COMPUTER VERIFICATION
OF THE ANKENY–ARTIN–CHOWLA CONJECTURE
FOR ALL PRIMES LESS THAN 100 000 000 000

A. J. VAN DER POORTEN, H. J. J. TE RIELE, AND H. C. WILLIAMS

Abstract. Let \( p \) be a prime congruent to 1 modulo 4, and let \( t, u \) be rational integers such that \( (t + u\sqrt{p})/2 \) is the fundamental unit of the real quadratic field \( \mathbb{Q}(\sqrt{p}) \). The Ankeny-Artin-Chowla conjecture (AAC conjecture) asserts that \( p \) will not divide \( u \). This is equivalent to the assertion that \( p \) will not divide \( B_{(p-1)/2} \), where \( B_n \) denotes the \( n \)th Bernoulli number. Although first published in 1952, this conjecture still remains unproved today. Indeed, it appears to be most difficult to prove. Even testing the conjecture can be quite challenging because of the size of the numbers \( t, u \); for example, when \( p = 4094470441 \), then both \( t \) and \( u \) exceed \( 10^{330000} \). In 1988 the AAC conjecture was verified by computer for all \( p < 10^{9} \). In this paper we describe a new technique for testing the AAC conjecture and we provide some results of a computer run of the method for all primes \( p \) up to \( 10^{11} \).

1. Introduction

Let \( p \) denote a prime such that \( p \equiv 1 \pmod{4} \) and let \( \varepsilon = (t + u\sqrt{p})/2 \) (\( > 1 \)) be the fundamental unit in the real quadratic number field \( \mathbb{Q}(\sqrt{p}) \). In 1952 Ankeny, Artin and Chowla \cite{2} asked whether \( p \nmid u \) always and noted that \( p \nmid u \) for \( p < 2000 \) \((p \equiv 5 \pmod{8})\). This question was written in the form of a conjecture by Mordell \cite{15}, and has since become known as the Ankeny-Artin-Chowla conjecture (AAC conjecture). The conjecture is equivalent to stating that \( p \nmid u \) if \( t, u \) are the least positive integers such that

\[ t^2 - pu^2 = \pm 4. \]

It arose ultimately from expressions which were derived in \cite{2} for the value of \( hu/t \) modulo \( p \), where \( h \) is the class number of \( \mathbb{Q}(\sqrt{p}) \). One of these results is

\[ hu/t \equiv B_{(p-1)/2} \pmod{p}, \tag{1.1} \]

where \( B_n \) here denotes the \( n \)th Bernoulli number. Mordell \cite{15} noted that this was proved only for \( p \equiv 5 \pmod{8} \) in \cite{2}; it was later established for all \( p \equiv 1 \pmod{4} \) by Ankeny and Chowla \cite{4}. However, this result had been proved earlier (1948) by Kiselev \cite{12}. Ankeny and Chowla \cite{3} also noted that \( h < p \); hence, \( p | u \) if and only if \( p | B_{(p-1)/2} \), a fact also noted earlier by Kiselev \cite{12} and Carlitz \cite{7}.
Ankeny, Artin and Chowla also announced that
\[ 2hu/t \equiv (A + B)/p, \]
where

\[ A = \prod_{0 < r < p} r, \quad B = \prod_{0 < n < p} n, \]

and \((\frac{p}{2}) = 1, (\frac{p}{4}) = -1\). This was proved later by Carlitz [7]. Unfortunately, there does not seem to be any fast method of verifying the AAC conjecture for a given \( p \) by making use of either (1.1) or (1.2). The work of Fillebrown [9] suggests that computing Bernoulli numbers is very expensive, and there is no method known currently for computing \( A + B \mod p^2 \). Indeed, \( AB = (p - 1)! \) and it has only recently been possible to compute the values of the Wilson quotients \( w_p = ((p - 1)! + 1)/p \) up to \( 5 \times 10^8 \) (see Crandall, Dilcher and Pomerance [8]). In fact, in all previous attempts to verify the AAC conjecture for all primes \( p < L \), the value of \( u \) was computed modulo \( p \). We summarize this work in Table 1.1.

Ankeny, Artin and Chowla [1] also announced that
\[ 2hu/t \equiv (A + B)/p, \]
where

\[ A = \prod_{0 < r < p} r, \quad B = \prod_{0 < n < p} n, \]

and \((\frac{p}{2}) = 1, (\frac{p}{4}) = -1\). This was proved later by Carlitz [7]. Unfortunately, there does not seem to be any fast method of verifying the AAC conjecture for a given \( p \) by making use of either (1.1) or (1.2). The work of Fillebrown [9] suggests that computing Bernoulli numbers is very expensive, and there is no method known currently for computing \( A + B \mod p^2 \). Indeed, \( AB = (p - 1)! \) and it has only recently been possible to compute the values of the Wilson quotients \( w_p = ((p - 1)! + 1)/p \) up to \( 5 \times 10^8 \) (see Crandall, Dilcher and Pomerance [8]). In fact, in all previous attempts to verify the AAC conjecture for all primes \( p < L \), the value of \( u \) was computed modulo \( p \). We summarize this work in Table 1.1.

### Table 1.1. Verification of AAC for all \( p < L \).

<table>
<thead>
<tr>
<th>( L )</th>
<th>Investigator(s)</th>
<th>Date</th>
<th>Machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>Ankeny, Artin, Chowla ( p \equiv 5 \mod 8 ) only [2]</td>
<td>1952</td>
<td>—</td>
</tr>
<tr>
<td>100 000</td>
<td>Goldberg [10]</td>
<td>1954</td>
<td>SEAC</td>
</tr>
<tr>
<td>6 270 714</td>
<td>Beach, Williams, Zarnke [6]</td>
<td>1971</td>
<td>IBM 360-65</td>
</tr>
<tr>
<td>100 028 010</td>
<td>Soleng [20]</td>
<td>1986</td>
<td>Cyber 171</td>
</tr>
<tr>
<td>1 000 000 000</td>
<td>Stephens, Williams [21]</td>
<td>1988</td>
<td>Amdahl 5850</td>
</tr>
</tbody>
</table>

Ankeny, Artin and Chowla did not provide any algebraic justification that would suggest a negative response to their question. Perhaps the conjecture is true because of considerations that seem far from our understanding; however, one might ask whether the data so far collected really should be persuasive in making one believe in this conjecture. It is certainly a most tempting conjecture to test, particularly if one subscribes to the familiar “log log argument”. This reasoning is based on the seemingly reasonable assumptions that the probability that \( p | u \) is \( 1/p \) and that trials for different \( p \) values are independent events. It then follows that we might expect that the number of exceptions to the conjecture in the interval \( [x, y] \) is given by

\[ \sum_{\substack{x \leq p \leq y \\ p \equiv 1 \mod 4}} \frac{1}{p} \approx \frac{1}{2} \log \left( \frac{\log y}{\log x} \right) = N(x, y). \]

For \( x = 5, y = 10^9 \), we get \( N(x, y) = 1.28 \) and for \( x = 5, y = 10^{11} \), we get only a small increase in \( N(x, y) \) to 1.37. Thus, even if the AAC conjecture is false, one is not entirely surprised that there are no counterexamples up to \( 10^9 \). This also makes the AAC conjecture a most tempting conjecture to test, since it seems that there might be an exception to it within a range that modern computers would have the capability to search.

The purpose of this paper is to present a new algorithm for verifying the AAC conjecture for a given prime \( p \). We will also describe the implementation and
running of this algorithm on two fast computers. Our computer runs allowed us to verify the AAC conjecture for all primes between $10^9$ and $10^{11}$; hence, we now know that the conjecture holds for all $p < 10^{11}$.

The strategy employed in devising our algorithm is based on the following simple observation. If

$$\varepsilon^k = (X + Y \sqrt{p}) / 2 = (X_k + Y_k \sqrt{p}) / 2 \quad (X_1 = t, Y_1 = u)$$

and $p \nmid k$, then $p | u$ if and only if $p | Y$. This is very easy to see on expanding the $k$th power of $(t + u \sqrt{p}) / 2$ by the binomial theorem and noting that

$$2^{k-1} Y \equiv k t^{k-1} u \pmod{p}.$$  

We estimate a value of $\log_2 \varepsilon^k$ for some $k$ such that $p \nmid k$ and use the infrastructure ideas of Shanks [18] to determine a value of $\eta \in \mathbb{Z}$ such that $p | Y$ if and only if $p | \eta$. To determine this estimate we make use of the analytic class number formula

$$2h R = \sqrt{p} L(1, \chi_p),$$  

where $R (= \log \varepsilon)$ is the regulator of $K = \mathbb{Q}(\sqrt{p})$ and $L(1, \chi_p)$ is the Dirichlet $L$-function for $K$ evaluated at $s = 1$. We also note that $h < \sqrt{p}$ (see, for example, Slavutskii [19]).

Thus, our intention, then, is to compute quickly an estimate $E$ for $\log_2 \varepsilon^k$ and use this to determine whether or not $p$ divides $Y$, as above. The value of $\varepsilon^k$ for any $k$ could, of course, be determined from the continued fraction expansion of $(1 + \sqrt{p}) / 2$. However, a priori, one would have to compute so many terms of the continued fraction that the running time would be prohibitive. Instead, with knowledge of $E$, we can use Shank's “infrastructure” to greatly accelerate our search through the continued fraction to determine an accurate value of $\log_2 \varepsilon^k$. Since we need only determine $Y \pmod{p}$, we don’t need to compute the integers $X$ and $Y$, which will usually be unmanageably enormous. Thus, we determine a number $\eta \pmod{p}$, such that $p$ divides $Y$ if and only if $p$ divides $\eta$. Given an accurate value of $\log_2 \varepsilon^k$, we can compute $\eta$ via the infrastructure of the principal ideal class of $\mathbb{Q}(\sqrt{p})$. Thus, our overall algorithm is made up of three components.

1. Find an estimate $E$ of $h R_2$, where $R_2 = \log_2 \varepsilon$, by estimating $L(1, \chi_p)$ and using (1.3).
2. Use $E$ to determine an integral multiple $k R_2$ of $R_2$ and check that $k R_2 < 8p$. This value of $k$ will likely be $h$, but whether it is or not, our estimate is probably sufficiently good that $k$ is not very different from $h \ (< \sqrt{p})$. Thus, it is most likely that $p \nmid k$. Since $R_2 = \log_2 \varepsilon \geq \log_2 \left(\sqrt{p - 4} + \sqrt{p}\right) > 8$ for the values of $p$ in our search range, our check that $k R_2 < 8p$ ensures that $p \nmid k$.
3. Compute $\eta = \eta(k R)$ and verify that $\eta \neq 0$.

2. Estimation of $h R_2$

It is well known that we can write $L(1, \chi_p)$ in its Euler product form as

$$L(1, \chi_p) = \prod_q \left(1 - \chi_p(q)/q\right)^{-1},$$

where the product is taken over all the primes $q$ and the character $\chi_p(q)$ is the same as the Kronecker symbol $(p/q)$. Bach [5] has developed a technique for estimating.
\[ \log L(1, \chi_p) \] which has been found to be very effective in practice (see §2 of Jacobson, Lukes and Williams [11]). For some suitable \( T \), we compute
\[ C(T) = \sum_{i=0}^{T-1} (T + i) \log(T + i) \]
and
\[ a_j = (T + j) \log(T + j)/C(T) \quad (j = 0, 1, 2, \ldots, T - 1). \]

By Theorem 9.2 of [5], we have (under the Extended Riemann Hypothesis for \( L(s, \chi_p) \))
\[ \left| \log L(1, \chi_p) - \sum_{i=0}^{T-1} a_i \log B(T + i) \right| < A(T, p), \]
where
\[ B(x) = \prod_{q<x} (1 - \chi_p(q)/q)^{-1}, \]
\[ A(T, p) = \frac{A \log p + B}{\sqrt{T \log T}}, \]
and \( A, B \) here are constants which are explicitly given in Table 3 of [5]. The important item to note here is that \( A(T, p) \) will be quite small for even modest values of \( T \) because \( \log p \) will not be large compared to \( \sqrt{T} \).

Let
\[ S(T, p) = \sum_{i=0}^{T-1} a_i \log B(T + i). \]

As pointed out in [11], we can write this as
\[ S(T, p) = \sum_{q<2T-1} w(q) \log [q/(q - (p/q))], \]
where
\[ w(q) = \begin{cases} 1, & \text{when } q < T, \\ \sum_{j=q-T+1}^{T-1} a_j, & \text{when } T \leq q < 2T - 1. \end{cases} \]

Since we will need to evaluate \( S(T, p) \) for many values of \( p \), it is convenient to precompute and store in a large table the quadratic residues and nonresidues of \( p \) together with the values of \( w(q) \log[q/(q + 1)] \), and \( w(q) \log[q/(q - 1)] \), for all the primes \( q < 2T - 1 \). It is then a simple matter to compute \( S(T, p) \) by doing only table look-ups and additions. Our estimate \( E \) for \( hR_2 \) is then computed as
\[ E = \frac{\sqrt{p}}{\log 4} \exp(S(T, p)). \]

It should be emphasized here that this method for finding \( E \) usually provides a much better result than what Bach’s estimate for the error suggests. Also, although on average the error decreases with increasing \( T \), in many cases the error for small \( T \) (say \( T = 100 \)) is comparable to the error for larger \( T \) (say \( T = 5000 \)). We illustrate these remarks in Table 2.1.
Table 2.1. Some experiments with Bach’s method to estimate $hR_2$.

<table>
<thead>
<tr>
<th>T</th>
<th>E</th>
<th>E/R_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p = 9999999241$</td>
<td>100</td>
<td>374191.1</td>
</tr>
<tr>
<td>$R_2 = 377424.5$</td>
<td>200</td>
<td>377174.2</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>382290.3</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>377796.8</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>375368.2</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>377872.1</td>
</tr>
<tr>
<td>$p = 9999999253$</td>
<td>100</td>
<td>152887.6</td>
</tr>
<tr>
<td>$R_2 = 50890.1$</td>
<td>200</td>
<td>155708.1</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>154600.0</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>154297.7</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>153518.1</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>152657.5</td>
</tr>
<tr>
<td>$p = 9999994117$</td>
<td>100</td>
<td>268436.8</td>
</tr>
<tr>
<td>$R_2 = 9998.8$</td>
<td>200</td>
<td>268043.1</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>271177.0</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>271498.3</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>271266.8</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>270060.7</td>
</tr>
</tbody>
</table>

We next need to know how to use $E$ to find $kR_2$. For this we will require some results concerning continued fractions and their relationship to the ideals in the ring $\mathcal{O}_K$ of algebraic integers in $K = \mathbb{Q}(\sqrt{-p})$.

3. Continued fractions and ideals

In this section we will briefly review some well-known results concerning the ideals of $\mathcal{O}_K$ and continued fractions. For proofs of these results we refer the reader to Stephens and Williams [21], Mollin [14] or Williams and Wunderlich [22].

By $\langle a_0, a_1, a_2, \ldots, a_n, \ldots \rangle$ we denote the simple continued fraction

$$
a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots + \cfrac{1}{a_n + \ddots}}}}
$$

The partial quotients $a_i \geq 1$ ($i \geq 0$) and the convergents $C_n = \langle a_0, a_1, a_2, \ldots, a_n \rangle$ are given by $C_n = A_n/B_n$, where $A_{-2} = 0, B_{-2} = 1, A_{-1} = 1, B_{-1} = 0$ and

$$
A_{i+1} = a_{i+1}A_i + A_{i-1},
$$

$$
B_{i+1} = a_{i+1}B_i + B_{i-1} \quad (i = -1, 0, 1, \ldots).
$$

Note that $B_0 = 1, B_1 = a_1$ and $B_i \geq 1$ for $i \geq 0$. Also

$$
A_nB_{n-1} - B_nA_{n-1} = (-1)^{n+1}.
$$
Let $P, Q, D \in \mathbb{Z}$ such that $D > 0$, $\sqrt{D} \not\in \mathbb{Q}$ and $Q|D - P^2$. The continued fraction of $\phi = \phi_0 = (P + \sqrt{D})/Q$ is given by

$$\phi = (a_0, a_1, a_2, \ldots, a_{n-1}, \phi_n).$$

The partial quotients are determined by making use of the formulas

$$P_{i+1} = a_i Q_i - P_i,$$
$$Q_{i+1} = (D - P_{i+1}^2)/Q_i = Q_{i-1} - a_i (P_{i+1} - P_i),$$
$$a_{i+1} = \lfloor (P_{i+1} + d)/Q_{i+1} \rfloor = \lfloor (P_{i+1} + \sqrt{D})/Q_{i+1} \rfloor,$$

where $d = \lfloor \sqrt{D} \rfloor, P_0 = P, Q_0 = Q, a_0 = [\phi_0]$. Also,

$$\phi_n = (P_n + \sqrt{D})/Q_n.$$

Note that

$$\phi_{i+1} = \frac{1}{\phi_i - a_i};$$

hence, $\phi_i > 1$ when $i > 0$. At some point in the computation of the continued fraction of $\phi$, we must find some $\phi_k$ such that $\overline{\phi}_k < 0$; furthermore, this value of $k$ will be $O(\log |Q|/\sqrt{D})$.

If we put $\theta_1 = 1$ and define

$$\theta_k^{-1} = \prod_{i=1}^{k-1} \phi_i \quad (k > 1),$$

then

$$\theta_i = (-1)^{i-1} (A_{i-2} - \phi B_{i-2})$$

and

$$\theta_i \overline{\theta}_i = (-1)^{i-1} Q_{i-1}/Q_0.$$

We also put $\Psi_1 = 1$ and define

$$\Psi_k = \prod_{i=1}^{k-1} \psi_i \quad (k > 1),$$

where

$$\psi_i = \frac{1}{(\overline{\phi}_i)^{-1}} = \frac{1}{(P_i + \sqrt{D})/Q_i-1};$$

hence,

$$\Psi_i = \overline{\theta}_i = |A_{i-2} - \overline{\phi} B_{i-2}|.$$

Since

$$\phi = \frac{\phi_{i-1} A_{i-2} + A_{i-3}}{\phi_{i-1} B_{i-2} + B_{i-3}} \quad (i \geq 1),$$

we get

$$A_{i-2} - \overline{\phi} B_{i-2} = \frac{A_{i-2} B_{i-3} - A_{i-3} B_{i-2}}{\phi_{i-1} B_{i-2} + B_{i-3}}$$

thus, by (3.1) and (3.4) we get

$$\Psi_i = 1/|\phi_{i-1} B_{i-2} + B_{i-3}|.$$

\footnote{Here, as is customary, we use $\overline{\alpha}$ to denote the conjugate of $\alpha$ in $K$.}
Let \([\alpha, \beta]\) denote the module \(\{\alpha x + \beta y : x, y \in \mathbb{Z}\}\). If \(D\) is squarefree and if we put \(\omega = (1 + \sqrt{D})/2\) when \(D \equiv 1 \pmod{4}\) or \(\omega = \sqrt{D}\) otherwise, then the maximal order \(\mathcal{O}_K\) (ring of algebraic integers of \(K\)) is given by \(\mathcal{O}_K = [1, \omega]\). Any ideal of \(\mathcal{O}_K\) can be written as \(a = [L(a), \beta]\), where \(L(a)\) is the least positive rational integer in \(a = b + c\omega (b, c \in \mathbb{Z})\), and \(c|b|L(a), L(a)|\beta\omega\). If \(c = 1\), we say that \(a\) is primitive. A primitive ideal is said to be reduced if \(L(a)\) is a minimum in \(a\); that is, there does not exist any nonzero \(\alpha \in a\) such that \(|\alpha| < L(a)\) and \(|\overline{\alpha}| < L(a)\).

**Theorem 3.1.** An ideal \(a\) of \(\mathcal{O}_K\) is reduced if and only if there exists some \(\beta \in a\) such that \(a = [L(a), \beta]\), where \(\beta > L(a)\) and \(-L(a) < \beta < 0\).

If \(a = [L(a), \beta]\), we define \(\overline{a}\) to be the ideal \([L(a), \overline{\beta}]\).

**Theorem 3.2.** If \(a\) is a reduced ideal of \(\mathcal{O}_K\), then so is \(\overline{a}\).

**Proof.** Let \(a = [L(a), \beta]\). If \(\overline{a}\) is not reduced, there must exist some \(\alpha \in \overline{a}\) such that \(|\alpha| < L(a)\) and \(|\overline{\alpha}| < L(a)\). But since \(\overline{\alpha} \in a\) and \(a\) is reduced, this is impossible.

By our previous observations it is easy to see that any primitive ideal \(a\) of \(\mathcal{O}_K\) can be put in the form \([Q/r, (P + \sqrt{D})/r]\) where \(Q, P \in \mathbb{Z}\), \(r = 2\) if \(D \equiv 1 \pmod{4}\) or \(r = 1\) otherwise. Furthermore, \(Q|D - P^2\). Hence we can expand \((P + \sqrt{D})/Q\) into a continued fraction and produce a sequence of ideals

\[
a_1 = (a_1), a_2, a_3, \ldots,
\]

where

\[
a_i = [Q_{i-1}/r, (P_{i-1} + \sqrt{D})/r].
\]

All of these ideals lie in the same equivalence class. Indeed,

\[
(Q_0a_1)(a_i) = (Q_{i-1})a_1.
\]

Thus, by (3.6) and (3.4) we get

\[
(Q_0)s_i = (Q_0\Psi_i)a_1.
\]

We are now able to mention some useful theorems.

**Theorem 3.3.** If \(a_k\) is reduced, then \(L(a_k) < 2\sqrt{D}/r\).

**Theorem 3.4.** If \(L(a_k) < \sqrt{D}/r\), then \(a_k\) is reduced.

**Theorem 3.5.** If \(\overline{a}_k < 0\), then \(a_{k+1}\) is reduced.

If we begin the sequence (3.8) with an ideal \(a\), such as \(\mathcal{O}_K\) itself, which is already reduced, then the sequence is completely periodic and is made up exclusively of the reduced ideals that are equivalent to \(a\). If, moreover, \(l\) is the least positive integer such that \(a_1 = a_{l+1}\), then it is readily shown that for any positive integer \(k\)

\[
\epsilon^k = \Psi_{lk+1}.
\]

Furthermore, when \(a = \overline{a}\) (\(a\) is an ambiguous ideal), there is a symmetry property, namely \(\overline{a}_{i+1} = a_{i+1}\), by which we are able to compute \(\epsilon\) by looking only halfway through the cycle \(a_1, a_2, \ldots, a_l\).

**Theorem 3.6.** If \(a_1 = a\) is reduced and ambiguous, there must exist a least positive integer \(s\) such that either \(P_s = P_{s+1}\) or \(Q_s = Q_{s+1}\). If \(P_s = P_{s+1}\), then \(l = 2s\) and

\[
\epsilon = \Psi_{s+1}/[\overline{\Psi}_{s+1}] = Q_0\Psi_{s+1}^2/Q_s.
\]
If $Q_s = Q_{s+1}$, then $l = 2s + 1$,
\begin{equation}
\epsilon = \Psi_{s+2}/|\Psi_{s+1}| = Q_0 \Psi_{s+1} \Psi_{s+2}/Q_s,
\end{equation}
and
\begin{equation}
R_2 = \log_2 \epsilon = \log_2(Q_0 \psi_{s+1}/Q_s) + 2 \sum_{i=1}^s \log_2 \psi_i.
\end{equation}

4. Some results concerning ideals and continued fractions

In order to develop our algorithms, we will need some further results concerning ideals and continued fractions. We first require a simple lemma.

**Lemma 4.1.** If $k > 1$ and $\phi_k < 0$, then $0 < Q_k < 2\sqrt{D}$, $|P_k| < \sqrt{D}$ and $Q_{k-1} > 0$. If $-1 < \phi_k < 0$, then $P_k > 0$.

**Proof.** Since $\phi_k > 1$, we have $\phi_k - \phi_k = 2\sqrt{D}/Q_k > 1$; hence, $0 < Q_k < 2\sqrt{D}$. Since $P_k + \sqrt{D} > Q_k > 0$ and $P_k - \sqrt{D} < 0$, we must also have $|P_k| < \sqrt{D}$. Since $Q_k Q_{k-1} = D - P_k^2$, we get $Q_{k-1} > 0$. Finally, since $2P_k/Q_k = \phi_k + \phi_k$, we see that $P_k > 0$ when $\phi_k > -1$.

Our next result and its converse provide us with a simple criterion for determining when $a_k$ is reduced.

**Theorem 4.2.** If $k \geq 1$, $Q_{k-1} > 0$ and $a_k$ is reduced, then $-1 < \phi_k < 0$ and $\psi_k > 1$.

**Proof.** We know that $L(a_k) = Q_{k-1}/r$ and, by Theorem 3.3, that $L(a_k) < 2\sqrt{D}/r$ when $a_k$ is reduced. Hence, $0 < Q_{k-1} < 2\sqrt{D}$. Put
\[
\gamma = L(a_k) \phi_k^{-1} = (\sqrt{D} - P_k)/r = (-a_{k-1}Q_{k-1} + P_{k-1} + \sqrt{D})/r \in a_k.
\]
Since $\phi_k > 1$, we get $0 < \gamma < L(a_k)$ which means that $0 < \sqrt{D} - P_k < Q_{k-1} < 2\sqrt{D}$; consequently, $P_k + \sqrt{D} > 0$ and $Q_{k} > 0$. Since $a_k$ is reduced, we must have $|\gamma| > L(a_k)$. It follows that $|\phi_k| < 1$. Also, since $D - P_k^2 = Q_k Q_{k-1} > 0$, we have $|P_k| < \sqrt{D}$ and $\phi_k < 0$.

**Theorem 4.3.** If $-1 < \phi_k < 0$ ($k \geq 1$), then $Q_{k-1} > 0$ and $a_k$ is reduced.

**Proof.** By (3.2), we have
\[
\phi_k = \frac{1}{\phi_k - 1 - a_{k-1}},
\]
hence, we must have $\phi_k - 1 - a_{k-1} < -1$. Thus
\[
a_{k-1} - \phi_k - 1 = \phi_k - 1 + (\sqrt{D} - P_{k-1})/Q_{k-1} > 1.
\]
Now $Q_{k-1} > 0$ by Lemma 1.1 and $a_k = [Q_{k-1}/r, (P_{k-1} - \sqrt{D})/r] = [L(a_k), \beta]$, where $L(a_k) = Q_{k-1}/r$ and $\beta = a_{k-1}Q_{k-1}/r + (\sqrt{D} - P_{k-1})/r > L(a_k)$. Note further that
\[
\beta = a_{k-1}Q_{k-1}/r - (\sqrt{D} + P_{k-1})/r \quad \text{and} \quad a_{k-1} = [(P_{k-1} + \sqrt{D})/Q_{k-1}] ;
\]
hence, $-L(a_k) < \beta < 0$. By Theorem 3.2, we know that $a_k$ is reduced, and by Theorem 4.1 we know that $a_k$ is reduced.
We next suppose that
\[ a = [Q/r, (P + \sqrt{D})/r], \]
where \( Q > 0 \) and \( 0 < P < Q \). Notice that any ideal of \( \mathcal{O}_K \) must have such a representation.

**Theorem 4.4.** If \( k > 0 \) is the least integer such that \( \overline{\alpha}_k < 0 \), then \( \Psi_i \leq 1 \) for \( 1 \leq i \leq k \).

**Proof.** The theorem is certainly true if \( i = 1 \). If \( k \geq i = 2 \), then \( \Psi_1 = \psi_1 = |P_1 + \sqrt{D}|/Q_0 \). Since \( \overline{\alpha}_1 > 0 \), we cannot have \( \alpha_i \) reduced by Theorem 4.2, hence, by Theorem 3.4 we must have \( Q_0 > \sqrt{D} \), and therefore \( 0 \leq a_0 < 1 \). If \( a_0 = 0 \), then \( P_1 = -P_0 \) and \( \psi_1 = | -P_0 + \sqrt{D}|/Q_0 \). In this case \( 0 < \sqrt{D}/Q_0 < 1 \) and \( 1 < -P_0/Q_0 < 0 \); hence, \( \psi_1 < 1 \). If \( a_0 = 1 \), then \( P_1 = Q_0 - P_0 \) and \( 1 + \sqrt{D} = Q_0 - P_0 + \sqrt{D} > 0 \). If \( P_0 = \sqrt{D} \), then \( 0 < \sqrt{D}/Q_0 < 1 \) if \( P_0 < \sqrt{D} \), then \( \overline{\alpha}_0 < 0 \), which by Theorem 3.5 means that \( \alpha_k \) must be reduced, a contradiction. If \( k \geq i \geq 3 \), then we have \( \overline{\alpha}_{i-1} > 0 \), \( B_{i-2} > 1 \), \( B_{i-3} > 1 \); hence, by (3.7) we get \( \Psi_i < 1 \).

**Corollary 4.5.** If, in the sequence of ideals (3.8), \( \alpha_i \) is not reduced, then \( \Psi_i \leq 1 \).

**Proof.** Since \( \alpha_i \) is not reduced, we must have \( \overline{\alpha}_{i-1} > 0 \) by Theorem 3.5. Thus, \( k > i - 1 \) or \( i \leq k \).

**Corollary 4.6.** If, in the sequence (3.8), \( \alpha_i \) is the first reduced ideal, then \( \Psi_i \leq 1 \).

**Proof.** Since \( \alpha_i \) is reduced, we must have \( i \leq k + 1 \) by Theorem 3.5. If \( i \leq k \), the result follows from the theorem. Suppose \( i = k + 1 \). If \( \overline{\alpha}_k < -1 \), then \( |\psi_k| < 1 \) and \( \Psi_i = |\psi_k|, \Psi_k < 1 \); if \( -1 < \overline{\alpha}_k < 0 \), then \( \alpha_i \) is a reduced ideal by Theorem 4.3, contradicting the definition of \( \alpha_i \).

In developing the algorithms that follow, it is essential to be able to perform baby-steps (the process of moving through the sequence (3.8) one step at a time) and giant-steps (the process of moving through the sequence (3.8) by taking several baby-steps at once). In what follows we will describe a simple procedure for taking baby-steps, and in the next section we will show how to take giant steps. We will assume that \( \alpha_i \) is reduced.

We define \( \zeta_j = \zeta(\alpha_j) \) and \( \rho_j = \rho(\alpha_j) \) by
\[ 2^{\zeta_j - 1} < \Psi_j < 2^{\zeta_j}, \quad \rho_j = 2^{\zeta_j}/\Psi_j. \]

We now have the following baby-step algorithm.

**Algorithm 4.7.** Given \( \alpha_j, \zeta_j, \rho_j \); compute \( \alpha_{j+1}, \zeta_{j+1}, \rho_{j+1} \).

1. \( \alpha_{j+1} = [Q_j/r, (P_j + \sqrt{D})/r], \quad \chi_j = (\sqrt{D} - P_j)/Q_j. \)
2. Put \( \rho = \rho_j \chi_j, \quad \zeta = \zeta_j. \)
3. **while** \( \rho < 1 \)
   
   \( \rho = 2 \rho, \quad \zeta = \zeta + 1 \)

   **end while**

4. \( \rho_{j+1} \leftarrow \rho, \quad \zeta_{j+1} \leftarrow \zeta. \)

Note that the process of determining \( Q_j, P_j \) (\( \alpha_{j+1} \)) from \( Q_{j-1}, P_{j-1} \) (\( \alpha_j \)) is given in §3.
Since all of the ideals in (3.8) are reduced, we must have

\[ Q \]

some

Proof. In this case, these ideals are generated by the associated continued fraction algorithm. By (3.9) we can write

\[ b \]

where

\[ a \]

Thus,

If

by Lemma 4.1 and (3.5).

If

which means that

\[ a \]

continued fraction algorithm into a very specific order. This organization was called the "infrastructure" of the class by Shanks, the discoverer of this phenomenon.

License or copyright restrictions may apply to redistribution; see http://www.ams.org/journal-terms-of-use
Thus, we can find a reduced principal ideal of distance $x$ from $b_1 = [1, \omega]$ by performing about $x/\delta_x$ multiply-reduction steps, using an ideal $b_s$ as a multiplier, instead of the roughly $x/1.186569$ (Lévy’s law, see [14] pp. 243 – 244) baby-steps that would likely be required. This is the process of taking giant-steps (of size $\delta_x$).

We will now show how this idea can be used in the development of some algorithms.

**Algorithm 5.1.** Given $b(x), b(y), \rho(x), \rho(y)$; compute $b(x + y), \rho(x + y)$.

1. Compute $(u)a_1 = b(x)b(y)$ (say by using the technique described in §3 of [21]). Put $\rho_1 = u\rho(x)\rho(y)$.
2. $a_1 := [Q_0/r, (P_0 + \sqrt{D})/2], \ a_0 = [(P_0 + d)/Q_0], \ i \leftarrow 1$.
3. while $\rho_i \geq 1$
   
   $P_i = a_{i-1}Q_{i-1} - P_{i-1}$
   
   $Q_i = (D - P_i^2)/Q_{i-1}$
   
   $a_i = [(P_i + d)/Q_i]$
   
   $\rho_{i+1} = \rho_i\left(\sqrt{D} - P_i\right)/Q_i$
   
   $i \leftarrow i + 1$

end while

4. $b(x + y) = a_{i-1} = [Q_{i-2}/r, (P_{i-2} + \sqrt{D})/r], \rho(x + y) = \rho_{i-1}$.

**Proof (of correctness of Algorithm 5.1).** We have $b_s = b(x), b_t = b(y)$ for some $s, t$, and $\rho_t = \rho_s/\Psi_t$. Put $j = i - 1$ for the value of $j$ produced after the execution of step 3. We must have $\rho_{j+1} < 1$ and $a_j = (\Psi_j \Psi_s \Psi_t/u)$. Let $k$ be the least positive integer such that $a_k$ is reduced.

Case 1. $(j < k)$. Here $j + 1 \leq k$ and $a_j$ is not reduced: hence, by Corollary 4.5 we have $\Psi_j \leq 1$. If $a_{j+1}$ is reduced, then $j + 1 = k$ and $\Psi_{j+1} \leq 1$ by Corollary 4.6. If $a_{j+1}$ is not reduced, then we also have $\Psi_{j+1} \leq 1$. Thus, $\Psi_j \leq \rho_1$ and $\Psi_{j+1} \leq \rho_1$, a contradiction.

Case 2. $(j \geq k)$. In this case $a_j$ is reduced and $a_j = b_m$, where $\Psi_m = \Psi_j \Psi_s \Psi_t/u$ and $\Psi_{m+1} = \Psi_j \Psi_s \Psi_t/u$. It follows that

$$\Psi_m \leq 2^{x+y}, \quad \Psi_{m+1} > 2^{x+y};$$

hence $a_j = b_m = b(x + y)$. Also, $\rho(x + y) = 2^{x+y}/\Psi_m = \rho_1/\Psi_j = \rho_j$. \hfill \Box

**Algorithm 5.2.** $(D > 9)$ Given some real $x \geq 1$, compute $b(x), \rho(x)$.

1. Put $j = \lceil \log_2 x \rceil, \ y = x/2^j$.
2. Put $b(y) = b_1, \rho(y) = 2^y$.
3. for $m = 1$ to $j$
   
   $b(y) \leftarrow b(2y)$
   
   $\rho(y) \leftarrow \rho(2y)$
   
   $y \leftarrow 2y$

end for

4. $b(x) \leftarrow b(y), \rho(x) \leftarrow \rho(y)$. 

**Proof (of correctness of Algorithm 5.2).** Since $0 < y \leq 1$, we know by Corollary 4.10 that $b(y) = b_1$. Let $v = x/2^j$. Since $x = 2^jv$, we see that in step 4 we have $b(y) = b(2^jv) = b(x), \rho(y) = \rho(2^jv) = \rho(x)$. \hfill \Box

We will now show how to incorporate Algorithms 4.7, 5.1 and 5.2 into an algorithm which determines an integral multiple of $R_2$. We first need the following definition.
Case 1. (hence, (3.9), we define $\delta(a_i, a_1) = \log_2 \Psi_i$. If $a_1 = O_K$, we write $\delta(a_i)$ for $\delta(a_i, a_1)$.

Note that $\delta(b(x)) = x - \log_2 \rho(x)$. Also, it is easy to see that $\overline{b_j b_j} = (L(b_j))$; hence,

$$\overline{v_j} \Psi_j = L(b_j)\theta,$$

where $\theta$ is some positive unit of $K$. It follows that

$$\delta(\overline{v_j}) = -\delta(b_j) + tR_2 + \log_2 L(b_j), \quad (5.1)$$

where $t \in \mathbb{Z}$. If, for reals $a, b, c$ with $c \neq 0$, we say that $a \equiv b \pmod{c}$ whenever $(a - b)/c \in \mathbb{Z}$, we can write (5.1) as

$$\delta(\overline{v_j}) = -\delta(b_j) + \log_2 L(b_j) \quad (\text{mod } R_2). \quad (5.2)$$

Furthermore, since $\delta(b_j)$ is a strictly increasing function of $i$, we observe that if $\delta(b_k) \equiv \delta(b_j) + u \pmod{R_2} (k \geq j)$ and $0 \leq u \leq \delta(b_i, b_j)$, then $b_k \in \{b_j, b_{j+1}, \ldots, b_i\}$. 

Now let $\theta$ be any positive unit of $K$ such that $|\log_2 \theta - E| < K$. Then $\log_2 \theta = E - V$ with $|V| < K$ and $E \equiv V \pmod{R_2}$.

We are now able to present our algorithm for determining an integral multiple of $R_2$.

**Algorithm 5.4.** Find an integral multiple of $R_2$ from an estimate $E$.

1. Select (by trial) some parameter $c$ with $c > B = \lfloor \log_2(2\sqrt{D}/r) \rfloor$.
2. Compute $a_1 = b(E)$ and $\rho(E)$ (Algorithm 5.2). Compute the set of ideals $S = \{a_1, a_2, \ldots, a_t\}$ together with their associated $\rho$ and $\zeta$ values (Algorithm 4.7) until $\zeta > c + B$.
3. Compute the ideals $b(c), b(2c), \ldots$ and associated $\rho(c), \rho(2c), \ldots$ (Algorithms 4.7 and 5.1) until either $b(ic) \in S$ or $\overline{b(ic)} \in S$.
   (a) If $b(ic) \in S$, then $kR = E - ic + \zeta + \log_2 (\rho(ic)/\rho_j \rho(E))$, when $b(ic) = a_j$.
   (b) If $\overline{b(ic)} \in S$, then $kR = E + ic + \zeta - \log_2 (L(b(ic)) \rho(ic) \rho_j \rho(E))$, when $\overline{b(ic)} = a_j$.

**Proof (of correctness of Algorithm 5.4).** Put $m = \lfloor K/c \rfloor + 2$.

Case 1. $(V - \log_2 \rho(E) + B > 0)$. In this case we put $i = \lceil (V - \log_2 \rho(E) + B)/c \rceil$. Then

$$V - \log_2 \rho(E) + B = ic - f \quad (0 \leq f < c)$$

and

$$ic = V - \log_2 \rho(E) + B + f < V + B + c < K + 2c.$$
It follows that \( i \leq m \). Now
\[
\delta(b(ic)) - \delta(a_1) = ic - \log_2 \rho(ic) - E + \log_2 \rho(E)
\]
\[
\equiv ic - \log_2 \rho(ic) - V + \log_2 \rho(E) 
\]
\[
= f + B - \log_2 \rho(ic) \pmod{R_2}.
\]
Since \( \log_2 \rho(ic) < B \), we have \( f + B - \log_2 \rho(ic) > 0 \) and \( f + B - \log_2 \rho(ic) < c + B \).
Thus, we must have \( b(ic) \in S \). If \( b(ic) = a_j \), then
\[
\delta(b(ic)) \equiv \delta(a_1, a_j) + \delta(b(E)) \pmod{R_2}.
\]
From this we get
\[
ic - \log_2 \rho(ic) \equiv E - \log_2(\rho(E)) + \zeta_j - \log_2 \rho_j \pmod{R_2}.
\]
Case 2. \( (V - \log_2 \rho(E) + B \leq 0) \). If we put \( i = \lfloor \lfloor V - \log_2 \rho(E) \rfloor / c \rfloor \), then
\[
-V + \log_2 \rho(E) = ic + f \quad (0 \leq f < c),
\]
and since \( ic + f < K + B \), we see that \( i \leq m - 1 \). Furthermore, by (5.2)
\[
\delta(b(ic)) - \delta(a_1) \equiv -ic + \log_2(L(b(ic))\rho(ic)) - E + \log_2 \rho(E)
\]
\[
\equiv f + \log_2(L(b(ic))\rho(ic)) \pmod{R_2}.
\]
Since \( 0 < f + \log_2(L(b(ic))\rho(ic)) < c + B \), we must have \( \overline{b}(ic) \in S \). If \( a_j = \overline{b}(ic) \), then by (5.2)
\[
\delta(a_1, a_j) + \delta(b(E)) \equiv -\delta(b(ic)) + \log_2(L(b(ic))\rho(ic)) \pmod{R_2}.
\]
Thus, \( kR_2 = E + ic + \zeta_j - \log_2(L(b(ic))\rho(ic)\rho_j\rho(E)) \).

6. The algorithm for determining \( \eta \)

There remains the problem of determining whether or not \( p \mid Y \). Let \( b(x) = b_j = (\Psi'_j) \), where
\[
\Psi'_j = (W_j + Z_j\sqrt{D})/2
\]
and \( W_j, Z_j \in \mathbb{Z} \). If \( (W_j, D) = 1 \), we define a pair \((\xi(x), \eta(x))\) by:
1. \( \xi(x), \eta(x) \in \mathbb{Z} \);
2. \( 0 \leq \xi(x), \eta(x) < D \);
3. \( \xi(x), \eta(x) \) not both zero;
4. \( Z_j\xi(x) \equiv W_j\eta(x) \pmod{D} \).

Note that any particular pair \((\xi(x), \eta(x))\) for a given \( \Psi'_j \) is not unique. For if \((\xi, \eta)\) is any pair satisfying properties 1–4 above, then so does \((\xi_1, \eta_1)\), where \( \xi_1 \equiv a\xi, \eta_1 \equiv a\eta \pmod{D} \) and \((a, D) = 1 \). Also, if \( D \) is a prime \( p > 4 \), then \( (D, W_j) = (p, W_j) = 1 \). For if \( p \mid W_j \), we must have \( p \mid Q_{j-1} \) by (3.6) and (3.4), which, since \( 0 < Q_{j-1} < 2\sqrt{p} \) (Lemma 4.1), is impossible for \( p > 4 \). Finally, if \( D \) is a prime \( p \), then \( p \mid Z_1 \) if and only if \( \eta(x) = 0 \). For if \( \eta(x) = 0 \), then \( p \mid Z_1\xi(x) \). Since \( p \nmid \xi(x) \) by property 3, we must have \( p \mid Z_1 \). On the other hand, if \( p \mid Z_1 \), then \( p \mid W_j\eta(x) \) and we have already seen that \( p \nmid W_j \).

For \( b(x) \) above, we have \( b(2x) = \Psi'_m \), where
\[
\Psi'_m = \Psi_1(\Psi'_j)^2 / u
\]
Algorithm 6.1. Given $D$ when property 3. From these observations, we see that we may put $p$ which means that $Q$ when $G$, hence, we get

Putting $Q = (Q_0, D) = 1$ when $D = p$. Also, if $D = p$ and $\xi \equiv \eta \equiv 0 \pmod{p}$, then $p|\xi(x)|G$. If $p|G$, then $p|Q_{i+1}$ by (3.4) and (3.6). Since $p|Q_{i+1}$, we must have $p|Q_{i+1}$ when $p|Q_0$, which means that $p|G$. If $p|\xi(x)$, then $p|\eta$ implies that $p|\eta(x)$, which contradicts property 3. From these observations, we see that we may put

$$
\xi(2x) \equiv \xi(x)|G \pmod{p},
$$
$$
\eta(2x) \equiv 2\eta(x)|G + B\xi(x) \pmod{p},
$$

when $D = p$.

**Algorithm 6.1.** Given $D = p > 9$ and $x = kR_2$, compute $b(x)$, $\xi(x)$, $\eta(x)$.

1. Put $j = \lceil \log_2 x \rceil$, $y = x/2^j$ (0 ≤ $y < 1$).
2. Put $b(y) = b_1$, $\rho(y) = 2^6$, $\xi(y) = 1$, $\eta(y) = 0$.
3. for $m = 1$ to $j$
   
   Compute $(u) a_1 = b^2(y)$, $\rho_1 = u\rho(y)^2$.
   
   Put $a_1 = [Q_0/r, (P_0 + \sqrt{B})/r]$, $B_{-1} = 0$, $B_{-2} = 1$, $i = 1$,
   
   $a_0 = [(P_0 + \sqrt{B})/Q_0]$.
   
   while $\rho_i \geq 1$
   
   $P_i \equiv a_{i-1}Q_{i-1} - P_{i-1}, Q_i \equiv (p - P_i^2)/Q_{i-1}$
   
   $a_i \equiv [(P_i + \sqrt{B})]/Q_i$;
   
   $B_{i-1} \equiv a_{i-1}B_{i-2} + B_{i-3}$
   
   $\rho_{i+1} = \rho_i ([\sqrt{B} - P_i]/Q_i]$;
   
   $i \leftarrow i + 1$
   
end while
   
   $b(2y) = a_{i-1}$
   
   $\rho(2y) = \rho_{i-1}$
   
   $G \equiv P_{i+2}B_{i-3} + Q_{i-2}B_{i-4} \pmod{p}$
   
   $\xi(2y) \equiv \xi(y)|G \pmod{p}$
   
   $\eta(2y) \equiv 2\eta(y)|G + B_{i-3}\xi(y) \pmod{p}$
   
   $y \leftarrow 2y$
   
end for

4. $b(x) \leftarrow b(y)$, $\xi(x) \leftarrow \xi(y)$, $\eta(x) \leftarrow \eta(y)$

Note that if $x = kR_2$ where $k \in \mathbb{Z}^+$, then $b(x) = O_K$ and $L(b(x)) = 1$.

7. Implementation and computational results

The complete algorithm for testing the AAC conjecture was implemented in Fortran 77 and tested and run on an SGI O3 workstation and on one processor of an SGI Origin 2000 computer system at CWI in Amsterdam. Both of these machines support 64-bit arithmetic, which is particularly helpful in the third step.
of the overall algorithm (Algorithm 6.1). The program executes about four times
more quickly on the Origin 2000 than it does on the O2.

A basic step in the computations is the continued fraction evaluation,

\[ P_{i+1} = a_i Q_i - P_i, \]
\[ Q_{i+1} = (p - P_{i+1}^2)/Q_i, \]
\[ a_{i+1} = \lfloor (P_i + \sqrt{T})/Q_i \rfloor, \]

where it is known that \( Q_i \|(p - P_{i+1}^2). \) Special precautions were taken to guarantee
the correctness of this routine, taking into consideration that \( p \) can be as large as
\( 10^{11}, \) and using the relation \( Q_{i+1} = Q_{i-1} - a_i(P_{i+1} - P_i). \) Furthermore, we made
use of a computing trick of Head [10] to deal with integers that become as large as
\( p^2 \approx 10^{22} (= 2^{64}). \) This was very useful in the third phase of the procedure.

In view of the result of Lenstra [13], that computation of \( R \) can be done in
about \( p^{1/5} \) elementary operations, we put \( c = p^{1/5} \) in Algorithm 5.4. Since baby-steps
are much cheaper to compute than giant-steps, it was important to do some
experimentation to find the best value for \( T \) in the set \( S \) of Algorithm 5.4. To
this end, we introduced a parameter \( f \) and computed \( S \) until \( \zeta_t > fp^{1/5}. \) Since
\( p^{1/5} > 0.5 \log_2 p \) for \( p > 10^6, \) we have \( \zeta_t > c + B \) when \( f \geq 2. \) Usually we used
\( f = 3, \) but as \( p \) became larger, we occasionally used \( f = 10 \) and \( f = 20. \) We also
experimented with the value for \( T. \) We found that for values of \( p \) up to about
\( 6 \times 10^{10}, \) a value of \( T = 2000 \) worked reasonably well, but beyond that point we
used \( T = 5000. \) Thus our \( T, f \) pairs were usually (2000, 3) or (5000, 3), but when
we failed to find a value for \( kR_2 \) for a modest value of \( i \) such that \( \mathfrak{b}(ic) \) or \( \mathfrak{b}(ic) \in S, \)
we used a different parameter set. We usually bounded \( i \) in our program by 60.
When this failed to produce a value for \( kR_2, \) we tried \( T = 1000, f = 10, \) \( i \)-bound
= 200 or \( T = 2000, f = 20, \) \( i \)-bound = 500.

Of course, in running such a complex algorithm, it is essential to perform some
checks to ensure that the program is performing properly. We have already men-
tioned the simple check that our value for \( kR_2 \) be less than \( 8p, \) but we also always
checked that \( \mathfrak{b}(kR_2) = \mathfrak{b}_1 = [1, \omega] \) whenever we ran Algorithm 6.1. This was a
very useful confirmation that our value for \( kR_2 \) is correct. It was also a very cheap
check.

We less frequently carried out a more expensive check. From the continued fraction
evaluation of \( \left( 1 + \sqrt{T} \right)/2, \) we computed \( t, u \) modulo \( p \) and the value of \( R_2 \)
by using (3.10) and (3.11). (When \( D = p \equiv 1 \pmod{4}, \) we must always find some
\( s \) such that \( Q_s = Q_{s+1}. \) See, for example, Perron [17] pp. 106–108]. The actual
values of \( t \) and \( u \) can become enormous; for example, if \( p = 40,094,470,441, \) then
both \( t \) and \( u \) exceed \( 10^{330,000}. \) ) We next divided this value of \( R_2 \) into our computed
value of \( kR_2 \) to check that this is very close to an integer \( k. \) We then computed \( X_k \)
and \( Y_k \) modulo \( p \) by putting \( X_0 = 2, Y_0 = 0, X_1 \equiv t, Y_1 \equiv u \pmod{p} \) and using

\[ X_{n+1} = X_1 X_n + X_{n-1} \pmod{p}, \]
\[ X_{n+1} = X_1 Y_n + Y_{n-1} \pmod{p}. \]

We checked that the computed values for \( \xi(kR_2) \) and \( \eta(kR_2) \) satisfied

\[ \xi(kR_2)Y_k \equiv \eta(kR_2)X_k \pmod{p}. \]
As this check is very costly, we carried it out only for a small subset of the values \( p \) on which we ran our main program. This check was carried out successfully for every 100 000-th prime for which we verified the AAC conjecture.

In all our runs, we did not find a single counterexample of the conjecture; thus, we have confirmed the truth of the AAC conjecture for all primes between \( 10^9 \) and \( 10^{11} \). Computing times on the \( \Omega_2 \) and Origin 2000 were about 250 and 700 CPU hours, respectively. We used the \( \Omega_2 \) to search the range \( 10^9 \) to \( 10^9 \), and the Origin to search the range \( 9 \times 10^9 \) to \( 10^{11} \).

8. A Detailed Example

We will now illustrate how our algorithm works by using a nontrivial numerical example with \( p = 97,843,343,893 \). We put \( T = 1000 \) and obtain 
\[
S(T, p) = 1,475,146, 986,410, 691.
\]

We next put \( c = p^{1/5} = 157.7997, f = 10, \) and \( i \)-bound \( = 200 \). We find
\[
\alpha_1 = b(E) = \lfloor Q_0/2, (P_0 + \sqrt{p})/2 \rfloor
\]
with
\[
\begin{align*}
P_0 &= 295,721, & Q_0 &= 46,766, & \rho(E) &= 11.23627.
\end{align*}
\]

Furthermore, we compute \( \alpha_i \) for \( i = 2, 3, \ldots, 941 \) (\( \zeta_{941} \) is the first \( \zeta_i > f c = 1,577,9973 \)):

<table>
<thead>
<tr>
<th>( i )</th>
<th>( P_{i-1} )</th>
<th>( Q_{i-1} )</th>
<th>( \zeta_i )</th>
<th>( \rho_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>312,237</td>
<td>7,514</td>
<td>4</td>
<td>1.19714</td>
</tr>
<tr>
<td>3</td>
<td>311,425</td>
<td>114,162</td>
<td>11</td>
<td>1.84453</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>926</td>
<td>312,243</td>
<td>13,426</td>
<td>1,551</td>
<td>1.35561</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>939</td>
<td>81,187</td>
<td>23,294</td>
<td>1,576</td>
<td>1.30055</td>
</tr>
<tr>
<td>940</td>
<td>152,107</td>
<td>320,226</td>
<td>1,577</td>
<td>1.30525</td>
</tr>
<tr>
<td>941</td>
<td>168,119</td>
<td>217,282</td>
<td>1,578</td>
<td>1.73824</td>
</tr>
</tbody>
</table>

Next, we find \( b(c) = [92,354/2, (286,825 + \sqrt{p})/2] \) and compute at most 199 more ideals \( b(2c), b(3c), \ldots, b(200c) \) until \( b(ic) \) or \( \bar{b}(ic) \) is one of the previously determined \( \alpha_i \). We find

<table>
<thead>
<tr>
<th>( i )</th>
<th>( P'_{i-1} )</th>
<th>( Q'_{i-1} )</th>
<th>( \rho(ic) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>286,825</td>
<td>92,354</td>
<td>2.24634</td>
</tr>
<tr>
<td>2</td>
<td>282,267</td>
<td>97,594</td>
<td>4.45638</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>12</td>
<td>305,353</td>
<td>13,426</td>
<td>32.65983</td>
</tr>
</tbody>
</table>

and \( \bar{b}(12c) = \alpha_{926} \). That is \( Q_{i-1} = Q'_{j-1} \) and \( P_{i-1} \equiv -P'_{j-1} \) (mod \( Q_{i-1} \)) \( (i = 926, j = 12) \). We find, then, that
\[
kR_2 = E + ic + \zeta_j - \log_2 (L(b(ic))\rho(ic)\rho_j\rho(E)) = 989,833,617.
\]
Next, we compute \( b(kR_2) \), starting with \( b(y) = b_1 \), where \( y = kR_2/2^{20} = 0.9439789 \) and computing \( b(2y), b(4y), \ldots, b(2^{20}y) \). We find that \( b(kR_2) = [1, (312,799 + \sqrt{7})/2] = [1, \omega] \). Together with computing \( b(2^y) \) we also compute \( \xi(2^y) \) and \( \eta(2^y) \), finding

\[
\xi(kR_2) \equiv 73,973,607,135 \mod (p),
\eta(kR_2) \equiv 6,870,136,643 \mod (p).
\]

Since \( \eta(kR_2) \not\equiv 0 \mod (p) \), we have confirmed the AAC conjecture for \( p = 97 \).

To run our expensive check we compute the continued fraction expansion of \( \omega = (1 + \sqrt{7})/2 \) until two consecutive \( Q \) values are equal. We find that \( Q_s = Q_{s+1} \) for \( s = 96,929 \). We also find \( R_2 = 329,944,539 \), and on dividing this into \( kR_2 \) obtain \( k = 3,000,000,000 \). Furthermore, we get \( t \equiv 84,779,576,991, u \equiv 38,999,918,048 \mod (p) \). We then compute \( X_3 \equiv 13,063,766,902 \mod (p), Y_3 \equiv 78,686,933,642 \mod (p) \), and finally verify that

\[
\xi(kR_2)Y_3 \equiv \eta(kR_2)X_3 \mod (p).
\]

**References**


CENTRE FOR NUMBER THEORY RESEARCH, MACQUARIE UNIVERSITY, SYDNEY, NSW 2109, AUSTRALIA
E-mail address: alf@math.mq.edu.au

CWI, KRUISLAAN 413, 1098 SJ AMSTERDAM, THE NETHERLANDS
E-mail address: Herman.te.Riele@cwi.nl

DEPT. OF COMPUTER SCIENCE, UNIVERSITY OF MANITOBA, WINNIPEG, MANITOBA CANADA R3T 2N2
E-mail address: williams@cs.umanitoba.ca