U.S. DEPARTMENT OF COMMERCE
National Technical Information Service

AD-A029 130

Analysis of the Binary Euclidean Algorithm

Carnegie-Mellon Univ.

June 1976

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ANALYSIS OF THE BINARY EUCLIDEAN ALGORITHM		5. TYPE OF REPORT & PERIOD COVERED Interim
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard P. Brent		8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-0370 NR 044-422
9. PERFORMING ORGANIZATION NAME AND ADDRESS Carnegie-Mellon University Computer Science Dept. Pittsburgh, PA 15213		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, VA 22217		12. REPORT DATE June 1976
		13. NUMBER OF PAGES 37
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

None

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PRICES SUBJECT TO CHANGE

ANALYSIS OF THE BINARY EUCLIDEAN ALGORITHM

Richard P. Brent

June 1976

DEPARTMENT
of
COMPUTER SCIENCE

Carnegie-Mellon University

ANALYSIS OF THE BINARY EUCLIDEAN ALGORITHM

Richard P. Brent

June 1976

To appear in New Directions and Recent Results in Algorithms and Complexity, Edited by J. F. Traub, Academic Press, 1976.

ANALYSIS OF THE BINARY EUCLIDEAN ALGORITHM

Richard P. Brent
Australian National University
and Carnegie-Mellon University

1. Introduction

The binary Euclidean algorithm of Silver and Terzian [62] and Stein [67] finds the greatest common divisor (GCD) of two integers, using the arithmetic operations of subtraction and right shifting (i.e., division by 2). Unlike the classical Euclidean algorithm, no divisions are required. Thus, an iteration of the binary algorithm is faster than an iteration of the classical algorithm on many binary computers.

The classical algorithm has been exhaustively analyzed from the time of Gauss: see, for example, Dixon [70, 71], Gauss [12], Heilbronn [68], Khinchin [35a, 35b, 36], Kusmin [28], Lévy [29], Szüsz [61], Tonkov [74] and Wirsing [74]. A good survey is given in Knuth [69]. The theory of the binary algorithm is much less satisfactory. Knuth [69] analyzed a "lattice-point" model which is, unfortunately, only a crude and pessimistic approximation to the actual algorithm. In this paper we analyze a continuous model of the binary algorithm and find the expected number of iterations. The results agree with the observed behavior of the algorithm much better than those predicted by Knuth's "lattice-point" model.

The binary Euclidean algorithm for finding the GCD of positive integers u and v is given in Knuth [69, Sec. 4.5.2, Alg. B]. After steps B1 to B5 of the algorithm have been performed once, the problem is reduced to that of finding

This research was supported in part by the National Science Foundation under Grant MCS75-222-55 and the Office of Naval Research under Contract N00014-76-C-0370, NR 044-422.

the GCD of two odd integers. Thus, we assume here that u and v are odd, and the algorithm is as follows.

RS Binary Algorithm

```
     [n ← 0;]
L1:  t ← |u - v|;
     if t = 0 then return u as the GCD and halt;
L2:  t ← t/2;
     if t is even then go to L2;
L3:  [n ← n + 1;]
     if u ≥ v then u ← t else v ← t;
     go to L1.
```

The statements in square brackets are not essential. We say that one "iteration" is one execution of step L3, so n counts the number of iterations. To distinguish the different values taken by the variables u and v, we let u_n be the value of u at iteration n, etc. Step L2 is executed twice as often as step L3, on the average, but the L2 loop merely shifts t right until it becomes odd, and this may be done efficiently on a binary computer.

Let $x_n = \min(u_n, v_n)/\max(u_n, v_n)$, and let $F_n(x)$ be the probability distribution function of x_n. We assume that u_0 and v_0 are uniformly and independently distributed in $(0, N)$ (with the constraint that they are odd), and consider the continuous approximation obtained by letting $N \to \infty$. In Section 2 we derive a recurrence relation for the continuous distributions $F_n(x)$.

In Section 3 we show that $F_n(x) = \alpha_n(x)\lg(x)^* + \beta_n(x)$, where $\alpha_n(x)$ and $\beta_n(x)$ are analytic and satisfy certain recurrence relations. An explicit expression for $\alpha_n(x)$ is given

*Throughout this paper, lg(x) denotes $\log_2(x)$.

in Section 4.

In Section 5 we consider the equivalent recurrence $f_{n+1} = Tf_n$ for the probability density functions $f_n(x) = F_n'(x)$. We show that $\|f_{n+2}-f_{n+1}\| < \|f_{n+1}-f_n\|$ for a certain norm. Numerical evidence, described in Section 7, suggests that convergence is rapid. Thus, it is likely that f_n tends to a limiting density f_∞, though we have not been able to prove this.

The expected number of iterations is asymptotically $K\lg(N)$ for large N, and an expression for the constant K is derived in Section 6. The theoretical value of $K \simeq 0.706$ agrees with values obtained numerically for moderate values of N. The numerical results are described in Section 7.

Finally, in Section 8 we consider another algorithm which uses only shifts and subtractions. The algorithm uses left shifts (i.e., multiplication by 2) instead of right shifts, so we call it the left-shift binary Euclidean algorithm (LS algorithm for short). We show that the expected number of iterations is slightly greater than for the (right-shift) binary Euclidean algorithm. However, the LS algorithm is worth considering for use on a computer with a "normalize" instruction, as the left-shifting loop may be replaced by one instruction. Either of the binary algorithms could be implemented in hardware (or microprogrammed) with approximately the same expense as integer division.

We consider only single-precision integer GCD computations here. For polynomial and multiple-precision integer GCD algorithms, see Collins [74], Schönhage [71] and Knuth [69].

2. The Recurrence for F_n

For notational simplicity we write u for u_n and u' for

u_{n+1}, etc. Also, there is no loss of generality in assuming that $u \geq v$. The iteration terminates if $u = v$, so we assume that $u > v$. Thus, $x = v/u$, $t = 2^{-k}(u-v)$, and $x' = \min(t,v)/\max(t,v)$, where $k \geq 1$ is chosen so that t is an odd integer.

Let $P(E)$ denote the probability of an event E. By definition, $F_{n+1}(y) = P(x' \leq y)$, but $x' = \min(t/v, v/t)$, so

$$(2.1) \quad F_{n+1}(y) = P(t/v \leq y \vee v/t \leq y)$$

$$(2.2) \quad = P(t \leq vy \vee v \leq ty).$$

It may be shown that, for $K = 1,2,\ldots,$

$$(2.3) \quad \lim_{N\to\infty} P(k = K) = 2^{-K}.$$

Thus,

$$(2.4) \quad F_{n+1}(y) = \sum_{k=1}^{\infty} 2^{-k}\, P(2^{-k}(u-v) \leq vy \vee v \leq 2^{-k}(u-v)y)$$

$$(2.5) \quad = \sum_{k=1}^{\infty} 2^{-k} P(2^{-k}(1-x) \leq xy \vee x \leq 2^{-k}(1-x)y).$$

Since $x \in (0,1)$, we have $2^{-k}(1-x) \leq xy$ iff $x \geq 1/(1+2^k y)$, and $x \leq 2^{-k}(1-x)y$ iff $x \leq 1/(1+2^k/y)$. Also, assuming $y \in (0,1)$, we have $1/(1+2^k/y) \leq 1/(1+2^k y)$. Thus, from (2.5),

$$(2.6) \quad F_{n+1}(y) = \sum_{k=1}^{\infty} 2^{-k}[1-P(1/(1+2^k/y) \leq x \leq 1/(1+2^k y))].$$

Since x has distribution function F_n, this gives the interesting recurrence relation

$$(2.7)\begin{cases} F_{n+1}(y) = 1 + \sum_{k=1}^{\infty} 2^{-k}\left[F_n\left(\frac{y}{2^k+y}\right) - F_n\left(\frac{1}{1+2^k y}\right)\right], \\ F_0(y) = y, \end{cases}$$

for $n \geq 0$ and $y \in [0,1]$.

The corresponding recurrence for the classical algorithm is

$$(2.8) \quad G_{n+1}(x) = \sum_{k=1}^{\infty} [G_n(1/k) - G_n(1/(k+x))].$$

This was derived by Gauss [12], who conjectured that

$$(2.9) \quad \lim_{n\to\infty} G_n(x) = \lg(1+x),$$

which was proved by Kusmin [28]. Sharper results were later obtained by Lévy [29] and Szüsz [61]. Finally, Wirsing [74] proved that

$$(2.10) \quad G_n(x) = \lg(1+x) + O(\lambda^n x(1-x))$$

as $n \to \infty$, uniformly for all $x \in [0,1]$, where $\lambda \simeq 0.3036630029$ is a certain constant in $(0,1)$.

We conjecture that a similar result holds for $F_n(x)$. For a reason which will be clear later, the term $x(1-x)$ in (2.10) must be replaced by $x|\ln(x)|$.

<u>Conjecture 2.1</u>

There exists $F_\infty(x) = \lim_{n\to\infty} F_n(x)$, and

$$(2.11) \quad F_n(x) = F_\infty(x) + O(\lambda^n x|\ln(x)|)$$

as $n \to \infty$, uniformly for all $x \in (0,1]$, where λ is some constant in (0,1).

The theoretical evidence for Conjecture 2.1 is given in the next three sections, and some numerical evidence is given in Section 7.

Differentiating (2.7), we obtain the recurrence

$$
(2.12)\quad \begin{cases} f_{n+1}(x) = \displaystyle\sum_{k=1}^{\infty} \left[\left(\frac{1}{2^k+x}\right)^2 f_n\left(\frac{x}{2^k+x}\right) + \left(\frac{1}{1+2^k x}\right)^2 f_n\left(\frac{1}{1+2^k x}\right)\right] \\ f_0(x) = 1 \end{cases}
$$

for the probability density functions $f_n(x) = F_n'(x)$, $x \in (0,1]$, $n \geq 0$. The recurrences (2.7) and (2.12) are equivalent, but in Section 3 we prefer to work with (2.7) and consider the form of $F_n(x)$. Results for $f_n(x)$ are easily deduced by differentiation.

3. The Distribution Functions F_n

The following theorem gives the form of $F_n(x)$ for finite n.

Theorem 3.1

For all $n \geq 0$ and $x \in (0,1]$,

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

where $\alpha_n(x)$ and $\beta_n(x)$ are analytic and regular in $|x| < 1$, and $\alpha_n(0) = \beta_n(0) = 0$. Also, $\alpha_0(x) = 0$ and

$$
(3.2)\quad 2\alpha_{n+1}(2x) - \alpha_{n+1}(x) = \alpha_n\left(\frac{x}{1+x}\right) - 3f_n(1)x.
$$

Proof

Define $D_0(x) = 0$ and

$$(3.3)\quad D_{n+1}(x) = \sum_{k=1}^{\infty} 2^{-k} F_n\left(\frac{1}{1+2^k x}\right) .$$

We assume that

$$(3.4)\quad F_m(x) = \alpha_m(x)\lg(x) + \beta_m(x),$$

$$(3.5)\quad D_m(x) = 1 + \gamma_m(x)\lg(x) + \delta_m(x),$$

$$(3.6)\quad D_m(1/x) = \epsilon_m(x)\lg(x) + \phi_m(x),$$

and

$$(3.7)\quad F_m\left(\frac{1}{1+x}\right) = 1 + \eta_m(x)$$

for $m < n$, where $\alpha_m(x),\ldots,\eta_m(x)$ are analytic and regular for $|x| < 1$, and vanish at $x = 0$. We shall prove the corresponding result for $m = n$, so (3.1) will follow by induction. The results (3.4) to (3.7) are trivially true for $m = 0$, so we may assume $n > 0$.

From (2.7) and (3.3) we have

$$(3.8)\quad F_n(x) = 1 + D_n(1/x) - D_n(x),$$

so if $\alpha_n(x),\ldots,\phi_n(x)$ are regular at $x = 0$ we must have

$$(3.9)\quad \alpha_n(x) = \epsilon_n(x) - \gamma_n(x)$$

and

$$(3.10)\quad \beta_n(x) = \phi_n(x) - \delta_n(x).$$

From (3.3) we also have

$$(3.11)\quad 2D_n\left(\frac{1}{2x}\right) - D_n\left(\frac{1}{x}\right) = F_{n-1}\left(\frac{x}{1+x}\right) ,$$

so in the same way we find that

$$(3.12)\quad 2\epsilon_n(2x) - \epsilon_n(x) = \alpha_{n-1}\left(\frac{x}{1+x}\right)$$

and

$$(3.13)\quad 2\epsilon_n(2x) + 2\phi_n(2x) - \phi_n(x)$$

$$= \beta_{n-1}\left(\frac{x}{1+x}\right) - \alpha_{n-1}\left(\frac{x}{1+x}\right)\ \lg(1+x).$$

By the inductive hypothesis, the right sides of (3.12) and (3.13) are analytic and regular at $x = 0$. Let the Taylor expansion of $\alpha_m(x)$ be

$$(3.14)\quad \alpha_m(x) = \sum_{j=1}^{\infty} \alpha_{m,j} x^j,$$

and similarly for $\beta_m(x),\ldots,\eta_m(x)$. By equating coefficients we see that analytic solutions $\epsilon_n(x)$ and $\phi_n(x)$ satisfying (3.12) and (3.13) exist, and are given by

$$(3.15)\quad \epsilon_{n,j} = \frac{(-1)^j}{2^{j+1}-1} \sum_{k=1}^{j} (-1)^k \alpha_{n-1,k}\binom{j-1}{k-1}$$

and

$$(3.16)\quad \phi_{n,j} = \left(\frac{1}{2^{j+1}-1}\right)\left[-2^{j+1}\epsilon_{n,j} + \sum_{k=1}^{j-1}\left(\frac{2^{k+1}-1}{\ln 2}\right)\frac{(-1)^{j+k}\epsilon_{n,k}}{j-k}\right.$$

$$\left. + \sum_{k=1}^{j} (-1)^{j+k}\beta_{n-1,k}\binom{j-1}{k-1}\right],$$

where $j = 1,2,\ldots$. Thus, $\epsilon_n(x)$ and $\phi_n(x)$ are determined by $\alpha_{n-1}(x)$ and $\beta_{n-1}(x)$, and are analytic and regular in $|x| < 1$.

From (3.3) and (3.8),

$$(3.17)\quad F_n(y) = 1 - \frac{1}{2} F_{n-1}\left(\frac{1}{1+2y}\right) + \frac{1}{2} F_{n-1}\left(\frac{y}{2+y}\right) - \frac{1}{2} D_n(2y) + \frac{1}{2} D_n\left(\frac{2}{y}\right).$$

Substituting $y = 1/(1+x)$ gives

$$(3.18)\quad F_n\left(\frac{1}{1+x}\right) = 1 - \frac{1}{2} F_{n-1}\left(\frac{1+x}{3+x}\right) + \frac{1}{2} F_{n-1}\left(\frac{1}{3+2x}\right) - \frac{1}{2} D_n\left(\frac{2}{1+x}\right) + \frac{1}{2} D_n(2+2x).$$

By the inductive hypothesis,

$$(3.19)\quad F_{n-1}\left(\frac{1}{3}+y\right) = \alpha_{n-1}\left(\frac{1}{3}+y\right)\lg\left(\frac{1}{3}+y\right) + \beta_{n-1}\left(\frac{1}{3}+y\right),$$

so substituting $y = \left(\frac{2x}{3(3+x)}\right)$ and $\left(\frac{-2x}{3(3+2x)}\right)$ gives power series for $F_{n-1}\left(\frac{1+x}{3+x}\right)$ and $F_{n-1}\left(\frac{1}{3+2x}\right)$ respectively. Also,

$$(3.20)\quad D_n\left(\frac{2}{1+x}\right) = \epsilon_n\left(\frac{1+x}{2}\right)\lg\left(\frac{1+x}{2}\right) + \phi_n\left(\frac{1+x}{2}\right)$$

and

$$(3.21)\quad D_n(2+2x) = -\epsilon_n\left(\frac{1}{2+2x}\right)\lg(2+2x) + \phi_n\left(\frac{1}{2+2x}\right).$$

Thus, $F_n\left(\frac{1}{1+x}\right) = 1 + \eta_n(x)$,

where $\eta_n(x)$ is analytic and regular in $|x| < 1$.

It remains to consider $\gamma_n(x)$ and $\delta_n(x)$. From (3.3),

$$(3.22)\quad 2D_n\left(\frac{x}{2}\right) - D_n(x) = 1 + \eta_{n-1}(x),$$

so

$$(3.23)\quad 2\gamma_n\left(\frac{x}{2}\right) - \gamma_n(x) = 0$$

and

(3.24) $2\delta_n\left(\frac{x}{2}\right) - \delta_n(x) - 2\gamma_n\left(\frac{x}{2}\right) = \eta_{n-1}(x).$

Thus, we have the analytic solutions

(3.25) $\gamma_n(x) = \gamma_{n,1}x = -\eta_{n-1,1}x = f_{n-1}(1)x$

and

(3.26) $\delta_{n,j} = \left(\frac{1}{2^{1-j}-1}\right)\eta_{n-1,j}$

for $j \geq 2$. The constant $\delta_{n,1}$ may be determined from the relations $\beta_{n,1} = \phi_{n,1} - \delta_{n,1}$ and

(3.27) $F_n\left(\frac{1}{2}\right) = 1 - \frac{1}{2}F_{n-1}\left(\frac{1}{2}\right) - \frac{1}{4}F_{n-1}\left(\frac{1}{3}\right) + \frac{3}{4}D_n(2),$

obtained from (3.10) and (3.17) respectively.

We have now proved (3.4) to (3.7) for $m = n$, so the first part of the theorem follows by induction. (3.2) follows easily from (3.9), (3.12) and (3.25), so the proof is complete.

It is interesting to obtain an explicit formula for $F_1(x)$. First we need a lemma.

<u>Lemma 3.1</u>

If

(3.28) $D_1(x) = \sum_{k=1}^{\infty} 2^{-k}/(1+2^k x),$

then

(3.29) $D_1(x) = x\lg x + 1 + \frac{x}{2} - \frac{x^2}{1+x} - \sum_{j=2}^{\infty} \frac{(-x)^j}{2^{j-1}-1}$

for $0 < x < 2$, and

(3.30) $$D_1(1/x) = -\sum_{j=1}^{\infty} \frac{(-x)^j}{2^{j+1}-1}$$

for $|x| < 2$.

<u>Proof</u>

From (3.5) and (3.6), we have

(3.31) $$D_1(x) = 1 + \gamma_1(x)\lg(x) + \delta_1(x)$$

and

(3.32) $$D_1(1/x) = \epsilon_1(x)\lg(x) + \phi_1(x).$$

Since $\alpha_0(x) = 0$ and $\beta_0(x) = x$, (3.15) gives $\epsilon_1(x) = 0$, and (3.16) gives $\phi_{1,j} = (-1)^{j+1}/(2^{j+1}-1)$. This establishes (3.30).

From (3.25), $\gamma_1(x) = x$. Also, since $\eta_0(x) = 1/(1+x)$, (3.26) gives

(3.33) $$\delta_{1,j} = (-1)^j/(2^{1-j}-1)$$

for $j \geq 2$. Thus

(3.34) $$D_1(x) = x\lg x + 1 + \delta_{1,1}x - \sum_{j=2}^{\infty} \frac{(-x)^j}{1-2^{1-j}} .$$

The series in (3.34) converges for $|x| < 1$. Subtracting and adding $\frac{x^2}{1+x} = \sum_{j=2}^{\infty} (-x)^j$ gives

(3.35) $$D_1(x) = x\lg x + 1 + \delta_{1,1}x - \frac{x^2}{1+x} - \sum_{j=2}^{\infty} \frac{(-x)^j}{2^{j-1}-1} ,$$

where the last series converges for $|x| < 2$. By analytic continuation, (3.35) holds for $0 < x < 2$. The constant

$\delta_{1,1} = \frac{1}{2}$ may be determined by equating (3.30) and (3.35) with $x = 1$. Thus, (3.29) follows.

Corollary 3.2

$$F_1(x) = -x\lg(x) + \frac{x(5x-1)}{6(1+x)} + 3\sum_{j=2}^{\infty} \frac{(-x)^j\, 2^{j-1}}{(2^{j-1}-1)(2^{j+1}-1)} .$$

Proof

This follows from (3.8) and Lemma 3.1.

In principle we could obtain $F_2(x)$, $F_3(x)$, etc. in the same way as $F_1(x)$. However, the details become very complicated. The situation is similar for the classical algorithm: see Knuth [69].

Corollary 3.3

For all $n \geq 0$ and some $x \in [0,1]$, $F_{n+1}(x) \neq F_n(x)$.

Proof

Suppose, by way of contradiction, that $F_{n+1}(x) = F_n(x)$ for all $x \in [0,1]$. From Corollary 3.2, $n \neq 0$. From Theorem 3.1, $\alpha_{n+1}(x) = \alpha_n(x)$. Thus, from (3.2),

$$(3.36) \quad \alpha_{n-1}\left(\frac{x}{1+x}\right) - 3\, f_{n-1}(1)x = \alpha_n\left(\frac{x}{1+x}\right) - 3f_n(1)x$$

for $|x| < 1$.

Substituting $y = x/(1+x)$ we obtain

$$(3.37) \quad \alpha_n(y) - \alpha_{n-1}(y) = 3(f_n(1) - f_{n-1}(1))y/(1-y)$$

for $|y| < \frac{1}{2}$. By analytic continuation, (3.37) holds for $|y| < 1$. However, from (3.2) it follows that $\alpha_n(y)$ and $\alpha_{n-1}(y)$ are regular at $y = 1$, so we must have $f_n(1) = f_{n-1}(1)$, and thus $\alpha_n(y) = \alpha_{n-1}(y)$. Continuing in this way, we finally obtain $\alpha_1(x) = \alpha_0(x)$, which contradicts Corollary 3.2 ($\alpha_1(x) = x$, $\alpha_0(x) = 0$). Thus, the original assumption was false, and $F_{n+1}(x) \neq F_n(x)$ for some $x \in [0,1]$.

4. Solution of the Recurrence for α_n

In this section we solve the recurrence (3.2) explicitly. The method used here can obviously be generalized. However, we have not been able to solve the recurrence for $\beta_n(x)$ analytically.

Define $p(0) = 0$,

$p(2n) = p(n)$,

and $p(2n+1) = p(n) + 1$.

Thus, $p(n)$ is the number of one-bits in the binary representation of $n \geq 0$.

Theorem 4.1

Suppose $\alpha_0(x) = 0$ and

$$(4.1) \quad 2\alpha_{n+1}(2x) - \alpha_{n+1}(x) = \alpha_n\left(\frac{x}{1+x}\right) + c_{n+1}x$$

for $n \geq 0$, where $c_1, c_2, \ldots$ are constants, $c_0 = c_{-1} = \ldots = 0$, and $\alpha_{n+1}(x)$ is analytic and regular at $x = 0$. Then

$$(4.2) \quad \alpha_n(x) = \frac{x}{4} \sum_{k=0}^{\infty} 2^{-k} \sum_{j=0}^{2^k-1} \frac{c_{n-p(j)}}{2^k+jx}$$

for all $n \geq 0$ and all $x \notin (-\infty, -1]$.

Note

(4.1) is the same as (3.2) if $c_{n+1} = -3f_n(1)$ for $n \geq 0$. Thus, (4.2) gives an explicit solution of (3.2) in terms of $f_0(1), f_1(1), \ldots, f_{n-1}(1)$.

Proof of Theorem 4.1

The result is true for $n = 0$, and the analytic solution of (4.1) which is regular at $x = 0$ is clearly unique. Thus, it is sufficient to verify that if $\alpha_n(x)$ and $\alpha_{n+1}(x)$ are defined by (4.2) then (4.1) holds. From (4.2) we have

$$2\alpha_{n+1}(2x) - \alpha_{n+1}(x) - \alpha_n\left(\frac{x}{1+x}\right)$$

$$= \frac{x}{4} \sum_{k=-1}^{\infty} 2^{-k} \sum_{j=0}^{2^{k+1}-1} \frac{c_{n+1-p(j)}}{2^k+jx}$$

$$- \frac{x}{4} \sum_{k=0}^{\infty} 2^{-k} \sum_{j=0}^{2^k-1} \frac{c_{n+1-p(j)}}{2^k+jx}$$

$$- \frac{x}{4} \sum_{k=0}^{\infty} 2^{-k} \sum_{j=2^k}^{2^{k+1}-1} \frac{c_{n+1-p(j)}}{2^k+jx}$$

$$= c_{n+1}x,$$

since $p(2^k+j) = p(j) + 1$ for $0 \le j < 2^k$. Thus, the result follows.

Corollary 4.1

Suppose $\lim\limits_{n\to\infty} f_n(1) = f_\infty(1)$ exists. Then $\lim\limits_{n\to\infty} \alpha_n(x) = \alpha_\infty(x)$ exists, and

$$(4.3) \quad \alpha_\infty(x) = \frac{-3f_\infty(1)}{2} \phi(x),$$

where

$$(4.4)\quad \psi(x) = \frac{x}{2}\sum_{k=0}^{\infty} 2^{-k}\sum_{j=0}^{2^k-1}\frac{1}{2^k+jx}$$

is analytic, regular for $x \notin (-\infty, -1]$, and satisfies

$$(4.5)\quad 2\psi(2x) = \psi(x) + \psi\left(\frac{x}{1+x}\right) + 2x.$$

Also, $\psi(x) = \sum_{j=1}^{\infty} (-1)^{j-1}\psi_j x^j$, where $\psi_1 = 1$ and

$$(4.6)\quad \psi_n = \frac{1}{2(2^n-1)}\sum_{k=1}^{n-1}\psi_k\binom{n-1}{k-1}$$

$$(4.7)\quad\quad = \frac{1}{2n}\sum_{k=0}^{n-1}\frac{B_k}{2^{k+1}-1}\binom{n}{k}$$

for $n \ge 2$. [Here B_0, B_1,... are Bernoulli numbers.]

<u>Proof</u>

Let $d_n = \max_{m\ge n}|f_m(1) - f_\infty(1)|$, so $d_0 \ge d_1 \ge \ldots$ and $\lim_{n\to\infty} d_n = 0$. For convenience, let $d_{-1} = d_{-2} = \ldots = 0$.

From (4.2),

$$(4.8)\quad |\alpha_{n+1}(x) - \alpha_\infty(x)| < \frac{3|x|}{4}\sum_{k=0}^{\infty} 2^{-k}\sum_{j=0}^{2^k-1}\frac{d_{n-p(j)}}{|2^k+jx|}.$$

Thus, since $p(j) \le k$ for $j < 2^k$, we have

$$(4.9)\quad |\alpha_{n+1}(x)-\alpha_\infty(x)| \le \frac{3|x|}{4}\sum_{k=0}^{\infty} 2^{-k}\sum_{j=0}^{2^k-1}\frac{d_{n-k}}{|2^k+jx|}$$

For simplicity we assume x is real and positive, though a similar proof goes through for complex $x \notin (-\infty, -1]$. From (4.9) we have

$$(4.10) \quad |\alpha_{n+1}(x) - \alpha_\infty(x)| \le \frac{3x}{4} \sum_{k=0}^{\infty} 2^{-k} d_{n-k}$$

Given $\epsilon > 0$, there exists m such that $d_m \le \epsilon$. Thus, for $n \ge \max(m, m+\lg(d_0/\epsilon))$, we have

$$\sum_{k=0}^{\infty} 2^{-k} d_{n-k} \le \sum_{k=0}^{n-m} 2^{-k} d_{n-k} + \sum_{k=n-m+1}^{\infty} 2^{-k} d_{n-k}$$

$$\le 2\epsilon + 2^{m-n} d_0 \le 3\epsilon$$

Thus, $\lim_{n\to\infty} \alpha_n(x)$ exists, and the limit is given by (4.3) and (4.4).

The recurrence (4.5) may be verified as in the proof of Theorem 4.1, and equating coefficients gives (4.6). Also, substituting

$$(4.11) \quad \frac{1}{2^k + jx} = 2^{-k} \sum_{n=0}^{\infty} (-2^{-k} jx)^n$$

in (4.4) and equating coefficients gives (for $n > 1$)

$$(4.12) \quad \psi_n = \frac{1}{2} \sum_{k=1}^{\infty} 2^{-k(n+1)} \sum_{j=1}^{2^k-1} j^{n-1},$$

so (4.7) follows from ex. 1.2.11.2.4 of Knuth [68].

<u>Corollary 4.2</u>

Suppose $\lim_{n\to\infty} f_n(1) = f_\infty(1)$ exists, and that

(4.13) $f_n(1) = f_\infty(1) + O(\lambda^n)$

as $n \to \infty$, where $\lambda \in (\frac{1}{2}, 1)$. Then

(4.14) $\alpha_n(x) = \alpha_\infty(x) + O(\lambda^n x)$

and

(4.15) $\alpha_n'(x) = \alpha_\infty'(x) + O(\lambda^n)$

as $n \to \infty$, uniformly for all $x \in [0,1]$.

Proof

From (4.10),

$$|\alpha_{n+1}(x) - \alpha_\infty(x)| = O(\lambda^n x) \sum_{k=0}^{\infty} (2\lambda)^{-k},$$

and $2\lambda > 1$, so the last series is convergent. The proof of (4.15) is similar.

5. Some Convergence Results

We define a linear operator T, mapping the Banach space $L_1(0,1)$ into itself, by

(5.1) $$Tf(x) = \sum_{k=1}^{\infty} \left[\left(\frac{1}{2^k+x}\right)^2 f\left(\frac{x}{2^k+x}\right) + \left(\frac{1}{1+2^k x}\right)^2 f\left(\frac{1}{1+2^k x}\right) \right].$$

Thus, (2.12) is

(5.2) $f_{n+1} = Tf_n$.

We write $f \geq 0$ if $f(x) \geq 0$ for almost all $x \in [0,1]$ (in the sense of Lebesgue measure). Note that T is a positive operator, i.e., $Tf \geq 0$ whenever $f \geq 0$.

For $f \in L_1(0,1)$, $\|f\|$ is the norm of f, i.e.,

$$\|f\| = \int_0^1 |f(x)|dx.$$

The norm of a linear operator L is defined by

$$\|L\| = \sup\{\|Lf\| \mid f \in L_1(0,1),\ \|f\| = 1\}.$$

Theorem 5.1

For all $f \in L_1(0,1)$,

(5.3) $\|Tf\| \le \|f\|$.

Also, if $f \ge 0$ then

(5.4) $\|Tf\| = \|f\|$.

Proof

From (5.1),

$$(5.5)\quad \|Tf\| \le \sum_{k=1}^{\infty}\left[\int_0^1\left(\frac{1}{2^k+x}\right)^2\left|f\left(\frac{x}{2^k+x}\right)\right|dx + \int_0^1\left(\frac{1}{1+2^kx}\right)^2\left|f\left(\frac{1}{1+2^kx}\right)\right|dx\right].$$

With the change of variables $y = \frac{x}{2^k+x}$ in the first integral, and $y = \frac{1}{1+2^kx}$ in the second, this gives

$$\|Tf\| \le \sum_{k=1}^{\infty} 2^{-k}\left[\int_0^{\frac{1}{1+2^k}} |f(y)|dy + \int_{\frac{1}{1+2^k}}^1 |f(y)|dy\right]$$

$$= \sum_{k=1}^{\infty} 2^{-k}\int_0^1 |f(y)|dy = \|f\| .$$

This proves (5.3). To prove (5.4), we merely note that all the inequalities in the proof of (5.3) become equalities if $f \geq 0$.

Corollary 5.1

$\|T\| = 1.$

Proof

This is immediate from Theorem 5.1 and the definition of $\|T\|$.

We would like to prove that the iteration (5.2) converges to a fixed-point of T. Unfortunately, the theorems of Schauder (see Simmons [63]) and Krein and Rutman [43] are not applicable, because $\{f \in L_1(0,1) \mid \|f\| = 1\}$ is not compact. Thus, we have only been able to prove the weaker result given in Corollary 5.2.

Theorem 5.2

Suppose that f is continuous on (0,1), changes sign at least once, does not vanish on any finite subinterval of (0,1), and there exists $\epsilon > 0$ such that $f(x) = 0$ has no solution $x \in (0,\epsilon]$. Then

(5.6) $\|Tf\| < \|f\|.$

Proof

Suppose, by way of contradiction, that $\|Tf\| = \|f\|$. Thus, all inequalities in the proof of Theorem 5.1 must be equalities. Hence, for all $k \geq 1$ and all $x \in (0,1)$, we have

$$(5.7) \quad f\left(\frac{x}{2^k+x}\right) f\left(\frac{1}{1+2^k x}\right) \geq 0.$$

By assumption, f(x) changes sign at some point $\varphi \in (0,1)$.

There exists $K \geq 1$ such that $\varphi > \frac{1}{1+2^K}$. Suppose $k \geq K$, so $\varphi > \frac{1}{1+2^k}$. Then there exists $x_k \in (0,1)$ satisfying $\varphi = 1/(1+2^k x_k)$. Thus, from (5.7), f must also change sign at $y_k = x_k/(2^k+x_k) < 2^{-k}$. Since k may be arbitrarily large, this contradicts the hypotheses of the theorem. Thus, (5.6) must hold.

Corollary 5.2

Let $e_n = f_{n+1} - f_n$. Then

$$(5.8) \quad \|e_{n+1}\| < \|e_n\|$$

for all $n \geq 0$.

Proof

From (5.2), $e_{n+1} = Te_n$, so we have only to show that e_n satisfies the conditions of Theorem 5.2. From Theorem 3.1, $e_n(x) = \hat{\alpha}_n(x)\lg(x) + \hat{\beta}_n(x)$, where $\hat{\alpha}_n(x)$ and $\hat{\beta}_n(x)$ are analytic. Also, from Corollary 3.3, $e_n(x)$ does not vanish identically. Thus, $e_n(x)$ is continuous on (0,1) and does not vanish on any finite subinterval of (0,1).

Since

$$(5.9) \quad \int_0^1 e_n(x)dx = \int_0^1 f_{n+1}(x)dx - \int_0^1 f_n(x)dx = 0$$

but $\|e_n\| > 0$, $e_n(x)$ must change sign at least once on (0,1). Finally, from Theorem 3.1 we see that $e_n(x)$ has constant sign on $(0,\epsilon]$, for some sufficiently small $\epsilon > 0$. Thus, the conditions of Theorem 5.2 are satisfied, and the result follows.

From numerical evidence we conjecture that

$$(5.10)\quad \|e_{n+1}\| \le \lambda \|e_n\|$$

for some $\lambda \in (0,1)$. Unfortunately, Corollary 5.2 does not imply (5.10). If (5.10) is true then (f_n) is a Cauchy sequence and the limit f_∞ exists.

Corollary 5.3

For all $n \ge 20$, and all $x \in [0,1]$,

$$(5.11)\quad |F_{n+1}(x) - F_n(x)| \le \|e_n\| < 10^{-10}.$$

Proof

$|F_{n+1}(x) - F_n(x)| = |\int_0^x e_n(y)dy| \le \|e_n\|$, but numerical results (described in Section 7) show that $\|e_{20}\| < 10^{-10}$, so the result follows from Corollary 5.2.

From now on we assume that the limiting distribution $F_\infty(x)$ exists. In view of Corollary 5.3, we may use $F_{20}(x)$ instead of $F_\infty(x)$ for all practical purposes.

6. The Expected Number of Iterations

We use the notation of Section 2. Let $s = u+v$ and $s' = u'+v'$. Note that

$$(6.1)\quad s/s' = (u+v)/(u'+v') = 2^k\left[\frac{1+x}{1+(2^k-1)x}\right].$$

Since $k \ge 1$, $s/s' \ge 2$, so the maximum number of iterations is at most $\lfloor \lg(N) \rfloor$. The example $u = 2^m-1$, $v = 1$ shows that this bound is attainable. For another example see Knuth [69], exs. 4.5.2.27-28.

Let E_n be the expected value of $\ln(s/s')$. From (6.1),

$$E_n = \sum_{k=1}^{\infty} 2^{-k} \int_{x=0}^{x=1} \ln\left[\frac{2^k(1+x)}{1+(2^k-1)x}\right] dF_n(x)$$

$$= \sum_{k=1}^{\infty} 2^{-k}\left[\ln 2 - \int_0^1 \left(\frac{1}{1+x} - \frac{2^k-1}{1+(2^k-1)x}\right) F_n(x)\,dx\right],$$

so

(6.2) $E_n = \ln 2 + \int_0^1 \Phi(x) F_n(x)\,dx,$

where

(6.3) $\Phi(x) = \sum_{k=2}^{\infty} \left[\frac{1-2^{-k}}{1+(2^k-1)x}\right] - \frac{x}{2(1+x)}$.

The expected value of $\ln(s_0/s_n)$ is $\sum_{j=0}^{n-1} E_j$. Thus, assuming the existence of $E_\infty = \lim_{j\to\infty} E_j$, the expected number of iterations for odd integers u_0, $v_0 \le N$ is asymptotically $K \lg(n)$ as $N \to \infty$, where

(6.4) $K = \ln(2)/E_\infty$.

Approximating E_∞ by E_{40} and evaluating the integral in (6.2) numerically gives

(6.5) $K \simeq 0.705971246102$.

In the next section we give some numerical evidence which suggests that the expected number of iterations is $K \lg(n) + O(1)$. This is not surprising if Conjecture 2.1 holds, for then $E_n = E_\infty + O(\lambda^{-n})$.

7. Numerical Results

The recurrence relation (2.7) was solved numerically by three different methods. All computations were performed on a Univac 1108 using double-precision (60-bit fraction), and the numerical results given by the different methods agreed to the accuracy expected.

A. The Recursive Method

This is the most obvious method. $F_n(x)$ is evaluated recursively, using the recurrence (2.7) with the infinite sums truncated after the terms become negligible. The method is only useful for small n, as the computation time increases exponentially with n.

B. The Discretization Method

If $F_n(x)$ is known at a finite set of points, say $x_0 = 0 < x_1 < x_2 < \ldots < x_m = 1$, then we can use the recurrence (2.7) to approximate $F_{n+1}(x)$ at the same set of points, using linear or quadratic interpolation to approximate $F_n(x)$ at points $x \neq x_j$ for $j \le m$. Computations were performed with a uniform grid, i.e., $x_j = jh$, where $h = 1/m$. (It might be more efficient to use a non-uniform grid, because of the logarithmic singularity of $F_n'(x)$ at the origin.) Using several different h, we found that the error in the computed value of $F_n(x)$ was $O(h)$, for fixed n and x. The accuracy could be improved to $O(h^2)$ or better by using Richardson extrapolation. For example, using m = 1920, 3840 and 7680, we obtained $F_n(x)$ to eight decimal places (8D) for $n \le 20$.

C. The Power Series Method

In Section 3 we showed that $F_n(x) = \alpha_n(x)\lg(x) + \beta_n(x)$, where the coefficients $\alpha_{n,j}$ and $\beta_{n,j}$ in the power series $\alpha_n(x) = \sum_{j=0}^{\infty} \alpha_{n,j}x^j$ and $\beta_n(x) = \sum_{j=0}^{\infty} \beta_{n,j}x^j$ satisfy certain recurrence relations. Thus, it is possible to compute the coefficients $\alpha_{n,j}$ and $\beta_{n,j}$ by working with suitably truncated power series. To avoid numerical difficulties it is essential to stay well within the radius of convergence of each series, which ensures that the truncated terms are negligible. This is always possible. With the series truncated after the first 100 terms, we computed $F_n(x)$ to 12D, and the results agreed with those computed by the discretization method. The value K = 0.705971246102 should be correctly rounded to 12D.

Table 7.1 gives $F_n(x)$ to 4D for x = 0.1(0.1)0.9 and n = 1(1)5. It is clear that the distributions $F_n(x)$ converge rapidly. Table 7.2 gives the limit $F_\infty(x)$ to 10D for various x. The computed values of $F_n(x)$ differ by less than 10^{-12} for all n ≥ 20.

Table 7.3 gives the coefficients $\alpha_{\infty,j}$, $\beta_{\infty,j}$ and $\xi_{\infty,j}$ in the power series $\alpha_\infty(x)$, $\beta_\infty(x)$, and $\xi_\infty(x) = F_\infty(1+x)$, for $j \le 20$. Note that the coefficients alternate in sign, and their absolute values decrease monotonically, for $j \ge 2$.

The values given in Tables 7.2 and 7.3 confirm several identities which may be derived theoretically, for example:

$$7F_\infty(\tfrac{1}{2}) + F_\infty(\tfrac{1}{3}) = 2F_\infty(\tfrac{1}{5}) + 2F_\infty(\tfrac{1}{4}) + F_\infty(\tfrac{2}{3}) + 3,$$

$$3\xi_{\infty,1} = -6\xi_{\infty,2} = -2\alpha_{\infty,1},$$

and

$$18\beta_{\infty,2} + 3\beta_{\infty,1} + (10 + 3/\ln(2))\alpha_{\infty,1} = 0.$$

Table 7.1: Values of $F_n(x)$ to 4D

x	$F_1(x)$	$F_2(x)$	$F_3(x)$	$F_4(x)$	$F_5(x)$
0.1	0.3329	0.2871	0.2772	0.2753	0.2750
0.2	0.4967	0.4478	0.4370	0.4349	0.4346
0.3	0.6111	0.5666	0.5567	0.5548	0.5544
0.4	0.6989	0.6611	0.6526	0.6510	0.6507
0.5	0.7699	0.7394	0.7325	0.7312	0.7310
0.6	0.8294	0.8060	0.8007	0.7997	0.7995
0.7	0.8805	0.8637	0.8599	0.8592	0.8590
0.8	0.9251	0.9144	0.9120	0.9115	0.9114
0.9	0.9646	0.9595	0.9584	0.9581	0.9581

Table 7.2: Values of $F_\infty(x)$ to 10D

x	$F_\infty(x)$	x	$F_\infty(x)$
0.1	0.2750116116	1/3	0.5886652481
0.2	0.4345648990	2/3	0.8400418266
0.3	0.5544181563	1/4	0.4981238639
0.4	0.6507109442	3/4	0.8860223000
0.5	0.7309648721	1/6	0.3370894190
0.6	0.7994844345	5/6	0.9275771715
0.7	0.8590163978	1/12	0.2420627866
0.8	0.9114387997	5/12	0.6650572783
0.9	0.9580992159	7/12	0.7887496125
1.0	1.0000000000	11/12	0.9653900331

Table 7.3: The Coefficients $\alpha_{\infty,j}$, $\beta_{\infty,j}$ and $\xi_{\infty,j}$

j	$\alpha_{\infty,j}$	$\beta_{\infty,j}$	$\xi_{\infty,j}$
0	0.000000	0.000000	1.000000
1	-0.596884	0.765619	0.397923
2	0.099481	0.347519	-0.198961
3	-0.056846	-0.191979	0.111631
4	0.035529	0.138115	-0.067966
5	-0.023839	-0.105276	0.044193
6	0.016962	0.082567	-0.030365
7	-0.012663	-0.066260	0.021861
8	0.009823	0.054283	-0.016369
9	-0.007853	-0.045299	0.012666
10	0.006428	0.038417	-0.010072
11	-0.005361	-0.033033	0.008194
12	0.004540	0.028739	-0.006795
13	-0.003893	-0.025255	0.005725
14	0.003375	0.022384	-0.004890
15	-0.002953	-0.019989	0.004225
16	0.002605	0.017966	-0.003688
17	-0.002315	-0.016242	0.003247
18	0.002071	0.014760	-0.002881
19	-0.001864	-0.013476	0.002574
20	0.001686	0.012357	-0.002313

For integers u and v, let b(u,v) be the number of iterations required by the binary Euclidean algorithm as described in Section 1. Let

$$B(N) = \sum_{\substack{0<v<u<N \\ u,v \text{ odd}}} b(u,v)$$

and

$$\mathcal{B}(N) = 2B(N)/(\lfloor N/2 \rfloor(\lfloor N/2 \rfloor - 1)).$$

Thus, $\mathcal{B}(N)$ is the average number of iterations required for distinct, odd u and v less than N. Table 7.4 gives B(N), $\mathcal{B}(N)$ and $\Delta(N) = \mathcal{B}(N) - \mathcal{B}(N/2)$ for $N = 2^3, 2^4, \ldots, 2^{15}$. From the results of Sections 6 and 7, we expect $\Delta(N)$ to converge to $K = 0.705971246\ldots$ as $N \to \infty$. In fact, the values given in Table 7.4 satisfy $0 < K - \Delta(N) < 2\,\lg(N)/N$, and give the approximation

$$\mathcal{B}(N) \simeq K\lg(N) - 0.93.$$

Table 7.4: Exact Counts for Small N (algorithm RS)

N	B(N)	$\mathcal{B}(N)$	$\Delta(N)$
2^3	10	1.6667	0.6667
2^4	60	2.1429	0.4762
2^5	341	2.8417	0.6988
2^6	1701	3.4294	0.5878
2^7	8254	4.0942	0.6648
2^8	38692	4.7603	0.6661
2^9	178046	5.4548	0.6945
2^{10}	804192	6.1475	0.6927
2^{11}	3586234	6.8469	0.6994
2^{12}	15822368	7.5484	0.7015
2^{13}	69216057	8.2532	0.7048
2^{14}	300540247	8.9579	0.7047
2^{15}	1296893644	9.6632	0.7053

8. Other "Binary" Euclidean Algorithms

As well as the algorithm described above, there are several other "binary" variants of the Euclidean algorithm.

For example, Harris [70] suggested an algorithm which uses both division and right shifting, and requires less iterations than the classical algorithm, on the average. Yao and Knuth [75] considered the "subtractive" Euclidean algorithm, which requires neither shifts nor divisions. In this section we analyze the "left-shift" algorithm (LS) mentioned at the end of Section 1. For positive integers u and v, even or odd, the algorithm is as follows.

<u>LS Binary Algorithm</u>

L0: if $u < v$ then interchange u and v;
if $u = v$ or $v = 0$ then return u as the GCD and halt;
$t \leftarrow v$;
while $2t \le u$ do $t \leftarrow 2t$;

L1: $u \leftarrow u - t$;
go to L0.

The interchanging of u and v can be avoided by duplicating some of the code. The "while" loop merely shifts t left until its leading one bit is in the same position as that of u, or one position to the right of it. This may be done with a floating-point "normalize" instruction, possibly followed by one right shift.

We say that an iteration is one execution of step L1. The expected number of iterations is given by the following theorem.

<u>Theorem 8.1</u>

If integers u, v are chosen uniformly and independently in $(0,N]$, the expected number of iterations of algorithm LS is asymptotically $K_2 \lg(N)$ as $N \to \infty$, where

$$(8.1)\quad K_2 = 12(\ln(2)/\pi)^2\, c \simeq 0.875837091,$$

$$(8.2)\quad c = \sum_{j=1}^{\infty} p(j)\lg\left[\frac{(j+1)^2}{j(j+2)}\right],$$

and p(j) is defined in Section 4.

Proof

We shall only sketch the proof. Suppose $u > v > 0$ and we perform one iteration of the classical Euclidean algorithm, i.e., we find $q = \lfloor u/v \rfloor$, $r = u-qv$, set $u \leftarrow v$ and $v \leftarrow r$. Then the new values of u and v would be obtained after exactly p(q) iterations of algorithm LS. [Let

$$q = \sum_{j=1}^{p(q)} 2^{m_j},$$

where $m_1 > m_2 > \dots > m_{p(q)} \geq 0$. If $1 \leq j \leq p(q)$, then the j-th execution of step L1 of algorithm LS replaces the current u by u-t, where $t = 2^{m_j}v$.]

Let the regular continued fraction for u/v be

$$(8.3)\quad u/v = q_0 + 1/q_1 + 1/ \dots + 1/q_k,$$

so the classical algorithm requires k+1 iterations. From the above discussion, algorithm LS requires $\sum_{j=0}^{k} p(q_j)$ iterations (actually one less if $q_k = 1$, because of our test "if u = v ...").

Let $E_2(N)$ be the expected number of iterations for algorithm LS, and $E_c(N)$ be the expected number for the classical algorithm. Thus, (8.4) $\lim_{N\to\infty} E_2(N)/E_c(N) = \lim_{n\to\infty} \lim_{N\to\infty} \bar{p}(q_n)$,

where $\bar{p}(q_n)$ is the expected value of $p(q_n)$. From results like those of Khinchin [35a, 35b, 36],

(8.5) $\lim_{n\to\infty} \lim_{N\to\infty} \bar{p}(q_n) = c$,

where c is given by (8.2). [Intuitively, the probability that $q_n = q$ is about $\lg\left[\frac{(j+1)^2}{j(j+2)}\right]$, from (2.9).] Also,

(8.6) $E_c(N) \sim 12(\ln(2)/\pi)^2 \lg(N)$

as $N \to \infty$ (see Knuth [69]). Thus, the result follows from (8.4).

The constant c is difficult to evaluate numerically from (8.2). The following lemma is much better for numerical purposes. Using (8.8), we found

(8.7) $c \simeq 1.49930818096$

very easily.

<u>Lemma 8.1</u>

If c is defined by (8.2), then

(8.8) $c = 2 + \sum_{j=1}^{\infty} \lg\Gamma(1+2^{-j})$

(8.9) $= 2 - \frac{1}{\ln(2)}\left[\gamma - \sum_{j=2}^{\infty} \frac{(-1)^j\zeta(j)}{j(2^j-1)}\right]$

$$(8.10) \qquad = 1 + \frac{1}{2\ln(2)}\left[\ln(\pi) - \gamma + 2\sum_{j=2}^{\infty} \frac{(-1)^j \zeta(j)}{j2^j(2^j-1)}\right].$$

Here, $\gamma = 0.5772\ldots$ is Euler's constant, $\Gamma(x)$ is the Gamma function, and $\zeta(j)$ is the Riemann Zeta function.

Sketch of Proof

Splitting the sum in (8.2) into odd and even indices, and using $p(2j+1) = p(j) + 1$ and $p(2j) = p(j)$, gives

$$(8.11) \qquad c = \sum_{j=0}^{\infty} \lg\left[\frac{1+1/(2j+1)}{1+1/(2j+2)}\right] + \sum_{j=1}^{\infty} p(j)\lg\left[\frac{1+1/(2j)}{1+1/(2j+2)}\right].$$

Continuing the splitting process eventually gives

$$(8.12) \qquad c = \sum_{k=1}^{\infty}\sum_{j=0}^{\infty} \lg\left[\frac{1+1/(2^k(j+\frac{1}{2}))}{1+1/(2^k(j+1))}\right].$$

From Stirling's approximation,

$$(8.13) \qquad \prod_{j=0}^{n} [1+x/(j+y)] \sim n^x\Gamma(y)/\Gamma(x+y)$$

as $n \to \infty$, so (8.12) gives

$$(8.14) \qquad c = \sum_{k=1}^{\infty} \lg\left[\frac{\Gamma(\frac{1}{2})\Gamma(1+2^{-k})}{\Gamma(\frac{1}{2}+2^{-k})}\right].$$

From the well-known identity

$$(8.15) \qquad \Gamma(x)\Gamma(x+\tfrac{1}{2}) = \Gamma(2x)\Gamma(\tfrac{1}{2})2^{1-2x}$$

with $x = \frac{1}{2} + 2^{-k}$, it is easy to show that

$$(8.16) \quad \sum_{k=1}^{\infty} \lg[\Gamma(\tfrac{1}{2})/\Gamma(\tfrac{1}{2}+2^{-k})] = 2,$$

so (8.8) follows from (8.14).

Suppose $|x| < 1$, $n \geq 1$. We have

$$(8.17) \quad \ln\Gamma(1+x) = \begin{pmatrix}\ln\Gamma(n+x) \\ -\ln\Gamma(n)\end{pmatrix} - \sum_{k=1}^{n-1} \ln(1+x/k)$$

$$(8.18) \quad = \lim_{n\to\infty}\left[x\ln(n) - \sum_{k=1}^{n-1} \ln(1+x/k)\right]$$

$$(8.19) \quad = -\gamma x + \sum_{j=2}^{\infty} (-x)^j \frac{\zeta(j)}{j} .$$

(8.9) follows from (8.8) by putting $x = 2^{-k}$ in (8.19) and summing over $k = 1, 2, \ldots$. The proof of (8.10) is similar.

Numerical Results for Algorithm LS

For integers u and v, let $b_2(u,v)$ be the number of iterations required by algorithm LS,

$$(8.20) \quad B_2(N) = \sum_{0<v<u\leq N} b_2(u,v),$$

$$(8.21) \quad \mathcal{B}_2(N) = 2B_2(N)/[N(N-1)],$$

and

(8.22) $\Delta_2(N) = \mathcal{B}_2(N) - \mathcal{B}_2(N/2)$.

Table 8.1 gives $B_2(N)$, $\mathcal{B}_2(N)$ and $\Delta_2(N)$ for $N = 2^2, 2^3, \ldots, 2^{12}$ (compare Table 7.4 for algorithm RS).

Table 8.1: Exact Counts for Small N (algorithm LS)

N	$B_2(N)$	$\mathcal{B}_2(N)$	$\Delta_2(N)$
2^2	8	1.3333	0.3333
2^3	55	1.9643	0.6310
2^4	305	2.5417	0.5774
2^5	1625	3.2762	0.7345
2^6	8135	4.0352	0.7590
2^7	39282	4.8329	0.7977
2^8	184670	5.6578	0.8249
2^9	851566	6.5096	0.8519
2^{10}	3860856	7.3712	0.8615
2^{11}	17268497	8.2383	0.8671
2^{12}	76392955	9.1090	0.8707

From Theorem 8.1, we expect

(8.23) $$\lim_{N\to\infty} \Delta_2(N) = K_2 \simeq 0.875837,$$

and the numerical results support this prediction.

Summary

Table 8.2 summarizes the average and worst-case behavior of four algorithms: the classical algorithm, the RS and LS binary algorithms, and the subtractive algorithm of Yao and Knuth [75]. The subtractive algorithm is of theoretical interest only. The choice of which of the other three algorithms is to be preferred depends on the instruction set and instruction timing of the machine used.

Table 8.2: Comparison of Various Euclidean GCD Algorithms

Algorithm	Average iterations*	Maximum iterations*
Classical	0.5842lg(N)	1.4404lg(N)
RS Binary	0.7060lg(N)	lg(N)
LS Binary	0.8758lg(N)	1.4404lg(N)
Subtractive	$0.2921(\lg(N))^2$	N

*Notes: 1. Lower order terms are neglected (in most cases they are O(1)).

2. An iteration of one algorithm (e.g., the binary algorithm) may take less time than an iteration of another algorithm (e.g., the classical algorithm).

Acknowledgment

I would like to thank Frank de Hoog for his assistance with equation (4.4), and Don Knuth both for his encouragement and for an independent proof of equation (4.7).

References

Collins [74] Collins, G. E., "The Computing time of the Euclidean Algorithm," SIAM J. Computing 3 (1974), 1-10.

Dixon [70] Dixon, J. D., "The Number of Steps in the Euclidean Algorithm," J. Number Theory 2 (1970), 414-422.

Gauss [12] Gauss, C. F., "Brief an Laplace vom 30 Jan. 1812," Carl Friedrich Gauss Werke, Bd. X_1, Gottingen, 371-374.

Harris [70] Harris, V. C., "An Algorithm for Finding the Greatest Common Divisor," Fibonacci Quar. 8 (1970), 102-3.

Heilbronn [68] Heilbronn, H. A., "On the Average Length of a Class of Finite Continued Fractions," in Abhandlungen aus Zahlentheorie und Analysis, VEB Deutscher Verlag, Berlin, 1968, 87-96.

Khinchin [35a] Khinchin, A., "Continued Fractions," Moscow, 1935 (English translation by P. Wynn, P. Noordhoff, Groningen, 1963).

Khinchin [35b] Khinchin, A., "Metrische Kettenbruchprobleme," Compos. Math. 1 (1935), 361-382.

Khinchin [36] Khinchin, A., "Zur Metrischen Kettenbruchtheorie," Compos. Math. 3 (1936), 276-285.

Knuth [68] Knuth, D. E., "The Art of Computer Programming," Vol. 1, Addison-Wesley, Menlo Park, 1968, Section 1.2.11.

Knuth [69] Knuth, D. E., "The Art of Computer Programming," Vol. 2, Addison-Wesley, Menlo Park, 1969, Sections 4.5.2 and 4.5.3.

Krein and Rutman [43] Krein, M. G. and Rutman, M. A., "Linear Operators Leaving Invariant A Cone in a Banach Space," Uspekhi Mat. Nauk (N.S) 3, 1 (23) (1943), 3-95 (in Russian).

Kusmin [28] Kusmin, R. O., "Sur un Problème de Gauss," Atti del Congresso Internationale dei Matematici 6 (Bologna, 1928), 83-89.

Lévy [29] Lévy, P., "Sur les Lois de Probabilité dont Dependent les Quotients Complets et Incomplets d'une Fraction Continue," Bull. Soc. Math. France 57 (1929), 178-194.

Schönhage [71] Schönhage, A., "Schnelle Berechnung von Kettenbruchentwicklungen," Acta Informatica 1 (1971), 139-144.

Silver and Terzian [62] Silver, R. and Terzian, J., unpublished, 1962 (see Knuth [69], page 297).

Simmons [63] Simmons, G. F., "Introduction to Topology and Modern Analysis," McGraw-Hill, New York, 1963, Appendix 1.

Stein [67] Stein, J., "Computational Problems Associated with Racah Algebra," J. Comput. Phys. 1 (1967), 397-405.

Szüsz [61] Szüsz, P., "Über einen Kusminschen Satz," Acta Math. Acad. Sci. Hungar. 12 (1961), 447-453.

Tonkov [74] Tonkov, T., "On the Average Length of Finite Continued Fractions," Acta Arith. 26 (1974), 47-57.

Wirsing [74] Wirsing, E., "On the Theorem of Gauss-Kusmin-Lévy and a Frobenius-Type Theorem for Function Spaces," Acta Arith. 24 (1974), 507-528.

Yao and Knuth [75] Yao, A. C. and Knuth, D. E., Analysis of the Subtractive Algorithm for Greatest Common Divisors, Report STAN-CS-75-510, Computer Science Dept., Stanford University, Sept. 1975.