A practical deterministic method for zero recognition of polynomial terms

Daniel Richardson,
Computer Science, Bath University,
Bath BA2 7AY, UK
email: masdr@bath.ac.uk,
and
Ahmed El-Sonbaty,
email: mapaaeee@bath.ac.uk
Computer Science, Bath University

January 3, 2003

Abstract

The uniformity conjecture postulates a relationship between the syntactic length of expressions built up from the natural numbers using field operations, radicals and exponentials and logarithms, and the smallness of non zero complex numbers defined by such expressions. The uniformity conjecture claims that if the expressions are written in an expanded form in which all the arguments of the exponential function have absolute value bounded by 1, then a small multiple of the syntactic length gives a bound for the number of decimal places needed to distinguish the defined number from zero.

A polynomial term is a tree with operators $*, +, -$ on the interior nodes and natural numbers and variables on the frontier. We decide whether or not such a tree represents the zero polynomial by substituting algebraically independent real numbers for the variables and approximating the resulting constant with the accuracy which we believe, according to the uniformity conjecture, is sufficient to distinguish it from zero if it is not zero.

The result is that if a polynomial term has $s(T)$ nodes, and the natural numbers on the frontier have logarithmic height bounded by $h(T)$, then the bit complexity of deterministic zero recognition is bounded by $O\left(s(T)M\left(s(T)^2h(T)\right)\right)$, where $M(n)$ is the bit complexity of multiplication of two n digit natural numbers.

1 Introduction

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

In the section below, the family of nested radical exponential-logarithmic expressions, (exp-log expressions for short), 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.

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 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 $-A$, $\exp(A)$ 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 natural 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.
4. If A is defined, then $\exp(A)$ is defined with meaning e^A.
5. If A is defined, and $V(A) \neq 0$, then $\log(A)$ is defined, as the branch of the logarithm base e so that $-\pi < im(\log(A)) \leq \pi$.
6. If A is defined and and $V(A) \neq 0$, and n is a canonical representation of a natural number bigger than 1, then $A^{1/n}$ is defined and equal to $exp(log(A)/n)$.

The operator V is called evaluation.

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

A field with good closure properties including the closed form numbers is the field of elementary numbers. These are numbers of the form $q(\alpha)$, where q is in $\mathbf{Q}[x_1, \cdots, x_n]$, and $\alpha \in \mathbf{C}^n$ is a non singular solution of a system of equations $(p_1, \ldots, p_n) = 0$ and each p_i is in $\mathbf{Z}[x_1, \ldots, x_n, e^{x_1}, \ldots, e^{x_n}]$. It has been shown that this is an effective field i.e., equality is decidable if the Schanuel conjecture is true. See [Richardson97], [Richardson2001].

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 nested radical exponential and logarithmic expressions depends on a choice of canonical representation for the natural numbers. Assume in the following that we represent the natural numbers in the usual way base 10. We define the length of an natural number in this representation to be the number of digits base 10 which are used.

Each exp-log expression may be considered as a tree with representations of natural numbers on the frontier and operators among $\{+, -, *, /\,, \sqrt[n]{\ }, \exp, \log\}$ on the interior nodes. We allow $-$ to have arity either 1 or 2. The radical sign has arity 2, and one of its arguments must be a natural number in canonical form. We define the length of an expression to be the sum of the number of interior nodes, i.e. the number of operators, and the sum of the lengths of the representations of natural numbers on the frontier. We use $length(E)$ to denote the length of expression E.

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 $g : Exp \to \mathbf{R}_+$, where Exp is the set of nested radical exponential and logarithmic expressions, so that if x is a closed form number represented by an expression A, and $x \neq 0$ then $\mid x \mid > 10^{-g(A)}$*

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 gap function.

An important question is : How does the evaluation operator, V, behave with respect to the two natural measures we have, namely, the length of an expression and the logarithm of the absolute value of complex numbers ? For some more discussion of this question, (with a slightly different definition of length) one can refer to [Richardson2000].

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 nth roots, we do not have nth powers. So we can not easily write a short expression for the result of a sequence of repeated squarings, for example.)

Definition 2 *We consider an expression E to be a subexpression of itself. We will say that an expression E is in expanded form if for any exponential subexpression e^A of E, we have $\mid V(A) \mid \leq 1$.*

Uniformity Conjecture: *If E is an expression in expanded form, and $V(E) \neq 0$, then $\mid V(E) \mid$ is bigger than $10^{-C\ length(E)}$, where C is some constant between 1 and 2.*

C is called the uniformity constant. If a base other than 10 were used, this would be reflected in a different uniformity constant.

Roughly speaking, the conjecture says that the amount of base 10 precision which is needed to discriminate the value of an expanded form expression from zero is proportional to the length of the expression. If we define $\rho(E) = |log_{10}|V(E)||\,/length(E)$, the conjecture says that $\rho(E) \leq 2$ for E in expanded form. We can prove that there exist expressions E in expanded form with $\rho(E) > 1$, but we have not been able to find any explicit example.

For discussion of the evidence supporting this conjecture, and the background of related ideas, see [Richardson2000], [Richardson2001a], [Richardson and Langley].

As an example, consider the famously small $3 * log(640320)/\sqrt{163} - \pi$. This is only zero to 15 decimal places, whereas its syntactic length is 16 plus the length of π. We can substitute $\pi = log(-1)/(-1)^{1/2}$ in which case the length of π would be 8.

3 Polynomial Terms

Consider representation of polynomials as trees with $+, -, *$ on the interior nodes and with variables and natural numbers on the the leaves. Such a representation is significantly more succinct than any of the usual canonical representations of $\mathbf{Z}[x_1, \ldots, x_n]$. On the other hand, it is much weaker than the standard straight line program representation. It is known that there is a probabilistic polynomial time method for zero recognition of straight line programs. See [Ibarra and Moran]. On the other hand, there is no known deterministic polynomial time solution for this problem. This question was discussed in in the PhD thesis of Bill Naylor [Naylor], who constructs GCD for polynomials represented as SLP (straight line programs). In this connection, see also [Kaltofen]. In general, the zero recognition problem for non canonical representations of polynomials is of much interest in computer algebra. See [Zippel] for a discussion. In this article we solve it by substituting algebraically independent numbers in place of the variables and assuming the Uniformity Conjecture. If k is the length of the resulting nested exp-log expression we could recognise whether the given polynomial was zero by approximation with decimal precision $2k$. The resulting algorithm is not only polynomial time in theory, it is also feasible in practice. It compares very well in practice with other known deterministic algorithms.

For example, let $q(x)$ be some univariate polynomial with integer coefficients. Let $p_n(x_1, \ldots, x_n) = \prod_{i=1}^{n} q(x_i)$. The length of p_n increases only linearly with n. If we are given the fact that p_n has degree less than d in each variable, we could deterministically verify that p_n was not the zero polynomial by evaluating it on the cross product $(\{1, 2, ..., d\})^n$. We need d^n elements in the product to avoid all the roots. It seems therefore that the deterministic method of zero recognition by substitution of integers uses a number of substitutions which must increase exponentially with the number of variables. However, if we can compute with algebraically independent numbers, we only need one substitution. This gives us an extra motivation for learning more about how to compute with classical non algebraic numbers.

A related probabilistic method which can be used is independently to pick n integers $(x_1, \ldots, x_n)$ at random in the range $[0, d * n * N]$, where d is an upper bound on the degree and N is some large multiplier, and then substitute these integers into the polynomial and evaluate. If N is chosen sufficiently large, we can make the probability of an error (caused by accidentally choosing a root of the polynomial) very small, in the order of $(1/N)$, provided that the polynomial is independent of the random number generator. See [Zippel]. Thus if we want the probability of an error to be, for example, less than 10^{-100}, we need to evaluate the polynomial with n arguments of length no more than $100 + log(n) + log(d)$. Suppose that the integers on the frontier have logarithmic height no more than $h(T)$. Assume that there are no more than $s(T)$ nodes in the tree. Thus the largest integer in the computation has length no more than $O(s(T) * Max(h(T), 100 + log(n) + log(d)))$. There are $s(T)$ computations to be done. So the total complexity is bounded by $O(s(T)M(s(T)Max(h(T), 100 + log(n) + log(d))))$, where $M(k)$ is the complexity of multiplication of two numbers of length k. This probabilistic technique is faster than the deterministic method described below by a factor which is approximately $M(s(T)^2h(T))/M(s(T)h(T))$.

4 Algebraic Independence

Definition 3 *Complex numbers $a_1, \ldots, a_n$ are algebraically independent if $p(a_1, \ldots, a_n) \neq 0$ for all not identically zero polynomials p in $\mathbf{Z}[x_1, \ldots, x_n]$.*

4.1 Lindemann's Theorem.

Theorem 1 *For any distinct algebraic numbers $\alpha_1, \ldots, \alpha_n$ and non zero algebraic numbers $\beta_1, \ldots, \beta_n$ we have*

$$\beta_1 e_1^{\alpha} + \cdots + \beta_n e_n^{\alpha} \neq 0$$

.

Corollary 1 *If the algebraic numbers $\alpha_1, \ldots, \alpha_n$ are linearly independent over $\mathbf{Q}$, then $e^{\alpha_1}, \ldots, e^{\alpha_n}$ are algebraically independent.*

For proof, see [Baker].

The following is well known. See [Besicovitch].

Lemma 1 *If $p_1, \ldots, p_n$ are different square free natural numbers, then $1/\sqrt{p_1}, \ldots, 1/\sqrt{p_n}$ are linearly independent independent over the rationals.*

Note. We use the reciprocals of the radicals above and not the radicals themselves to fulfil the requirements of applying the uniformity conjecture i.e., to have the value of the exponent no more than 1 so that the resulting expressions are in expanded form.

By induction on the number of nodes of a polynomial term one can prove the following

Lemma 2 *If T is a polynomial term with real numbers on the frontier, $s(T)$ nodes, and the real numbers are bounded by $10^{h(T)}$ then the value $|T|$ represented by T is bounded by $10^{h(T)s(T)}$.*

Theorem 2 *(Using the uniformity conjecture). Suppose polynomial $p(x_1, \ldots, x_n)$ is represented by polynomial term $T(x_1, \ldots, x_n)$ with $s(T)$ nodes and logarithmic height $h(T)$. Then p is the zero polynomial if and only if*

$$\mid p(e^{p_1^{-1/2}}, \cdots, e^{p_n^{-1/2}}) \mid < 10^{-k}$$

where $k = 2 * s(T) * max(h(T), n+1)$ and $p_1, \ldots, p_n$ are the first square free natural numbers >1.

Proof.

By the theorem and lemma above, $e^{p_1^{-1/2}}, \ldots, e^{p_n^{-1/2}}$ are algebraically independent. So p is the zero polynomial if and only if $p(e^{p_1^{-1/2}}, \ldots, e^{p_n^{-1/2}}) = 0$. According to the uniformity conjecture, $p(e^{p_1^{-1/2}}, \ldots, e^{p_n^{-1/2}}) = 0$ if and only if $\left|p(e^{p_1^{-1/2}}, \ldots, e^{p_n^{-1/2}})\right| < 10^{-2\,k}$, where k is the length of the expression for $p(e^{p_1^{-1/2}}, \ldots, e^{p_n^{-1/2}})$.

There are $s(T)$ nodes in the tree for the polynomial $p(x_1, \ldots, x_n)$. In the corresponding tree for the constant which results from substitution, $p(e^{p_1^{-1/2}}, \ldots, e^{p_n^{-1/2}})$, the interior nodes contribute length 1 each, the frontier nodes with natural numbers on them contribute length at most $h(T)$ each, and the frontier nodes with exponentials on them contribute length at most $n+1$ each, since the first n square free numbers all have length no more than n. (This follows from the Bertrand postulate, which states that the n th prime number is always below 2^n. See [Zippel].)

5 Complexity of Approximation

Using the results in [Borwein, Borwein 1988], [Brent] we can get $e^{1/\sqrt{p}}$ to precision $n >$ length(p) decimal places in $O(M(n))$ bit operations, where $M(n)$ is the bit complexity of multiplying two n-digit natural numbers.

We have a bound of $B = 10^{s(T)\,h(T)}$ on the numbers which appear in the tree (See lemma 1)

At each operation $+, *, -$ within the polynomial term we lose no more than $O(s(T)h(T))$ decimal places. To do one operation of multiplication for example, xy to k decimal places, we have

$$|(x+\epsilon_1)(y+\epsilon_2) - xy| < (\epsilon_1+\epsilon_2)\max(|x|,|y|)$$

and therefore we would need $(\epsilon_1+\epsilon_2)10^{s(T)\,h(T)} < 10^{-k}$

It suffices, according to the above theorem, to finish the computation with precision $k = 2 * s(T) * \max(h(T), n+1)$.

The precision needed for the whole computation is bounded by

$$2\ s(T)\max(h(T), n+1) + O(s(T)(s(T)h(T)))\ =\ O\left(s(T)^2\, h(T)\right)$$

Since we have $s(T)$ interior nodes (operations), the bit complexity of the computation is $O\left(s(T)\, M(s(T)^2\ h(T))\right)$.

Theorem 3 *(Assuming the uniformity conjecture). Suppose a polynomial $p(x_1,\ldots,x_n)$ is represented by polynomial term $T(x_1,\ldots,x_n)$ with $s(T)$ nodes and logarithmic height $h(T)$. Then the bit complexity of deciding whether or not p is the zero polynomial is bounded by $O\left(s(T)\, M(s(T)^2\ h(T))\right)$*

6 Testing the Derivatives, and Computing Coefficients

We can also decide the zero equivalence of the derivatives in polynomial time using algebraically independent numbers and assuming the uniformity conjecture.

Theorem 4

$$D_x^m p \equiv 0 \iff p_1v_1 - p_2v_2 + p_3v_3 - \cdots + (-1)^m\, p_{m+1}v_{m+1} = 0$$

where $p_i := p(a_i, b_1, \ldots, b_n)$; $i = 1, 2, \ldots, m+1$ and $a_1, \ldots, a_{m+1}, b_1, \ldots, b_n$ are as usual algebraically independent. The general form of v_i is given by

$$\begin{aligned} v_i &= det(V(a_1,\ldots,a_{i-1},a_{i+1},\ldots,a_{m+1})) \\ &= \prod_{\substack{1\le j<k\le m+1 \\ j,k\notin\{i\}}} (a_k - a_j) \end{aligned}$$

Proof. The m^{th} derivative of p with respect to x is identically zero if and only if the vector of values $(p_1,\ldots,p_{m+1})$ is a linear combination of the vectors of powers up to $m-1$ of $a_1,\ldots,a_{m+1})$.

Here, we use the expansion of the determinant of the Vandermonde matrix $V(a_1,\ldots,a_{i-1},a_{i+1},a_{m+1})$ as a product of differences. The Vandermonde matrix is given below.

$$\begin{bmatrix} 1 & a_1 & a_1^2 & a_1^3 & \cdots & a_1^{m-1} \\ 1 & a_2 & a_2^2 & a_2^3 & \cdots & a_2^{m-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & a_{i-1} & a_{i-1}^2 & a_{i-1}^3 & \cdots & a_{i-1}^{m-1} \\ 1 & a_{i+1} & a_{i+1}^2 & a_{i+1}^3 & \cdots & a_{i+1}^{m-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & a_{m+1} & a_{m+1}^2 & a_{m+1}^3 & \cdots & a_{m+1}^{m-1} \end{bmatrix}$$

For more details about the Vandermonde matrix and related things in computer algebra one may consult [Zippel].

Corollary 2 *If $p \in \mathbf{Z}[x,y_1,\ldots,y_m]$ is represented as polynomial term T with $s(T)$ nodes and $h(T)$ height then we can decide whether the nth derivative with respect to x is zero in time polynomial in $s(T),h(T)$ and n.*

This seems to depend on the the special form of the Vandermonde determinant, which allows a fairly compact representation as a term. We do not know whether or not any determinant with polynomial entries can be represented as a polynomial term with size bounded polynomially in the size of the determinant and the size of the entries. (It has been conjectured that this is not the case. See [Bürgisser].)

We can use the last corollary to determine the degree of x in T in polynomial time. This can be done using derivative testing until we get the first identically zero derivative.

We can also compute the coefficients as polynomial terms.

Theorem 5 *Let $T(x)$ be a polynomial term with variables $x,y_1,\ldots,y_n$. Suppose that $T(x)$ could be rewritten in the form $c_0 + c_1x + \ldots + c_nx^n$, where $c_0,\ldots,c_n$ do not involve x. Then $transpose(c_1,\ldots,c_n) = A^{-1}transpose(T(0),\ldots,T(n))$ where $A = (a_{ij})_{n+1 \text{ by } n+1}$, and $a_{ij} = (i-1)^{j-1}$.*

Proof. We note that the matrix A is invertible, and is in fact another Vandermonde matrix.

Corollary 3 *Each coefficient of $T(x)$ can be written as*
$c_i = a_{i0}T(0) + \ldots + a_{in}T(n)$
where a_{ij} is rational, with numerator and denominator less in absolute value than n^n.

It seems that we can do other algebraic computations e.g., finding GCD of polynomials represented as polynomial terms if we work in the field of fractions.

We remark that our zero recognition algorithm for polynomial terms is in complexity class NC, since there exists an efficient parallelisation of evaluation of such terms. See [Brent74], [Brent].

7 Analysis of the Algorithm

What are the properties of $e^{1/\sqrt{p_1}}, \ldots, e^{1/\sqrt{p_n}}$ that were needed to apply this algorithm ?

- calculating their kth digit can be done quickly in the sense of Borwein see [Borwein, Borwein 1988]. See,as well, [Brent].
- They are algebraically independent.
- We assumed also the Uniformity Conjecture. We can use a weaker version of the uniformity conjecture and still get a useful result. For example, in Theorem 2, we need

$$\left|p\left(e^{1/\sqrt{p_1}}, \ldots, e^{1/\sqrt{p_n}}\right)\right| < 10^{-k} \to p\left(e^{1/\sqrt{p_1}}, \ldots, e^{1/\sqrt{p_n}}\right) = 0.$$

  We would still have a polynomial complexity test if we defined k to be any polynomial in $s(T)$ and $h(T)$.

We, thus, can use any list of numbers satisfying these properties to recognise zero polynomials. So either the zero recognition problem for polynomial terms can be solved deterministicly in polynomial time, or there is no such list of numbers (which seems highly unlikely, even if we do not believe the uniformity conjecture).

7.1 Further Work

- What is a good lower bound for the value of a non singular determinant whose entries are exp-log expressions? Is the precision necessary to distinguish such a value from zero polynomial in the size of the determinant and the lengths of the entries? A more specific question: Does there exist a non singular matrix whose entries are exp-log expressions, but whose determinant is smaller in absolute value than 10^{-m}, where m is the sum of the lengths of the entries of the matrix? (This is related to the question of whether or not the matching problem is in NC.)
- Consider

$$\begin{aligned}\Gamma_n = \{&(x_1, \ldots, x_n) \in \mathbf{C}^n : \forall k\, \exists \text{polynomial term } T(x_1, \ldots, x_n) \\ &\text{with } s(T) \text{ nodes and height } h(T) \text{ so that} \\ &T \not\equiv 0 \text{ but } |T(x_1, \ldots, x_n)| < 10^{-k(s(T)h(T))}\}\end{aligned}$$

  Does Γ_n have measure 0 ?
  Note that if $(x_1, \ldots, x_n) \notin \Gamma_n$ and have decimal expansion which can be computed in polynomial time then they can be used to test zero equivalence of polynomial term.

The following result is a first step toward establishing the measure theoretic conjecture mentioned above, essentially implying that good test substitutions are common.

Let $deg(T, x_i)$ be the degree of variable x_i in polynomial term T. Suppose T has variables $x_1, \ldots, x_n$. Define $d(T) = \sum_{i=1}^{n} deg(T, x_i)$.

Let $\{|T| < 10^{-k}\}$ be the subset of R^n in which $|T| < 10^{-k}$.

Theorem 6 *If polynomial term T is not identically zero, then the intersection of the unit cube in R^n with $\{|T| < 10^{-k}\}$ has Lebesgue measure $\leq 2d(T)10^{-k/d(T)}$.*

Proof.

We use induction on n and the fact that if $|a_1 \ldots a_d| > |b_1 \ldots b_d|$ then for some i we must have $|a_i| > |b_i|$.

Let x be one of the variables in T, $d = deg(T, x)$, $D = d(T) - d$.

For almost all values of the variables,

$T = a_d(x - \alpha_1) \ldots (x - \alpha_d)$

where $\alpha_1, \ldots, \alpha_d$ depend algebraically on the other variables.

$\{|T| < 10^{-k}\}$

$= \{|a_d|\,|m(x)| < 10^{-k}\}$

$\subseteq \{|a_d| < 10^{-k\alpha}\} \cup \{|m(x)| < 10^{-k\beta}\}$

where $\alpha = D/(d+D)$, $\beta = d/(d+D)$, and $m(x) = (x-\alpha_1)\ldots(x-\alpha_d)$.

Let L be Lebesgue measure, and I^n the unit cube in R^n.

We apply our induction hypothesis to get an upper bound on $L(I^n \cap \{|a_d| < 10^{-k\alpha}\}$.

$L(I^n \cap \{|T| < 10^{-k}\}$

$\leq L(I^n \cap \{|a_d| < 10^{-k\alpha}\} + L(I^n \cap |m(x)| < 10^{-k\beta}\}$

$\leq 2D10^{-k\alpha/D} + 2d10^{-k\beta/d}$

$\leq 2(d + D)10^{-k/(d+D)}$

which verifies the theorem.

It is a consequence of this that if $k(T)$, mapping polynomial terms into natural numbers, is such that $\sum d(T)10^{-k(T)}$ (taken over all polynomial terms) converges to a finite limit, then for almost all values of the variables in the unit cube there is a number N so that for all non zero polynomial terms T, $d(T) > N \rightarrow |T| \geq 10^{-k(T)d(T)}$.

References

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

[Besicovitch] A.S. Besicovitch, On the linear independence of fractional powers of integers, JLMS, 1940, vol 15, pp3-6

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

[Borwein 1988] , On The Complexity of Familiar Functions and Numbers, J. M. Borwein and P. B. Borwein, SIAM Review, Vol.30, No 4, December 1988, pp 589-601.

[BCSS] Blum,Cucker,Shub,Smale, *Complexity and Real Computation*, Springer.

[Naylor] Bill Naylor, *Polynomial GCD using straight line program representation*,Thesis 1999, University of Bath.

[Brent74] R.P. Brent, The parallel evaluation of generic arithmetic expressions, Journal of the ACM 21 (1974), pp201-206

[Brent] Richard P. Brent, Multiple-precision zero- finding methods and the complexity of elementary functions, Analytic Computational Complexity, J.F.Traub ed., Academic Press 1975, pp. 151-176.

[Bürgisser] Peter Bürgisser, *Completeness and reduction in algebraic complexity theory, Algorithms and computations in mathematics, Vol. 7.*

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

[Ibarra and Moran] Equivalence of straight-line programs, J.ACM 30, 1983, pp 217-228.

[Kaltofen] E. Kaltofen, Greatest common divisors of polynomials given by straight line programs, JACM, 1988, pp 231-264

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

[Richardson97] Daniel Richardson, How to Recognise Zero, J. Symbolic Computation (1997), 24, pp 627-645.

[Richardson2001] Daniel Richardson, Multiplicative Independence of Algebraic Numbers and Expressions, Journal of pure and applied algebra, 164, 2001, pp231-245

[Richardson2000] Daniel Richardson, The Uniformity Conjecture, Proceedings of Computability and Complexity in Analysis, CCA2000, September 17-19, Swansea, Wales. Also in associated Springer lecture notes in commuter science Vol 2064, pp 253-272. See www.bath.ac.uk/ masdr/unif.dvi.

[Richardson2001a] Testing the Uniformity Conjecture, www.bath.ac.uk/ masdr/testu.dvi

[Richardson and Langley] Some observations about familiar numbers, ISSAC2002

[Zippel] Richard Zippel, *Effective Polynomial Computation*, Kluwer Academic Publishers, 1993.