Revisiting the Binary Euclidean Algorithm*

Richard P. Brent
Computing Laboratory
University of Oxford

rpb@comlab.ox.ac.uk
http://www.comlab.ox.ac.uk

*Presented at University of Warwick, 12 June 1998.

Warwickt typeset using LaTeX

Abstract

The binary Euclidean algorithm is a variant of the classical Euclidean algorithm. It avoids divisions and multiplications, except by powers of two, so is potentially faster than the classical algorithm on a binary machine. In this talk I will describe the classical and binary algorithms, and compare their worst case and average case behaviour. In particular, I will correct some small but significant errors in the literature, discuss some recent results of Brigitte Vallée, and describe a numerical computation which verifies Vallée's conjecture to 44 decimal places.

Outline

- The classical Euclidean algorithm
  - The algorithm
  - Worst case
  - Continued fractions
  - Continuous model of Gauss
  - Results of Kuz'min, Lévy, et al
- The binary Euclidean algorithm
  - The algorithm
  - Worst case
  - Continuous model
  - Conjectured/empirical results
  - An error in the literature
  - Useful operators
  - Recent results of Vallée
  - Confirmation of a conjecture
  - Open problems

Notation

$\lg(x)$ denotes $\log_2(x)$.

$\mathrm{Val}_2(u)$ denotes the dyadic valuation of the positive integer u, i.e. the greatest integer j such that $2^j \mid u$.

The Classical Euclidean Algorithm

The Euclidean algorithm finds the greatest common divisor (GCD) of two positive integers m and n. It is one of the best known of all algorithms. Knuth (§1.1) gives the algorithm as:

Algorithm E

E1. $r \leftarrow m \bmod n$

E2. If $r = 0$ terminate with n as the result.

E3. Set $m \leftarrow n$, $n \leftarrow r$, and go to E1

Of course, Euclid did not describe the algorithm in this way. In fact, it is not quite clear what Euclid intended at step E1. For a translation of Euclid's description, see Knuth, §4.5.2.

The algorithm was probably known about 200 years before Euclid. Nevertheless, we shall call Algorithm E the "classical Euclidean algorithm" or just the "classical algorithm".

One-line Version

while $n \neq 0$ do $\binom{m}{n} \leftarrow \binom{n}{m \bmod n}$; return m.

Relation to Continued Fractions

Assume $m \geq n$. The first execution of step E1 gives

$$m = q \times n + r$$

where q is the quotient and r is the remainder on division of m by n. By definition,

$$0 \leq r < n.$$

Define $n_0 = m$, $n_1 = n$, $n_2 = r = m \bmod n$. Suppose step E3 is executed k times. Then

$$n_j = q_j \times n_{j+1} + n_{j+2}$$

holds for $j = 0, 1, \ldots, k$ and $n_{k+2} = 0$. We can write this as

$$\frac{n_j}{n_{j+1}} = q_j + 1 \Big/ \frac{n_{j+1}}{n_{j+2}}$$

so, in the usual notation for continued fractions,

$$\frac{m}{n} = q_0 + 1/q_1 + 1/q_2 + \cdots + 1/q_k.$$

The Worst Case

We have seen that there is an intimate connection between the classical Euclidean algorithm for computing $\mathrm{GCD}(m, n)$ and the continued fraction expansion of the rational number m/n.

The worst case for the classical algorithm occurs when all the quotients q_j are 1. This happens when the inputs are consecutive Fibonacci numbers. (These are defined by $F_0 = 0$, $F_1 = 1$, $F_{j+2} = F_{j+1} + F_j$ for $j \geq 0$.)

For example,

$$(m, n) = (F_6, F_5) = (8, 5) \to (5, 3) \to (3, 2) \to (2, 1).$$

Since $F_k \sim \rho^k/\sqrt{5}$, where $\rho = \frac{\sqrt{5}+1}{2} \simeq 1.618$, the worst case number of iterations of the classical algorithm is

$$\log_\rho N + O(1),$$

where $N = \max(m, n)$.

The Continuous Model of Gauss

To investigate the *average* behaviour of the classical algorithm, we can restrict attention to the case $0 < m < n$ (so $q_0 = 0$). We assume that $n = N$ is large and that m is equally likely to take the values $\{1, 2, \ldots, N-1\}$. Thus, m/n will be approximately uniformly distributed in $(0, 1)$, and the sequence of quotients $q_1, q_2, \ldots$ will "look like" the quotients in the continued fraction expansion of a uniformly distributed random number.

Gauss's Recurrence

Suppose $x_0 \in (0, 1)$, and x_0 has a continued fraction expansion

$$x_0 = 1/q_1 + 1/q_2 + \ldots + 1/(q_k + x_{k+1}),$$

where $q_1, \ldots, q_k$ are positive integers and $x_{k+1} \in (0, 1)$.

We can express the probability distribution function $F_{k+1}(x)$ of x_{k+1} in terms of the distribution function $F_k(x)$ of x_k. Using the fact that $1/x_k = q_k + x_{k+1}$, we see that

$$\begin{aligned} F_{k+1}(x) &= \Pr(0 \le x_{k+1} \le x) \\ &= \sum_{q \ge 1} \Pr(q \le 1/x_k \le q + x) \\ &= \sum_{q \ge 1} \Pr\left(\frac{1}{q+x} \le x_k \le \frac{1}{q}\right) \\ &= \sum_{q \ge 1} \left(F_k\left(\frac{1}{q}\right) - F_k\left(\frac{1}{q+x}\right)\right) \end{aligned}$$

The Limiting Distribution

To investigate average case behaviour in the continuous model, we assume $F_0(x) = x$ for $x \in (0, 1)$ (the uniform distribution of m/n), and consider $F_k(x)$ as $k \to \infty$.

If we *assume* that a limit distribution $F(x) = \lim_{k\to\infty} F_k(x)$ exists, then $F(x)$ satisfies the functional equation

$$F(x) = \sum_{q \ge 1} \left(F\left(\frac{1}{q}\right) - F\left(\frac{1}{q+x}\right)\right).$$

Gauss noticed the simple solution

$$F(x) = \lg(1 + x).$$

However, Gauss was not able to prove convergence. It was eventually proved by Kuz'min (1928). Sharper results were proved by Lévy (1929), Wirsing (1974), and Babenko (1978).

Babenko's Theorem

The definitive result, due to Babenko (1978), is

$$F_k(x) = \lg(1 + x) + \sum_{j \ge 2} \lambda_j^k \Psi_j(x),$$

where $|\lambda_2| > |\lambda_3| \ge |\lambda_4| \ge \cdots$,

$$-\lambda_2 = \lambda = 0.3036630028 \cdots$$

is called *Wirsing's constant*, and the $\Psi_j(x)$ are certain analytic functions (see Knuth, §4.5.3 for more details).

The expected number of iterations is

$$\sim \frac{12 \ln 2}{\pi^2} \ln N \sim 0.8428 \ln N \sim 0.5842 \lg N$$

which can be compared with

$$\sim \log_\rho N \sim 2.0781 \ln N \sim 1.4404 \lg N$$

for the worst case.

Remainder of the Talk

In the time remaining, I will describe what progress has been made towards a similar analysis of the *binary* Euclidean algorithm (to be defined).

The Binary Euclidean Algorithm

The idea of the *binary* Euclidean algorithm is to avoid the "division" operation $r \leftarrow m \bmod n$ of the classical algorithm, but retain $O(\log N)$ worst (and average) case.

We assume that the algorithm is implemented on a binary computer so division by a power of two is easy. In particular, we assume that the "shift right until odd" operation

$$u \leftarrow u/2^{\mathrm{Val}_2(u)}$$

or equivalently

$$\text{while even}(u) \text{ do } u \leftarrow u/2$$

can be performed in constant time (although time $O(\mathrm{Val}_2(u))$ would be sufficient).

There are several almost equivalent ways to define the algorithm. For simplicity, let us assume that u and v are *odd* positive integers.

Definition of the Binary Algorithm

Here u, v are odd, positive integers.

Algorithm B

B1. $t \leftarrow |u - v|$;
if $t = 0$ terminate with result u

B2. $t \leftarrow t/2^{\mathrm{Val}_2(t)}$

B3. if $u \geq v$ then $u \leftarrow t$ else $v \leftarrow t$;
go to B1.

History

The binary Euclidean algorithm is attributed to Silver and Terzian (unpublished, 1962) and Stein (1967). However, it seems to go back almost as far as the classical Euclidean algorithm. Knuth (§4.5.2) quotes a translation of a first-century AD Chinese text *Chiu Chang Suan Shu* on how to reduce a fraction to lowest terms:

> If halving is possible, take half.
> Otherwise write down the denominator and the numerator, and subtract the smaller from the greater.
> Repeat until both numbers are equal.
> Simplify with this common value.

This looks very much like Algorithm B !

Another Formulation

It will be useful to rewrite Algorithm B in the following equivalent form (using pseudo-Pascal):

Algorithm V { Assume $u \le v$ }

while $u \ne v$ **do**
  begin
  while $u < v$ **do**
    begin
    $j \leftarrow \mathrm{Val}_2(v-u)$;
    $v \leftarrow (v-u)/2^j$;
    end;
  $u \leftrightarrow v$;
  end;
return u.

Continued Fractions

Vallée [13] shows a connection between Algorithm V and continued fractions of a certain form:

$$\frac{u}{v} = 1/a_1 + 2^{k_1}/a_2 + 2^{k_2}/\ldots/a_r + 2^{k_r}\ ,$$

where a_j is odd, $k_j > 0$, and $0 < a_j < 2^{k_j}$.

The Worst Case

At step B1, u and v are odd, so t is even. Thus, step B2 always reduces t by at least a factor of two. Using this fact, it is easy to show that step B3 is executed at most

$$\lfloor \lg(u+v) \rfloor$$

times (Knuth, exercise 4.5.2.37). Thus, if $N = \max(u,v)$, step B3 is executed at most

$$\lg(N) + O(1)$$

times.

Exercise

Even if step B2 is replaced by single-bit shifts

$$\text{while even}(t)\ \text{do}\ t \leftarrow t/2$$

the overall worst case is still $O(\log N)$.

Extended Binary Algorithm

It is possible to give an *extended* binary GCD algorithm which computes multipliers α and β such that

$$\alpha u + \beta v = \mathrm{GCD}(u,v)$$

(Bojanczyk and Brent [1], 1987).

Systolic Binary Algorithm

For hardware implementation, there is a systolic array variant of the binary GCD algorithm (Brent and Kung [4], 1985). This takes time $O(\log N)$ using $O(\log N)$ 1-bit processors and nearest-neighbour communication. The overall bit-complexity is $O(\log N)^2$.

A Heuristic Continuous Model

To analyse the expected behaviour of Algorithm B, we can follow what Gauss did for the classical algorithm. This was first attempted in my 1976 paper [2] and there is a summary in Knuth (Vol. 2, *third* edition, §4.5.2).

Assume that the initial inputs u_0, v_0 to Algorithm B are uniformly and independently distributed in (0, N), apart from the restriction that they are odd. Let (u_n, v_n) be the value of (u, v) after n iterations of step B3.

Let

$$x_n = \frac{\min(u_n, v_n)}{\max(u_n, v_n)}$$

and let $F_n(x)$ be the probability distribution function of x_n (in the limit as $N \to \infty$). Thus $F_0(x) = x$ for $x \in [0, 1]$.

We assume that $\mathrm{Val}_2(t)$ takes the value k with probability 2^{-k} at step B2. (Vallée does not make this assumption – we will discuss her rigorous analysis later.)

The Recurrence for F_n

Consider the effect of steps B2 and B3. We can assume that $u > v$ so $t = u - v$. If $\mathrm{Val}_2(t) = k$ then $X = v/u$ is transformed to

$$\begin{aligned} X' &= \min\left(\frac{u-v}{2^k v}, \frac{2^k v}{u-v}\right) \\ &= \min\left(\frac{1-X}{2^k X}, \frac{2^k X}{1-X}\right) . \end{aligned}$$

It follows that $X' < x$ iff

$$X < \frac{1}{1 + 2^k/x} \ \text{ or } \ X > \frac{1}{1 + 2^k x} .$$

Thus, the recurrence for $G_n(x) = 1 - F_n(x)$ is

$$G_{n+1}(x) = \sum_{k \ge 1} 2^{-k} \left(G_n\left(\frac{1}{1+2^k/x}\right) - G_n\left(\frac{1}{1+2^k x}\right)\right) ,$$

and $G_0(x) = 1 - x$ for $x \in [0, 1]$.

The Recurrence for f_n

Differentiating the recurrence for G_n we obtain (formally) a recurrence for the probability density $f_n(x) = F_n'(x) = -G_n'(x)$:

$$\begin{aligned} f_{n+1}(x) &= \sum_{k \ge 1} \left(\frac{1}{x + 2^k}\right)^2 f_n\left(\frac{x}{x+2^k}\right) \\ &+ \sum_{k \ge 1} \left(\frac{1}{1 + 2^k x}\right)^2 f_n\left(\frac{1}{1+2^k x}\right) . \end{aligned}$$

Operator Notation

The recurrence for f_n may be written as

$$f_{n+1} = \mathcal{B}_2 f_n,$$

where the operator $\mathcal{B}_2$ is the case $s = 2$ of a more general operator $\mathcal{B}_s$ which will be defined later.

Conjectured and Empirical Results

In my 1976 paper [2] I gave numerical and analytic evidence that $F_n(x)$ converges to a limiting distribution $F(x)$ as $n \to \infty$, and that $f_n(x)$ converges to the corresponding probability density $f(x) = F'(x)$ (note that $f = \mathcal{B}_2 f$ so f is a "fixed point" of the operator $\mathcal{B}_2$). Assuming the existence of F, it is shown in [2] that the expected number of iterations of Algorithm B is $\sim K \lg N$ as $N \to \infty$, where $K = 0.705\ldots$ is a constant defined by

$$K = \ln 2 / E_\infty ,$$

and

$$E_\infty = \ln 2 + \int_0^1 \left(\sum_{k=2}^{\infty} \left(\frac{1 - 2^{-k}}{1 + (2^k - 1)x} \right) - \frac{1}{2(1+x)} \right) F(x)\, dx .$$

A Simplification

We can simplify the expression for K to obtain

$$K = 2/b \ ,$$

where

$$b = 2 - \int_0^1 \lg(1-x) f(x) \, dx \ .$$

Using integration by parts we obtain an equivalent expression

$$b = 2 + \frac{1}{\ln 2} \int_0^1 \frac{1 - F(x)}{1 - x} \, dx \ .$$

For the proofs, see Knuth, third edition, §4.5.2.

An Error in the Literature

In (Brent, 1976) I claimed that, for all $n \ge 0$ and $x \in (0, 1]$,

$$F_n(x) = \alpha_n(x) \lg(x) + \beta_n(x) \ ,$$

where $\alpha_n(x)$ and $\beta_n(x)$ are analytic and regular in the disk $|x| < 1$. However, *this is incorrect*, even in the case $n = 1$.

The error appeared to go unnoticed until 1997, when Don Knuth was revising Volume 2 in preparation for publication of the third edition. Knuth computed the constant K using recurrences for the analytic functions $\alpha_n(x)$ and $\beta_n(x)$, and I computed K directly using the defining integral and recurrences for $F_n(x)$. Our computations disagreed in the 14th decimal place ! Knuth found

$$K = 0.70597\ 12461\ 019\underline{45\ 99986} \cdots$$

but I found

$$K = 0.70597\ 12461\ 019\underline{16\ 39152} \cdots$$

Some Detective Work

After a flurry of emails we tracked down the error. It was found independently, and at the same time (within 24 hours), by Flajolet and Vallée.

The source of the error is illustrated by Lemma 3.1 of my 1976 paper [2], which is wrong (and corrected in the solution to ex. 4.5.2.29 of Knuth, third edition).

The *Mellin transform* of a function $g(x)$ is defined by

$$g^*(s) = \int_0^\infty g(x) x^{s-1} dx \ .$$

If $f(x) = \sum_{k \ge 1} 2^{-k} g(2^k x)$ then the Mellin transform of f is

$$f^*(s) = \sum_{k \ge 1} 2^{-k(s+1)} g^*(s) = \frac{g^*(s)}{2^{s+1} - 1} \ .$$

Under suitable conditions we can apply the Mellin inversion formula to obtain

$$f(x) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} f^*(s) x^{-s} ds \ .$$

Applying these results to $g(x) = 1/(1+x)$, whose Mellin transform is $g^*(s) = \pi / \sin \pi s$ when $0 < \mathcal{R}s < 1$, we find

$$f(x) = \sum_{k \ge 1} \frac{2^{-k}}{1 + 2^k x}$$

as a sum of residues of

$$\left(\frac{\pi}{\sin \pi s} \right) \frac{x^{-s}}{2^{s+1} - 1}$$

for $\mathcal{R}s \le 0$. This gives

$$f(x) = 1 + x \lg x + \frac{x}{2} + x P(\lg x) - \frac{2}{1} x^2 + \frac{4}{3} x^3 - \cdots \ ,$$

where

$$P(t) = \frac{2\pi}{\ln 2} \sum_{n=1}^{\infty} \frac{\sin 2n\pi t}{\sinh(2n\pi^2 / \ln 2)} \ .$$

The “Wobbles” Caused by $P(t)$

$P(t)$ is a very small periodic function:
$|P(t)| < 7.8 \times 10^{-12}$ for real t.

In [2, Lemma 3.1], the term $xP(\lg x)$ is omitted. Essentially, we only considered poles on the real axis and ignored those at $s = -1 \pm 2\pi i n/\ln 2$, $n = 1, 2, \ldots$. Because the residues at these poles are tiny (thanks to the sinh term in the denominator) numerical computations performed using single-precision floating-point arithmetic did not reveal the error.

An Analogy

Ramanujan made a similar error when he gave a formula for $\pi(x)$ (the number of primes $\le x$) which essentially ignored the residues of $x^s\zeta'(s)/\zeta(s)$ arising from zeros of $\zeta(s)$ off the real axis.

Some Useful Operators

Operators $\mathcal{B}_s$, $\mathcal{U}_s$, $\widetilde{\mathcal{U}}_s$, $\mathcal{V}_s$, useful in the analysis of the binary Euclidean algorithm, are defined on suitable function spaces by

$$\mathcal{U}_s[f](x) = \sum_{k\ge1} \left(\frac{1}{1+2^k x}\right)^s f\left(\frac{1}{1+2^k x}\right),$$

$$\widetilde{\mathcal{U}}_s[f](x) = \left(\frac{1}{x}\right)^s \mathcal{U}_s[f]\left(\frac{1}{x}\right),$$

$$\mathcal{B}_s = \mathcal{U}_s + \widetilde{\mathcal{U}}_s,$$

$$\mathcal{V}_s[f](x) = \sum_{k\ge1} \sum_{\substack{a \text{ odd},\\ 0<a<2^k}} \left(\frac{1}{a+2^k x}\right)^s f\left(\frac{1}{a+2^k x}\right).$$

In these definitions s is a complex variable, and the operators are called Ruelle operators [10]. They are linear operators acting on certain function spaces.

The case $s = 2$ is of particular interest. $\mathcal{B}_2$ encodes the effect of one iteration of the inner “while” loop of Algorithm V, and $\mathcal{V}_2$ encodes the effect of one iteration of the outer “while” loop.

Relations between Operators

$\mathcal{B}_2$ (denoted T) was introduced in my 1976 paper [2], and generalised to $\mathcal{B}_s$ by Vallée. $\mathcal{V}_s$ was introduced by Vallée. We shall call $\mathcal{B}_2$ the *binary Euclidean operator* and $\mathcal{V}_s$ *Vallée's operator*. Not surprisingly, the operators are related, as the following Lemma and Theorem show.

Lemma 1

$$\mathcal{V}_s = \mathcal{V}_s\widetilde{\mathcal{U}}_s + \mathcal{U}_s.$$

The following Theorem (which follows from the Lemma) gives a simple relationship between $\mathcal{B}_s$, $\mathcal{V}_s$ and $\mathcal{U}_s$.

Theorem 1

$$(\mathcal{V}_s - \mathcal{I})\mathcal{U}_s = \mathcal{V}_s(\mathcal{B}_s - \mathcal{I}).$$

Algorithmic interpretation

Algorithm V gives an interpretation of Lemma 1 in the case $s = 2$. If the input density of $x = u/v$ is $f(x)$ then execution of the inner “while” loop followed by the exchange of u and v transforms this density to $\mathcal{V}_2[f](x)$. However, by considering the first iteration of this loop (followed by the exchange if the loop terminates) we see that the transformed density is given by

$$\mathcal{V}_2\widetilde{\mathcal{U}}_2[f](x) + \mathcal{U}_2[f](x),$$

where the first term arises if $u < v$ without an exchange, and the second arises if an exchange occurs.

A Conjecture of Vallée

Let $\lambda = f(1)$, where f is the limiting probability density (conjectured to exist) as above. Vallée (see Knuth, third edition, §4.5.2(61)) conjectured that

$$\frac{\lambda}{b} = \frac{2 \ln 2}{\pi^2} \,,$$

or equivalently that

$$K = \frac{4 \ln 2}{\pi^2 \lambda} \,. \tag{1}$$

Vallée proved the conjecture under the assumption that the operator $\mathcal{B}_s$ satisfies a certain spectral condition. We have verified the conjectures numerically to 44 decimal places.

Recent Results of Vallée

Using her operator $\mathcal{V}_s$, Vallée recently *proved* that

$$K = \frac{2 \ln 2}{\pi^2 g(1)} \sum_{\substack{a \text{ odd}, \\ a>0}} 2^{-\lfloor \lg a \rfloor} G\left(\frac{1}{a}\right)$$

where g is a nonzero fixed point of $\mathcal{V}_2$ (i.e. $g = \mathcal{V}_2 g \neq 0$) and $G(x) = \int_0^x g(t)\, dt$. This is yet another expression for K (the only one which has been proved).

Warning: G here is not the same as $G(x) = 1 - F(x)$! Unfortunately Knuth and Vallée use incompatible notation.

Because $\mathcal{V}_s$ can be proved to have nice spectral properties, the existence and uniqueness (up to scaling) of g can be proved rigorously.

Fixed Points of some Operators

It follows immediately from Theorem 1 that, if

$$g = \mathcal{U}_2 f,$$

then

$$(\mathcal{V}_2 - \mathcal{I})g = \mathcal{V}_2(\mathcal{B}_2 - \mathcal{I})f.$$

Thus, if f is a fixed point of the operator $\mathcal{B}_2$, then g is a fixed point of the operator $\mathcal{V}_2$. From the recent result of Vallée [13, Prop. 4] we know that $\mathcal{V}_2$, acting on a certain Hardy space $\mathcal{H}^2(\mathcal{D})$, has a unique positive dominant simple eigenvalue 1, so g must be (a constant multiple of) the corresponding eigenfunction (provided $g \in \mathcal{H}^2(\mathcal{D})$). Also, from the definitions of $\mathcal{B}_2$ and $\mathcal{V}_2$, we have

$$\lambda = f(1) = 2g(1) = 2 \sum_{k \geq 1} \left(\frac{1}{1+2^k}\right)^2 f\left(\frac{1}{1+2^k}\right),$$

which is useful for proving the consistency of two of the expressions for K given above.

Numerical results

Using an improvement of the "discretization method" of [2], and the MP package with the equivalent of more than 50 decimal places (50D) working precision, we computed the limiting probability density f, then K, $\lambda = f(1)$, and $K\lambda$. The results were

$$\begin{aligned}
K &= 0.7059712461\ 0191639152\ 9314135852\ 8817666677 \\
\lambda &= 0.3979226811\ 8831664407\ 6707161142\ 6549823098 \\
K\lambda &= 0.2809219710\ 9073150563\ 5754397987\ 9880385315
\end{aligned}$$

These are believed to be correctly rounded values.
One of Vallée's conjectures is that $K\lambda = 4 \ln 2/\pi^2$. The computed value of $K\lambda$ agrees with $4 \ln 2/\pi^2$ to 40 decimals (in fact to 44 decimals).

Conclusion and Open Problems

Since Vallée's work (to appear in ANTS III Proceedings [12] and *Algorithmica* [13]) analysis of the average behaviour of the binary Euclidean algorithm has a rigorous foundation. However, some interesting open questions remain.

For example, does the binary Euclidean operator $\mathcal{B}_2$ have a unique positive dominant simple eigenvalue 1? Vallée [13, Prop. 4] has proved the corresponding result for her operator $\mathcal{V}_2$.

In order to estimate the speed of convergence of f_n to f (assuming f exists), we need more information on the spectrum of $\mathcal{B}_2$. What can be proved ? Preliminary numerical results indicate that the sub-dominant eigenvalue(s) are a complex conjugate pair:
$\lambda_2 = \overline{\lambda}_3 = 0.1735 \pm 0.0884i$, with
$|\lambda_2| = |\lambda_3| = 0.1948$ to 4D.

Acknowledgements

Thanks to Don Knuth for encouraging me to correct and extend my 1976 results for the third edition of *Seminumerical Algorithms*, to Brigitte Vallée for sharing her conjectures and results with me, and to Philippe Flajolet for his notes on Mellin transforms.

Further details of the new results presented in this talk will appear in a Technical Report [5].

References

[1] Adam W. Bojanczyk and Richard P. Brent, A systolic algorithm for extended GCD computation, *Comput. Math. Applic.* 14 (1987), 233–238.

[2] Richard P. Brent, Analysis of the Binary Euclidean Algorithm, *New Directions and Recent Results in Algorithms and Complexity*, (J. F. Traub, editor), Academic Press, New York, 1976, 321–355.

[3] Richard P. Brent and H. T. Kung, Systolic VLSI arrays for linear-time GCD computation, in *VLSI 83* (F. Anceau and E. J. Aas, editors), North-Holland, Amsterdam, 1983, 145–154.

[4] Richard P. Brent and H. T. Kung, A systolic VLSI array for integer GCD computation, in *ARITH-7, Proc. Seventh Symposium on Computer Arithmetic* (K. Hwang, editor), IEEE/CS Press, 1985.

[5] Richard P. Brent, *Further analysis of the Binary Euclidean algorithm*, Technical Report, Oxford University Computing Laboratory, to appear.

[6] Hervé Daudé, Philippe Flajolet and Brigitte Vallée, An analysis of the Gaussian Algorithm for Lattice Reduction, Proc. ANTS'94, Lecture Notes in Computer Science 877, Springer-Verlag, 1994, 144–158. Extended version in *Combinatorics, Probability and Computing* **6** (1997), 397–433.

[7] Philippe Flajolet and Brigitte Vallée, Continued Fraction Algorithms, Functional Operators and Structure Constants, *Theoretical Computer Science* **194** (1998), 1–34.

[8] Carl F. Gauss, Brief an Laplace vom 30 Jan. 1812, *Carl Friedrich Gauss Werke*, Bd. $\mathbf{X}_1$, Gottingen, 371–374.

[9] Donald E. Knuth, *The Art of Computer Programming, Volume 2: Seminumerical Algorithms* (third edition). Addison-Wesley, Menlo Park, 1997.

[10] David Ruelle, *Thermodynamic formalism*, Addison Wesley, Menlo Park, 1978.

[11] Brigitte Vallée, Opérateurs de Ruelle–Mayer généralisés et analyse des algorithmes de Gauss et d'Euclide, *Acta Arithmetica* **81** (1997), 101–144.

[12] Brigitte Vallée, The complete analysis of the Binary Euclidean Algorithm, manuscript, Jan. 1998 (to appear in Proc. ANTS'98, Springer-Verlag, 1998).

[13] Brigitte Vallée. Dynamics of the Binary Euclidean Algorithm: functional analysis and operators, manuscript, Feb. 1998 (to appear in *Algorithmica*).