{2}$ and ϕ. This is easily seen since:

$$P(x) = x \Rightarrow P(0) = 0$$

$$P(x) = x^2 - 2 \Rightarrow P(\sqrt{(2)}) = 0$$

$$P(x) = x^2 - x - 1 \Rightarrow P(\phi) = P(\frac{\sqrt{(5)}+1}{2}) = 0$$

Numbers that are not algebraic are referred to as transcendental numbers. Such numbers include the mathematical constants π and e.
As of this writing, it remains an open question whether π^π, π^e, e^e and the Euler-Mascheroni constant γ are algebraic or transcendental.

Minimal Polynomial

Each algebraic number α has a corresponding minimal polynomial P that fulfills the following three rules [38]:

1. The coefficient of the highest term is 1 (P is monic)
2. α is a root of P ($P(\alpha) = 0$)
3. P is the simplest (lowest degree) polynomial that fulfills 1) and 2) (hence the name "minimal")

It is clear that all algebraic numbers must have a minimal polynomial. Let α be an algebraic number. Then P exists, such that $P(\alpha) = 0$:

$$P(\alpha) = \frac{c_n}{d}\alpha^n + \frac{c_{n-1}}{d}\alpha^{n-1} + \cdots + \frac{c_0}{d}\alpha^0 = 0 \Rightarrow$$

$$P(\alpha) = \alpha^n + \frac{c_{n-1}}{c_n}\alpha^{n-1} + \cdots + \frac{c_0}{c_n}\alpha^0 = 0$$

P fulfills both rule 1) and 2). This shows that at least one of such polynomials exists. One of such polynomials must have the smallest degree. Also the minimal polynomial is unique.

Theorem 1. *The minimal polynomial of α is unique*

Proof. Let $p(x)$ and $q(x)$ be two minimal polynomials of α (of same degree). Then

$$p(\alpha) = q(\alpha) = 0$$

Consider now $r(x)$ defined by:

$$r(x) = \frac{1}{c}(p(x) - q(x))$$

where c is the leading coefficient of $p(x) - q(x)$ (to turn r monic). Clearly, both c and r are non-zero, since $p(x) \neq q(x)$ and p and q are of same degree.

Now, $r(\alpha) = 0$ and the degree of r is lower than the degree of p and q (since p and q are monic).

This yields two cases for r:

(i) r is reducible

(ii) r is irreducible

(ii) is trivial, since r is a *smaller* minimal polynomial; leading to a contradiction. In (i) some irreducible factor of r must have α as root, rendering this factor a *smaller* minimal polynomial. □

Normal numbers

A normal number n is any real number whose digits are uniformly distributed in any base. Hence, the probability of any digit, digram, trigram and so on must be constant of all combinations thereof. For example, the probability of seeing the sequence "123" in base 10 must be the same as seeing the sequence "432" or "999".

The Champernowne constant is an example of a number that is normal in one base. It is generated artificially from concatenating digits in some base b. As an example, I list the Champernowne constant in bases 2 and 10:

$C_2 = 0.11011100101110111 10\ldots$

$C_{10} = 0.1234567891011121314\ldots$

However, this number is only guaranteed to be normal in the specified base (here 2 and 10).

In this project I will investigate whether numbers that are both algebraic and irrational are normal, disregarding the chosen base. This property has been conjectured by Émile Borel but has never been proved nor rejected.

1.2 Sources of Algebraic Irrationals

In order to test whether algebraic irrationals are normal, I need numbers that are both algebraic and irrational. I focus on Pisot-Vijayaraghavan numbers and Salem numbers, since these classes of numbers have been subject to previous research in this area [36]. Furthermore I add 15 randomly generated polynomials.

Pisot-Vijayaraghavan Numbers

A Pisot-Vijayaraghavan number [17] (herein after called Pisot numbers) is any real algebraic number $a > 1$ in which the roots of its minimal polynomial (other than a) are strictly less than 1 in magnitude. That is, only one root can have a longer distance to 0 than 1.

The magnitude (or distance from 0) of a complex number $C = a + bi$ is defined as $\sqrt{a^2 + b^2}$, which simplifies to $\mid a \mid$ for real numbers.

It is clear that every $n \in \mathbb{N}$ is a Pisot number, since the minimal polynomial of such an n would be $x - n$, a first degree polynomial with just one root. However, in this project I need irrational roots, not natural.
An example of an irrational Pisot number is $\phi = \frac{\sqrt{5}+1}{2}$ (the golden ratio). ϕ's minimal polynomial is $x^2 - x - 1$, which root of magnitude larger than 1 is $\phi \approx 1.618$ (the second root is $\frac{1}{\phi} \approx 0.618$).

It has been shown that all roots of polynomials of the form $x^n(x^2 - x - 1) + 1$ and $x^n(x^2 - x - 1) + (x^2 - 1)$ are Pisot numbers. In this project I use the first 39 Pisot numbers published in [36].
The property that only one real root of a Pisot number lies outside the unit circle simplifies the root finding methods, by guaranteeing one unique root, with the largest absolute value. This eliminates the possibility of multiple roots (e.g. 0 in x^2) in the searched interval (outside $[-1:1]$). Multiple roots can slow down the convergence of root finding methods that rely on the first or second derivative function.

Salem Numbers

Recall that a Pisot Number was any real algebraic number, $a > 1$, for which the other roots of the minimal polynomial are strictly less than 1 (< 1).

A Salem number is defined slightly differently. The other roots of the minimal polynomial needs to be less than or equal to 1 (≤ 1) and at least one of these has to be equal to 1 (Pisot numbers are not also Salem numbers).

In this project, I use the first 47 Salem numbers published in [33].

Salem numbers have the same convenient properties with respect to root finding as the Pisot numbers.

Randomly Generated Polynomials

In order to test Borel's conjecture on numbers not discussed in other research papers, I generate 15 randomly chosen polynomials with random roots. Well, the polynomials are not actually chosen at random, however the coefficients are chosen in a way that renders them "randomish".

The random polynomials are on the following form:

$$p(x) = x^n + [\sum_{i=1}^{n-1} c_i x^i] - 1$$

Where c_i is some positive integer coefficient. The polynomial is monic, since the leading coefficient is zero. It has at least one rational root, since $p(0) = -1$ whereas $p(1) > 0$. Since both the leading and the ending coefficient are 1, only 1 and -1 can be rational roots. 1 is excluded since $p(1) > 0$. This leaves -1 as a potential rational roots. I have tested the 15 generated polynomials and verified that 1 does not occur as a root in any of them. However, this may be the case for other values of n.

I define the nth ($1 \leq n \leq 15$) random polynomial in the following way:

- it is a polynomial of $n + 16$ degree (has $n + 16 - 1$ variables).

- its coefficients are selected by seeding Python's `random` module with n and hereafter using its `randint` method to produce "random" coefficients in the interval $[1; 2^{31} - 1]$. Coefficients are generated in a bottom-up manner, starting with c_1 and ending with c_{n-1}).

Seeding the Python's random module with n ensures that the polynomials can be easily regenerated (and the results reproduced). At the same time, it guarantees that the polynomials have not been *conveniently* chosen (e.g. to produce wanted statistical results or perform well in the algorithms).

CHAPTER 2

ROOT FINDING ALGORITHMS

2.1 Introduction

As described in Section 1 algebraic irrationals are roots of polynomials, and can thereby be computed using root finding algorithms.

Initially the project was beriched by two Intel Dual Core2 servers, with 2 and 8 GB ram respectively. The project was later expanded with an Intel Quadro Core server, also with 8 GB ram (yielding a total of 8 cores).

An upper bound on the precision of the roots that can be computed is 2^{36} bits (2^{33} bytes or 8 gigabytes). This upper bound is based solely on how long a root the computer memory can maintain, however it may be constrained by both computation time and memory usage requirements of the algorithms being used.

In the following sections, I review 3 root finding algorithms, and discuss their strengths and weaknesses, with respect to the setting described above. All the algorithms are based on iterative approximation of the sought root.

Specifically I will compare them on the terms of

Time: The perfect root finding algorithm is fast on both small and large polynomials.

Memory: The servers used in this project have maximum 8 GB memory available.

Ideally, the algorithm used will have a low memory footprint, allowing several computations to run simultaneously on the multicored servers. This would allow parallel computation of the roots which could yield a severe performance boost.

2.2 MPsolve

MPsolve is an open source MultiPrecision Polynomial Solver, developed by Dario Bini and Giuseppe Fiorentino. It uses GMPlib as backend, and is faster than both Mathematica's NSolve, MAPLE's fsolve and PARI's rootpol [15].

In this project, it is used both as a comparison, with regards to speed, and as a reference with regard to correctness.

2.3 Newton's Method

Newton's method [30] is the classical method for approximation of roots in a differential valued function (such as a polynomial). It works iteratively, by adjusting its position by a direction vector and thereby moving closer towards the root in each iteration.

The direction vector is defined by $\frac{p(x_0)}{p'(x_0)}$, where p is the polynomial function and p' its first derivative. The iteration then becomes:

$$x_i - \frac{p(x_{i-1})}{p'(x_{i-1})}$$

This method can be understood as simply *following* the direction of the function, however it has one simple problem: It may converge to a local minimum instead of the wanted zero. To avoid this problem, it is necessary to provide an initial x_0, that is *reasonably* close to the wanted zero.

Given such a *reasonable* initial x_0, Newton's method will converge quadratically. Each iteration contains a single division, rendering this method very fast.

2.4 Gupta-Mittal

[21; 20; 22; 23] describe an algorithm for computing the roots of monic polynomials using only symbolic manipulation. It exploits that the root can be approximated by the ratio of two distinct letters in a carefully expanded sequence.

In particular, an alphabet A of $2m$ letters is used, where m is the degree of the polynomial. The letters in A have no particular semantics, except that each letter $l \in A$ has a negative counterpart $\sim l \in A$.

An initial word w_0 is chosen as a single letter:

$$w_0 := \{l_0\}$$

This word is now repeatedly expanded by replacing each letter with a new word chosen by a carefully constructed replacement function R:

$$w_{i+1} := R(w_i)$$

When the needed precision is reached the root can be computed as the following ratio:

$$\frac{|l_0| - |\sim l_0|}{|l_1| - |\sim l_1|}$$

where $|\ l\ |$ is the number of occurrences of letter l. In other words, the root is approximated by the ratio between the counts of two distinct letters (which letters are chosen does not matter).

The Replacement Function

In order to define the replacement function R, a replacement matrix is defined on the polynomial coefficients, $a_0, a_1, \ldots, a_m$:

$$M_s = \begin{pmatrix} -a_1 & -a_2 & -a_3 & \dots & -a_{m-1} & -a_m \\ a_0 & 0 & 0 & \dots & 0 & 0 \\ 0 & a_0 & 0 & \dots & 0 & 0 \\ 0 & 0 & a_0 & \dots & 0 & 0 \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & a_0 & 0 \end{pmatrix}$$

M_s has m columns, however the replication function will need one column for each letter ($2m$ in total). To meet this demand, M_s is extended into M:

$$M = \begin{pmatrix} M_s & 0 \\ 0 & M_s \end{pmatrix}$$

The replacement function is now defined as a function that repeats each $l_i \in A$, M_{ij} times:

$$R(l_j) := l_1 \times M_{1j} \mid\mid l_2 \times M_{2j} \mid\mid l_3 \times M_{3j} \dots l_m \times M_{mj}$$

where $||$ is the concatenation operator and repeating a letter $-k$ times $(l \times (-k))$ is defined as $(\sim l) \times k$. Since each column only contains two non-zero elements, this rule reduces to:

$$R(l_j) := l_1 \times M_{1j} \mid\mid l_j \times a_0$$

R will expand letter l_j to a word of size:

$$\mid R(l_j) \mid = \sum_{i=1}^{m} abs(M_{ij})$$

This size will be at least 2, if all coefficients are not 0. In this case the space usage will be at least $O(2^n)$, where n is the number of replacements performed.

Numeric Computations

In order to avoid the housekeeping of building and maintaining extremely long words, we simply keep track of the letter counts. A word can then be represented by a vector of m letter counts:

$$V = (|A_0| - | \sim A_0|, \dots, |A_m| - | \sim A_m|)$$

A_0 is a special case in the replacement rules, since it is the only letter that appears in each rule. When replacing each letter in the word W_n, the number of A_1's will equal:

$$|A_1|' = |A_1| + |A_1| \sum_{i=1}^{m} a_i$$

The rest of the letters only appear in two rules each, their own replacement rule and the previous one. The count of each $A_i \neq A_1$ becomes:

$$|A_i|' = |A_{i-1}| + |A_i|$$

This yields the following vector in iteration i:

$$R(W) = \begin{pmatrix} |A_1| + a_i \sum_{i=1}^{m} |A_i| \\ |A_1| + |A_2| \\ |A_2| + |A_3| \\ \dots \\ |A_{m-1}| + |A_m| \end{pmatrix}$$

$$= \begin{pmatrix} (a_1+1)|A_1| & +a_1|A_2| & \dots & +a_1|A_{i-1}| & +a_1|A_i| & \dots & +a_1|A_{m-1}| & +a_1|A_m| \\ . & . & . & . & . & . & . & . \\ 0 & +0 & \dots & +|A_{i-1}| & +|A_i| & \dots & +0 & +0 \\ . & . & . & . & . & . & . & . \\ 0 & +0 & \dots & +0 & +0 & \dots & +|A_{m-1}| & +|A_m| \end{pmatrix}$$

The iterations can now be summed to:

$$R \cdot \Big(R \cdot (R \cdot \ldots \cdot (R \cdot W))\Big) = \Big(R \cdot R \cdot R \cdot \ldots \cdot R\Big) \cdot W = R^n \cdot W$$

Gupta-Mittal has two serious performance issues. The matrix, which is computed to a new power in each iteration, contains $O(n^2)$ elements. Hence, both memory and time usage depend on the degree of the polynomial.

Moreover, the iteration of $R^n \cdot V$ is identical to that of the Power Iteration [28], which is known to converge with a ratio of $\frac{\lambda_1}{\lambda_2}$. Hence, the time usage also depends on coefficients of the polynomial (since they imply λ_1 and λ_2).

2.5 Laguerre's Method

Laguerre's method [8; 30] is an algorithm for finding the roots of polynomials in cubic time. It is based on the assumption that all other roots are located some distance δ away from the one wanted.

The algorithm locates the root of the polynomial p as follows:
Since p is a polynomial, its can be described by it roots r_i:

$$p(x) = c(x - r_0)(x - r_1)(x - r_2)\dots(x - r_n)$$

Consider now the functions G and H defined as:

$$G := \frac{d}{dx} \ln |\, p(x) \,| = \frac{1}{x - r_0} + \frac{1}{x - r_1} + \cdots + \frac{1}{x - r_n}$$

$$H := -\frac{d^2}{dx^2} \ln |\, p(x) \,| = \frac{1}{(x - r_0)^2} + \frac{1}{(x - r_1)^2} + \cdots + \frac{1}{(x - r_n)^2}$$

Since all uninteresting roots are at least δ away from r_0, we can approximate these terms by introducing $b = x - r_i$ ($i > 0$) [8]:

$$G := \frac{d}{dx} \ln \mid p(x) \mid = \frac{1}{x - r_0} + \frac{1}{b} + \cdots + \frac{1}{b} = \frac{1}{x - r_0} + \frac{n-1}{b}$$
$$= \frac{p'(x)}{p(x)}$$
$$H := -\frac{d^2}{dx^2} \ln \mid p(x) \mid = \frac{1}{(x - r_0)^2} + \frac{1}{b^2} + \cdots + \frac{1}{b^2} = \frac{1}{(x - r_0)^2} + \frac{n-1}{b^2}$$
$$= G^2 - \frac{p''(x)}{p(x)}$$

These two functions yield an approximation of the distance to r_0. Solving them with respect to $x - r_0$ yields [8]:

$$a = x - r_0 = \frac{n}{G \pm \sqrt{(n-1)(nH - G^2)}}$$

where a is the factor used to revise the current approximation of the root r_0. Specifically, the next approximation x_i is gained by $x_i = x_{i-1} - a$.
Laguerre's Method converges in $O(3^n)$, tripling the number of correct digits in each iteration [8]. Hence, the precision of the approximation in iteration i is 3 times higher than that of iteration $i - 1$. This convergence is independent of the degree of the polynomial.

CHAPTER 3

IMPLEMENTATION

3.1 Introduction

In the following sections, I give a short introduction to the software libraries and general optimizations used in the implementation. In Section 3.2, I give a quick overview of the computation process followed by a description of how the roots are stored securely on the Amazon S3 file storage service in Section 3.4. In Section 3.3, I describe how the correctness of the implemented algorithms are verified. In the preceding sections (3.5, 3.6 and 3.7), I describe the implementation details of each algorithm discussed in Section 2. In Section 3.9, I compare the performance of the implementations.

Software Packages

The following software has been used in computation of roots and statistics (the versions used are listed in the bibliography):

Python: All the implementations are written the general purpose programming language Python. The language is known for its high-level concise syntax as well as its large collection of scientific libraries [5].

PostgreSQL: An object-relational database management system, known to scale very well with large amounts of data [4].

psycopg2: Thread-safe PostgreSQL adaptor for Python, with full support for Python's DBAPI 2.0 and all PostgreSQL and Python data types.

GMPLib: The Gnu MultiPrecision Library is known world-wide for its ability to handle arbitrary precision arithmetics efficiently. This includes support for both integers, rationals and decimals [1].

GMPy: Python wrapper for GMPLib [2].

NumPy: Fundamental scientific computing package for Python. Includes functions for matrix and vector manipulation [3].

SciPy: Scientific computing package for Python. Includes statical functions for computation of various distributions [6].

SymPy: Python library for symbolic mathematics. Includes functions for polynomial representation and manipulation [7].

General Optimizations

In order to speed up GMPLib even further, the Intel Core2 assembler patches from [29] have been applied; resulting in a general performance boost of $30 - 40\%$. The correctness of this patchset has been verified using the tests included in the GMP package, and that of the GMPy Python wrapper.

The important algorithms from GMPLib includes:

Multiplication through Fermat style fast Fourier transformation (FFT) proposed by Schonhage and Strassen. This yields multiplication in $O(n \ln(n) \ln(\ln(n)))$ time.

Division using divide and conquer in $O(M(n) \ln(n))$ time, where $M(n)$ is the time used to multiply two n sized numbers.

Square Root computed using Karatsuba, originally proposed by Paul Zimmerman. It runs in $O(M(n/2))$.

Base Conversion in sub-quadratic time using a divide and conquer approach.

Much time was consumed trying to optimize the original C implementation of the Gupta-Mittal algorithm. The source code was poorly documented, and the description in [17] uses the same approach as the original Gupta-Mittal paper (no comparison to Power iterations). It took several weeks to discover that the bottleneck was located deep inside the PARI/gp system, and that the performance boost wanted would require a total rewrite. After porting the original version to the GMPLib backend and implementing Picarte iterations [31] in C, I had gained the insight needed to reimplement the algorithm in a small Python version; without losing any performance. This reimplementation reduced the number of code lines by 90% (628 lines to 58 lines; including verification).

Both Newton's and Laguerre's methods took less than a day to implement in optimized versions.

3.2 Pipeline

The framework used to compute the roots takes care of uploading them to Amazon S3, in several bases, for later use. Then the statistical methods are applied and the results stores in a PostgreSQL database.

The following steps describe the process of computing the statistics for a particular root (e.g. $x^3 - x - 1$):

1. Compute the root to the desired precision;
   Store the result on Amazon S3.

2. Convert the root to the chosen bases;
   Store the result on Amazon S3.

3. Count digit occurrences;
   Store the result in the local database.

4. Apply statistical methods on the counts;
   Present the result (as tables or graphs).

3.3 Verification

To ensure correct computations each implementation has been verified to produce correct digits. More specifically, each implemented method is verified in the following way:

1. They correctly compute $\sqrt{2}$ to 2^{20} digits
2. They correctly compute the first Pisot number to 2^{20} bits
3. They correctly compute the first 39 Pisot numbers to 2^{15} bits
4. A runtime check verifies the first 2^{15} bits of any computed root.

Verification is done using MPsolve, which is believed to work correctly. Failure of verification of the first 2^{15} bits will result in an irrevocable error in the application.

3.4 Amazon S3 Storage

I use Amazon's cloud file system S3 for storing the computed roots. In order to ensure the integrity of the uploaded files, such uploads are handled in the following manner:

1. Compute a fingerprint (SHA-1) of the original file content
2. Compress the original file (Gzip) and split it into chunks of 16 MBs
3. Compute a fingerprint (SHA-1) of each chunk
4. Upload chunks and fingerprints to S3

Downloading data from S3 is done in reverse; verifying all fingerprints to ensure file integrity. Restoring is done in a temporary location, allowing thread safe operations. Restoring is implemented to be atomic (one final move operation restores the data).

All uploads are verified by attempting to download the data after upload completion. If the downloaded data fails the integrity checks (one or more fingerprints fail) the upload action fails. This was necessary to implement, since S3 is not a stable service (it may occasionally error e.g. by losing the connection).

Storing Roots on S3

In order to store roots on the Amazon S3 storage, it is necessary to convert these into binary data. Oddly, the built-in method of GMPy, `binary()`, which produces base-256 digits is significantly slower to reload roots than the base-2 or hexadecimal digits equivalents.
Because of this oddity I convert numbers to hexadecimal digits before uploading them to S3. Due to compression, the effect on file size is insignificant.

3.5 Newton's Method

Newton's Method is the simplest of those implemented and the implementation reflects this. The performance is gained from using GMP as backend for all computations.

The only performance optimization applied is to expand the precision of the `mpf` floating point type representing the current approximation of the root on-the-fly.

Recall that Newton's method converges quadratically. In iteration i the precision is therefore adjusted to:

$$\max(4096, 4 \cdot 2^i)$$

Hence, the first 9 iterations use a precision of 4096, whereas later iterations use a precision four times the amount of expected correct bits.

3.6 Gupta-Mittal

Gupta-Mittal is by far the most complex algorithm implemented, with respect to the operations used during the computations. It uses both polynomial transformations and matrix multiplication, and is the only implemented algorithm that has a running time which depends on both the degree and the coefficients of the polynomial.

The implementation presented in this paper is inspired by [36]. However, it does have some significant optimizations, rendering it severely faster.

GMP

The old version was implemented on top of PARI/gp which uses GMPlib as backend. Since PARI/gp does not support base conversion natively, the implementation would have to output the number in base 10 for later conversion.

In order to loose the overhead of PARI/gp and unneeded base conversions, the new implementation runs directly on GMPlib.

NumPy

The NumPy package is used in the implementation to perform operations on matrices and vectors. The generic implementation of the module conveniently supports GMP integers as data type in its routines. NumPy greatly reduces the number of code-lines to maintain, by reducing operations such as the matrix `power function` to a one-line function call.

An alternate multithreaded `power function` that can exploit SMP systems, has also been implemented. This alternate implementation runs four times as fast on the quadro-core server.

Division

The floating point division in GMP 4.2.4 (or older) is slow. There are plans to implement division by multiplication of inverses (obtained through Newton iterations), however at the time of this writing, this is not part of any stable release.

To circumvent the GMP floating point division the new implementation performs division through Picarte iterations, operating solely on integers [31].

Picarte iterations approximate the multiplicative inverse, like Newton iterations, but are faster when the sought precision k is larger than the size of the number to invert n.

Let $M(a, b)$ be the time needed to multiply two numbers of size a and b. Where Newton iterations use $O(M(k, k))$ time to approximate the inverse of precision k bits, Picarte iterations do this in $O(M(n, k))$ time. The division in Gupta-Mittal needs to divide two numbers of size n with precision n. However, to obtain a division precision of n bits, it is necessary to compute the inverse to at least $4n$ bits; yielding $4n = k$, hence giving Picarte an advantage.

3.7 Laguerre's Method

Laguerre's Method (or just Laguerre) is implemented in a straight forward manner, using GMP. However, there is one interesting feature in the implementation. Recall that one of the conditions of Laguerre, is that the other roots are located some distance away from the interesting root. Because of this, Laguerre needs a *good* starting guess in order to ensure convergence.

To find such an initial guess, I use MPsolve to compute 4096 bits of the root from which I let Laguerre converge.

Since Laguerre operates on floating points, it is necessary to specify the roots precision. A naive solution would be to specify the expected resulting precision of $2 \cdot 3^N$ from the beginning, however, this would require operations on high precision floating point numbers throughout each iteration. Even in the first iteration, when the precision is no better than the initial guess.

To speed up the Laguerre iterations, I extend the precision of the floating point containing the root in each iteration. At iteration i, the precision will be extended to $\min(4096, 2 \cdot 3^i)$, which matches the expected precision of the convergence.

In the original Laguerre's Method algorithm, the following division operations are used:

G = pdx(x) / p(x)
H = G^2 − pdx2(x) / p(x)

where `pdx` and `pdx2` are the first and second derivative functions respectively and "^" denotes the power function.

Instead of dividing by $p(x)$ twice, I compute the inverse of $p(x)$ and divide by multiplication. This variation uses a single division and is slightly faster:

```
pinv = 1 / p(x)
G = pdx(x) * pinv
H = G^2 − pdx2(x) * pinv
```

where * denotes multiplication.

The algorithm may run significantly faster if the computation of the inverse was computed through Picarte iterations, as was seen with Gupta-Mittal. However, such computation is non-trivial, since it requires converting the root to an integer, hence losing track of the floating point.

3.8 Handling Multiple Roots

As of introducing randomly generated polynomials, I introduced the possibility of multiple roots. That is, a root with multiplicity more than 1. For example, the root 1 has multiplicity 2, in the polynomial:

$$p(x) = x^2 - 2x + 1$$

Since $x^2 - 2x + 1 = (x - 1)^2$ [26].

In this section I describe a simple algorithm that converts a polynomial with one or more multiple roots into one where all roots have multiplicity 1.

Consider some polynomial p with at least one multiple root:

$$p(x) = (x - r_0)(x - r_1) \dots (x - r_i)^j \dots (x - r_{n-1})(x - r_n)$$

The derivative of p also has root r_i, however with one lower multiplicity [26]:

$$p'(x) = (x - s_0)(x - s_1) \dots (x - r_i)^{j-1} \dots (x - s_{m-1})(x - s_m)$$

The idea is now to compute the greatest common divisor (GCD) between p and p'. This results in the polynomial g consisting of all common roots between p and p':

$$g(x) = (x - r_i)^{j-1}$$

Checking for multiple roots is simple, as $g(x)$ will be the unit polynomial, 1, iff there are no multiple roots. Consider now the quotient:

$$q(x) = \frac{p(x)}{g(x)} = \frac{(x - r_0) \dots (x - r_i)^j \dots (x - r_n)}{(x - r_i)^{j-1}}$$
$$= (x - r_0) \dots (x - r_i) \dots (x - r_n)$$

Obviously, $g(x)$ divides $p(x)$ evenly from being the greatest common divisor of p and p'. However, $g(x)$ was constructed as a product of the multiple roots in p (with one lower multiplicity) thus eliminating the multiple roots.

This algorithm has been implemented using `SymPy`'s division operator (for GCD) and is used on the random polynomials. However, none of them contained any multiple roots.

3.9 Performance

In this section, I compare the different methods with respect to time and memory usage (as described in Section 2). Apparently, performance comparison of arbitrary precision root finding algorithms are hard to come by. Since I was unable to find any suitable comparison (only found being [40]), I had to test the algorithms myself.

Table Explanation

Each table compares the results of the methods on two parameters:

Time: the absolute CPU time used to compute the result.

Rate: the convergence rate in bits per CPU second. When computing the Pisot numbers, the rate is relative to that of the Power Method (Table 3.1 and 3.2). However, when computing the roots of the randomly generated polynomials, the Power Method is faced out due to performance issues and Newton's method is used as reference instead (Table 3.3 and 3.4).

To improve readability the method names have been abbreviated to:

GM: Gupta-Mittal (old implementation)

MP: MPsolve

PM: Power Method (new implementation of Gupta-Mittal)

NT: Newton's Method

LM: Laguerre's Method

Each row represents the precision of size 2^N, however this may not be the actual precision computed. Since the convergence of Gupta-Mittal and the Power Method depends on the relation between lambda values, these two methods may return a much lower precision. On the other hand, because of Laguerre's cubic convergence, this method may return a much higher precision (it runs until *at least* precision 2^N is obtained). This is the reason for choosing convergence rate, as opposed to CPU timings alone.

Results

The four methods has been tested by computing the roots of the following polynomials:

Pisot-1: $x^3 - x - 1$

Pisot-10: $x^8 - x^7 - x^6 + x^2 - 1$

Random-1:

$x^{17} + 3098990841x^{16} + 1912923433x^{15} + 9045431x^{14} + 3273968024x^{13} + 1858720404x^{12} + 3589583788x^{11} + 121751485x^{10} + 403123856x^9 + 3387540998x^8 + 2798570508x^7 + 1930549423x^6 + 2127877496x^5 + 1095513124x^4 + 3280387010x^3 + 3639700185x^2 + 577090035x - 1$

Random-10:

$x^{26} + 4219432775x^{25} + 3297838299x^{24} + 571136783x^{23} + 2842608299x^{22} + 2945752650x^{21} + 2898951944x19 + 1218130971x^{18} + 1638985230x^{17} + 2590683947x^{16} + 3694363524x^{15} + 191368206x^{14} + 4280179691x^{13} + 4092317463x^{12} + 1073727551x^{11} + 1407773507x^{10} + 2236257872x^9 + 688180705x^8 + 2806643162x^7 + 3537287273x^6 + 3493188175x^5 + 885185167x^4 + 2482883232x^3 + 1842064464x^2 + 2454155457x - 1$

Conclusion

Gupta-Mittal (or Power Method) is very slow on small polynomials and even slower on large ones. Its quadratic memory usage renders it unable to compute the first Pisot number to more than 2^{26} bits precision (on the available hardware). Gupta-Mittal is 8 times slower on the Pisot-10 than on Pisot-1. And it is completely useless on the randomly generated polynomials, that combines high degree with large coefficients.

However, the new implementation is much faster than the one presented in [36]. Their performance comparison claims computation of Pisot-1 and Pisot-10 to a precision of $2^{18.3}$ bits (100,000 decimals) in 1.9 and 19.5 seconds respectively. An increase of a factor 10. The implementation presented in this project computes the same digits to a precision of 2^{19} (524,000 decimals) in just 1.1 and 4.4 seconds. An increase of a factor 4. Even though the hardware is different, it is clear that the re-implementation scales significantly better with the degree of the polynomial.

Another interesting factor is how well the implementation scales with the precision. The comparison in [36] shows an increase in time usage from computing 50,000 decimals to 100,000 decimals of a factor of 2.29 (on Pisot-10). However, doubling the precision from 524,000 decimals to 1,048,000 in the new implementation only increases the time usage by a factor 2.03 (an increase from 262,000 decimals to 524,000 yields a factor 1.96).

Since Gupta-Mittal is slow by design my choice lies between Newton's and Laguerre's methods. Both perform well on all the tested polynomials and neither has more than linear memory usage. I recommend both as general purpose root finding methods, as long as multiple roots are handled with care (Section 3.8). I choose to use Laguerre's method to compute the roots, because it converges about twice as fast as Newton's method on large polynomials.

Table 3.1: Performance Comparison on Pisot-1

$(x^3 - x - 1)$

	GM		MP		PM		NT		LM	
N	Rate	Time	Rate	Time	Rate	Time	Rate	Time	Rate	Time
15	0.01	38.68	24.00	0.02	1.00	0.48	6.00	0.08	9.61	0.06
16	0.00	154.11	10.60	0.05	1.00	0.53	8.83	0.06	11.94	0.08
17	??	??	3.87	0.16	1.00	0.62	6.20	0.10	9.86	0.17
18	??	??	1.60	0.48	1.00	0.77	5.92	0.13	6.12	0.17
19	??	??	0.91	1.16	1.00	1.05	5.00	0.21	4.95	0.43
20	??	??	0.63	2.72	1.00	1.72	4.00	0.43	4.05	0.43
21	??	??	0.46	6.54	1.00	2.99	3.22	0.93	3.03	1.50
22	??	??	0.38	15.81	1.00	6.07	2.89	2.10	2.48	5.58
23	??	??	0.35	38.46	1.00	13.27	2.76	4.81	2.69	5.62
24	??	??	??	??	1.00	30.03	2.61	11.49	2.40	21.37
25	??	??	??	??	1.00	65.80	2.45	26.86	2.04	82.59
26	??	??	??	??	1.00	142.92	2.32	61.71	2.21	82.82

Table 3.2: Performance Comparison on Pisot-10

$(x^8 - x^7 - x^6 + x^2 - 1)$

	GM		MP		PM		NT		LM	
N	Rate	Time	Rate	Time	Rate	Time	Rate	Time	Rate	Time
15	0.04	40.23	13.65	0.11	1.00	0.69	4.84	0.31	5.82	0.31
16	0.01	160.27	6.31	0.30	1.00	0.87	5.41	0.35	9.22	0.37
17	??	??	3.10	0.90	1.00	1.28	6.80	0.41	14.77	0.51
18	??	??	2.17	2.27	1.00	2.26	9.28	0.53	12.55	0.53
19	??	??	1.70	5.67	1.00	4.42	12.18	0.79	20.53	0.95
20	??	??	1.66	11.76	1.00	8.98	14.81	1.32	20.86	0.95
21	??	??	1.58	27.01	1.00	19.65	15.11	2.83	24.18	2.69
22	??	??	1.36	67.78	1.00	42.46	15.99	5.78	23.06	9.14
23	??	??	1.32	160.11	1.00	97.37	16.09	13.17	25.93	9.32
24	??	??	??	??	1.00	222.48	16.32	29.68	24.64	33.62
25	??	??	??	??	1.00	507.75	15.93	69.38	21.70	130.66
26	??	??	??	??	1.00	1143.79	16.37	152.07	24.42	130.78

Table 3.3: Performance Comparison on Random-1

$(x^{17} + 3098990841x^{16} + 1912923433x^{15} + 9045431x^{14} + 3273968024x^{13} + 1858720404x^{12} + 3589583788x^{11} + 121751485x^{10} + 403123856x^{9} + 3387540998x^{8} + 2798570508x^{7} + 1930549423x^{6} + 2127877496x^{5} + 1095513124x^{4} + 3280387010x^{3} + 3639700185x^{2} + 577090035x - 1)$

	GM		MP		PM		NT		LM	
N	Rate	Time	Rate	Time	Rate	Time	Rate	Time	Rate	Time
15	0.00	3203.07	28.28	0.39	0.00	57.80	1.00	11.03	9.53	1.39
16	??	??	10.03	1.12	??	??	1.00	11.23	13.49	1.50
17	??	??	3.46	3.34	??	??	1.00	11.56	15.47	2.02
18	??	??	1.43	8.37	??	??	1.00	11.99	8.76	1.85
19	??	??	0.66	19.72	??	??	1.00	13.05	7.35	3.60
20	??	??	0.35	44.19	??	??	1.00	15.68	4.79	3.32
21	??	??	0.23	93.45	??	??	1.00	21.38	3.51	9.25
22	??	??	0.15	226.44	??	??	1.00	34.39	2.71	28.96
23	??	??	0.12	519.60	??	??	1.00	64.69	2.61	28.30
24	??	??	??	??	??	??	1.00	130.12	2.18	102.06
25	??	??	??	??	??	??	1.00	288.47	1.87	395.05
26	??	??	??	??	??	??	1.00	608.31	1.98	394.89

Table 3.4: Performance Comparison on Random-10

$(x^{26} + 4219432775x^{25} + 3297838299x^{24} + 571136783x^{23} + 2842608299x^{22} + 2945752650x^{21} + 2898951944x19 + 1218130971x^{18} + 1638985230x^{17} + 2590683947x^{16} + 3694363524x^{15} + 191368206x^{14} + 4280179691x^{13} + 4092317463x^{12} + 1073727551x^{11} + 1407773507x^{10} + 2236257872x^{9} + 688180705x^{8} + 2806643162x^{7} + 3537287273x^{6} + 3493188175x^{5} + 885185167x^{4} + 2482883232x^{3} + 1842064464x^{2} + 2454155457x - 1)$

	GM		MP		PM		NT		LM	
N	Rate	Time	Rate	Time	Rate	Time	Rate	Time	Rate	Time
16	??	??	50.27	2.61	??	??	1.00	131.21	59.41	3.98
17	??	??	17.73	7.42	??	??	1.00	131.59	74.73	4.76
18	??	??	6.23	21.28	??	??	1.00	132.60	37.18	4.82
19	??	??	2.52	53.43	??	??	1.00	134.49	37.04	7.36
20	??	??	1.28	108.36	??	??	1.00	139.00	19.07	7.39
21	??	??	0.64	232.59	??	??	1.00	148.88	13.39	16.91
22	??	??	0.35	488.39	??	??	1.00	171.00	7.64	51.04
23	??	??	0.20	1108.69	??	??	1.00	223.67	4.85	52.63
24	??	??	??	??	??	??	1.00	351.42	3.25	185.15
25	??	??	??	??	??	??	1.00	636.94	2.29	714.63
26	??	??	??	??	??	??	1.00	1221.19	2.18	719.52

CHAPTER 4

STATISTICAL ANALYSIS

4.1 Introduction

In this section, I investigate whether the computed roots behave like normal numbers. I do this using several common statistical tests that are known to peak on non-uniform distributions.

There are several widely used methods to test statistical properties, such as the Pearson's χ^2 Test [35], the Generalized Serial Test [19], Kullback-Leibler Divergence [27] and Kolmogorov-Smirnov [32; 25]. In this paper I will focus on Pearson's χ^2 Test, the Generalized Serial Test and Kologorov-Smirnov, since these three seem to be the standard approach when dealing with goodness-of-fit ([39; 19] use the two first methods to test the normality of e and $\sqrt{2}$). I choose to analyze the numbers in bases 2, 3, 5, 7 and 10. The bases 2 and 10 is chosen due to the work of previous papers (Section 0.3).

If a number a is normal in base b and $\log(b) / \log(b')$ is a rational number, then a is normal in base b' too [37]. Hence, if a number is normal in base 2 it is normal in bases 2, 4, 8, 16 etc. When analysis numbers in base 2 and 3, bases 4, 8, 9, 16, 27 can be omitted. This leaves 5, 6, 7 as possibilities where from 5 and 7 was selected.
The statistical experiments being analyzed in this paper consist of:

Target: The root being analyzed

Events: Occurrences of digit blocks of equal size
(e.g. "123", "222", "322")

Block size: The size of blocks that are used as events
(also referred to as "runlength" or "window size")

Counts: Each event has an observed count (number of occurrences) and an expected count. The different goodness-of-fit tests use the differences in these values, to weigh the distribution.

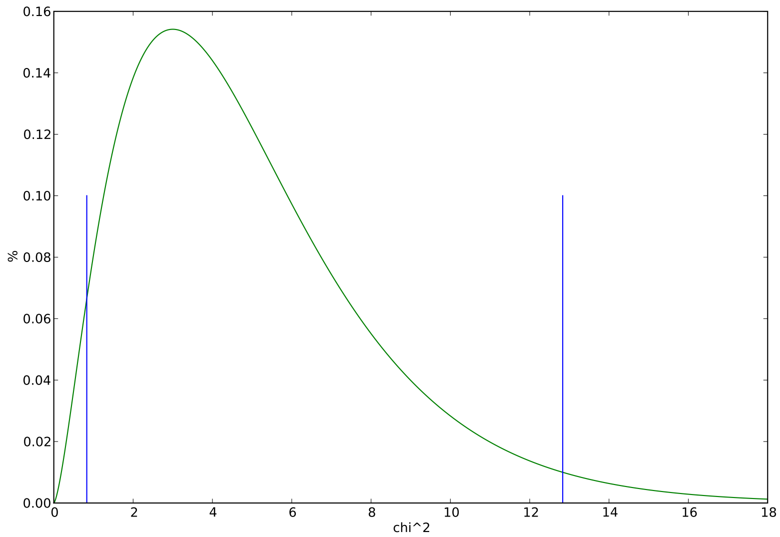
Frequencies: Frequencies are normalized counts, i.e. counts divided by the size of the target.

As an example, I list the first 4 blocks of size 5 in the first Pisot number (base 10), using a prefix of size 15 (only the first 15 digits are examined):

$$1.\underline{32471}7957244746$$
$$1.3\underline{24717}957244746$$
$$1.32\underline{47179}57244746$$
$$1.324\underline{71795}7244746$$

Each occurrence of a block is considered an event. Each event is counted

Figure 4.1: χ^2 distribution with 5 degrees of freedom



and the accumulated counts are analyzed using statistical methods.

4.2 Statistical Methods

In the following sections, I introduce the statistical methods I use to determine whether the computed digits correspond to what is expected of a normal number.

In order to compare the results to what is expected, each method has an associated "acceptance interval". The acceptance interval defines an interval, in which observed values are believed to be reasonable. Values outside the acceptance interval are believed to be extreme and unlikely. In general, this interval contains the 95% least extreme (most likely) values, hence excluding the 5% most extreme.

The graph in Figure 4.1 shows a χ^2 distribution with 5 degrees of freedom. Two vertical markers outline the acceptance interval. The x-axis indicates the χ^2 values, while the y-axis is the probability of seeing the specific x values in an equidistribution. The interval indicates the most likely 95%, hence excluding the most extreme 5% (the least and highest 2.5%).

Pearson's χ^2 test

The χ^2 test [35] computes the distance from the observed frequency of each event and the expected frequency of these:

$$\chi_s^2 = \sum_{E_o} \frac{(E_o - E_e)^2}{E_e}$$

where s is the block size, and E_o and E_e are the observed and expected mass of event e respectively.
I compute the acceptance intervals using the inverse cumulative density function (percent point function) provided by SciPy's `chi2` module. For example, the acceptance interval for the graph from the previous section is $[0.83; 12.83]$.

Generalized Serial Test

[19] describes an alternative to Pearson's χ^2 test called the Generalized Serial Test. It is defined from the ψ^2 value, which equals χ^2 when an equidistribution is expected. The Generalized Serial Test is defined as:

$$\nabla^2 \chi_s^2 = \chi_s^2 - 2\chi_{s-1}^2 + \chi_{s-2}^2$$

The expected $\nabla^2 \chi_s^2$ value is defined by the degrees of freedom, found by:

$$df(\nabla^2 \chi_s^2) = b^s - 2b^{s-1} + b^{s-2}$$

Where b is the base and s the block size.
For example, the expected $\nabla^2 \chi^2$ value in base 10 using block size 3 is:

$$10^3 - 2 \cdot 10^2 + 10^1 = 1000 - 200 + 10 = 810$$

However, the $\nabla^2 \chi^2$ distribution is asymptotically equal to that of χ^2. Hence, the acceptance interval is also asymptotically equal, and can be computed with SciPy's `chi2` module, yielding $[733.0; 890.8]$. A computation of the expectance value gives 809.33 a difference of 0.08%.

Kolmogorov-Smirnov

The Kolmogorov-Smirnov (KS) test [25] compares distances between the accumulated distribution values. If $O(e)$ defines the observed count of event e, and $E(e)$ the expected count, the KS test can be summarized to:

$$D^+ = \max_{i=1}^{|E|} \left(\frac{i}{N} - F(e_i) \right)$$

$$D^- = \max_{i=1}^{|E|} \left(F(e_i) - \frac{i-1}{N} \right)$$

where $\mid E \mid$ is the number of possible events. In short, the KS test will go through each event e comparing the probability of seeing an event lower than or equal to e with what was expected.

[25, Knuth] shows that multiplying D^+ and D^- with a factor of $\sqrt{N}$ normalizes the result in a way that renders them independent of N. These normalized variants are denoted by:

$$K^+ = \sqrt{N} \max_{i=1}^{|E|} \left(\frac{i}{N} - F(e_i) \right)$$

$$K^- = \sqrt{N} \max_{i=1}^{|E|} \left(F(e_i) - \frac{i}{N} \right)$$

In this paper, I use Knuth's variant [25]. The `kone` module of the SciPy package does support computation of the survival function (called `sf`), however an exact computation is too slow for the values of N used in this project. Instead, the following approximation (proposed by Knuth) is used:

$$sf(N,p) \approx \sqrt{\frac{1}{2} \ln(\frac{1}{1-p})} - \frac{1}{6 \cdot \sqrt{N}}$$

The approximation has an error of $O(\frac{1}{N})$; hence the approximation closer to the exact value for larger values of N. Knuth suggests that the approximation should only be used for $N > 30$. In this project, the smallest value of N is 2^{20}, much larger than 30, so it should be safe to use the approximation.

To verify that the approximation is precise enough, I have compared the values with those from SciPy. The result is presented in Table 4.2. The largest error is observed when $N = 2^{20}$ and is $\approx 0.000004\%$ (very small). In contrast, the approximation is much faster than the exact calculation.

Table 4.1: Exact and Approximated Survival Function for $p = 0.95$

Method	N	Time	sf-value
Exact	2^{20}	0.82s	1.22371060127
Approx	2^{20}	0.00s	1.22371065492
Exact	2^{21}	1.63s	1.22375830014
Approx	2^{21}	0.00s	1.22375832635
Exact	2^{22}	3.27s	1.22379202227
Approx	2^{22}	0.00s	1.22379203513
Exact	2^{23}	6.52s	1.22381586711
Approx	2^{23}	0.00s	1.22381587084
Exact	2^{24}	17.38s	1.22383272201
Approx	2^{24}	0.00s	1.22383272524

4.3 Implementation Concerns

Since the methods rely on summations of fractions, I needed to minimize the risk of accumulated rounding errors. I try to minimize this risk by:

1. Accumulating the block counts as integers (and computing the frequencies as needed).

2. Performing all internal floating point operations using 128-bit precision (using GMPLib).

By storing the block counts in the database, I avoid any errors in the permanent part of the computation. If any rounding error is discovered in later computations, it is possible to extract the precise block counts and recompute the statistics with a higher precision.

I consider the risk of a significant floating point error to be small, since 128-bit is far from the 64-bit the servers support natively, and very far from the precision I use in the resulting tables. However, it is impossible to completely avoid this problem.

4.4 Block sizes

In this section, I look at frequency deviations with different block sizes and how they affect each other. Specifically I investigate whether examining all block sizes of a particular root is necessary if the largest block size passes the statistical tests. Say that the first Pisot number passes all tests when the block size is 10. Is it then necessary to test whether this is true for block sizes 9, 8, 7, . . . , 2, 1?

Frequency Deviation Analysis of Smaller Block Sizes

Recall that the frequency of some block ("123") is the count of this block, normalized by size of the prefix (the examined part of the root). The deviation is the frequency divided by the expected frequency. In the following sections I prove that this deviation cannot grow when the block size is lowered.

Definition 1. *A "wrapped" sequence of size N is a sequence in which blocks are wrapped around the edge. As a result, the number of blocks does not vary as the blocks size changes.*

Example 1. Let the sequence $s =$ "12345" and let s be wrapped
Blocks of length 2 of s are: "12", "23", "34", "45", "51".
Blocks of length 3 of s are: "123", "234", "345", "451", "512".
"51" and "512" are examples of blocks that "wrap" around the edge of s. These blocks are only included because s is wrapped.

The following definitions allow convenient access of the most and least frequent block in a sequence:

Definition 2. *$C_s(b)$ denotes the number of occurrences of block b in sequence s.*

Definition 3. $\max_l(s)$ *denotes a block of length l in s that has the highest number of occurrences. Put precisely:*

$$\forall b \in blocks_l(s) : C_s(b) \leq C_s(\max_l(s))$$

where $blocks_l(s)$ are the l-lengthed blocks of s. Several blocks may fulfill this requirement, in which case one selected arbitrary.

Symmetrically, I define access for the least frequent block:

Definition 4. $\min_l(s)$ *denotes a block of length l in s that has the the lowest number of occurrences. Put precisely:*

$$\forall b \in blocks_l(s) : C_s(b) \geq C_s(\min_l(s))$$

where $blocks_l(s)$ are the l-lengthed blocks of s. Several blocks may fulfill the requirement, in which case one is selected arbitrary.

Example 2. The sequence $s =$"1112" has:

$$\max_2(s) = 11, C_s(\max_2(s)) = 3$$

On the other hand, both "12" and "21" occur exactly once. However, it does not matter which one of these is chosen as $\min_2(s)$:

$$\min_2(s) = 12, C_s(\min_2(s)) = 1$$

In this example, only digits 1 and 2 were used. If 0 had been allowed (but not occurred) $C_s(\min_2(s))$ would have been zero, while $\min_2(s)$ would have been some combination containing 0 (e.g. "00" or "01").

Lemma 1. The most frequent block of length $l-1$ occurs at most b times more than of length l:

$$C_s(\max_{l-1}(s)) \leq b \cdot C_s(\max_l(s))$$

Proof. Assume the opposite, namely $C_s(\max_{l-1}(s)) > b \cdot C_s(\max_l(s))$. Since the sequence s is wrapped, we can write $C_s(\max_{l-1}(s))$ as a sum of all bs blocks of length l that has $\max_{l-1}(s)$ as a prefix:

$$C_s(\max_{l-1}(s)) = \sum_{i=1}^{b} C_s(bs_i) > b \cdot C_s(\max_l(s)) \Rightarrow$$

$$\sum_{i=1}^{b} C_s(\max_l(s)) > b \cdot C_s(\max_l(s)) \Rightarrow$$

$$b \cdot C_s(\max_l(s)) > b \cdot C_s(\max_l(s))$$

which is a contradiction. □

Lemma 2. The least frequent block of length $l-1$ occurs at least b times more than that of length l:

$$C_s(\min_{l-1}(s)) \geq b \cdot C_s(\min_l(s))$$

Proof. Assume the opposite, namely $C_s(\min_{l-1}(s)) < b \cdot C_s(\min_l(s))$. Since the sequence s is wrapped, we can write $C_s(\min_{l-1}(s))$ as a sum of all bs blocks of length l that has $\min_{l-1}(s)$ as a prefix:

$$C_s(\min_{l-1}(s)) = \sum_{i=1}^{b} C_s(bs_i) < b \cdot C_s(\min_l(s)) \Rightarrow$$

$$\sum_{i=1}^{b} C_s(\min_l(s)) < b \cdot C_s(\min_l(s)) \Rightarrow$$

$$b \cdot C_s(\min_l(s)) < b \cdot C_s(\min_l(s))$$

which is a contraction. □

Theorem 2. *The highest frequency deviation with block size $r-1$ is maximum that of block size l:*

$$\frac{C_s(\max_{l-1})}{b^{-(l-1)}N} \leq \frac{C_s(\max_l)}{b^{-l}N}$$

Proof.

$$\frac{C_s(\max_{l-1}(s))}{b^{-(l-1)}N} \leq \frac{C_s(\max_l(s))}{b^{-l}N} \Rightarrow$$
$$\frac{C_s(\max_{l-1}(s))}{b^{-(l-1)}} \leq \frac{C_s(\max_l(s))}{b^{-l}} \Rightarrow$$
$$b^{-1}C_s(\max_{l-1}(s)) \leq C_s(\max_{l}(s)) \Rightarrow$$
$$C_s(\max_{l-1}(s)) \leq b \cdot C_s(\max_{l}(s))$$

which was proved in Lemma 1. □

Theorem 3. *The lowest frequency deviation with block size $l-1$ is minimum that of block size l:*

$$\frac{C_s(\min_{l-1})}{b^{-(l-1)}N} \geq \frac{C_s(\min_l)}{b^{-l}N}$$

Proof.

$$\frac{C_s(\min_{l-1}(s))}{b^{-(l-1)}N} \geq \frac{C_s(\min_l(s))}{b^{-l}N} \Rightarrow$$
$$\frac{C_s(\min_{l-1}(s))}{b^{-(l-1)}} \geq \frac{C_s(\min_l(s))}{b^{-l}} \Rightarrow$$
$$b^{-1}C_s(\min_{l-1}(s)) \geq C_s(\min_{l}(s)) \Rightarrow$$
$$C_s(\min_{l-1}(s)) \geq b \cdot C_s(\min_{l}(s))$$

which was proved in Lemma 2. □

Theorem 4. *The absolute deviation of the maximum frequency with block size $l-1$ is maximum that of l:*

$$\left| 1 - \frac{C_s(\max_{l-1})}{b^{-(l-1)}N} \right| \leq \left| 1 - \frac{C_s(\max_l)}{b^{-l}N} \right|$$

Proof. Since both fractions are at least 1, both sides are at most 0, which yields:

$$\left| 1 - \frac{C_s(\max_{l-1})}{b^{-(l-1)}N} \right| \leq \left| 1 - \frac{C_s(\max_l)}{b^{-l}N} \right| \Rightarrow$$
$$\frac{C_s(\max_{l-1})}{b^{-(l-1)}N} - 1 \leq \frac{C_s(\max_l)}{b^{-l}N} - 1 \Rightarrow$$
$$\frac{C_s(\max_{l-1})}{b^{-(l-1)}N} \leq \frac{C_s(\max_l)}{b^{-l}N}$$

which was proved in Theorem 2. □

Theorem 5. *The absolute deviation of the minimum frequency with block size $l-1$ is minimum that of l:*

$$\left| 1 - \frac{C_s(\min_{l-1})}{b^{-(l-1)}N} \right| \geq \left| 1 - \frac{C_s(\min_l)}{b^{-l}N} \right|$$

Proof. Since both fractions are at most 1, the two terms are at least 0, which yields:

$$\left| 1 - \frac{C_s(\max_{l-1})}{b^{-(l-1)}N} \right| \geq \left| 1 - \frac{C_s(\max_l)}{b^{-l}N} \right| \Rightarrow$$

$$\frac{C_s(\max_{l-1})}{b^{-(l-1)}N} - 1 \geq \frac{C_s(\max_l)}{b^{-l}N} - 1 \Rightarrow$$

$$\frac{C_s(\max_{l-1})}{b^{-(l-1)}N} \geq \frac{C_s(\max_l)}{b^{-l}N}$$

which was proved in Theorem 3. □

Theorem 4 and 5 show that lowering the block size cannot result in more extreme normalized frequencies. Since the statistical methods are based on these frequencies, it is sufficient to examine the results for the highest block size that passes the tests. Specifically, if a numbers passes the tests at block size 10 it is unnecessary to test it on block sizes $9, 8, 7, \ldots, 2, 1$.

4.5 Statistical Results

In this section, I go through the anomalies found in the statistical results. Anomalies are defined as distribution values, that lie outside the corresponding expectancy interval, when using a confidence value of 5%.

The results has been ordered in tables that list the statistics for each distribution in each base and prefix size. The statistics correspond to the highest block size that yields equidistributed digit sequences on the largest prefix. Hence, the f distribution has no anomalies on the largest prefix. If f yields anomalies with a block size of 6, only sizes 1, 2, 3, 4 and 5 are used. The maximum block size has been set to 10 for practical reasons. However, the distribution f can still have anomalies in the statistical results since different prefix sizes are used.

The following two tables list the results of Pisot-1 in base 10:

Table 4.2: Pisot-1, Base 10 - Confidence Intervals
$(x^3 - x - 1)$

Method	Min	Exp.	Max
f	0.025	1.000	1.975
χ^2	99124.378	99998.333	100877.411
$\nabla^2\chi^2$	80213.026	80999.333	81790.763
KS	0.000	0.589	1.224

Table 4.3: Pisot-1, Base 10 - Results
$(x^3 - x - 1)$

Prefix	BS	f	χ^2	$\nabla^2\chi^2$	KS
$2^{21.00}$	5	▲ **2.098** ▲	100642.332	81205.502	0.973
$2^{25.00}$	5	1.231	100136.143	81328.546	0.969
$2^{26.00}$	5	1.186	▲ **100912.361** ▲	81654.033	0.637
$2^{26.58}$	5	1.145	100684.676	81676.922	0.790
$2^{27.00}$	5	0.883	100495.342	81589.467	0.836
$2^{27.32}$	5	0.894	100547.201	81653.131	0.515

The anomalies are marked with black triangles whose directions indicate whether the result is lower or higher than expected (triangle facing up indicates a result higher than expected and vice versa).

Only block sizes, that did not yield anomalies in the distribution defined by f has been used.

When filtering out large block sizes using the distribution f all numbers reached the same maximums, specifically:

Table 4.4: Largest block size used for the different bases

Base	Largest Block Size
2	10
3	10
5	7
7	6
10	5

Hence, when discussing a number in base 10 only block sizes 1,2,3,4,5 has been used.

In general the numbers tested have very few anomalies. The anomalies found are only seen in one or two of the four distributions and phase out as the prefix size grows.

An example of this is the first Pisot number. Base 2 and 3 have no significant anomalies, although in base 5 there are 4 prefix sizes that yield anomalies in the χ^2 distribution. However, the largest prefix shows no anomalies. This suggests that the lower prefix sizes are too small for the asymptotic distributions. In base 7 the Kolmogorov-Smirnov distribution peaks even on the largest prefix of $2^{27.81}$ bits. Hence, the first $2^{27.81}$ bits of the first Pisot number behaves normal in 4 of out 5 bases.

This scenario repeats with the first Salem number. Here Kolmogorov-Smirnov peaks in base 5; the only base that has anomalies at highest prefix size. Base 2 has several anomalies in the χ^2 distribution, but they are phased out as the prefix size reaches $2^{29.25}$ bits.

Likewise the first randomly generated number, Random-1, has anomalies in several bases. However, only in base 2 does the anomaly occur in the largest prefix. All other anomalies disappear in larger prefixes.

Table 4.5 list the observed anomalies in the largest prefix (numbers with less than 3 anomalies have been filtered out). The frequency distribution f has been left out, since it has no anomalies with the highest prefix (it would be an empty row).

Table 4.5: List of anomalies for each numbers

	Base					Distribution		
Number	2	3	5	7	10	χ^2	$\nabla^2\chi^2$	KS
Pisot-5	x				x	5	5	10
Pisot-7	x	x	x	x	x	3,5	2	7,10
Pisot-11	x			x	x		2,10	7,10
Pisot-14	x	x				2		2,3
Pisot-15		x		x		3,7	7	
Pisot-19	x	x		x		3	2	3,7
Pisot-24			x	x	x	7	7	5,10
Pisot-25	x		x		x		2	5,10
Pisot-30	x				x	2,10	10	2
Pisot-33	x	x				2		2,3
Pisot-37	x	x	x			5		2,3
Salem-2	x		x	x		2	10	7
Salem-5		x	x			3		3,5
Salem-16				x	x	10	7	7
Salem-17		x	x	x		3,7	7	5
Salem-21	x	x				2		2,3
Salem-24	x	x	x				2	3,5
Salem-32	x	x		x		2		2,3,7
Salem-42		x	x		x		5	3,5,10
Salem-43			x	x		5,7	7	
Random-5			x	x	x	5		7,10
Random-6	x		x	x		2,7		5
Random-13	x		x	x		7		2,5
Random-14	x		x			2	2,5	2

Even though there are several anomalies they are located in only a few of the 101 analyzed numbers. As with the anomalies in Pisot-1, that phased out with higher prefix sizes, it is likely that these anomalies may phase out with prefix sizes higher than what is used in this project ($2^{29.5}$).

Notice how the Serialized $\nabla^2\chi^2$ test does not fail as often than the two other tests. The reason for this oddity is unknown, however it relaxes the anomalies of the other methods. Also, base 10 seem to have fewer anomalies than the other bases tested.

The overview tables list the distribution results at highest prefix size along with the largest prefix needed to avoid anomalies in the remaining prefix sizes. That is, the smallest prefix for which larger prefixes larger prefixes do not yield anomalies. An example is the overview table of Pisot-1:

Table 4.6: Summary: pisot-1

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	0.995	1076.496	274.790	0.845	$2^{21.00}$	$2^{25.00}$	$2^{29.00}$	$2^{21.00}$
3	0.945	59085.813	26301.079	0.543	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{26.00}$
5	1.074	78876.007	50380.457	0.449	$2^{21.00}$	$2^{27.81}$	$2^{27.00}$	$2^{21.00}$
7	0.901	117277.824	85913.799	▲ **1.237** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
10	0.894	100547.201	81653.131	0.515	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{21.00}$

The table shows that a prefix size of 2^{29} is needed to avoid anomalies in base 2. It also shows that the Kolmogorov-Smirnov distribution has anomalies at highest prefix size in base 7.

In general there is no perfect prefix size that avoid anomalies in all the tested numbers. However, the anomalies are typically in a single distribution. Also, the f distribution has no anomalies at the largest prefix in any of the tested numbers.

An interesting observation is that several of the numbers avoid anomalies at a very low prefix size. For example Pisot-6 in base 7 has no anomalies at a prefix size of just 2^{21} (the smallest tested).

Another observation is that there is no significant different in the results of Pisot, Salem or randomly generated numbers. All of the groups have anomalies, however none of them in any particular base. The anomalies seem to spread randomly among the digits tested.

Also, the simple frequency distribution f shows no anomaly at highest prefix size in *any* number. Meaning that when the largest prefix of $2 \cdot 3^{18} \approx 2^{29.5}$ bits is used the digit frequencies are resembles an equidistributed within an error of 5%. This result is the same despite different block sizes, bases and polynomials.

The general picture gained from these tables is that the numbers may very well be normal. It is not possible to reject the conjecture from the statistics collected in this project.

CHAPTER 5

CONCLUSION AND FUTURE WORK

I have showed that the 101 algebraic irrational numbers analyzed in this project resembles normal numbers. Specifically, I showed that the first 39 Pisot numbers, the first 47 Salem numbers along with 15 randomly generated algebraic irrational numbers have χ^2, $\nabla^2\chi^2$ and Kolmogorov-Smirnov distributions that quantitatively resemble what could be expected from normal numbers, when using a confidence interval of 0.05. I have showed that no significant difference is observable between bases or Pisot, Salem and randomly generated roots. The distribution of anomalies seem itself normal.

I show that the digit frequencies of all the tested numbers resembles an equidistribution within an error margin of 5%. This is true for all 101 numbers in all 5 bases and in all block sizes.

This result supports Émil Borel's conjecture that all algebraic irrational numbers are normal. First by leaving no pattern in the observed anomalies, despite number and base used and secondly by leaving showing no anomalies in the frequency distributions.

The framework implemented has been designed to allow extensions like the addition of other root finding algorithms or other statistical methods. This could be the basis of other projects investigating Borel's conjecture.

If the conjecture is true, roots of algebraic irrationals could be used to build perfect pseudo-random number generators (PRNGs). One could imagine a system that outputs random numbers by constructing a polynomial from a given *seed* and hereafter compute its roots.

Another interesting application that has its roots in cryptography is a secure hash function. As with the PRNG, a polynomial could be constructed from the input. The hash (or "fingerprint") could then be computed as a specific part of the root. If the polynomials are chosen carefully, it might be possible to prove the sought properties, such as hash-collision resistant.

Based on such a hash function, one could build a perfect block cipher, whose output resembles random permutations, solving the problems of insecure encryption ciphers. It would be interesting to see further work in this specific field.

CHAPTER 6

BIBLIOGRAPHY

6.1 Bibliography

[1] Gmplib: Gnu multiprecision library. http://gmplib.org. version 4.2.4.

[2] Gmpy: Gmplib python wrapper. http://code.google.com/p/gmpy/. version 1.04.

[3] Numpy: Fundamental scientific computing package for python. http://numpy.scipy.org/. version 1.1.1.

[4] Postgresql: object-relational sql database management. http://postgresql.org/. version 8.3.4-2.2.

[5] The python programming language. http://python.org. version 2.5.2.

[6] Scipy: Scientific computing package for python. http://scipy.org. version 0.6.0.

[7] Sympy: Python library for symbolic mathematics. http://code.google.com/p/sympy/. version 0.6.5-git.

[8] F. S. Acton. Numerical Methods that Work. MMA, illustrated and revised edition, 1990. ISBN-10: $08 - 8385 - 450 - 3$.

[9] M. Artin. Algebra. Prentice Hall, 1991. ISBN-10: 01-3004-763-5.

[10] D. H. Bailey. The computation of pi to 29,360,000 decimal digit using borwein's quartically convergent algorithm. Mathematics of Computations, 50:283–296, 1988.

[11] D. H. Bailey and R. E. Crandall. On the random character of fundamental constant expansions. Experimental Mathematics, 10:175–190, 2001.

[12] D. H. Bailey and R. E. Crandall. Random generators and normal numbers. Experimental Mathematics, 11:527–546, 2002.

[13] W. A. Beyer, N. Metropolis, and J. R. Neergaard. Square roots of integers 2 to 15 in various bases 2 to 10: 88062 binary digits or equivalent. Mathematics of Computations, 23:679, 1969.

[14] W. A. Beyer, N. Metropolis, and J. R. Neergaard. Statistical study of digits of some square roots of integers in various bases. Mathematics of Computations, 24:455–473, 1970a.

[15] D. Bini and G. Fiorentino. Mpsolve benchmarks. http://www.dm.unipi.it/cluster-pages/mpsolve/bench.htm. Seen online: Jun, 2009.

[16] D. G. Champernowne. The construction of decimals normal in the scale of ten. Journal of the London Mathematical Society, 8:254–260, 1933.

[17] A. Decomps-Guilloux. Pisot and Salem Numbers. Birkhauser, 1993. ISBN-10: 08-1762-648-4.

[18] M. Émile Borel. Les probabilités dénombrables et leurs applications arithmétiques. Rendiconti del Circolo Matematico di Palermo, 27(1):247–271, december 1909.

[19] I. J. Good and T. N. Gover. Corrigendum: The generalized serial test and the binary expansion of $\sqrt{2}$. Journal of the Royal Statistical Society. Series A (General), 131(3):434, 1968.

[20] A. K. Gupta and A. K. Mittal. A counting method for finding rational approximates to arbitrary order roots of integers, 1999.

[21] A. K. Gupta and A. K. Mittal. A 'replacement sequence' method for finding the largest real root of an integer monic polynomial, 1999.

[22] A. K. Gupta and A. K. Mittal. Integer sequences associated with integer monic polynomial, 2000.

[23] A. K. Gupta and A. K. Mittal. Symbolic computation of the roots of any polynomial with integer coefficients, 2000.

[24] R. Isaac. On the simple normality to base 2 of sqrt(s), for s not a perfect square. 2005.

[25] D. E. Knuth. Art of Computer Programming, Volume 2: Seminumerical Algorithms. Addison-Wesley Professional, 3 edition, November 1997.

[26] S. G. Krantz. Handbook of Complex Variables. Birkhauser Boston, 1 edition, 1999.

[27] S. Kullback and R. A. Leibler. On information an sufficiency. The Annals of Mathematical Statistics, 22(1):79–86, 1951.

[28] S. J. Leon. Linear Algebra With Applications. Pearson Prentice Hall, 7th edition, 2006. ISBN-10: 01-3200-306-6.

[29] J. W. Martin. Gmp intel core2 assembler optimisation patchset. http://math.jmu.edu/~martin, 2006. Seen online: Jun, 2009.

[30] W. R. Mekwi. Iterative methods for roots of polynomials. Master's thesis, University of Oxford, 2001.

[31] M. Monsalve and C. Gutierrez. Fast inverse for big numbers: Picarte's iteration. Computer Science Department, Universidad de Chile.

[32] A. M. Mood, F. A. Graybill, and D. C. Boes. Introduction to the Theory of Statistics. McGraw-Hill Companies, 3 edition, 1974.

[33] M. Mossinghoff. Small salem numbers. http://www.cecm.sfu.ca/∼mjm/Lehmer/lists/SalemList.html. Seen online: Jun, 2009.

[34] R. K. Pathria. A statistical study of randomness among the first 10,000 digits of π. Mathematics of Computations, 16:188–197, 1962.

[35] R. L. Plackett. Karl pearson and the chi-squared test. International Statistical Review, 51:59–72, 1983.

[36] M. Printzlau and M. B. Pedersen. When roots are random. Dept. of Computer Science, University of Copenhagen, June 2007.

[37] W. Schmidt. On normal numbers. Pacific Journal of Mathematics, 10:661–672, 1960.

[38] I. Stewart and D. Tall. Algebraic Number Theory. Chapman and Hall, 1987.

[39] R. G. Stoneham. A study of the 60,000 digits of the transcendental "e". American Mathematical Monthly, 72:483–506, 1965.

[40] J. A. Tropp. Comparison of rootfinding methods. www.math.lsa.umich.edu/∼annacg/courses/m471-f07/handout/m471-f07-rootfinding.pdf. Seen online: Jun, 2009.

LIST OF TABLES

APPENDIX A

STATISTICAL RESULTS

A.1 Statistical Summary

Overview

In this section I present a summation of the results. Each table presents the accumulated results for the root as long as the largest indices, for which the resulting suffix of the computed root fulfills the expectations of *chi*^{2}, $\nabla\chi^2$ and Kolmigorov-Smirnov respectively.

The following table illustrates the layout:

		All Blocks			Largest Index		
Base	N	χ^2	$\nabla\chi^2$	KS	χ^2	$\nabla\chi^2$	KS
2	2^{29}	0.1	1.0	0.1	20000	25000	30000
3	2^{29}	0.1	1.0	0.1	20000	25000	30000
5	2^{29}	0.1	1.0	0.1	20000	25000	30000
7	2^{29}	0.1	1.0	0.1	20000	25000	30000
10	2^{29}	0.1	1.0	0.1	20000	25000	30000

Table A.1: Summary: pisot-1

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	0.995	1076.496	274.790	0.845	$2^{21.00}$	$2^{25.00}$	$2^{29.00}$	$2^{21.00}$
3	0.945	59085.813	26301.079	0.543	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{26.00}$
5	1.074	78876.007	50380.457	0.449	$2^{21.00}$	$2^{27.81}$	$2^{27.00}$	$2^{21.00}$
7	0.901	117277.824	85913.799	▲ **1.237** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
10	0.894	100547.201	81653.131	0.515	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{21.00}$

Table A.2: Summary: pisot-2

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	▼ **928.751** ▼	246.995	0.883	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{21.00}$
3	0.948	59357.597	26521.404	1.065	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.58}$
5	1.082	78115.870	50229.221	▲ **1.623** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.00}$
7	1.102	117223.040	86398.919	1.155	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.58}$
10	1.112	99218.928	80591.122	0.979	$2^{21.00}$	$2^{25.00}$	$2^{26.00}$	$2^{26.00}$

Table A.3: Summary: pisot-3

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1003.809	261.937	0.367	$2^{21.00}$	$2^{28.58}$	$2^{21.00}$	$2^{28.91}$
3	1.053	58755.138	25999.977	▲ **1.311** ▲	$2^{21.00}$	$2^{27.81}$	$2^{21.00}$	$2^{28.70}$
5	1.070	▲ **79004.961** ▲	50423.973	0.696	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{25.00}$
7	1.109	117325.537	86164.367	1.071	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.107	99751.228	80678.378	1.037	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.00}$

Table A.4: Summary: pisot-4

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1030.134	253.899	0.517	$2^{21.00}$	$2^{28.91}$	$2^{21.00}$	$2^{21.00}$
3	1.052	59259.344	26193.642	▲ **1.273** ▲	$2^{21.00}$	$2^{28.17}$	$2^{21.00}$	$2^{28.70}$
5	0.924	78103.399	49729.891	1.027	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.098	117297.735	86256.492	1.133	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.893	99887.607	81340.137	1.012	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{25.00}$

Table A.5: Summary: pisot-5

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1084.890	278.741	0.954	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{28.17}$
3	1.053	▼ **58095.795** ▼	▼ **25748.410** ▼	0.807	$2^{21.00}$	$2^{28.70}$	$2^{28.70}$	$2^{21.00}$
5	0.928	77977.789	49796.112	1.040	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.00}$
7	0.903	116738.204	85970.056	0.635	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.111	100161.275	80879.437	▲ **1.235** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$

Table A.6: Summary: pisot-6

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	968.117	226.789	1.018	$2^{21.00}$	$2^{29.00}$	$2^{29.32}$	$2^{26.58}$
3	1.052	58720.831	▼ **25700.657** ▼	1.143	$2^{21.00}$	$2^{28.32}$	$2^{28.70}$	$2^{28.58}$
5	1.077	78076.673	49867.672	1.101	$2^{21.00}$	$2^{26.58}$	$2^{26.58}$	$2^{27.81}$
7	0.889	117586.907	87070.705	0.724	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.103	99449.450	80541.111	0.658	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{21.00}$

Table A.7: Summary: pisot-7

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	976.740	▲ **306.865** ▲	0.520	$2^{21.00}$	$2^{25.00}$	$2^{29.46}$	$2^{28.46}$
3	0.910	▲ **60443.933** ▲	26435.494	0.843	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{21.00}$
5	1.071	▲ **79060.325** ▲	50388.623	0.996	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{27.58}$
7	1.100	117698.429	86236.963	▲ **1.603** ▲	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{27.81}$
10	1.121	100408.353	81722.524	▲ **1.789** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$

Table A.8: Summary: pisot-8

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	▼ **934.118** ▼	241.816	0.768	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{29.09}$
3	0.950	▼ **57920.070** ▼	25819.094	0.399	$2^{21.00}$	$2^{28.70}$	$2^{28.58}$	$2^{27.32}$
5	0.928	77832.400	49802.419	0.881	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.105	117927.252	86679.098	0.833	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.879	99658.849	80724.208	0.947	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{21.00}$

Table A.9: Summary: pisot-9

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1036.813	243.422	1.099	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{29.39}$
3	1.051	▲ **59852.535** ▲	26458.042	0.601	$2^{21.00}$	$2^{28.70}$	$2^{27.00}$	$2^{21.00}$
5	1.077	78285.312	50297.780	0.522	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.00}$
7	0.899	▼ **116530.168** ▼	85840.327	0.988	$2^{21.00}$	$2^{27.81}$	$2^{21.00}$	$2^{21.00}$
10	1.110	99774.512	81069.696	0.693	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{21.00}$

Table A.10: Summary: pisot-10

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	964.057	244.192	0.767	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{29.09}$
3	0.947	59257.447	26212.141	0.765	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{28.00}$
5	1.077	78338.529	50585.181	0.827	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$	$2^{21.00}$
7	1.100	116831.029	▼ **85472.247** ▼	0.681	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$	$2^{21.00}$
10	1.111	100055.055	81213.335	0.771	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.11: Summary: pisot-11

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.005	944.427	▼ **206.287** ▼	0.846	$2^{21.00}$	$2^{29.39}$	$2^{29.46}$	$2^{27.00}$
3	0.948	59141.289	26129.312	0.594	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
5	0.926	78471.326	49724.663	1.082	$2^{21.00}$	$2^{21.00}$	$2^{27.58}$	$2^{26.00}$
7	1.109	117672.272	86721.187	▲ **1.342** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
10	1.112	100542.750	▲ **81863.076** ▲	▲ **1.441** ▲	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$	$2^{27.32}$

Table A.12: Summary: pisot-12

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1088.148	239.703	0.925	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{21.00}$
3	1.054	59347.437	26319.546	1.150	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.58}$
5	1.075	▲ **79010.531** ▲	50111.508	0.718	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{26.58}$
7	0.906	117664.658	86626.753	0.788	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.110	99269.040	▼ **80158.848** ▼	0.978	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$

Table A.13: Summary: pisot-13

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	951.411	▼ **212.834** ▼	0.986	$2^{21.00}$	$2^{28.81}$	$2^{29.46}$	$2^{21.00}$
3	0.951	59481.116	26339.497	0.594	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
5	1.074	78214.403	49894.631	0.915	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.115	117305.098	86118.118	0.965	$2^{21.00}$	$2^{21.00}$	$2^{27.58}$	$2^{21.00}$
10	1.112	▼ **99076.498** ▼	80432.332	1.053	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$	$2^{27.00}$

Table A.14: Summary: pisot-14

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	▲ **1132.094** ▲	285.647	▲ **2.262** ▲	$2^{21.00}$	$2^{29.46}$	$2^{27.32}$	$2^{29.46}$
3	1.049	59640.403	26442.227	▲ **1.338** ▲	$2^{21.00}$	$2^{21.00}$	$2^{26.58}$	$2^{28.70}$
5	0.929	77469.826	49797.515	1.088	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
7	1.103	117532.404	86164.631	0.658	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.136	99913.908	80720.118	0.653	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{21.00}$

Table A.15: Summary: pisot-15

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1021.896	249.455	0.801	$2^{21.00}$	$2^{27.58}$	$2^{21.00}$	$2^{27.00}$
3	0.952	▼ **57885.721** ▼	26061.008	0.743	$2^{21.00}$	$2^{28.70}$	$2^{27.00}$	$2^{21.00}$
5	1.084	78201.757	50039.043	0.536	$2^{21.00}$	$2^{25.00}$	$2^{25.00}$	$2^{21.00}$
7	0.895	▼ **116522.483** ▼	▼ **85386.859** ▼	0.968	$2^{21.00}$	$2^{27.81}$	$2^{27.81}$	$2^{21.00}$
10	0.875	100583.693	81473.486	1.026	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{21.00}$

Table A.16: Summary: pisot-16

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.003	946.259	274.394	0.497	$2^{21.00}$	$2^{28.91}$	$2^{21.00}$	$2^{21.00}$
3	0.945	59562.916	26311.996	0.909	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
5	0.924	78361.260	50031.120	1.217	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{25.00}$
7	1.096	116809.264	85970.879	0.665	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{25.00}$
10	1.122	▲ **101066.089** ▲	▲ **81793.767** ▲	0.678	$2^{21.00}$	$2^{27.32}$	$2^{27.32}$	$2^{21.00}$

Table A.17: Summary: pisot-17

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	980.824	224.429	0.624	$2^{21.00}$	$2^{28.17}$	$2^{26.58}$	$2^{28.00}$
3	0.952	58378.044	25922.069	0.774	$2^{21.00}$	$2^{25.00}$	$2^{28.17}$	$2^{21.00}$
5	0.923	77945.883	50216.655	0.595	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	0.901	▼ **116652.377** ▼	85995.426	1.153	$2^{21.00}$	$2^{27.81}$	$2^{21.00}$	$2^{21.00}$
10	1.110	100786.957	81739.975	1.122	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.18: Summary: pisot-18

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	975.481	256.946	1.216	$2^{21.00}$	$2^{28.32}$	$2^{21.00}$	$2^{29.39}$
3	1.049	59207.287	26678.708	▲ **1.404** ▲	$2^{21.00}$	$2^{21.00}$	$2^{28.58}$	$2^{28.70}$
5	1.082	78638.125	50463.201	0.840	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.112	117065.895	85924.873	0.941	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.888	99846.787	81186.367	0.489	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.19: Summary: pisot-19

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	0.995	1051.279	▼ **210.964** ▼	0.876	$2^{21.00}$	$2^{28.58}$	$2^{29.46}$	$2^{21.00}$
3	1.052	▼ **58017.458** ▼	25864.487	▲ **1.455** ▲	$2^{21.00}$	$2^{28.70}$	$2^{28.58}$	$2^{28.70}$
5	0.917	78294.195	49988.208	0.688	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{21.00}$
7	0.901	117117.721	86395.527	▲ **1.473** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
10	1.099	99325.540	80670.553	0.923	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{21.00}$

Table A.20: Summary: pisot-20

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.005	1039.973	246.044	0.785	$2^{21.00}$	$2^{28.70}$	$2^{21.00}$	$2^{28.00}$
3	1.053	59261.632	26057.547	1.103	$2^{21.00}$	$2^{27.58}$	$2^{27.32}$	$2^{21.00}$
5	1.085	78125.005	49751.917	1.043	$2^{21.00}$	$2^{26.00}$	$2^{25.00}$	$2^{21.00}$
7	0.899	117338.291	86503.556	1.219	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.58}$
10	1.111	99999.695	80936.720	0.869	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.21: Summary: pisot-21

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1104.054	259.908	▲ **1.400** ▲	$2^{21.00}$	$2^{29.17}$	$2^{21.00}$	$2^{29.46}$
3	1.050	58658.171	26321.618	0.712	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
5	0.918	78190.707	50493.320	▲ **1.303** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.00}$
7	1.094	116953.959	86253.242	0.895	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.116	99492.288	80586.449	0.903	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.22: Summary: pisot-22

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1039.077	278.556	0.755	$2^{21.00}$	$2^{21.00}$	$2^{29.09}$	$2^{26.00}$
3	0.944	▲ **60046.122** ▲	26048.644	0.944	$2^{21.00}$	$2^{28.70}$	$2^{27.32}$	$2^{27.00}$
5	1.070	78001.943	50218.581	0.479	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{25.00}$
7	1.107	117849.848	86518.463	0.870	$2^{21.00}$	$2^{25.00}$	$2^{25.00}$	$2^{27.32}$
10	0.888	100268.210	81237.856	0.818	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.00}$

Table A.23: Summary: pisot-23

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1091.383	301.610	0.541	$2^{21.00}$	$2^{28.00}$	$2^{29.32}$	$2^{28.32}$
3	1.052	59179.181	▼ **25612.615** ▼	0.403	$2^{21.00}$	$2^{26.00}$	$2^{28.70}$	$2^{21.00}$
5	0.924	78213.784	50032.425	1.092	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.00}$
7	0.900	118436.895	86772.202	1.150	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.110	99395.370	80810.793	0.606	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.24: Summary: pisot-24

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.005	958.567	249.307	0.706	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
3	1.051	58556.458	26173.039	0.732	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{21.00}$
5	0.924	78170.562	50357.307	▲ **1.573** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.00}$
7	1.098	▲ **119434.993** ▲	▲ **87323.040** ▲	1.019	$2^{21.00}$	$2^{27.81}$	$2^{27.81}$	$2^{21.00}$
10	1.139	100238.907	81306.015	▲ **1.296** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.58}$

Table A.25: Summary: pisot-25

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1045.099	▼ **211.961** ▼	0.490	$2^{21.00}$	$2^{28.91}$	$2^{29.46}$	$2^{21.00}$
3	1.053	59298.120	26410.440	0.666	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{21.00}$
5	1.081	77840.709	49901.373	▲ **1.836** ▲	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{28.00}$
7	1.100	118232.896	86727.352	0.908	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.119	99542.496	80698.758	▲ **1.259** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$

Table A.26: Summary: pisot-26

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1072.844	263.761	1.102	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
3	1.048	59331.821	26384.923	0.796	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{21.00}$
5	0.922	78883.708	50417.368	1.170	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.00}$
7	1.096	118444.061	86386.642	0.741	$2^{21.00}$	$2^{27.58}$	$2^{21.00}$	$2^{21.00}$
10	1.112	99350.711	80748.011	1.147	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{26.00}$

Table A.27: Summary: pisot-27

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.995	1090.123	267.830	▲ **1.499** ▲	$2^{21.00}$	$2^{29.09}$	$2^{26.58}$	$2^{29.46}$
3	0.946	59528.961	26092.512	0.751	$2^{21.00}$	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$
5	1.080	78282.287	50040.199	0.914	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{21.00}$
7	0.901	118343.200	86938.173	0.827	$2^{21.00}$	$2^{27.58}$	$2^{21.00}$	$2^{21.00}$
10	0.894	99597.939	80956.361	1.105	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.28: Summary: pisot-28

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1090.198	268.939	0.575	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.00}$
3	0.946	58512.731	26178.533	1.066	$2^{21.00}$	$2^{28.46}$	$2^{21.00}$	$2^{27.00}$
5	0.923	77898.092	50090.241	▲ **1.421** ▲	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{28.00}$
7	0.899	117529.503	86192.895	▲ **1.374** ▲	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{27.81}$
10	1.109	99433.469	80951.802	0.782	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.29: Summary: pisot-29

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	936.465	236.723	0.682	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$	$2^{21.00}$
3	1.055	59079.405	26398.514	1.050	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{21.00}$
5	1.078	78216.643	49903.686	▲ **1.800** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.00}$
7	1.101	117390.104	86656.480	▲ **1.556** ▲	$2^{21.00}$	$2^{21.00}$	$2^{25.00}$	$2^{27.81}$
10	1.116	99922.922	81042.892	0.913	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.30: Summary: pisot-30

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.995	▲ **1148.213** ▲	263.182	▲ **1.412** ▲	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{29.46}$
3	1.056	58743.674	26437.646	0.852	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{28.00}$
5	0.926	78205.400	50151.180	0.650	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.098	118194.289	86832.696	0.571	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.106	▲ **101566.154** ▲	▲ **82157.386** ▲	0.650	$2^{21.00}$	$2^{27.32}$	$2^{27.32}$	$2^{21.00}$

Table A.31: Summary: pisot-31

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	▼ **881.780** ▼	248.951	0.744	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{21.00}$
3	1.051	59278.214	26029.648	0.737	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
5	1.077	▲ **78936.453** ▲	50198.693	0.631	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{25.00}$
7	1.102	117674.965	86117.105	0.763	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.899	100417.225	81597.472	0.717	$2^{21.00}$	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$

Table A.32: Summary: pisot-32

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.994	1093.997	277.184	1.212	$2^{21.00}$	$2^{26.00}$	$2^{29.00}$	$2^{28.58}$
3	1.052	59328.891	26238.788	0.813	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.58}$
5	1.118	78106.858	49913.314	1.210	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.58}$
7	1.097	118157.101	86624.563	▲ **1.791** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
10	1.109	100451.462	81268.542	1.157	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.58}$

Table A.33: Summary: pisot-33

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	▲ **1209.974** ▲	268.252	▲ **1.690** ▲	$2^{21.00}$	$2^{29.46}$	$2^{28.17}$	$2^{29.46}$
3	0.948	58868.925	26292.559	▲ **1.760** ▲	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{28.70}$
5	1.072	77983.132	49698.252	1.114	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.58}$
7	1.105	117305.956	86467.313	0.560	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.106	99712.121	80917.438	1.148	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.00}$

Table A.34: Summary: pisot-34

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1061.875	296.423	0.717	$2^{21.00}$	$2^{29.25}$	$2^{29.25}$	$2^{27.00}$
3	0.945	59384.474	25930.658	1.120	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.00}$
5	0.919	78455.207	50217.564	▲ **1.250** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.00}$
7	1.099	117528.175	86077.406	1.195	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.58}$
10	1.105	99430.293	80660.088	0.888	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.35: Summary: pisot-35

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1024.666	259.538	0.578	$2^{21.00}$	$2^{28.81}$	$2^{21.00}$	$2^{28.58}$
3	1.049	58891.812	26113.407	1.062	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{28.58}$
5	1.073	78123.770	50361.636	0.949	$2^{21.00}$	$2^{25.00}$	$2^{25.00}$	$2^{21.00}$
7	1.099	117501.430	86925.433	0.612	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{21.00}$
10	0.888	100714.818	81249.474	1.087	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.00}$

Table A.36: Summary: pisot-36

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.005	974.008	248.842	▲ **1.260** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{29.46}$
3	1.052	59347.127	26287.953	0.551	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
5	1.079	77727.516	49740.083	0.856	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.108	117362.637	85856.754	▲ **1.353** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
10	0.879	100172.526	81121.366	0.611	$2^{21.00}$	$2^{26.00}$	$2^{25.00}$	$2^{21.00}$

Table A.37: Summary: pisot-37

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1013.827	250.482	▲ **1.430** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{29.46}$
3	0.952	59040.535	26341.521	▲ **1.564** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.70}$
5	1.086	▼ **77250.010** ▼	49560.054	0.701	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{26.58}$
7	1.107	118542.149	87200.427	0.822	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$	$2^{21.00}$
10	1.106	99508.820	80292.018	0.765	$2^{21.00}$	$2^{25.00}$	$2^{27.00}$	$2^{21.00}$

Table A.38: Summary: pisot-38

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1010.513	248.403	0.414	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{21.00}$
3	1.054	59121.797	26441.458	0.468	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{21.00}$
5	1.079	▼ **77283.437** ▼	49893.508	0.593	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{21.00}$
7	0.902	117392.202	86064.032	1.137	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$
10	0.893	99369.077	80315.412	0.790	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.39: Summary: pisot-39

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1109.669	241.051	1.008	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{29.32}$
3	1.049	58781.137	26114.090	0.867	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{21.00}$
5	1.086	77791.350	49651.697	1.162	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
7	1.101	117895.209	86589.526	0.675	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.114	100268.701	81016.963	0.621	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.40: Summary: salem-1

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.005	1045.352	289.371	1.059	$2^{21.00}$	$2^{29.17}$	$2^{28.91}$	$2^{21.00}$
3	0.950	58650.472	26060.544	0.535	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{21.00}$
5	1.075	77592.459	49591.469	▲ **1.330** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.00}$
7	1.095	117970.892	86604.983	0.915	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{21.00}$
10	1.106	99541.881	80541.823	1.212	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.41: Summary: salem-2

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.005	▲ **1123.524** ▲	230.356	0.739	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{28.91}$
3	1.051	58464.559	26046.837	0.704	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{27.81}$
5	0.920	78803.566	50235.933	1.056	$2^{21.00}$	$2^{27.81}$	$2^{26.00}$	$2^{27.32}$
7	1.093	118434.080	86717.454	▲ **1.482** ▲	$2^{21.00}$	$2^{27.58}$	$2^{27.58}$	$2^{27.81}$
10	1.120	100762.833	▲ **81855.385** ▲	0.757	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$

Table A.42: Summary: salem-3

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1054.413	260.264	0.637	$2^{21.00}$	$2^{29.09}$	$2^{21.00}$	$2^{26.00}$
3	1.046	▼ **57861.840** ▼	26187.052	1.038	$2^{21.00}$	$2^{28.70}$	$2^{21.00}$	$2^{27.58}$
5	1.070	77638.261	50036.439	0.455	$2^{21.00}$	$2^{25.00}$	$2^{26.58}$	$2^{21.00}$
7	0.906	▼ **116639.349** ▼	86101.666	0.678	$2^{21.00}$	$2^{27.81}$	$2^{21.00}$	$2^{27.00}$
10	1.105	99869.800	80636.246	0.726	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.43: Summary: salem-4

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1066.053	▲ **303.270** ▲	0.672	$2^{21.00}$	$2^{28.58}$	$2^{29.46}$	$2^{27.32}$
3	0.946	59288.286	26227.765	▲ **1.908** ▲	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$	$2^{28.70}$
5	0.927	77825.399	50207.646	0.914	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.111	117388.677	86421.987	0.919	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.893	100144.457	81152.747	0.854	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.44: Summary: salem-5

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1079.696	252.670	1.220	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{29.39}$
3	1.050	▼ **58188.950** ▼	26209.327	▲ **1.806** ▲	$2^{21.00}$	$2^{28.70}$	$2^{21.00}$	$2^{28.70}$
5	0.915	77952.076	49997.716	▲ **1.680** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.00}$
7	0.902	117916.171	86090.981	0.687	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.892	99954.524	81006.618	1.191	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.00}$

Table A.45: Summary: salem-6

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1055.699	243.522	1.183	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{29.32}$
3	0.946	58988.798	25998.073	0.751	$2^{21.00}$	$2^{27.32}$	$2^{28.17}$	$2^{21.00}$
5	1.082	▼ **77271.173** ▼	49440.290	0.561	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{21.00}$
7	1.111	117602.341	86093.148	0.725	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.894	100543.241	81428.538	0.654	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.46: Summary: salem-7

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.005	1113.261	283.710	0.874	$2^{21.00}$	$2^{28.58}$	$2^{21.00}$	$2^{28.70}$
3	0.945	59209.874	26329.870	0.813	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.00}$
5	1.081	▼ **77070.688** ▼	▼ **49331.396** ▼	0.874	$2^{21.00}$	$2^{28.00}$	$2^{28.00}$	$2^{26.00}$
7	1.099	117528.896	86650.649	0.683	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.099	100386.747	81071.100	0.690	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.47: Summary: salem-8

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	994.715	247.153	▲ **1.292** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{29.46}$
3	0.949	58878.890	26157.534	▲ **1.391** ▲	$2^{21.00}$	$2^{27.00}$	$2^{26.58}$	$2^{28.70}$
5	0.924	78328.356	50079.309	0.822	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.114	117729.962	86983.014	1.218	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.110	99575.093	80757.412	0.608	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.48: Summary: salem-9

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.997	941.898	254.857	0.561	$2^{21.00}$	$2^{29.39}$	$2^{21.00}$	$2^{21.00}$
3	1.048	▲ **59940.100** ▲	26079.400	0.928	$2^{21.00}$	$2^{28.70}$	$2^{21.00}$	$2^{21.00}$
5	1.075	78146.646	50167.475	0.580	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	0.899	117671.308	86231.010	0.576	$2^{21.00}$	$2^{26.00}$	$2^{26.58}$	$2^{21.00}$
10	1.113	100585.050	81412.015	0.707	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.49: Summary: salem-10

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	▼ **863.674** ▼	232.756	0.628	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{25.00}$
3	0.952	59491.704	26379.931	0.516	$2^{21.00}$	$2^{28.46}$	$2^{21.00}$	$2^{21.00}$
5	1.077	▼ **77152.674** ▼	49596.690	0.931	$2^{21.00}$	$2^{28.00}$	$2^{27.81}$	$2^{21.00}$
7	1.100	117650.568	86157.622	0.731	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.886	99988.610	81359.057	0.598	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.50: Summary: salem-11

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	965.401	241.655	0.929	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{29.25}$
3	1.059	59355.769	26323.204	0.666	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{25.00}$
5	0.921	77738.321	49817.797	0.903	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	0.901	117458.759	86232.286	1.055	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$
10	1.111	100248.184	81486.944	0.914	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.51: Summary: salem-12

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	▼ **865.426** ▼	241.231	1.068	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{29.39}$
3	0.952	59303.836	26248.855	0.635	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
5	1.075	78413.411	50215.512	0.973	$2^{21.00}$	$2^{26.00}$	$2^{26.00}$	$2^{21.00}$
7	1.101	117214.988	86279.864	0.733	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{21.00}$
10	1.110	99751.085	80648.689	0.722	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.52: Summary: salem-13

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1038.481	▲ **305.137** ▲	0.950	$2^{21.00}$	$2^{29.00}$	$2^{29.46}$	$2^{27.00}$
3	1.052	58931.310	26177.966	▲ **1.608** ▲	$2^{21.00}$	$2^{27.81}$	$2^{21.00}$	$2^{28.70}$
5	1.075	78458.218	49964.078	0.981	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.102	117899.070	86930.092	0.746	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.887	100232.954	81515.674	0.671	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.58}$

Table A.53: Summary: salem-14

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1042.778	283.091	0.484	$2^{21.00}$	$2^{28.81}$	$2^{29.32}$	$2^{21.00}$
3	0.952	59262.251	26296.702	1.068	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{28.58}$
5	0.919	77830.150	49751.618	0.702	$2^{21.00}$	$2^{27.00}$	$2^{27.32}$	$2^{21.00}$
7	1.098	116812.794	86268.714	0.550	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{25.00}$
10	1.105	99530.330	80610.286	0.875	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.54: Summary: salem-15

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.005	1044.047	240.751	0.974	$2^{21.00}$	$2^{28.32}$	$2^{29.09}$	$2^{21.00}$
3	1.050	58794.752	26155.528	0.389	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{25.00}$
5	1.079	77906.386	49748.953	0.562	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.102	117373.915	86964.468	0.599	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{21.00}$
10	0.891	99334.887	80803.784	0.937	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{21.00}$

Table A.55: Summary: salem-16

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1023.781	255.001	0.775	$2^{21.00}$	$2^{21.00}$	$2^{28.32}$	$2^{26.00}$
3	0.953	59486.959	25937.515	0.717	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
5	1.075	78697.266	50198.728	0.676	$2^{21.00}$	$2^{27.58}$	$2^{21.00}$	$2^{21.00}$
7	0.902	116870.212	▼ **85383.469** ▼	▲ **1.361** ▲	$2^{21.00}$	$2^{27.00}$	$2^{27.81}$	$2^{27.81}$
10	0.896	▼ **99039.329** ▼	80213.566	0.957	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$	$2^{21.00}$

Table A.56: Summary: salem-17

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	0.995	1111.470	262.102	0.614	$2^{21.00}$	$2^{29.39}$	$2^{26.58}$	$2^{21.00}$
3	1.047	▼ **58348.287** ▼	25941.996	0.708	$2^{21.00}$	$2^{28.70}$	$2^{21.00}$	$2^{28.46}$
5	1.072	77465.567	49480.988	▲ **1.297** ▲	$2^{21.00}$	$2^{21.00}$	$2^{25.00}$	$2^{28.00}$
7	0.898	▼ **116607.830** ▼	▼ **85341.361** ▼	0.831	$2^{21.00}$	$2^{27.81}$	$2^{27.81}$	$2^{21.00}$
10	1.122	100141.730	81233.622	1.130	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.57: Summary: salem-18

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	▲ **1130.296** ▲	252.203	▲ **2.160** ▲	$2^{21.00}$	$2^{29.46}$	$2^{27.32}$	$2^{29.46}$
3	1.055	58749.257	26302.475	0.621	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.00}$
5	1.078	77652.003	50027.770	1.179	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.00}$
7	1.101	117097.129	86250.553	1.157	$2^{21.00}$	$2^{27.58}$	$2^{25.00}$	$2^{27.58}$
10	1.106	99411.098	80263.105	0.670	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.58: Summary: salem-19

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1073.593	▲ **322.056** ▲	0.707	$2^{21.00}$	$2^{29.00}$	$2^{29.46}$	$2^{21.00}$
3	1.046	59118.811	26490.588	0.497	$2^{21.00}$	$2^{21.00}$	$2^{28.46}$	$2^{21.00}$
5	1.072	▲ **78951.474** ▲	50072.472	0.816	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{26.00}$
7	0.895	118030.489	86936.001	0.707	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{21.00}$
10	1.109	99961.576	80995.121	0.588	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.59: Summary: salem-20

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	▼ **905.265** ▼	245.748	0.493	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{28.81}$
3	1.056	59065.773	26470.684	1.087	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
5	1.079	78181.911	50183.842	0.877	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.107	117423.774	86474.933	0.576	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.125	99332.413	▼ **80187.061** ▼	0.890	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$

Table A.60: Summary: salem-21

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.994	▲ **1147.148** ▲	274.799	▲ **1.542** ▲	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{29.46}$
3	0.946	59267.483	26683.374	▲ **1.343** ▲	$2^{21.00}$	$2^{27.81}$	$2^{28.58}$	$2^{28.70}$
5	1.082	78127.678	50131.796	0.964	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.102	117153.160	86113.071	0.910	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.00}$
10	1.107	100327.274	81204.122	0.961	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.61: Summary: salem-22

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.005	▲ **1171.564** ▲	273.715	1.038	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{29.25}$
3	1.047	59081.568	26413.569	0.836	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.00}$
5	1.080	78040.151	50188.010	0.744	$2^{21.00}$	$2^{27.00}$	$2^{27.00}$	$2^{27.32}$
7	1.105	117210.056	86342.972	0.647	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{25.00}$
10	0.890	100187.303	80781.231	▲ **1.401** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$

Table A.62: Summary: salem-23

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.995	992.365	256.646	0.571	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
3	0.950	▼ **58328.351** ▼	26107.164	1.149	$2^{21.00}$	$2^{28.70}$	$2^{21.00}$	$2^{28.58}$
5	0.928	77407.341	49810.300	0.772	$2^{21.00}$	$2^{27.81}$	$2^{21.00}$	$2^{21.00}$
7	0.900	117378.360	87137.809	1.095	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$
10	1.104	99561.374	80593.938	▲ **1.502** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$

Table A.63: Summary: salem-24

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	964.266	▼ **209.107** ▼	0.499	$2^{21.00}$	$2^{27.00}$	$2^{29.46}$	$2^{21.00}$
3	0.953	59713.482	26529.647	▲ **1.406** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.70}$
5	0.918	77949.427	50376.275	▲ **1.408** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.00}$
7	0.897	116965.299	85913.137	1.083	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.901	99729.107	80506.314	0.836	$2^{21.00}$	$2^{26.00}$	$2^{26.58}$	$2^{27.00}$

Table A.64: Summary: salem-25

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.005	1042.681	274.446	1.041	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
3	0.947	58600.004	26089.673	0.561	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{21.00}$
5	1.079	78860.608	50271.098	0.523	$2^{21.00}$	$2^{27.58}$	$2^{27.00}$	$2^{27.00}$
7	1.097	117154.881	86727.977	0.453	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.893	99888.560	81255.076	▲ **1.368** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$

Table A.65: Summary: salem-26

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1029.486	237.477	▲ **1.334** ▲	$2^{21.00}$	$2^{27.81}$	$2^{21.00}$	$2^{29.46}$
3	1.055	58683.942	26335.316	▲ **1.272** ▲	$2^{21.00}$	$2^{28.46}$	$2^{21.00}$	$2^{28.70}$
5	1.074	77961.523	50287.533	0.817	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.111	116853.156	86122.257	1.123	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{27.58}$
10	1.119	100194.954	81361.350	1.085	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.58}$

Table A.66: Summary: salem-27

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	977.130	253.290	0.853	$2^{21.00}$	$2^{29.17}$	$2^{21.00}$	$2^{28.32}$
3	1.050	58964.544	26154.121	0.720	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
5	1.081	78098.723	50238.096	▲ **1.256** ▲	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{28.00}$
7	0.899	117927.578	86517.967	▲ **1.513** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
10	1.109	100278.997	80937.791	0.770	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.67: Summary: salem-28

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.003	959.891	251.290	0.569	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.00}$
3	0.947	59496.050	26271.712	▲ **1.248** ▲	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{28.70}$
5	1.079	▲ **79053.750** ▲	50301.508	0.766	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{27.00}$
7	0.898	117740.844	86391.396	0.581	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.896	99358.442	80672.116	1.141	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{21.00}$

Table A.68: Summary: salem-29

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1038.251	217.125	0.640	$2^{21.00}$	$2^{21.00}$	$2^{29.00}$	$2^{21.00}$
3	1.050	59363.523	26523.947	0.634	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
5	1.086	78684.097	50056.379	0.730	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$	$2^{26.00}$
7	1.102	118465.305	▲ **87454.459** ▲	0.752	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$	$2^{21.00}$
10	1.110	100193.533	81008.958	▲ **1.242** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$

Table A.69: Summary: salem-30

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	▼ **884.410** ▼	232.130	▲ **1.492** ▲	$2^{21.00}$	$2^{29.46}$	$2^{28.00}$	$2^{29.46}$
3	1.048	59134.715	26539.708	1.031	$2^{21.00}$	$2^{28.32}$	$2^{21.00}$	$2^{28.46}$
5	1.078	77597.130	49890.594	0.589	$2^{21.00}$	$2^{27.81}$	$2^{21.00}$	$2^{21.00}$
7	0.891	117834.175	86517.844	1.220	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{27.58}$
10	1.104	99348.957	80283.455	0.673	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.70: Summary: salem-31

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	946.965	246.726	0.533	$2^{21.00}$	$2^{29.17}$	$2^{21.00}$	$2^{28.17}$
3	1.047	59040.004	26117.649	0.959	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
5	0.926	78089.673	50326.351	▲ **1.279** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.00}$
7	1.101	117656.004	86029.740	1.124	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.890	100521.589	81182.949	0.521	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.71: Summary: salem-32

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	▲ **1174.006** ▲	276.957	▲ **1.455** ▲	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{29.46}$
3	1.051	59168.913	26247.502	▲ **1.239** ▲	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{28.70}$
5	1.077	77776.059	49934.819	1.086	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.103	117782.506	86660.608	▲ **1.658** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
10	1.113	99646.531	80507.537	0.717	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.72: Summary: salem-33

	Total				Index			
Base	f	χ^2	$\nabla\chi^2$	KS	f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1111.112	273.407	1.215	$2^{21.00}$	$2^{29.00}$	$2^{21.00}$	$2^{21.00}$
3	0.948	59554.304	26307.052	1.142	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{21.00}$
5	1.080	78583.303	49758.832	0.700	$2^{21.00}$	$2^{26.00}$	$2^{26.00}$	$2^{21.00}$
7	1.098	117139.331	85997.862	0.774	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$
10	0.892	100248.694	81212.230	0.917	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.73: Summary: salem-34

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1093.320	264.936	1.076	$2^{21.00}$	$2^{29.17}$	$2^{29.00}$	$2^{28.70}$
3	1.056	▼ **58299.013** ▼	26234.174	0.508	$2^{21.00}$	$2^{28.70}$	$2^{21.00}$	$2^{21.00}$
5	0.926	77812.623	50262.600	0.859	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	0.905	116873.686	85732.746	0.613	$2^{21.00}$	$2^{27.58}$	$2^{27.58}$	$2^{21.00}$
10	1.111	99532.638	80598.059	0.669	$2^{21.00}$	$2^{26.00}$	$2^{26.58}$	$2^{27.00}$

Table A.74: Summary: salem-35

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.995	▲ **1171.656** ▲	275.741	0.897	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{29.17}$
3	0.947	59571.254	26220.279	0.576	$2^{21.00}$	$2^{28.58}$	$2^{21.00}$	$2^{21.00}$
5	1.081	78386.320	50126.800	0.827	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{21.00}$
7	1.098	117698.608	86981.355	▲ **1.373** ▲	$2^{21.00}$	$2^{21.00}$	$2^{27.00}$	$2^{27.81}$
10	1.110	99986.211	81118.497	0.689	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.75: Summary: salem-36

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1056.292	281.402	0.669	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{28.32}$
3	0.952	59036.141	26026.186	0.767	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{28.00}$
5	1.075	78456.229	49708.678	0.706	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$	$2^{21.00}$
7	1.098	117841.446	86355.910	1.066	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.58}$
10	1.118	▲ **100984.477** ▲	81478.643	0.878	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$	$2^{21.00}$

Table A.76: Summary: salem-37

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1045.009	245.624	▲ **1.476** ▲	$2^{21.00}$	$2^{29.17}$	$2^{28.17}$	$2^{29.46}$
3	1.053	59280.986	26259.659	▲ **1.983** ▲	$2^{21.00}$	$2^{27.58}$	$2^{21.00}$	$2^{28.70}$
5	0.928	77563.139	49834.953	1.167	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	0.903	118016.936	86807.984	0.914	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.103	100630.099	81412.652	0.739	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.77: Summary: salem-38

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1025.554	247.030	0.647	$2^{21.00}$	$2^{29.00}$	$2^{21.00}$	$2^{21.00}$
3	0.945	58897.856	26271.415	0.740	$2^{21.00}$	$2^{28.17}$	$2^{21.00}$	$2^{21.00}$
5	1.074	77672.475	49888.207	0.701	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	0.902	118081.674	86830.076	▲ **1.794** ▲	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{27.81}$
10	1.112	100283.775	81048.155	▲ **1.429** ▲	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{27.32}$

Table A.78: Summary: salem-39

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1010.366	238.916	0.666	$2^{21.00}$	$2^{28.17}$	$2^{21.00}$	$2^{28.58}$
3	0.949	58570.040	25926.462	0.519	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{21.00}$
5	1.080	78256.549	50031.024	0.826	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{21.00}$
7	0.902	117410.218	86421.176	0.710	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.108	99839.277	81189.565	0.635	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.79: Summary: salem-40

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.006	▲ **1123.307** ▲	255.480	0.918	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{28.32}$
3	0.952	▲ **59931.210** ▲	26581.583	0.967	$2^{21.00}$	$2^{28.70}$	$2^{28.17}$	$2^{21.00}$
5	1.071	77997.405	49891.370	0.772	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	0.905	117259.612	86348.342	0.553	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{21.00}$
10	1.118	100165.394	81347.854	0.677	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.80: Summary: salem-41

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	▼ **911.512** ▼	277.061	0.779	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{28.17}$
3	0.948	58656.347	25986.522	0.770	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{21.00}$
5	1.076	77956.886	49709.246	0.822	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.107	117173.116	85906.129	0.876	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.892	99829.721	81018.800	0.949	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.81: Summary: salem-42

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1107.219	271.137	0.689	$2^{21.00}$	$2^{29.39}$	$2^{28.32}$	$2^{27.00}$
3	0.948	58623.477	26065.557	▲ **1.430** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.70}$
5	0.927	78691.428	▲ **50804.145** ▲	▲ **1.431** ▲	$2^{21.00}$	$2^{21.00}$	$2^{28.00}$	$2^{28.00}$
7	0.899	118516.541	86588.110	0.740	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.119	99942.284	81104.239	▲ **1.730** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$

Table A.82: Summary: salem-43

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	978.039	273.403	0.968	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{28.58}$
3	0.947	58829.271	26261.327	0.911	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.32}$
5	0.924	▼ **77229.010** ▼	49774.052	1.141	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{21.00}$
7	0.899	▲ **118977.122** ▲	▲ **87264.189** ▲	1.019	$2^{21.00}$	$2^{27.81}$	$2^{27.81}$	$2^{26.58}$
10	0.889	100163.768	80894.503	0.569	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.83: Summary: salem-44

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.005	1057.722	235.359	0.839	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{21.00}$
3	1.052	59584.046	▲ **26762.903** ▲	0.920	$2^{21.00}$	$2^{28.00}$	$2^{28.70}$	$2^{28.46}$
5	1.075	78551.475	50243.580	0.819	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{25.00}$
7	1.106	117581.565	86130.216	0.619	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{21.00}$
10	0.900	100433.775	80962.627	0.626	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.84: Summary: salem-45

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	989.825	244.857	0.856	$2^{21.00}$	$2^{26.00}$	$2^{27.00}$	$2^{27.81}$
3	1.053	59490.968	26269.913	0.881	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
5	0.919	77360.199	49627.413	1.120	$2^{21.00}$	$2^{27.81}$	$2^{21.00}$	$2^{27.81}$
7	1.098	117774.669	86847.865	0.681	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.118	99293.241	80375.452	0.684	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.85: Summary: salem-46

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1005.451	268.145	0.865	$2^{21.00}$	$2^{28.32}$	$2^{21.00}$	$2^{26.58}$
3	1.049	58558.005	26326.716	▲ **1.446** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{28.70}$
5	0.929	77956.472	49875.721	0.874	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.113	117096.255	86124.018	0.727	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{26.58}$
10	1.114	99539.290	80735.100	1.005	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.86: Summary: salem-47

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1089.195	281.740	0.959	$2^{21.00}$	$2^{29.17}$	$2^{21.00}$	$2^{29.17}$
3	0.952	59573.886	25821.601	1.186	$2^{21.00}$	$2^{27.32}$	$2^{28.58}$	$2^{27.00}$
5	0.926	77972.337	49909.024	0.700	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.101	▼ **116511.378** ▼	86092.312	▲ **1.279** ▲	$2^{21.00}$	$2^{27.81}$	$2^{21.00}$	$2^{27.81}$
10	1.103	100086.522	80901.164	0.878	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.87: Summary: random-1

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	▲ **1116.952** ▲	284.787	0.512	$2^{21.00}$	$2^{29.46}$	$2^{27.32}$	$2^{21.00}$
3	1.049	58981.153	26212.336	0.589	$2^{21.00}$	$2^{25.00}$	$2^{25.00}$	$2^{26.00}$
5	1.075	77654.690	50033.693	0.844	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	0.900	118246.871	86477.595	0.947	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$	$2^{21.00}$
10	1.109	99784.903	80636.767	0.985	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.88: Summary: random-2

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.995	1112.835	249.485	0.829	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{29.25}$
3	0.951	59281.199	26319.435	0.812	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{25.00}$
5	1.078	77971.292	49996.474	0.768	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.098	117858.020	86919.486	0.798	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{21.00}$
10	1.122	99346.923	80472.592	1.200	$2^{21.00}$	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$

Table A.89: Summary: random-3

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	▲ **1127.423** ▲	▲ **303.905** ▲	0.687	$2^{21.00}$	$2^{29.46}$	$2^{29.46}$	$2^{27.32}$
3	1.052	59674.173	26612.298	0.854	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{21.00}$
5	1.071	78129.156	50006.818	1.154	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{21.00}$
7	1.099	117650.409	86632.553	0.544	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.888	99313.448	80567.408	0.651	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.90: Summary: random-4

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.995	976.019	280.810	0.830	$2^{21.00}$	$2^{28.32}$	$2^{21.00}$	$2^{27.81}$
3	0.948	▼ **58362.742** ▼	25950.792	0.950	$2^{21.00}$	$2^{28.70}$	$2^{28.58}$	$2^{21.00}$
5	1.073	77708.590	50025.649	▲ **1.262** ▲	$2^{21.00}$	$2^{26.58}$	$2^{26.58}$	$2^{28.00}$
7	0.894	117005.325	86113.746	1.151	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.58}$
10	1.104	99792.950	81252.482	0.601	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.91: Summary: random-5

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1084.989	272.370	1.100	$2^{21.00}$	$2^{29.32}$	$2^{21.00}$	$2^{29.32}$
3	1.053	59371.909	26153.426	0.688	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
5	0.929	▲ **78977.918** ▲	50380.044	0.963	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{21.00}$
7	1.119	117733.091	86682.923	▲ **1.460** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
10	1.117	100776.876	81504.203	▲ **1.236** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$

Table A.92: Summary: random-6

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	▼ **834.826** ▼	227.661	0.684	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{21.00}$
3	1.053	59363.747	26142.124	0.750	$2^{21.00}$	$2^{27.00}$	$2^{21.00}$	$2^{27.58}$
5	0.926	77521.945	49909.277	▲ **1.404** ▲	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{28.00}$
7	1.097	▲ **118618.135** ▲	87159.969	0.930	$2^{21.00}$	$2^{27.81}$	$2^{21.00}$	$2^{26.00}$
10	0.895	100050.505	81205.988	0.930	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.93: Summary: random-7

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1032.800	273.843	0.532	$2^{21.00}$	$2^{29.09}$	$2^{21.00}$	$2^{25.00}$
3	1.048	58964.224	26561.236	0.560	$2^{21.00}$	$2^{28.32}$	$2^{21.00}$	$2^{27.81}$
5	1.079	77901.560	49916.141	0.695	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
7	1.108	118311.510	86988.098	0.756	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.00}$
10	1.112	99806.514	80727.482	▲ **1.708** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.32}$

Table A.94: Summary: random-8

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	▼ **897.812** ▼	▼ **213.401** ▼	0.732	$2^{21.00}$	$2^{29.46}$	$2^{29.46}$	$2^{21.00}$
3	0.954	59512.094	26235.420	0.465	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
5	0.923	▼ **77342.319** ▼	49605.799	0.973	$2^{21.00}$	$2^{28.00}$	$2^{27.81}$	$2^{21.00}$
7	1.103	117225.958	86171.368	0.860	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.881	99287.335	81136.218	1.015	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.95: Summary: random-9

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.005	▲ **1133.158** ▲	278.138	▲ **1.317** ▲	$2^{21.00}$	$2^{29.46}$	$2^{21.00}$	$2^{29.46}$
3	0.949	58610.723	26177.688	0.497	$2^{21.00}$	$2^{26.58}$	$2^{21.00}$	$2^{21.00}$
5	1.075	78137.549	49551.570	1.001	$2^{21.00}$	$2^{21.00}$	$2^{27.58}$	$2^{21.00}$
7	0.897	117770.252	86657.684	1.035	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{27.32}$
10	1.103	99946.302	80638.687	0.964	$2^{21.00}$	$2^{25.00}$	$2^{21.00}$	$2^{21.00}$

Table A.96: Summary: random-10

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.997	1000.379	▼ **204.223** ▼	1.058	$2^{21.00}$	$2^{28.17}$	$2^{29.46}$	$2^{29.32}$
3	1.051	59483.422	26370.893	1.050	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
5	0.923	78681.718	50263.685	0.828	$2^{21.00}$	$2^{27.81}$	$2^{25.00}$	$2^{21.00}$
7	1.099	116857.023	85994.213	▲ **1.640** ▲	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
10	1.107	100511.659	81519.102	0.620	$2^{21.00}$	$2^{26.00}$	$2^{26.00}$	$2^{21.00}$

Table A.97: Summary: random-11

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	1005.193	275.962	1.147	$2^{21.00}$	$2^{29.00}$	$2^{28.32}$	$2^{28.00}$
3	1.047	▼ **58327.121** ▼	25936.175	0.710	$2^{21.00}$	$2^{28.70}$	$2^{21.00}$	$2^{21.00}$
5	1.072	78364.968	50026.224	0.881	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.00}$
7	1.111	118400.555	86996.168	1.208	$2^{21.00}$	$2^{27.00}$	$2^{27.00}$	$2^{21.00}$
10	0.883	▲ **101136.873** ▲	81273.579	1.215	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$	$2^{27.00}$

Table A.98: Summary: random-12

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	997.461	225.276	0.989	$2^{21.00}$	$2^{28.00}$	$2^{21.00}$	$2^{28.00}$
3	1.050	59313.011	26354.985	0.912	$2^{21.00}$	$2^{27.58}$	$2^{21.00}$	$2^{27.81}$
5	0.921	77722.794	49611.110	▲ **1.230** ▲	$2^{21.00}$	$2^{26.00}$	$2^{21.00}$	$2^{28.00}$
7	0.905	117558.729	86484.050	0.664	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.00}$
10	1.110	99545.985	80729.531	0.805	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.99: Summary: random-13

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.996	1052.908	250.482	▲ **1.672** ▲	$2^{21.00}$	$2^{29.09}$	$2^{26.58}$	$2^{29.46}$
3	1.051	58414.265	26101.297	0.574	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{27.81}$
5	1.076	78017.406	50342.081	▲ **1.822** ▲	$2^{21.00}$	$2^{21.00}$	$2^{25.00}$	$2^{28.00}$
7	1.103	▲ **118682.664** ▲	87229.881	0.798	$2^{21.00}$	$2^{27.81}$	$2^{21.00}$	$2^{21.00}$
10	1.106	100376.490	81524.365	0.849	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.100: Summary: random-14

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	0.997	▼ **912.158** ▼	▼ **208.543** ▼	▲ **1.579** ▲	$2^{21.00}$	$2^{29.46}$	$2^{29.46}$	$2^{29.46}$
3	0.949	59107.317	26229.336	0.714	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
5	0.910	78571.862	▲ **50709.221** ▲	0.735	$2^{21.00}$	$2^{21.00}$	$2^{28.00}$	$2^{27.00}$
7	0.891	118076.145	86563.628	0.677	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	1.121	99563.780	80692.663	0.770	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$

Table A.101: Summary: random-15

Base	Total f	χ^2	$\nabla\chi^2$	KS	Index f	χ^2	$\nabla^2\chi^2$	KS
2	1.004	944.520	242.509	0.436	$2^{21.00}$	$2^{29.17}$	$2^{21.00}$	$2^{27.00}$
3	0.946	59230.717	26238.127	1.031	$2^{21.00}$	$2^{28.46}$	$2^{21.00}$	$2^{27.00}$
5	1.077	78301.951	49877.966	0.442	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{25.00}$
7	1.101	117504.795	86720.925	0.909	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$	$2^{21.00}$
10	0.896	▲ **101000.980** ▲	81733.464	▲ **1.245** ▲	$2^{21.00}$	$2^{27.32}$	$2^{21.00}$	$2^{27.32}$