Twenty years' analysis of the Binary Euclidean Algorithm

Richard P. Brent

1 Introduction

The real significance of Quicksort is its excellent *average case* behaviour. A large part of Tony Hoare's seminal 1962 paper [7] is devoted to a thorough average case analysis of quicksort. The formal correctness proofs appeared later, in [6, 8]. When invited to write a paper for this celebration of Tony Hoare's contributions to Computing Science, the choice of topic was easy. Given Tony's interest in Euclidean algorithms and average case analysis, my conclusion was that I should consider the average case behaviour of Euclidean algorithms. Fortunately, thanks to Don Knuth and Brigitte Vallée, I had something new to say about the *binary* Euclidean algorithm.

1.1 Outline

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 §2 we define the binary Euclidean algorithm and mention some of its properties, history and generalisations. In §3 we outline the heuristic model which was first presented in my 1976 paper [1]. Some of the results of that paper are mentioned (and simplified) in §4.

Average case analysis of the binary Euclidean algorithm lay dormant from 1976 until Vallée's recent analysis [12, 13] using some nontrivial functional analysis. In §§5–6 we discuss Vallée's results and conjectures. In §7 we give some numerical evidence for one of her conjectures. Some connections between Vallée's results and our earlier results are given in §8.

Finally, in §9 we take the opportunity to point out an error in the 1976 paper [1]. Although the error is theoretically significant and (when pointed out) rather obvious, it appears that no one noticed it until recently. The manner of its discovery is discussed in §9.

Due to space limitations, much had to be omitted from this chapter. For a more leisurely exposition, including proofs of Theorems 1–2 below, see [2].

1.2 Notation

$\lg(x)$ denotes $\log_2(x)$. N, n, a, k, u, v are positive integers. $\mathrm{Val}_2(u)$ denotes the dyadic valuation of the positive integer u, i.e. the greatest integer j such that $2^j \mid u$ (this is just the number of trailing zero bits in the binary representation of u). $f(x)$ is usually a probability density, and $F(x)$ is the corresponding probability distribution.

A word of warning: Brent [1], Knuth [9], and Vallée [11, 12, 13] use incompatible notation, so we can not be consistent with all of them. Knuth uses $G(x)$ for our $\widetilde{F}(x) = 1 - F(x)$, and Vallée sometimes interchanges our f and g.

2 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.

2.1 Definitions of the Binary Euclidean Algorithm

There are several almost equivalent ways to define the algorithm. It is easy to take account of the largest power of two dividing the inputs, using the relation

$$\mathrm{GCD}(u, v) = 2^{\min(\mathrm{Val}_2(u), \mathrm{Val}_2(v))}\, \mathrm{GCD}\left(u/2^{\mathrm{Val}_2(u)}, v/2^{\mathrm{Val}_2(v)}\right),$$

so for simplicity we assume that u and v are *odd* positive integers. Following is a simplified version of the algorithm given in Knuth [9, §4.5.2].

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.

2.2 History

The binary Euclidean algorithm is usually attributed to Silver and Terzian (unpublished, 1962) and independently Stein (1961-1967). However, it seems to go back much further. Knuth [9, §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 is essentially Algorithm B ! Hence, the binary algorithm is almost as old as the classical Euclidean algorithm.

3 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 [1]. 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 $x \in [0, 1]$.

3.1 A Plausible Assumption

We make the assumption that $\mathrm{Val}_2(t)$ takes the value k with probability 2^{-k} at step B2. The assumption is plausible because $\mathrm{Val}_2(t)$ at step B2 depends on the least significant bits of u and v, whereas the comparison at step B3 depends on the most significant bits, so one would expect the steps to be (almost) independent when N is large. (Vallée does not need to make this assumption. Her results are mentioned in §§5–6. They show that the assumption is correct in the limit as $N \to \infty$.)

3.2 The Recurrence for F_n

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

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

It follows that $X' < x$ iff

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

Thus, the recurrence for $\widetilde{F}_n(x) = 1 - F_n(x)$ is

$$\widetilde{F}_{n+1}(x) = \sum_{k\geq 1} 2^{-k} \left(\widetilde{F}_n \left(\frac{1}{1+2^k/x} \right) - \widetilde{F}_n \left(\frac{1}{1+2^k x} \right) \right) \tag{1}$$

and $\widetilde{F}_0(x) = 1 - x$ for $x \in [0,1]$.

3.3 The Recurrence for f_n

Differentiating the recurrence for $\widetilde{F}_n$ we obtain (formally) a recurrence for the probability density $f_n(x) = F_n'(x) = -\widetilde{F}_n'(x)$:

$$f_{n+1}(x) = \sum_{k\geq 1} \left(\frac{1}{x+2^k} \right)^2 f_n \left(\frac{x}{x+2^k} \right) + \sum_{k\geq 1} \left(\frac{1}{1+2^k x} \right)^2 f_n \left(\frac{1}{1+2^k x} \right) \, .$$

It was noted in [1, §5] that the coefficients in this recurrence are positive, and that the recurrence preserves the L_1 norm of positive functions.

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 is defined in §5.3.

4 Conjectured and Empirical Results

In my 1976 paper [1] 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 [1] 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 given by

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

and

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

4.1 A Simplification

We can simplify the expression for K to obtain

$$K = 2/b \, , \tag{2}$$

where

$$b = 2 - \int_0^1 \lg(1-x) f(x)\, dx\ . \tag{3}$$

Using integration by parts we obtain an equivalent expression

$$b = 2 + \frac{1}{\ln 2} \int_0^1 \frac{1 - F(x)}{1-x}\, dx\ . \tag{4}$$

My direct proof of (3)–(4) is given in Knuth [9, §4.5.2].

5 Another Formulation – Algorithm V

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.

5.1 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 $0 < a_j < 2^{k_j}$ (excluding the trivial case $u = v = 1$).

5.2 Some Details of Vallée's Results

Algorithm V has two nested loops. The outer loop exchanges u and v. Between two exchanges, the inner loop performs a sequence of subtractions and shifts which can be written as

$$\begin{aligned} v &\to u + 2^{b_1} v_1; \\ v_1 &\to u + 2^{b_2} v_2; \\ &\cdots \\ v_{m-1} &\to u + 2^{b_m} v_m \end{aligned}$$

with $v_m \leq u$.

If $x_0 = u/v$ at the beginning of an inner loop, the effect of the inner loop followed by an exchange is the rational $x_1 = v_m/u$ defined by

$$x_0 = \frac{1}{a + 2^k x_1},$$

where a is an odd integer given by $a = 1 + 2^{b_1} + 2^{b_1+b_2} + \cdots + 2^{b_1+\cdots+b_{m-1}}$, and the exponent k is given by $k = b_1 + \cdots + b_m$. Thus, the rational u/v, for $1 \leq u < v$, has a unique *binary continued fraction expansion* of the form

$$\frac{u}{v} = 1/a_1 + 2^{k_1}/a_2 + 2^{k_2}/\cdots + 2^{k_{r-1}}/(a_r + 2^{k_r})$$

Vallée studies three parameters related to this continued fraction

1. The height or the depth (i.e. the number of exchanges) r.
2. The total number of subtractions necessary to obtain the expansion; if $p(a)$ denotes the number of "1"s in the binary expansion of the integer a, it is equal to $p(a_1) + p(a_2) + \cdots + p(a_r)$. (Equivalently, the number of times step B2 of Algorithm B is performed.)
3. The total number of single-bit shifts, i.e. the sum of exponents of 2 in the numerators of the binary continued fraction, $k_1 + \cdots + k_r$.

Her results give the average values of these three parameters: they are asymptotically $A_i \ln N$ for certain computable constants A_1, A_2, A_3.

5.3 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 by

$$\mathcal{U}_s[f](x) = \sum_{k\geq 1} \left(\frac{1}{1+2^k x}\right)^s f\left(\frac{1}{1+2^k x}\right), \tag{5}$$

$$\widetilde{\mathcal{U}}_s[f](x) = \left(\frac{1}{x}\right)^s \mathcal{U}_s[f]\left(\frac{1}{x}\right), \tag{6}$$

$$\mathcal{B}_s = \mathcal{U}_s + \widetilde{\mathcal{U}}_s, \tag{7}$$

$$\mathcal{V}_s[f](x) = \sum_{k\geq 1} \sum_{\substack{a \text{ odd}, \\ 0<a<2^k}} \left(\frac{1}{a+2^k x}\right)^s f\left(\frac{1}{a+2^k x}\right). \tag{8}$$

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. See Vallée [12, 13] for a more detailed explanation.

5.4 History and Notation

$\mathcal{B}_2$ (denoted T) was introduced in my 1976 paper [1], and was generalised to $\mathcal{B}_s$ by Vallée. $\mathcal{V}_s$ was introduced by Vallée [12, 13]. We shall call

- $\mathcal{B}_s$ (or sometimes just $\mathcal{B}_2$) the *binary Euclidean operator* and
- $\mathcal{V}_s$ (or sometimes just $\mathcal{V}_2$) *Vallée's operator.*

5.5 Relation Between the Operators

The binary Euclidean operator and Vallée's operator are closely related, as the following results show. Proofs may be found in [2].

Lemma 1 $\mathcal{V}_s = \mathcal{V}_s\widetilde{\mathcal{U}}_s + \mathcal{U}_s$. □

Theorem 1 $(\mathcal{V}_s - \mathcal{I})\mathcal{U}_s = \mathcal{V}_s(\mathcal{B}_s - \mathcal{I})$. □

5.6 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 there is no exchange, and the second arises if an exchange occurs.

5.7 Fixed Points

It follows immediately from Theorem 1 that

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

Thus, if f is a fixed point of the operator $\mathcal{B}_2$, then $g = \mathcal{U}_2 f$ is a fixed point of the operator $\mathcal{V}_2$. From a 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 $g \in \mathcal{H}^2(\mathcal{D})$).

6 A Result of Vallée

Recall the constant K defined in §4. Using her operator $\mathcal{V}_s$, Vallée [13] recently proved that

$$K = \frac{2\ln 2}{\pi^2 g(1)} \sum_{\substack{a \text{ odd}, \\ a>0}} 2^{-\lfloor \lg a \rfloor} G\left(\frac{1}{a}\right) \tag{9}$$

where g is a nonzero fixed point of $\mathcal{V}_2$ (i.e. $g = \mathcal{V}_2 g \neq 0$) and $G(x) = \int_0^x g(t)\,dt$.

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

6.1 A Conjecture of Vallée

Let $\lambda = f(1)$, where f is the limiting probability density (conjectured to exist) as in §4. Vallée (see Knuth [9, §4.5.2(61)]) conjectured that $\lambda/b = 2\ln 2/\pi^2$, or equivalently that

$$K = \frac{4\ln 2}{\pi^2 \lambda} . \tag{10}$$

Vallée proved the conjecture under the assumption that the operator $\mathcal{B}_s$ satisfies a certain spectral condition which is known to be satisfied by $\mathcal{V}_s$.

7 Numerical Results

Using an improvement of the "discretization method" of my 1976 paper [1], with Romberg extrapolation and the equivalent of more than 50 decimal places (50D) working precision, we computed the limiting probability distribution F, then K (using (2) and (4)), $\lambda = f(1)$, 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.

Vallée's conjecture (10) 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. Details of the numerical computation are given in [2].

8 Some Relations Between Fixed Points

In this section we *assume* that f is a fixed point of the operator $\mathcal{B}_2$, $g = \mathcal{U}_2 f$ as in §5.7 is a fixed point of the operator $\mathcal{V}_2$, and both f and g are analytic functions

(not necessarily regular at $x = 0$). Using analyticity we extend the domains of f, g etc to include the positive real axis $(0, +\infty)$. Let

$$F(x) = \int_0^x f(t)\,dt \text{ and } G(x) = \int_0^x g(t)\,dt$$

be the corresponding integrals. By scaling, we can assume that $F(1) = 1$ but then we are not free to scale g (see (12) and (13) below).

From the definition (5) of $\mathcal{U}_s$, we have

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

so, integrating with respect to x and simplifying,

$$G(x) = \sum_{k=1}^{\infty} 2^{-k} \widetilde{F}\left(\frac{1}{1+2^k x}\right) . \tag{11}$$

Although our derivation of (11) assumes $x \in [0, 1]$, we can use (11) to give an analytic continuation of $G(x)$. Allowing x to approach $+\infty$, we see that there exists $\lim_{x\to+\infty} G(x) = G(+\infty)$ say, and

$$G(+\infty) = 1 . \tag{12}$$

From the definitions of $\mathcal{B}_2$ and $\mathcal{U}_2$, we have

$$f(1) = 2g(1) = 2\sum_{k\ge1} \left(\frac{1}{1+2^k}\right)^2 f\left(\frac{1}{1+2^k}\right) . \tag{13}$$

Using these results, it is easy to prove:

Theorem 2 Under the assumptions stated at the beginning of this section, the expressions (9) and (10) are equivalent. □

Remarks. As noted in §6.1, Vallée proved (10) under an assumption about the spectrum of $\mathcal{B}_s$. Our proof of Theorem 2 (given in detail in [2]) is more direct. We are not able to prove the equivalence of (2) and (10), but (as described in §7) it has been verified numerically to 40D.

9 Correcting an Error

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

$$F_n(x) = \alpha_n(x) \lg(x) + \beta_n(x) , \tag{14}$$

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 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

$$K = 0.70597\ 12461\ 01945\ 99986\cdots$$

but I found

$$K = 0.70597\ 12461\ 01916\ 39152\cdots$$

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 [1, Lemma 3.1], which is wrong (and corrected in the solution to ex. 4.5.2.29 of Knuth[9, third edition]). In order to explain the error, we need to consider Mellin transforms (a very useful tool in average-case analysis, see [3]).

9.1 Mellin Transforms and Mellin Inversion

The *Mellin transform* of a function $g(x)$ is defined by $g^*(s) = \int_0^\infty g(x)x^{s-1}dx$. It is easy to see that, if $f(x) = \sum_{k\geq 1} 2^{-k}g(2^k x)$, then the Mellin transform of f is

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

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 can express

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

as a sum of residues of

$$\Big(\frac{\pi}{\sin\pi s}\Big)\frac{x^{-s}}{2^{s+1}-1} \tag{15}$$

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

$$f(x) = 1 + x\lg x + \frac{x}{2} + xP(\lg x) - \frac{2}{1}x^2 + \frac{4}{3}x^3 - \cdots\ , \tag{16}$$

where

$$P(t) = \frac{2\pi}{\ln 2}\sum_{n=1}^{\infty}\frac{\sin 2n\pi t}{\sinh(2n\pi^2/\ln 2)}\ . \tag{17}$$

9.2 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 [1, Lemma 3.1], the term $xP(\lg x)$ in (16) was omitted. Essentially, the poles of (15) off the real axis at $s = -1 \pm 2\pi in/\ln 2$, $(n = 1, 2, \ldots)$ were ignored.

Because of the sinh term in the denominator of (17), the residues at the non-real poles are tiny, and numerical computations performed using single-precision floating-point arithmetic did not reveal the error.

9.3 Details of Corrections

The function $f(x)$ above is called $D_1(x)$ in [1]. In (3.29) of Lemma 3.1 of that paper, the expression for $D_1(x)$ is missing the term $xP(\lg x)$.

Equation (3.8) of the paper is (correctly)

$$F_n(x) = 1 + D_n(1/x) - D_n(x)$$

so in Corollary 3.2 the expression for $F_1(x)$ is missing a term $-xP(\lg x)$.

The statement following Corollary 3.2, that "In principle we could obtain $F_2(x), F_3(x)$, etc in the same way as $F_1(x)$" is dubious because it is not clear how to handle the terms involving $P(\lg x)$.

Corollary 3.3, that $F_{n+1} \neq F_n$, is probably correct, but the proof given is incorrect because it assumes the incorrect form (14) for $F_n(x)$.

10 Conclusion and Open Problems

Since Vallée's recent work [12, 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 $\mathcal{V}_2$. Are the various expressions for K given above all provably correct (only (9) has been proved) ? Is there an algorithm for the numerical computation of K which is asymptotically faster than the one we used to obtain the results of §7?

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.

It would be interesting to compute the spectra of $\mathcal{B}_2$ and $\mathcal{V}_2$ numerically, and compare with the classical case, where the spectrum is real and the eigenvalues appear to alternate in sign [4].

In order to give rigorous numerical bounds on the spectra of $\mathcal{B}_2$ and $\mathcal{V}_2$, we need to bound the error caused by making finite-dimensional approximations to these operators. This may not be so difficult for $\mathcal{V}_2$ as for $\mathcal{B}_2$.

Acknowledgements

Thanks to Don Knuth for encouraging me to correct and extend my 1976 results for the third edition of *Seminumerical Algorithms*, and Brigitte Vallée for sharing her conjectures and results with me before their publication.

References

[1] R. 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.

[2] Richard P. Brent, *Further analysis of the Binary Euclidean algorithm*, Technical Report, Oxford University Computing Laboratory, November 1999. `ftp://ftp.comlab.ox.ac.uk/pub/Documents/techpapers/Richard.Brent/rpb183tr.ps.gz`

[3] P. Flajolet and R. Sedgewick, *The Average Case Analysis of Algorithms: Mellin Transform Asymptotics*, Report 2956, INRIA Rocquencourt, August 1996. `http://pauillac.inria.fr/algo/flajolet/Publications/anacombi4.ps.gz`

[4] P. Flajolet and B. Vallée, On the Gauss-Kuzmin-Wirsing constant, manuscript, 29 October 1995. `http://pauillac.inria.fr/algo/flajolet/Publications/gauss-kuzmin.ps.gz`

[5] P. Flajolet and B. Vallée, Continued fraction algorithms, functional operators and structure constants, *Theoretical Computer Science* **194** (1998), 1–34. See also `http://www-rocq.inria.fr/algo/flajolet/Publications/RR2931.ps.gz`

[6] M. Foley and C. A. R. Hoare, Proof of a recursive program: Quicksort, *Comp. J.* **14**, 4 (1971), 391–395.

[7] C. A. R. Hoare, Quicksort, *Comp. J.* **5**, 1 (1962), 10–15.

[8] C. A. R. Hoare, Proof of a program: FIND, *Comm. ACM* **14**, 1 (1971), 39–45.

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

[10] D. Ruelle, *Thermodynamic Formalism*, Addison Wesley, Menlo Park, 1978.

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

[12] B. Vallée, The complete analysis of the binary Euclidean algorithm, *Proc. ANTS'98*, *Lecture Notes in Computer Science* **1423**, Springer-Verlag, 1998, 77–94.

[13] Brigitte Vallée, Dynamics of the binary Euclidean algorithm: functional analysis and operators, *Algorithmica* **22** (1998), 660–685. `http://www.info.unicaen.fr/~brigitte/Publications/bin-gcd.ps`