Some Observations about Familiar Numbers

Daniel Richardson,
Computer Science, Bath University,
Bath BA2 7AY, UK
email: masdr@bath.ac.uk,
and
Simon Langley,
University of the West of England,
Bristol BS16 1QY, UK
email: Simon.Langley@uwe.ac.uk

March 15, 2002

Abstract

We look at complex numbers which are values of classical closed form expressions, and consider the relationship between the length of an expression and the number of decimal places needed to distinguish its absolute value from zero (if it is non-zero).

1 Introduction

The exp-log expressions are, roughly speaking, those which can be constructed from expressions for natural numbers using the operators $\{+, -, *, /, \sqrt[n]{\cdot}, \exp, \log\}$.

In the section below, the family of exp-log expressions is described and the field of closed form numbers is defined. An expanded form is defined for the exp-log expressions, and the Uniformity Conjecture is stated. This claims that for expressions in expanded form, a small multiple of the syntactic length bounds the number of decimal places needed to distinguish the defined number from zero, if it is non-zero.

Some computational consequences are briefly mentioned. Some possible methods of searching for a counterexample are discussed. One of these methods, essentially picking expressions of a given length at random, is developed in more detail. Distributions of values obtained from such a search are described. No counterexamples have been found. Indeed the main empirical result is that moderate sized expressions which represent non zero numbers can be proved to be non zero using floating point arithmetic at a level of precision proportional to the length of the expression;

on the other hand, all the expressions representing zero which we tested could be proved to be zero in the computer algebra system we used.

2 Closed Form Numbers and the Uniformity Conjecture

2.1 Expressions

We assume, to begin with, some canonical representation for the natural numbers. Then the set of nested radical exponential and logarithmic expressions is the smallest set of expressions so that:

1. All the representations of the non-zero natural numbers are in the set.
2. If A and B are in the set so are $A + B$, $A - B$, and $A * B$, A/B.
3. If A is in the set, so are (A), $-A$, $\exp(A)$ and $\log(A)$
4. If A is in the set and n is a canonical representation of a natural number bigger than 1, then $A^{1/n}$ is in the set.

Each nested radical exponential and logarithmic expression E is either undefined, or is interpreted as a real or complex number $V(E)$, as follows.

1. If E is a representation of an natural number, $V(E)$ is that number.
2. The operators are given the usual precedence in the absence of brackets.
3. If A and B are defined, then $V(A + B), V(A - B), V(A * B)$ and $V(-A)$ are defined with the usual interpretation of the operators. If B is defined, and $V(B)$ is not zero, then $V(A/B)$ is defined, with the usual interpretation. $V((A))$ is the same as $V(A)$.
4. If A is defined, then $V(\exp(A))$ is defined with meaning $e^{V(A)}$.
5. If A is defined, and $V(A) \neq 0$, then $V(\log(A))$ is defined, as the branch of the logarithm base e so that $-\pi < \Im(V(\log(A))) \leq \pi$.
6. If A is defined and $V(A) \neq 0$, then $V(A^{1/n})$ is defined and equal to $V(\exp(\log(A)/n))$.

The operator V is called evaluation.

The complex numbers defined in this way are called *closed form* ([Chow]).

Please notice that although we have expressions here for n^{th} roots, we do not have expressions for n^{th} powers. If we want A^2, for example, we have to write it as $A * A$.

2.2 Length of an expression

Our set of exp-log expressions depends on a choice of canonical representation for the natural numbers. Assume that we have chosen some base b for representation of the natural numbers. We define the length of a number in this representation to be the number of digits base b which are used.

An exp-log expression may be viewed as an expression tree, which has operators on the interior nodes, and representations of the natural numbers on the leaves. We define the length of such an expression tree to be the sum of the number of operators and the sum of the lengths of the representations on the leaves.

$$
\begin{aligned}
\text{length}(A+B) &= \text{length}(A) + \text{length}(B) + 1\\
\text{length}(A-B) &= \text{length}(A) + \text{length}(B) + 1\\
\text{length}(-A) &= \text{length}(A) + 1\\
\text{length}(A*B) &= \text{length}(A) + \text{length}(B) + 1\\
\text{length}(A/B) &= \text{length}(A) + \text{length}(B) + 1\\
\text{length}((A)) &= \text{length}(A)\\
\text{length}(\sqrt[n]{A}) &= \text{length}(A) + \text{length}(n) + 1\\
\text{length}(\exp(A)) &= \text{length}(A) + 1\\
\text{length}(\log(A)) &= \text{length}(A) + 1\\
&.
\end{aligned}
$$

So, for example, in decimal notation, $4 - 3 * (10)^{1/8}$ would have length 8, since it has 5 digits and 3 operator symbols.

2.3 Gap Functions

Definition 1. *A gap function for the closed form numbers is a function $g : \boldsymbol{Exp} \to \boldsymbol{R}_+$, where $\boldsymbol{Exp}$ is the set of exp-log expressions, so that if $E \in \boldsymbol{Exp}$ and $V(E) \neq 0$ then $|V(E)| > 10^{-g(E)}$*

A gap function tells us the amount of decimal precision which is needed to distinguish a non-zero number from zero. Of course gap functions exist. We hope that there is a computable gap function, and even an easily computable one.

We have a natural scale, i.e. length, on expressions; and we also have a natural scale, i.e. the logarithm of the absolute value, on the complex numbers. We would like to know: How does the evaluation operator, V, behave with respect to these two scales?

2.4 Uniformity Conjecture

Using iterated exponentiation, it is possible to define very large numbers. Since we have division, it is also possible to define very small numbers with expressions involving iterated exponentiation, followed by division. There does not seem to be any other way to get very large numbers, or very small non-zero numbers. (Note that although we allow n^{th} roots, we do not have n^{th} powers. So we can not easily write a short expression for the result of a sequence of repeated squarings, for example.)

We will say that an expression E is in expanded form if, for any exponential subexpression $\exp(A)$ of E, we have $|V(A)| \leq 1$ (E is considered a subexpression of itself).

It appears that it is not possible to define very large numbers, or very small non-zero numbers using small expressions in expanded form.

This somewhat vague idea led to the original statement of the uniformity conjecture, as given below. See [Richardson 2000].

Uniformity Conjecture: *If E is an expression in expanded form, and $V(E) \neq 0$, then $|V(E)| \geq 1/N(k)$, where $k = \mathrm{length}(E)$, and $N(k)$ is the number of syntactically correct expanded form expressions of length $\leq k$.*

If we have S symbols for operators and digits in our alphabet, the number of syntactically correct expressions of length $\leq k$ is bounded by S^k, since we can encode expressions of length less than k as expressions of length k by padding on the right with non interpreted operator symbols. In case we use decimal notation, for example, S would be 17. In binary notation, S would be 9.

Assuming decimal notation, $10^k < N(k) < 10^{1.24k}$, since $1.24 > \log_{10}(17)$. Suppose $k > 4$. Consider a string of symbols of length k with digits on the ends, and digits at all the odd numbered positions and either digits or operators $+, -, *$ at interior even numbered positions. There are 13 choices for symbols at these interior even numbered positions, and 10 choices for the others. Any such string is syntactically correct. So, assuming $k > 4$, we get $10^{1.03k} < N(k) < 10^{1.24k}$.

Roughly speaking, the conjecture says that the amount of base S precision which is needed to discriminate the value of an expanded form expression from zero is proportional to the length of the expression.

The quantity $N(k)$ is not very nice to work with. So, in the following, we will consider:

Modified Uniformity Conjecture: *If E is an expression in expanded form, and using decimal notation for natural numbers, and $V(E) \neq 0$, then*

$|\log_{10}(|V(E)|)| \leq Ck$

where k is the length of E, and $C = \log_{10}(17)$.

We will say that a number occurs at level k if it has an exp-log expression representing it with length no more than k.

Example: the famous $3 * \log(640320)/\sqrt{163} - \pi$ is only zero to 15 decimal places, whereas its syntactic length is 15 plus the length of π. We can replace π with $\log(-1)/\sqrt{-1}$ with length 8.

The Uniformity conjecture is related to a family of conjectures, called Witness conjectures, originating with Joris van der Hoeven. See [van der Hoeven 97], [VDH2000]. It is reasonable to say that the Uniformity Conjecture is a special type of Witness conjecture.

It is also related to the concept of transcendence measure, which is discussed in the next section.

3 Transcendence Measure

If p is a polynomial with integral coefficients not all zero, we define $h(p)$, the logarithmic height of p, to the maximum of the logarithms of the absolute values of the coefficients.

A transcendence measure for a complex number x is usually defined to be a function $g(h,d)$ so that for all polynomials $p \in \boldsymbol{Z}[x]$ which are not identically zero and which have degree no more than d and logarithmic height no more than h, we have

$\log|p(x)| \geq -g(h,d)$

See [Lang], [Baker], [Waldschmidt], [Sprindzhuk]. A related notion is independence measure. If complex numbers $x_1,\ldots,x_n$ are algebraically independent, an independence measure for $x_1,\ldots,x_n$ is a function $g(h,d)$ so that for all polynomials $p \in \boldsymbol{Z}[x_1,\ldots,x_n]$ which are not identically zero and which have degree no more than d and logarithmic height no more than h, we have

$\log|p(x_1,\ldots,x_n)| \geq -g(h,d)$

Results of this kind can be obtained in an almost mechanical way by making effective versions of proofs of algebraic independence. See [Sert].

We can also define a somewhat broader notion of measure, as follows.

Let F be a field, which is a subfield of the complex numbers. We define a length function for F to be a function $\lambda : F \rightarrow \boldsymbol{N}$ so that:

1. $\lambda(1) = 1$
2. $\lambda(z+w) \leq \lambda(z) + \lambda(w) + 1$
3. $\lambda(z-w) \leq \lambda(z) + \lambda(w) + 1$
4. $\lambda(z*w) \leq \lambda(z) + \lambda(w) + 1$
5. $\lambda(z/w) \leq \lambda(z) + \lambda(w) + 1$

Definition 2. *We define a measure for F with respect to length λ to be a function $g : \boldsymbol{N} \rightarrow \boldsymbol{N}$ so that if $x \in F$ and $x \neq 0$, then*

$$|x| > 10^{-g(\lambda(x))}$$

We can relate this back to the uniformity conjecture in the following way. Let F be the closed form numbers, and suppose $x \in F$. Let E be an expanded form expression of minimal length so that $V(E) = x$. Define $\lambda(x)$ to be the length of E.

4 Computational Consequences

Most algorithms involving the real numbers involve branches depending on tests for equality or inequality of real numbers. In order to carry out such a test it is enough to be able to decide whether or not a defined number, x, is equal to zero. However, this may not be obvious. The solution adopted in most scientific programming is to find a good approximation to x, and if it seems to be zero to proceed as if it actually were zero, and then to attach a small warning about possible unreliability to the results. This is, so to speak, an engineering solution; to a mathematician it seems somewhat careless. After all, we would like to be able to believe the results of scientific programming. Of course the same problem occurs in all computer algebra systems. Within the domain of exp-log expressions, the uniformity conjecture would give us a principled way of applying the engineers' much ridiculed technique. Possibly the engineers were essentially correct all the time.

Another application would be to the problem of deciding whether or not a polynomial in $\boldsymbol{Z}[x_1, \ldots, x_n]$, given in some non-canonical way, for example as an expression tree, is identically zero. Let $\alpha_1, \ldots, \alpha_n$ be the reciprocals of the square roots of the first n prime numbers. By Lindemann's theorem, $e^{\alpha_1}, \ldots, e^{\alpha_n}$ are algebraically independent. If $p(x_1, \ldots, x_n)$ is in $\boldsymbol{Z}[x_1, \ldots, x_n]$, then p is identically zero if and only if $p(e^{\alpha_1}, \ldots, e^{\alpha_n}) = 0$. If we can compute a bound on the length of an expression for $p(e^{\alpha_1}, \ldots, e^{\alpha_n})$, and we can approximate this as a real value, we can use the uniformity conjecture to decide whether or not it is zero. Another method of solving this problem is to choose integral values at random, substitute these for the variables and evaluate. The uniformity conjecture would allow us to replace the random choice by suitably high precision approximation to fixed independent numbers. Note that high precision approximation to closed form numbers is not very difficult computationally. See [Borwein].

Another application of this sort is to the problem of determining the rank of an matrix with elements in a field, using parallel computation. See [Mulmuley]. The following simple algorithm is given by K. Mulmuley. Suppose A is square, symmetric, and n by n. (Mulmuley gives a transformation which reduces any case to this one.) Let X be a diagonal matrix in an indeterminate x such that $X_{ii} = x^{i-1}$, $(1 \leq i \leq n)$.

1. compute $Q(t) = det(tI - XA)$.
2. return $n - m$, where m is the highest degree such that t^m divides $Q(t)$.

The algorithm takes $O(\log^2 n)$ time using $O(n^{4.5})$ processors.

Mulmuley remarks that we can eliminate the indeterminate x and thus gain in speed by choosing for it a random value in the field containing the matrix entries, or in a suitable algebraic extension of it. If the field is a finitely generated subfield of the closed form numbers we could also, relying on the uniformity conjecture, replace x by a closed form number ξ which is transcendental over the field, and then replace both ξ and the field elements by suitably high precision floating point approximations. For an integral matrix, for example, we could take $\xi = e$. This idea will be developed in a subsequent article.

5 Looking for counterexamples

Assume decimal representation. Let E range over expressions in expanded form, with $V(E) \neq 0$.

The modified uniformity conjecture states that
$\sup\{|\log_{10}(|V(E)|| \,/\, \mathrm{length}(E)\} \leq \log_{10}(17)$.

Even if we do not believe this is plausible, we would like either to show that this quantity is unbounded or to find a good upper bound for it.

Define $\rho(E) = |\log_{10} |V(E)| \,/\, \mathrm{length}(E)|$

Definition 3. *We will say that an expression E is dominant if E is in expanded form, $V(E) \neq 0$, and $\rho(E) \geq \rho(D)$ for all D in expanded form with* $\mathrm{length}(D) \leq \mathrm{length}(E)$.

We will define $\rho^*(k)$ to be the supremum of $\rho(E)$ for E of length $\leq k$, and, as usual, E in expanded form and $V(E) \neq 0$. So, for example, $\rho^*(1) = \log_{10}(9)$ and $\rho^*(2) = (1/2)\log_{10}(99)$. In general, $\rho^*(k) \geq (1/k)\log_{10}(10^k - 1)$.

It is easy to establish the following.

Lemma 1. *1.* $10^{-\rho^*(k)k} \leq |V(E)| \leq 10^{\rho^*(k)k}$, *where E is in expanded form of length no more than k and $V(E) \neq 0$*

2. *Expressions of the following form are not dominant:* $\exp(A), -A, 1/A, A * B, A/B, \sqrt[n]{A}$.

Proof.

Since $\rho(E) = |\log_{10}|V(E)|\,/\,\mathrm{length}(E)|$,

$10^{-\rho(E)\,\mathrm{length}(E)} \leq |V(E)| \leq 10^{\rho(E)\,\mathrm{length}(E)}$, and so the first part follows. The second part is equally easy. e^A is not dominant, since dominant expressions have to be in expanded form. $A * B$ is not dominant since

$\rho(A * B) = |\log_{10}|V(A)V(B)||\,/(\mathrm{length}(A) + \mathrm{length}(B) + 1)$

and this is less than the maximum of $\rho(A)$ and $\rho(B)$.

□

So the only possibilities for dominant expressions are natural numbers and expressions of the form $\log(A), A + B, A - B$. Also, if E is dominant, and not a natural number, it is so by being small, rather than large. That is,

$|V(E)| < 10^{-\rho^*(k-1)k}$

where $k = \mathrm{length}(E)$.

The supremum of ρ is clearly at least 1. According to the modified uniformity conjecture, this supremum is no more than $\log_{10}(17)$. It can be shown that if α is a real irrational closed form number, and there are arbitrarily large partial quotients in the continued fraction expansion for α, then there is an expression E of the form $\alpha - p/q$ so that $\rho(E) > 1$. For example, we could take $\alpha = exp(1/9)$. The expressions constructed in this way are very large, of length at least a million in all cases we know, and are never, as far as we can determine, counterexamples to the uniformity conjecture. Although such expressions exist, we have not actually constructed one of them, one of the problems being the very high precision floating point arithmetic which would be needed. So far we have not encountered any moderate sized examples of expressions E with $\rho(E) > 1$.

5.1 Where to look for small numbers?

We are still thinking about a good way to organise this search. So far we have considered the following possibilities.

1. Exhaustive search at some level. In this case we would generate all expressions up to level k, for some small k. We would decide which expressions are defined, which ones represent zero, and which ones represent small non-zero values, and search for counterexamples among these. This has been done for $k = 5$ and $k = 6$, and the results reported in [Richardson3] . No counterexample was found.

2. Search for counterexamples of the form $\alpha - p/q$, where p/q is a rational approximation to irrational α found by continued fractions. This has been done for $\alpha = n^{1/m}$, and also for $\alpha = e^{n/m}$ and also for $\alpha = \log(n/m)$ for n and m natural numbers between 2 and 100, and the first 50 partial quotients in the continued fraction approximation. The results are reported in [Richardson3]. Again no counterexample was found. Some small numbers are reported, for example, $2^{1/8} - 494/453$. We are grateful to a referee who mentions the small number $109^{1/5} - 23/9$, found by Reyssat, and refers us to [Nitaj], which discusses the relationship between this kind of approximation and the abc conjecture.
3. Search for counterexamples among linear forms
   $a_1\alpha_1 + \cdots + a_n\alpha_n$
   where $a_1, \ldots, a_n$ take integer values, and $\alpha_1, \ldots, \alpha_n$ are closed form numbers which are linearly independent over $\boldsymbol{Q}$. The LLL algorithm or PSLQ can be used for this. See [Bailey], [De Weger]. Some trials of this kind have been done, but no counterexamples have been found. In the case of linear forms in logarithms, there is, again, a relationship with the abc conjecture. See [PP], [PP2].
4. Search for counterexamples of the form $F(1)$ where $F(x)$ is a function which has been constructed to have a zero of high multiplicity at 0. Techniques for constructing such functions are well developed in the theory of transcendental numbers. For example, one of the proofs of Lindemann's theorem begins by setting
   $F(x) = p_1(x)e^{\alpha_1 x} + p_2(x)e^{\alpha_2 x} + \cdots + p_m(x)e^{\alpha_m x}$, where $p_1(x), \ldots, p_m(x)$ are polynomials with unknown integral coefficients with degree bounded by n. Siegel's lemma, or some other method, is used to determine integral values of these coefficients so that $F(x)$ is not identically zero but has a zero of multiplicity approximately $nm/2$ at zero. As before, the end result is a linear form $F(1)$ with small absolute value.

It seems that some more empirical investigation of the distribution of values might help to direct the search.

6 Empirical Investigations

6.1 Methodology

The first search method described above, i.e. exhaustive generation of all expressions, is feasible only for very short expression lengths (especially when done within a computer algebra system). Our most recent approach has been to split the problem into two parts: rapidly generate many random expressions (typically 10^6/run) of a given length (currently between 5 and 30) and then use a computer algebra system (MuPAD) to process 'interesting' expressions.

We wrote a C++ program, `xpr` for the first part (all the software described here can be downloaded from `www.cems.uwe.ac.uk/~slangley`).

`xpr` generates large numbers of expressions with numbers and operators appearing with appropriate frequencies. The program was used in three ways: to obtain statistical information on the distribution of absolute values mod1 of expressions of various lengths (where $x \bmod 1 \equiv x - \lfloor x \rfloor$), to find 'small' valued expressions and to find expressions which appear to be zero.

For lengths in the range 10 - 30, a typical run would produce 1,000,000 expressions in a few seconds (on a 1GHz PC). Usually about a quarter of these would contain subexpressions with form A/E or $\log(E)$ where $V(E) = 0$ or fail to be in expanded form (i.e. including $\exp(E)$ where $|V(E)| > 1$).

Double precision complex numbers were used to evaluate the expressions and no attempt was made to adjust for rounding errors. Given the number of expressions generated it was assumed that any errors would be as randomly distributed as the values themselves.

We are interested in very small values and there is a possibility that these will be just the points where floating point errors become significant. Also though double precision floating point values can be as small as $\approx 10^{-300}$, the machine-ϵ (the smallest x such that $1 + x \neq 1$) is only $\approx 10^{-16}$ so computing values $\bmod 1$ is suspect.

To check the significance of this, several tens of thousand of expressions which were small mod 1 were fed into MuPAD and symbolically simplified. No anomalies were been found. Expressions with a floating point value of zero or close to machine-ϵ were always found via MuPAD to be identically zero.

In developing and testing the program we have also checked hundreds of small or probably zero expressions by eye. This is an un-exciting activity. Very few expressions required even pencil and paper to check and no even slightly interesting identities were found, a factor of $(1 - 1)$ in an expression probably accounts for most zeros.

6.2 Generating Random Expression

The program generates expressions top-down. To produce an expression of length n it first selects the topmost node of the expression tree: a number of length n, a root, a binary of unary operation and then fills in the lower level nodes recursively.

To compute the probability of a given type of top node requires computation of the number of expressions of a given length and the number of expressions with the given operator at the top node. To do this we used the following recurrences relations where $\mathcal{E}(n)$ is the number of expressions of length n, $\mathcal{N}(n)$ is the number of positive integers (base b) of length n and $\mathcal{U}(n)$, $\mathcal{B}(n)$ and $\mathcal{R}(n)$ are the number of expressions of length n in which the outer most operator is unary (one of exp, log or $-$), binary (one of $+, -, \times, \div$) or a root ($\sqrt{\cdot}$) respectively.

$$\mathcal{N}(n) = \begin{cases} 0 & \text{if } n = 0 \\ (b-1)b^{n-1} & \text{otherwise} \end{cases}$$

$$\mathcal{U}(n) = \begin{cases} 0 & \text{if } n < 2 \\ 3\mathcal{E}(n-1) & \text{otherwise} \end{cases}$$

$$\mathcal{B}(n) = \begin{cases} 0 & \text{if } n < 3 \\ 4\sum_{i=1}^{n-2} \mathcal{E}(i)\mathcal{E}(n-1-i) & \text{otherwise} \end{cases}$$

$$\mathcal{R}(n) = \begin{cases} 0 & \text{if } n < 3 \\ (b-2)\mathcal{E}(n-2) + \sum_{i=2}^{n-2} \mathcal{N}(i)\mathcal{E}(n-1-i) & \text{otherwise} \end{cases}$$

The slightly different form of $\mathcal{R}(n)$ reflects the requirement that in $\sqrt[s]{e}$, $s > 1$ and so there are $(b-2)$ single digit roots. Combining these gives

$$\mathcal{E}(n) = \mathcal{N}(n) + \mathcal{U}(n) + \mathcal{B}(n) + \mathcal{R}(n)$$

6.3 Results

For these experiments we used numbers in base 10 and generated random sample expressions of length 7, 11, 13, 17, 19, and 23. Each run produced 10^6 expressions of which about 7×10^5 appeared to be valid (in expanded form and not containing subexpressions equivalent to either $1/0$ or $\log(0)$). (The percentage of valid expressions appears remarkably consistent, one sequence of six runs produced 700616, 700560, 700905, 700054, 700484 and 700053 valid expressions. This initially lead us to doubt the randomness of the expression generator but we have found no faults.) The results are similar for all sizes and so we will illustrate them with a single example. The example in Fig. 1 shows the percentage of values mod1 in each percentile for about 6×10^5 expressions (i.e. 1% on the Y scale is about 6000 values) of length 17.

(This particular run produced an identity which was slightly less trivial than most: $\log(4) + \log(1/(6 * \exp(\log(1 * 2/\sqrt{9})))) = 0$.)

This plot is typical (though as the expression size increases the number of 'spikes' increases while their relative height decreases). For very short expression lengths (say < 5) there are zero height sections between the spikes.

The overall shape is not surprising. Consider the possible mod1 values for expressions of length 3. You can form fractions (with single digit denominators), roots (from 2 to 9) of single digit values, log of two digit numbers, $-\exp(1)$ and $\exp(-1)$. Most of these correspond to visible spikes on the graph. Now consider expressions of length 17 which can be written in the form $A \pm B$ where $\text{length}(B) = 3, \text{length}(A) = 13$ and $V(A) \in \boldsymbol{Z}$. There are $> 13^{13}$ integer expressions obtained by using only the operators $\{+, -, *\}$, all divisions integers by their factors, numbers with exact roots, expressions of the form $\exp(\log(A))$ or $\log(\exp(A))$ where $V(A) \in \boldsymbol{Z}$ and so on.

The spikes blot out other information which might be of interest. There are many ways of getting the value 0.5 - does the spike correspond

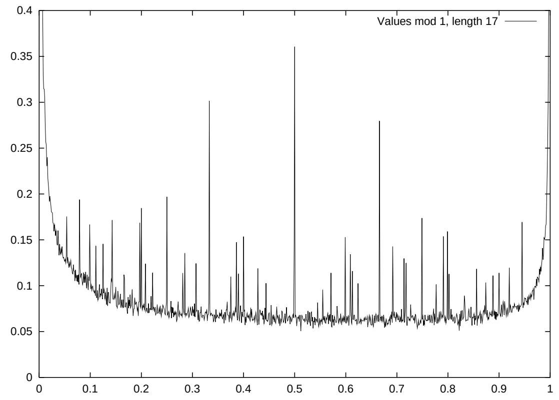

Figure 1: Distribution of values mod1

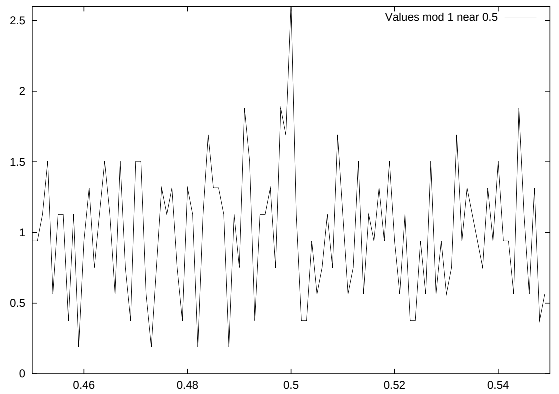

Figure 2: Distribution near 0.5 with duplicates removed

to values equal to 0.5 or are many different values clustered nearby? From the sample we extracted all values between 0.45 and 0.55 and eliminated all duplicates (strictly, rational duplicates were removed and floating point values which differed by less than 10^{-50}). This left 530 numbers. The plot in Fig. 2 shows their distribution with the Y scale being percentage frequency (1% represents about 5 numbers).

The central peak still appears significant but statistical analysis is required to decide if the rest of the graph is random.

We are also interested in measuring how close adjacent values are. If the Uniformity Conjecture is correct, the difference of two values corresponding to expressions with length 17 should be $\geq 10^{-1.24\times 35} \approx 10^{-44}$. Using the first 10,000 values of the same data we removed duplicates, sorted the resulting data and computed (to 50 bit precision) the differences between adjacent values.

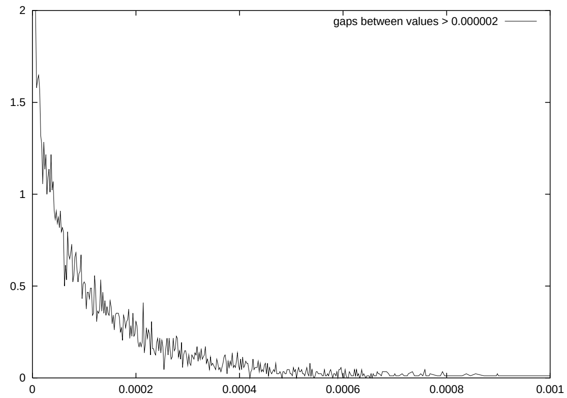

Figure 3: Intervals between values

The smallest gap found was $\approx 5 \times 10^{-20}$ and all differences after that were $\geq 10^{-14}$. About 5% of differences lay in the range $[10^{-14}, 2 \times 10^{-6}]$, larger gaps tail off rapidly after this (Fig. 3).

7 Conclusions

This article is intended to provoke questions or to elicit suggestions rather than to present conclusions. We still do not know how to organise a search for counterexamples to the uniformity conjecture. There seems to be some interesting mathematical structure in this area which is not understood. Computer algebra is a necessary tool for extending our understanding.

In our experiments we have generated millions of expressions of different sizes. In every run the smallest value has been either a simple rational number or an expression which is obviously dominated by a rational of the same length (e.g. $\log(2)/9938578$ as a length 10 expression). Generally, small non-zero values are very rare. When generating expressions of length L about 0.1% of non-zero values are $\leq 10^{-L+3}$ and though values $< 10^{-L+2}$ trivially exist we've never seen one.

Zero valued expressions are statistically rare, consistently being about 0.2% of the total. No nontrivial identities have been detected 'by eye' and in every case tested, now running to tens of thousands, where an expression has a value close to machine-ϵ or a value whose absolute value is below what is predicted by the Uniformity Conjecture, then MuPAD has shown it to be zero.

All the experiments we have done support the Uniformity Conjecture. However, if it is not correct we would be very pleased to hear of a counterexample! In the absence of that, we would be interested to hear of examples of expressions E of moderate size, so that E could be written on one sheet of paper, but E defines a non zero number, and $\rho(E) > 1$.

8 References

References

[Bailey] Helaman R. P. Ferguson, David H. Bailey, and Steve Arno, Analysis of PSLQ, An Integer Relation Finding Algorithm, preprint. See www.nersc.gov/ dhb.

[Baker] A. Baker, *Transcendental number theory*, CUP, 1975

[Borwein] J. M Borwein and P. B. Borwein, *Pi and the AGM*, John Wiley, Canadian Mathematical Society, 1987

[Chow] T. Y. Chow, What is a Closed-Form Number?, American Mathematical Monthly, Vol 106, No 5, 1999, pp 440-448

[De Weger] B. M. M. De Weger, Solving exponential Diophantine equations using lattice basis reduction algorithms, J. Number Theory 26 (1987) 325-367.

[HJLS] J. Håstad, B. Just, J.C. Lagarias, C.P.Schnorr, Polynomial time algorithms for finding integer relations among real numbers, SIAM J. Comput. 18 (1989). pp 859-881.

[HS] M. Hindry and J. Silverman, Diophantine Geometry, An Introduction, Springer Graduate Texts in Mathematics, 2000

[HST] E. Hlawka, J. Schoissengeier, R. Taschner, Geometric and Analytic Number Theory. Springer Verlag, 1991

[Lang] S. Lang, Introduction to Transcendental Numbers, Addison Wesley, 1966

[Mulmuley] K. Mulmuley, A fast parallel algorithm to compute the rank of a matrix over an arbitrary field, Combinatorica 7(1) (1987) pp 101-104

[Nitaj] , A. Nitaj, La Conjecture *abc*, Enseign. Math. 42 (1996), 3-24

[Philippon] P. Philippon, Sur des hauteurs alternatives, I, Math Ann. 289 (1991), pp 255-283; II Ann. Inst. Fourier 44 (1994), pp 1043-1065; III, J. Math. Pures Appl. 74 (1995), pp 345-365

[PP] P. Philippon, Quelques remarques sur des questions d'approximation diophantine, Bull Austral. Math. Soc., Vol 59(2), (1999), pp 323-334.

[PP2] P. Philippon, Addendum a quelques remarques sur des questions d'approximation diophantine, Bull Austral. Math. Soc., vol 61(1), (2000), pp 167-169.

[Richardson] Daniel Richardson, How to Recognise Zero, J. Symbolic Computation (1997), 24, pp 627-645

[Richardson2] Daniel Richardson, Multiplicative Independence of Algebraic Numbers and Expressions, Mega2000 conference Bath June 2000, also in Journal of Pure and Applied Algebra 164, 2001, pp 231-245, ISSN 0022-4049

[Richardson 2000] Daniel Richardson, The Uniformity Conjecture, Proceedings of Computability and Complexity in Analysis, CCA2000, September 17-19, Swansea, Wales. Also in*Springer lecture notes in Computer Science, Volume 2064*, ISBN 3-540-42197-1, pp 253-272.

[Richardson3] Testing the Uniformity Conjecture. www.bath.ac.uk/~masdr/testu.dvi, submitted for publication.

[Sprindzhuk] V. G. Sprinkzhuk, Achievements and Problems in Diophantine Approximation Theory, Russian Mathematical Surveys 35 (4), 1980, pp 1-80

[Sert] A. Sert, Une version effective du theoreme de Lindemann-Weierstrass par les determinants d'interpolation, Journal of Number Theory 76, pp 94-119 (1999)

[van der Hoeven 97] Joris van der Hoeven, Automatic Asymptotics, Ph.D. thesis, Ecole Polytechnique, 1997.

[VDH2000] Joris Van Der Hoeven, Zero-testing, witness conjectures and differential diophantine approximation, Preprint

[Waldschmidt] Michel Waldschmidt, Diophantine approximation on linear algebraic groups, Springer, Grundlehren der mathematischen Wissenschaften 326, ISBN 0072-7823