Twenty years' analysis of the Binary Euclidean Algorithm*

Dedicated to Tony Hoare on the occasion of his departure from Oxford

Richard P. Brent

Computing Laboratory
Oxford University

15 September 1999

*Presented at a Symposium in Celebration of the Work of Tony Hoare, Oxford, 13–15 September 1999.
 rpb183t

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.

I will describe the binary algorithm and consider its average case behaviour. In particular, I will describe the recent discovery of an error which I made in 1976, discuss some recent results of Brigitte Vallée, and describe a numerical computation which verifies a conjecture of Vallée to 40 decimal places.

Outline

- Introduction
- The binary Euclidean algorithm
  - First formulation (Algorithm B)
  - History
- Continuous model
  - Conjectured/empirical results
- Another formulation (Algorithm V)
- Useful operators
  - Relation between the operators
  - Fixed points
  - Vallée's conjectures and results
- Numerical results
- Tracking down an error
- Moral
- Conclusion and open problems

Introduction – Quicksort etc

Why a paper on average-case analysis of algorithms ?

Tony Hoare and I both have an interest in Euclidean algorithms, and, since the publication of his 1962 paper on Quicksort, Tony can hardly deny an interest in the average-case analysis of algorithms. Although Tony proved the *correctness* of Quicksort in his usual thorough and convincing manner, the real significance of Quicksort is its excellent *average case* behaviour. Tony was well aware of this: a large part of his 1962 paper is devoted to the average case analysis of quicksort.

Personally, I prefer Heapsort, since (unlike Quicksort) it has guaranteed worst case time bound $O(n \log n)$ to sort n items. After the Proceedings paper was written, it emerged in a tea-room conversation that Tony was closely involved in the discovery of Heapsort. He and Williams were in the same room; Williams found heapsort then Tony found a quicker sort!

The Topic $\cdots$

In any case, given Tony's interest in 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 (relatively) new to say about the *binary* (not the classical) Euclidean algorithm.

Notation

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

N, n, 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 $2^j \mid u$.

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.

Definition

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, so for simplicity we assume that u and v are *odd* positive integers.

Following is a simplified version of the algorithm given in Knuth, §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.

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 !

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

[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.

Plausible Assumption

We make the plausible (but not proved) assumption[2] that $\mathrm{Val}_2(t)$ takes the value k with probability 2^{-k} at step B2.

It 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.

[2] Vallée does not need to make this assumption. Her results will be mentioned 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 $\widetilde{F}_n(x) = 1 - F_n(x)$ is

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

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

The Recurrence for f_n

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

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

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 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 my 1976 paper 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

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

For my direct proof of (1), see Knuth, third edition, §4.5.2.

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 ≠ v do
   begin
   while u < v do
      begin
      j ← Val2(v − u);
      v ← (v − u)/2^j;
      end;
   u ↔ v;
   end;
return u.
```

Continued Fractions

Vallée (*Algorithmica*, 1998) 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 $0 < a_j < 2^{k_j}$.

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

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

$$\widetilde{\mathcal{U}}_s[f](x) = \left(\frac{1}{x} \right)^s \mathcal{U}_s[f]\left(\frac{1}{x} \right), \qquad (3)$$

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

$$\mathcal{V}_s[f](x) = \sum_{k \ge 1} \sum_{\substack{a \text{ odd}, \\ 0 < a < 2^k}} \left(\frac{1}{a + 2^k x} \right)^s f\left(\frac{1}{a + 2^k x} \right) . \qquad (4)$$

In these definitions s is a complex variable, and the operators are called Ruelle operators. 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.

History and Notation

$\mathcal{B}_2$ (denoted T) was introduced in my 1976 paper and was 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.*

Relation Between the Operators

The operators are closely related, as the following results show.

Lemma 1

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

An algebraic proof of Lemma 1 is given in the Proceedings.

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.

Consequence of Lemma 1

The following Theorem gives a simple relationship between $\mathcal{B}_s$, $\mathcal{V}_s$ and $\mathcal{U}_s$. The proof is immediate from Lemma 1 and the definitions of the operators.

Theorem 1

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

Fixed Points

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 a recent result of Vallée 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 $\mathcal{U}_2$, we have

$$\lambda = f(1) = 2g(1)$$

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

Some 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 the only expression for K which has been rigorously proved.

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.

Warning: Knuth uses $G(x)$ for our $\widetilde{F}(x) = 1 - F(x)$! Unfortunately Knuth and Vallée use incompatible notation. We have followed Vallée, with the exception that we do not assume the normalisation $G(1) = 1$.

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

$$K = \frac{4\ln 2}{\pi^2 \lambda} \,. \tag{5}$$

Vallée proved the conjecture under the assumption that the operator $\mathcal{B}_s$ satisfies a certain spectral condition.

Numerical Results

Using an improvement of the "discretization method" of my 1976 paper, 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.

Vallée's conjecture (5) 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!

Correcting an Error

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

$$F_n(x) = \alpha_n(x)\lg(x) + \beta_n(x) \,, \tag{6}$$

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, 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

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

It is easy to see that, if

$$f(x) = \sum_{k\ge1} 2^{-k} g(2^k x) \,,$$

then the Mellin transform of f is

$$f^*(s) = \sum_{k\ge1} 2^{-k(s+1)} g^*(s) = \frac{g^*(s)}{2^{s+1}-1} \,.$$

Mellin Inversion

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

$$f(x) = \sum_{k\ge1} \frac{2^{-k}}{1+2^k x}$$

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

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

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 Lemma 3.1 of my 1976 paper, the term $xP(\lg x)$ was omitted.

Essentially, the poles off the real axis at

$$s = -1 \pm \frac{2\pi i n}{\ln 2} \,, \quad n = 1, 2, \ldots$$

were ignored[3].

Because the residues at the non-real 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.

[3] In fact, the incorrect result was obtained without using Mellin transforms. If I had used them I probably would have obtained the correct result!

The Moral of this Tale

Computing results to (unreasonably) high precision by more than one method can be useful.

Small discrepancies should be explained, not ignored !

Conclusion and Open Problems

Since Vallée's recent work, 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 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.

Vallée has proved related results for some other algorithms (variants of the Euclidean algorithm, algorithms for computing the Jacobi symbol), but many analogous questions remain open.

Acknowledgements

Thanks to

- Don Knuth for encouraging me to correct and extend my 1976 results for the third edition of *Seminumerical Algorithms*,
- Brigitte Vallée for discussions and correspondence regarding her unpublished results and conjectures,
- Philippe Flajolet for his excellent notes on Mellin transforms.

Further details of the new results presented in this lecture will appear in a Technical Report.

References

[1] 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.

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

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

[4] Brigitte Vallée, The complete analysis of the Binary Euclidean Algorithm, *Proc. ANTS'98*, *Lecture Notes in Computer Science* **1423**, Springer-Verlag, 1998, 77–94.

[5] Brigitte Vallée, Dynamics of the binary Euclidean algorithm: functional analysis and operators, *Algorithmica* **22** (1998), 660–685.