CONJECTURES ASSOCIATED WITH COMPUTING THE CONTINUED FRACTION OF $\log_b a$

KEITH MATTHEWS

1. Shanks' algorithm

Let $a_0 = a > a_1 = b > 1$ be positive integers. In his article [1], Shanks gave an algorithm for computing the partial quotients of $\log_b a$. Construct two sequences $a_0, a_1, a_2, \ldots$ and $n_0, n_1, n_2, \ldots$, where the a_i are positive rationals and the n_i are positive integers, by the following rule: If $a_{i-1} > a_i > 1$, then

$$a_i^{n_{i-1}} \leq a_{i-1} < a_i^{n_{i-1}+1} \tag{1.1}$$

$$a_{i+1} = a_{i-1}/a_i^{n_{i-1}}. \tag{1.2}$$

Clearly (1.1) and (1.2) imply $a_i > a_{i+1} \geq 1$ and

$$a_{i+1} \leq a_1^{1/n_1 \cdots n_i}. \tag{1.3}$$

Then there are two possibilities:

(i) $a_{r+1} = 1$ for some $r \geq 1$. This implies a relation $a_0^q = a_1^p$ for positive integers p and q and so $\log_{a_1} a_0 = p/q$.

(ii) $a_{i+1} > 1$ for all i. In this case the decreasing sequence $\{a_i\}$ tends to $a \geq 1$. Also (1.3) implies $a = 1$ unless perhaps $n_i = 1$ for all sufficiently large i; but then (1.2) becomes $a_{i+1} = a_{i-1}/a_i$ and hence $a = a/a = 1$.

If $a_{i-1} > a_i > 1$, then from (1.1) we have

$$n_{i-1} = \left\lfloor \frac{\log a_{i-1}}{\log a_i} \right\rfloor. \tag{1.4}$$

Let $x_i = \log_{a_{i+1}} a_i$ if $a_{i+1} > 1$. Then we have

Lemma 1.1 If $a_{i+2} > 1$, then

$$x_i = n_i + \frac{1}{x_{i+1}}. \tag{1.5}$$

Date: 16th January 2019.

Proof. From (1.2), we have

$$\begin{aligned}\log a_{i+2} &= \log a_i - n_i \log a_{i+1}\\ 1 &= \frac{\log a_i}{\log a_{i+1}} \cdot \frac{\log a_{i+1}}{\log a_{i+2}} - n_i \cdot \frac{\log a_{i+1}}{\log a_{i+2}}\\ &= x_i x_{i+1} - n_i x_{i+1},\end{aligned}$$

from which (1.5) follows.

From Lemma 1.1, we deduce
Lemma 1.2

(a) If $\log_{a_1} a_0$ is irrational, then

$$x_i = n_i + \frac{1}{x_{i+1}} \text{ for all } i \geq 0.$$

(b) If $\log_{a_1} a_0$ is rational, with $a_{r+1} = 1$, then

$$x_i = \begin{cases} n_i + \frac{1}{x_{i+1}} & \text{if } 0 \leq i < r-1,\\ n_{r-1} & \text{if } i = r-1.\end{cases}$$

In view of the equation $\log_{a_1} a_0 = x_0$, Lemma 1.2 leads immediately to
Corollary 1.1
(1.6)

$$\log_{a_1} a_0 = \begin{cases} [n_0, n_1, \ldots] & \text{if } \log_{a_1} a_0 \text{ is irrational},\\ [n_0, n_1, \ldots, n_{r-1}] & \text{if } \log_{a_1} a_0 \text{ is rational and } a_{r+1} = 1.\end{cases}$$

Remark It is an easy exercise to show that

$$a_{2j} = \frac{a_0^{q_{2j-2}}}{a_1^{p_{2j-2}}}, \quad a_{2j+1} = \frac{a_1^{p_{2j-1}}}{a_0^{q_{2j-1}}} \tag{1.7}$$

where p_k/q_k is the k–th convergent to $\log_{a_1} a_0$.

Example 1.1 $\log_2 10$: Here $a_0 = 10,\ a_1 = 2$. Then $2^3 < 10 < 2^4$, so $n_0 = 3$ and $a_2 = 10/2^3 = 1.25$.

Further, $1.25^3 < 2 < 1.25^4$, so $n_1 = 3$ and $a_3 = 2/1.25^3 = 1.024$.

Also, $1.024^9 < 1.25 < 1.024^{10}$, so $n_2 = 9$ and

$$\begin{aligned} a_4 &= 1.25/1.024^9\\ &= 1250000000000000000000000000/1237940039285380274899124224\\ &= 1.0097419586\cdots\end{aligned}$$

Continuing we obtain

i	n_i	a_i	p_i/q_i
0	3	10	3/1
1	3	2	10/3
2	9	1.25	93/28
3	2	1.024	196/59
4	2	$1.0097419586\cdots$	485/146
5	4	$1.0043362776\cdots$	2136/643
6	6	$1.0010415475\cdots$	13301/4004
7	2	$1.0001628941\cdots$	28738/8651
8	1	$1.0000637223\cdots$	42039/12655
9	1	$1.0000354408\cdots$	70777/21306
10		$1.0000282805\cdots$	
11		$1.0000071601\cdots$	

$\log_2 10 = [3, 3, 9, 2, 2, 4, 6, 2, 1, 1, \ldots]$.

2. Some Pseudocode

In Table 1 we present pseudocode for the Shanks algorithm.

It soon becomes impractical to perform the calculations in multi-precision arithmetic, as the numerators and denominators of a_i grow rapidly.

In Algorithm 1, we are replacing a_i by $A[i]/c$, where $c > 1$ is an integer. If the condition `bb>e` were replaced by `bb>c`, the `A[i]` would decrease strictly until they reached `c`. Also `m[0]=n[0]` and we can expect a number of the initial `m[i]` to be partial quotients. The larger we take c, the more partial quotients will be produced.

3. Formal description of algorithm 1

We define two integer sequences

$$\{A_{i,c}\},\ i = 0, \ldots, l(c) \text{ and } \{m_{j,c}\},\ j = 0, \ldots, l(c) - 2,$$

as follows:

Let $A_{0,c} = c \cdot a_0, A_{1,c} = c \cdot a_1$. Then if $i \geq 1$ and $A_{i-1,c} > A_{i,c} > c$, we define $m_{i-1,c}$ and $A_{i+1,c}$ by means of an intermediate sequence $\{B_{i,r,c}\}$, defined for $r \geq 0$, by $B_{i,0,c} = A_{i-1,c}$ and

$$B_{i,r+1,c} = \left\lfloor \frac{cB_{i,r,c}}{A_{i,c}} \right\rfloor, r \geq 0. \tag{3.1}$$

Then $c \leq B_{i,r+1,c} < B_{i,r,c}$, if $B_{i,r,c} \geq A_{i,c} > c$ and hence there is a unique integer $m = m_{i-1,c} \geq 1$ such that

$$B_{i,m,c} < A_{i,c} \leq B_{i,m-1,c}.$$

Shanks' algorithm	Algorithm 1

<pre>input: integers a>b>1
output: n[0],n[1],...
s:= 0
a[0]:= a; a[1]:= b
aa:= a[0]; bb:= a[1]
while(bb > 1){
  i:=0
  while(aa ≥ bb){
    aa:= aa/bb
    i:= i+1
  }
  a[s+2]:= aa
  n[s]:= i
  t:= bb
  bb:= aa
  aa:= t
  s:= s+1
}</pre>
	<pre>input: integers a>b>1,t≥1
	output: m[0],m[1],...
	s:= 0; c:= b^t; e:= c+b*b*sqrt(c)
	A[0]:= a*c; A[1]:= b*c
	aa:= A[0]; bb:= A[1]
	while(bb > e){
	  i:=0
	  while(aa ≥ bb){
	    aa:= int(aa*c,bb)
	    i:= i+1
	  }
	  A[s+2]:= aa
	  m[s]:= i
	  t:= bb
	  bb:= aa
	  aa:= t
	  s:= s+1
	}</pre>

TABLE 1.

Then we define $A_{i+1,c} = B_{i,m,c}$. Hence $A_{i+1,c} \geq c$ and the sequence $\{A_{i,c}\}$ decreases strictly.

We believe that with $c = b^t, t \geq 1$, provided $A_{i+1,c} > c + b^2\sqrt{c}$, we have $m_{i-1,c} = n_i$. (See `http://www.numbertheory.org/php/log3.html` for a BCMATH program.)

At one stage the weaker condition $A_{i+1,c} > c + b\sqrt{c}$ also seemed to have the same property; however taking $(a, b) = (991, 2)$ and $t = 146, 147, 148$ gave a counter–example. The weaker condition has the charm of producing longer list of partial quotients. (See `http://www.numbertheory.org/php/log4.html` for a BCMATH program.)

We can extend the algorithm to $\log_b(a/d)$, where $a/d > 1$. In the pseudo code, we replace $\mathtt{c} = \mathtt{b}^\mathtt{t}$ by $\mathtt{c} = \mathtt{b}^\mathtt{t} * \mathtt{d}$ and $\mathtt{A}[0] = \mathtt{a} * \mathtt{b}^\mathtt{t}$.

Example 2.2 $a_0 = 3, a_1 = 2$. Here are the sequences $\{m_{i,c}\}$ for $\log_2 3$, with $c = 2^u, u = 1, \ldots, 50$, without using the cutoff $A[i+1] > c+b^2\sqrt{c}$, where the $A[i]$ are allowed to decrease to c:

```
1,1,
1,1,1,
1,1,1,1,
```

```
1,1,1,2,
1,1,1,2,
1,1,1,2,3,
1,1,1,2,2,2,
1,1,1,2,2,2,1,
1,1,1,2,2,2,1,2,
1,1,1,2,2,3,2,3,
1,1,1,2,2,3,2,
1,1,1,2,2,3,1,2, 1, 1,1,2,
1,1,1,2,2,3,1,3, 1, 1,3,1,
1,1,1,2,2,3,1,4, 3, 1,
1,1,1,2,2,3,1,4, 1, 9,1,
1,1,1,2,2,3,1,5,24, 1,2,
1,1,1,2,2,3,1,5, 3, 1,1, 2,7,
1,1,1,2,2,3,1,5, 2, 1,1, 5,3,1,
1,1,1,2,2,3,1,5, 2, 2,1, 3,1,16,
1,1,1,2,2,3,1,5, 2,15,1, 6,2
1,1,1,2,2,3,1,5, 2, 9,5, 1,2,
1,1,1,2,2,3,1,5, 2,13,1, 1,1, 6,1,2,2,
1,1,1,2,2,3,1,5, 2,17,2, 7,8,
1,1,1,2,2,3,1,5, 2,19,1,49,2, 1,
1,1,1,2,2,3,1,5, 2,22,4, 8,3, 4,  1,
1,1,1,2,2,3,1,5, 2,22,2, 1,3, 1,  3, 8,
1,1,1,2,2,3,1,5, 2,22,1, 6,3, 1,  1, 3, 4,  2,
1,1,1,2,2,3,1,5, 2,23,2, 1,1, 2,  1,12,17,
1,1,1,2,2,3,1,5, 2,23,3, 2,2, 2,  2, 1, 3,  2,
1,1,1,2,2,3,1,5, 2,23,2, 1,7, 2,  2,14, 1,  1, 6,
1,1,1,2,2,3,1,5, 2,23,2, 1,1, 1,  1, 1, 8,  2, 1,14,  2,
1,1,1,2,2,3,1,5, 2,23,2, 2,2, 1,  1, 2, 1,  1, 1, 1,  2, 1,  3, 1,
1,1,1,2,2,3,1,5, 2,23,2, 2,4, 2,  2, 1, 1,  4, 1,10,  1, 4,
1,1,1,2,2,3,1,5, 2,23,2, 2,2,11,  2, 9, 3,  3, 1, 2,  1, 2,
1,1,1,2,2,3,1,5, 2,23,2, 2,2,30,  3, 4, 1,  1, 8, 1,  4,
1,1,1,2,2,3,1,5, 2,23,2, 2,2, 7,  3, 4, 1, 23, 1, 5,  3,
1,1,1,2,2,3,1,5, 2,23,2, 2,2,17,  1, 2, 1,  1, 3, 1,430,
1,1,1,2,2,3,1,5, 2,23,2, 2,2,54, 22,10, 1,  1,12, 1,
1,1,1,2,2,3,1,5, 2,23,2, 2,2,55,  1,49, 1,  4, 7, 1,  1, 4,
1,1,1,2,2,3,1,5, 2,23,2, 2,1, 1,120, 1, 6,  1, 3, 2,  8, 2,  3, 3,
1,1,1,2,2,3,1,5, 2,23,2, 2,1, 1, 60, 5, 2, 22, 1, 3,  1, 1,  1, 3,2,
1,1,1,2,2,3,1,5, 2,23,2, 2,1, 1, 65, 1,22,  3, 2, 1, 10, 1,  3, 3,2,
1,1,1,2,2,3,1,5, 2,23,2, 2,1, 1, 59, 1, 6, 14, 3, 3,  1, 8,  5, 1,1,1,
1,1,1,2,2,3,1,5, 2,23,2, 2,1, 1, 56,10, 1,  7, 1, 2,  1, 2,  8, 3,1,1, 3,
1,1,1,2,2,3,1,5, 2,23,2, 2,1, 1, 56, 1, 3,741, 1,12,  1,12,  2,
1,1,1,2,2,3,1,5, 2,23,2, 2,1, 1, 56, 3, 1,  1, 1, 1,  8, 1,109, 2,1,2, 7,
1,1,1,2,2,3,1,5, 2,23,2, 2,1, 1, 56,16, 1,  1, 1, 2,  5, 1,  2, 1,5,6, 1, 2,1, 2,
1,1,1,2,2,3,1,5, 2,23,2, 2,1, 1, 55, 1,11,  1, 1, 1,  2, 1, 20,14,3,5,13,
1,1,1,2,2,3,1,5, 2,23,2, 2,1, 1, 55, 1, 6,  1, 1, 8,  1, 2,  4, 1,1,1, 1,16,1,14,1,1,
1,1,1,2,2,3,1,5, 2,23,2, 2,1, 1, 55, 1, 5,  3, 2, 7,  1, 2, 16, 2,1,1, 1, 1,2,
```

Example 2.2 $a_0 = 3, a_1 = 2$. Here are the sequences $\{m_{i,c}\}$ for $\log_2 3$, with $c = 2^u, u = 1, \ldots, 50$, using the cutoff $A[i+1] > c + b^2\sqrt{c}$. Note the monotonic increasing lengths of correct partial quotients. (We are actually using a variation of Algorithm 1, to ensure that in the few cases where $bc < c + b^2\sqrt{c}$ that $m[0] = n[0]$ is returned. (See `http://www.numbertheory.org/gnubc/logx` for a BC version with this modification.

```
1,
1,
1, 1,
1, 1,
1, 1, 1,
1, 1, 1,
1, 1, 1,
1, 1, 1, 2,
1, 1, 1, 2,
1, 1, 1, 2,
1, 1, 1, 2, 2,
1, 1, 1, 2, 2,
1, 1, 1, 2, 2,
1, 1, 1, 2, 2,
1, 1, 1, 2, 2, 3, 1,
1, 1, 1, 2, 2, 3, 1,
1, 1, 1, 2, 2, 3, 1,
1, 1, 1, 2, 2, 3, 1,
1, 1, 1, 2, 2, 3, 1,
1, 1, 1, 2, 2, 3, 1, 5,
1, 1, 1, 2, 2, 3, 1, 5,
1, 1, 1, 2, 2, 3, 1, 5, 2,
1, 1, 1, 2, 2, 3, 1, 5, 2,
1, 1, 1, 2, 2, 3, 1, 5, 2,
1, 1, 1, 2, 2, 3, 1, 5, 2,
1, 1, 1, 2, 2, 3, 1, 5, 2,
1, 1, 1, 2, 2, 3, 1, 5, 2,
1, 1, 1, 2, 2, 3, 1, 5, 2,
1, 1, 1, 2, 2, 3, 1, 5, 2,
1, 1, 1, 2, 2, 3, 1, 5, 2,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2, 2,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2, 1, 1,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2, 1, 1,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2, 1, 1,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2, 1, 1,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2, 1, 1,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2, 1, 1,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2, 1, 1,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2, 1, 1,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2, 1, 1,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2, 1, 1,
1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2, 1, 1, 55,
```

In fact

$$\log_2 3 = [1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2, 1, 1, 55, 1, 4, 3, 1, 1, 15, \ldots].$$

4. Another integer part algorithm

Algorithm 2. This algorithm is much slower than Algorithm 1 numerically, when it meets a large partial quotient. Also it seems to give the same output. However it is easier to describe and it may be capable of theoretical analysis. Let $a_0 > a_1 > 1$ be positive integers.

Let $A_0 = a \cdot c, A_1 = b \cdot c$, where $c = b^t$. If $A_{i-1} > A_i > c$, let $m_{i-1} \in \mathbb{N}$ be defined by

$$c \leq \frac{A_{i-1}c^{m_{i-1}}}{A_i^{m_{i-1}}} < A_i. \tag{4.1}$$

Then define A_{i+1} by

$$A_{i+1} = \left\lfloor \frac{A_{i-1}c^{m_{i-1}}}{A_i^{m_{i-1}}} \right\rfloor. \tag{4.2}$$

Then $c \leq A_{i+1} < A_i$ and eventually $A_i = c$.

Again it seems likely that if $A_{i+1} > c + b^2\sqrt{c}$, then $m_{i-1} = n_{i-1}$. We have found that both Algorithms 1 and 2 have the same output.

If we write $b_i = A_i/c$, then (4.1) and (4.2) give

$$b_i^{m_{i-1}} \leq b_{i-1} < b_i^{m_{i-1}+1} \tag{4.3}$$

$$b_{i+1} = \frac{1}{c}\left\lfloor \frac{b_{i-1}c}{b_i^{m_{i-1}}} \right\rfloor. \tag{4.4}$$

The resemblance to Shanks' algorithm is now more striking.

In fact, if we write $b_{i,c}$ and $m_{i,c}$, instead of b_i and m_i, then $b_{i,c} \to a_i$ and $m_{i,c} \to n_i$ as $c \to \infty$, where a_i and n_i are defined in (1.1) and (1.2).

We prove $b_{i,c} \to a_i$ as $c \to \infty$ by induction on $i \geq 0$. There is nothing to prove when $i = 0$ and $i = 1$, as $b_{0,c} = a_0$ and $b_{1,c} = a_1$. So let $i \geq 1$ and assume $b_{i-1,c} \to a_{i-1}$ and $b_{i,c} \to a_i$. Then from (4.3),

$$m_{i,c} = \left\lfloor \frac{\log b_{i-1,c}}{\log b_{i,c}} \right\rfloor \to \left\lfloor \frac{\log a_{i-1}}{\log a_i} \right\rfloor = n_i, \tag{4.5}$$

assuming that $\log_b a$ is not rational and hence $\log a_{i-1}/\log a_i$ is not an integer.

Then by (4.4), as

$$b_{i+1,c} = \frac{b_{i-1,c}}{b_{i,c}^{m_{i,c}}} - \frac{\theta_{i,c}}{c},$$

where $0 \leq \theta_{i,c} < 1$, it follows from the induction hypothesis and (4.5) that

$$b_{i+1,c} \to \frac{a_{i-1}}{a_i^{n_i}} = a_{i+1}.$$

5. A theorem

The next result is an attempt to obtain a result similar to (1.4), on the assumption that $A_{i,c}$ is "large". We believe that with the stronger assumption $A[i] > c + b^2\sqrt{c}, c = b^t$, that we always get the first alternative in (5.1).

Theorem 2.2 With $G_{i,c} = A_{i,c}/c$ and $A_{i,c} > c + \sqrt{c}$, we have

$$\left\lfloor \frac{\log G_{i-1,c}}{\log G_{i,c}} \right\rfloor = m_{i-1,c} \text{ or } 1 + m_{i-1,c}. \tag{5.1}$$

Proof. By inequalities (3.5) of our paper, with $r = m_{i-1,c}$, we get

$$\frac{G_{i-1,c}}{G_{i,c}^r} - \frac{(1 - \frac{1}{G_{i,c}^r})}{c(1 - \frac{1}{G_{i,c}})} < G_{i+1,c} \le \frac{G_{i-1,c}}{G_{i,c}^r}. \tag{5.2}$$

Also $1 \le G_{i+1,c}$ and (5.2) imply

$$1 \le \frac{G_{i-1,c}}{G_{i,c}^r}$$

and hence

$$r \le \left\lfloor \frac{\log G_{i-1,c}}{\log G_{i,c}} \right\rfloor. \tag{5.3}$$

If we now assume $A_{i,c} > c + \sqrt{c}$, we can deduce that

$$\left\lfloor \frac{\log G_{i-1,c}}{\log G_{i,c}} \right\rfloor \le r + 1. \tag{5.4}$$

For (5.2) and the inequality $G_{i+1,c} < G_{i,c}$ imply

$$\frac{G_{i-1,c}}{G_{i,c}^r} - \frac{1}{c(1 - \frac{1}{G_{i,c}})} < G_{i,c}. \tag{5.5}$$

If we now assume that $G_{i,c} > 1 + 1/\sqrt{c}$, (5.5) gives

$$\begin{aligned}
\frac{G_{i-1,c}}{G_{i,c}^r} &< G_{i,c} + \frac{1}{c\left(1 - \frac{1}{1+\frac{1}{\sqrt{c}}}\right)} \\
&= G_{i,c} + \frac{1}{\sqrt{c}}\left(1 + \frac{1}{\sqrt{c}}\right) \\
&< G_{i,c} + \frac{G_{i,c}}{\sqrt{c}} = G_{i,c}\left(1 + \frac{1}{\sqrt{c}}\right) < G_{i,c}^2.
\end{aligned}$$

Hence

$$\begin{aligned}
G_{i-1,c} &< G_{i,c}^{r+2} \\
\frac{\log G_{i-1,c}}{\log G_{i,c}} &< r + 2 \\
\left\lfloor \frac{\log G_{i-1,c}}{\log G_{i,c}} \right\rfloor &\le r + 1. \qquad (5.6)
\end{aligned}$$

Hence from (5.3) and (5.6), we have

$$\left\lfloor \frac{\log G_{i-1,c}}{\log G_{i,c}} \right\rfloor = r \text{ or } r+1.$$

6. The first two partial quotients of $\log_2(2^r+1)$

We conclude with some partial information about the continued fraction expansion of $\log_2(2^r+1)$.

If $r \geq 1$, $\log_2(2^r+1) = [r, n_1, \ldots]$, where

$$n_1 = \left\lfloor \log 2/\log\left(1+2^{-r}\right)\right\rfloor = \lfloor 2^r \log 2\rfloor \text{ or } \lfloor 2^r \log 2\rfloor + 1.$$

Both possibilities can occur. For example,

(a) $\log_2 3 = [1, 1, 1, 2, 2, 3, 1, 5, 2, 23, \ldots]$ ($\lfloor 2\log 2\rfloor = 1$);
(b) $\log_2 5 = [2, 3, 9, 2, 2, 4, 6, 2, 1, 1, \ldots]$ ($\lfloor 4\log 2\rfloor = 2$).

Proof. Assume $r \geq 1$. Then $2^r < 2^r + 1 < 2^{r+1}$ and hence

$$r < \log_2(2^r+1) < r+1,$$

so

$$\log_2(2^r+1) = r + \frac{1}{s},\ \ s > 1, \tag{6.1}$$

where

$$s = \log 2/\log\left(1+2^{-r}\right). \tag{6.2}$$

From (6.2) and the mean-value theorem, it follows that

$$\frac{1}{s}\log 2 = \log(2^r+1) - \log(2^r) = \frac{1}{2^r+\theta},$$

where $0 < \theta < 1$.

Hence $s = (2^r+\theta)\log 2$ and hence

$$2^r\log 2 < s < (2^r+1)\log 2 < 2^r\log 2 + 1.$$

Hence $n_1 = \lfloor s\rfloor = \lfloor 2^r\log 2\rfloor$ or $\lfloor 2^r\log 2\rfloor + 1$.

7. Acknowledgements

(i) This is a changed version of an earlier paper [2].
(ii) See `http://www.numbertheory.org/gnubc/logx` for a BC version of Algorithm 1.

References

[1] D. Shanks, *A logarithm algorithm*, Math. Tables and Other Aids to Computation 8 (1954). 60-64.

[2] K.R. Matthews, T. Jackson, *Heuristic versions of Shank's algorithm for computing the continued fraction of* $\log_b a$, Math. Tables and Other Aids to Computation 8 (1954). 60-64.

KEITH MATTHEWS
SCHOOL OF MATHEMATICS AND PHYSICS
UNIVERSITY OF QUEENSLAND
BRISBANE
AUSTRALIA 4072
E–mail: keithmatt@gmail.com