Serdica
Mathematical Journal
Сердика
Математическо списание

For further information on
Serdica Mathematical Journal
which is the new series of
Serdica Bulgaricae Mathematicae Publicationes
visit the website of the journal http://www.math.bas.bg/~serdica
or contact: Editorial Office
Serdica Mathematical Journal
Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
Telephone: (+359-2)9792818, FAX:(+359-2)971-36-49
e-mail: serdica@math.bas.bg

Serdica Math. J. **46** (2020), 121–134

Serdica
Mathematical Journal

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

MODULO q GREATEST COMMON DIVISOR ALGORITHMS

Sándor Szabó

Communicated by V. Drensky

Abstract. In this paper we are looking for fast gcd algorithms in certain quadratic number fields. These algorithms do not belong to the Euclidean algorithm family rather the proposed algorithms can be viewed as generalization of the binary gcd algorithm.

1. Introduction. If d is a greatest common divisor of the integers a and b, then there are integers u and v such that $d = ua + vb$. There is a well-known binary gcd algorithm to compute d and it can be used to compute the coefficients u and v too. We want to extend this binary algorithm to q-ary algorithms replacing the prime 2 by a prime power q. Further we are interested in extending these algorithms to quadratic number fields. Since we want to work not only in the ring of integers but in the ring of algebraic integers of certain imaginary quadratic field we will describe the basic procedure in a more general setting in an integral domain R. We assume that the unique factorization property holds in R. We need a function $N : R \to \{0, 1, 2, \ldots\}$, that is, N is map from the ring R to the nonnegative integers. We assume that N has the following properties

2020 *Mathematics Subject Classification:* Primary 11A05; Secondary 11Y16.
Key words: binary gcd algorithm, extended gcd algorithm.

(i) $N(ab) = N(a)N(b)$,

(ii) $N(a+b) + N(a-b) = 2N(a) + 2N(b)$,

(iii) $N(a) = 0$ implies $a = 0$

for each $a, b \in R$. Let us choose a prime p in R and let $q = p^\alpha$, where α is a positive integer. Consider the ideal $I = \langle q \rangle$ and let U be a complete set of representatives modulo I. The numbers relatively prime to q among these form a group under multiplication modulo I. Let T be denote this group. Let us define the numbers $\alpha(r, s)$ by

$$\alpha(r,s)s \equiv r \pmod{I} \tag{1}$$

for each $r, s \in T$. If the $\alpha(r, s)$ numbers can be chosen such that

$$2[N(\alpha(r,s)) + 1] \leq N(p) \tag{2}$$

holds for each $r, s \in T$, then one can construct a gcd algorithm.

The algorithm is the following.

I. Forward phase to compute d.

Step 1. (Initial step) Set $a_1 = a$, $b_1 = b$. (These are the initial values of the procedure.)

Step 2. (Iterated step) We compute a_{k+1}, b_{k+1}, d_k, C_k from the available values a_k, b_k.

We distinguish four cases.

(1) If $N(a_k) < N(b_k)$, then set $a_{k+1} = b_k$, $b_{k+1} = a_k$, $d_k = 1$, $C_k = \text{swap}$.

(2) If $b_k = 0$ or a_k, b_k are associates, then a_k is a greatest common divisor of a_k and b_k. In this case we set $d_k = a_k$, $C_k = \text{back}$, $u_k = 1$, $v_k = 0$ and the forward phase of the algorithm terminates.

(3) If b_k is a unit, then b_k is a greatest common divisor of a_k and b_k. In this case we set $d_k = b_k$, $C_k = \text{back}$, $u_k = 0$, $v_k = 1$ and the forward phase of the algorithm terminates.

(4) If none of (1), (2), (3) holds then we compute $r_k, s_k \in U$ for which

$$a_k \equiv r_k \pmod{I}, \; b_k \equiv s_k \pmod{I}.$$

Set $C_k = (r_k, s_k)$ and distinguish four cases depicted in Table 1. Then we return to Step 2.

II. Backward phase to compute u and v.

Step 1. (Iterated step) We compute u_{k-1}, v_{k-1} from the available values u_k, v_k, C_{k-1}. We distinguish two cases.

Table 1. The forwards phase of the algorithm

	$s_k \notin T$	$s_k \in T$
$r_k \notin T$	$a_{k+1} = a_k/p$ $b_{k+1} = b_k/p$ $d_k = p$	$a_{k+1} = a_k/p$ $b_{k+1} = b_k$ $d_k = 1$
$r_k \in T$	$a_{k+1} = a_k$ $b_{k+1} = b_k/p$ $d_k = 1$	$a_{k+1} = a_k - \alpha(r_k, s_k)b_k$ $b_{k+1} = b_k$ $d_k = 1$

Table 2. The backward phase of the algorithm

	$s_{k-1} \notin T$	$r_{k-1} \in T$

$r_{k-1} \notin T$	$u_{k-1} = u_k$
	$v_{k-1} = v_k$
		If $t_u \notin T$ then
		$u_{k-1} = u_k/p$
		$v_{k-1} = v_k$
		If $t_u \in T$ then
		$u_{k-1} = [u_k - \alpha(t_u, s_k)b_k]/p$
		$v_{k-1} = v_k + \alpha(t_u, s_k)a_k$

$r_{k-1} \in T$	If $t_v \notin T$ then
	$u_{k-1} = u_k$
	$v_{k-1} = v_k/p$
	If $t_v \in T$ then
	$u_{k-1} = u_k + \alpha(t_v, r_k)b_k$
	$v_{k-1} = [v_k - \alpha(t_v, r_k)a_k]/p$
		$u_{k-1} = u_k$
		$v_{k-1} = v_k - \alpha(r_{k-1}, s_{k-1})u_k$

(1) If $C_{k-1} = \text{swap}$, then set $u_{k-1} = v_k$, $v_{k-1} = u_k$.
(2) If $C_{k-1} = (r_{k-1}, s_{k-1})$, then we compute $t_u, t_v \in U$ for which

$$t_u \equiv u_k \pmod{I},\ t_v \equiv v_k \pmod{I}.$$

We consider four cases given in Table 2. Then return to Step 1 until $k = 1$ and the backward phase terminates.

Theorem 1. (i) *The algorithm terminates in finitely many steps.*

(ii) *If the algorithm terminates with $k = n$, then $d = d_1 \cdots d_n$.*

(iii) *The equation $d = ua + bv$ holds with the $u = u_1$, $v = v_1$, choices.*

(iv) *The algorithm terminates in at most*

$$\frac{2\log_2[N(a)N(b)]}{\log_2 N(p)}$$

steps not counting the "swap" steps.

P r o o f. (i) Let h_k be defined to be $N(a_k)N(b_k)$. Clearly h_k is a non-negative integer. We will call h_k the height at step k. Note that if $C_k = \text{swap}$, then $h_k = h_{k+1}$. Suppose that $C_k = (r_k, s_k)$. An inspection shows that

$$\begin{array}{lll} r_k \notin T,\ s_k \notin T & \text{implies} & h_{k+1} = h_k/N(p^2), \\ r_k \notin T,\ s_k \in T & \text{implies} & h_{k+1} = h_k/N(p), \\ r_k \in T,\ s_k \notin T & \text{implies} & h_{k+1} = h_k/N(p), \\ r_k \in T,\ s_k \in T & \text{implies} & h_{k+3} < h_k. \end{array}$$

Let us check the $r_k \in T$, $s_k \in T$ case. Now $a_{k+1} = a_k - \alpha(r_k, s_k)b_k$, $b_{k+1} = b_k$. There are two possibilities to deal with depending on $N(a_{k+1}) < N(b_{k+1})$ or $N(a_{k+1}) \geq N(b_{k+1})$. Since the cases are fairly similar we check only the first one. Suppose that $N(a_{k+1}) < N(b_{k+1})$. Consequently $a_{k+2} = b_{k+1}$, $b_{k+2} = a_{k+1}$. After this step $N(a_{k+2}) > N(b_{k+2})$ and b_{k+2} is divisible by p. Therefore $C_{k+2} = (r_{k+2}, s_{k+2})$ with $r_{k+2} \in T$, $s_{k+2} \notin T$. So $a_{k+3} = a_{k+2}$, $b_{k+3} = b_{k+2}/p$. Note that a_k, b_k are not associates since otherwise the algorithm terminates. From $N(a+b)+N(a-b) = 2N(a)+2N(b)$ it follows that $N(a+b) = 2N(a)+2N(b)-N(a-b)$. Therefore $N(a+b) \leq 2N(a)+2N(b)$ and equation holds only when $a = b$. Using that $N(a_k) \geq N(b_k)$ and (2) it follows that

$$\begin{aligned} h_{k+3} &= N(a_{k+3})N(b_{k+3}) \\ &= N[(a_k - \alpha(r_k, s_k)b_k)/p]N(b_k) \\ &= N[(a_k - \alpha(r_k, s_k)b_k)]N(b_k)/N(p) \\ &\leq [2N(a_k) + 2N(-\alpha(r_k, s_k)b_k)]N(b_k)/N(p) \\ &= [2N(a_k) + 2N(-\alpha(r_k, s_k))N(b_k)]N(b_k)/N(p) \\ &= [2N(a_k) + 2N(\alpha(r_k, s_k))N(b_k)]N(b_k)/N(p) \\ &\leq [2N(a_k) + 2N(\alpha(r_k, s_k))N(a_k)]N(b_k)/N(p) \\ &= 2[N(\alpha(r_k, s_k)) + 1]N(a_k)N(b_k)/N(p) \\ &\leq N(a_k)N(b_k) \\ &= h_k. \end{aligned}$$

Equation can hold only if $a_k = \alpha(r_k, s_k)b_k$. If $\alpha(r_k, s_k)$ is a unit then a_k and b_k are associates and the algorithm terminates. If $\alpha(r_k, s_k)$ is not a unit, then $N(a_k) > N(b_k)$ and so $h_{k+3} < h_k$.

(ii) The equation $d_n = \gcd(a_n, b_n)$ clearly holds. We assume that

$$d_k \cdots d_n = \gcd(a_k, b_k) \tag{3}$$

holds and by inspecting the cases we show that

$$d_{k-1} \cdots d_n = \gcd(a_{k-1}, b_{k-1}) \tag{4}$$

holds too.

If $C_{k-1} = \text{swap}$, then $a_k = b_{k-1}$, $b_k = a_{k-1}$, $d_{k-1} = 1$. Clearly $\gcd(a_k, b_k) = \gcd(a_{k-1}, b_{k-1})$ and (4) follows.

If $r_{k-1} \notin T$, $s_{k-1} \notin T$, then from Table 1, $a_k = a_{k-1}/p$, $b_k = b_{k-1}/p$, $d_{k-1} = p$. Hence

$$\begin{aligned} \gcd(a_{k-1}, b_{k-1}) &= p \gcd(a_k, b_k) \\ &= pd_k \cdots d_n \\ &= d_{k-1}d_k \cdots d_n \end{aligned}$$

as required.

If $r_{k-1} \notin T$, $s_{k-1} \in T$, then $a_k = a_{k-1}/p$, $b_k = b_{k-1}$, $d_{k-1} = 1$. Let g_k be a greatest common divisor of a_k, b_k and let g_{k-1} be a greatest common divisor of a_{k-1}, b_{k-1}. Now

$$g_k \mid a_k,\ g_k \mid b_k,\ g_{k-1} \mid \underbrace{a_{k-1}}_{pa_k},\ g_{k-1} \mid \underbrace{b_{k-1}}_{b_k}.$$

As $s_{k-1} \in T$, b_k is relatively prime to p and so g_{k-1} is prime to p. Then from $g_{k-1} \mid pa_k$ it follows that $g_{k-1} \mid a_k$. Now $g_{k-1} \mid a_k$, $g_{k-1} \mid b_k$ hold and implies that $g_{k-1} \mid g_k$. On the other hand from $g_k \mid b_k$ we get $g_k \mid b_{k-1}$ and from $g_k \mid a_k$ we get $g_k \mid pa_k = a_{k-1}$. Therefore $g_k \mid g_{k-1}$. Consequently

$$\begin{aligned} \gcd(a_{k-1}, b_{k-1}) &= \gcd(a_k, b_k) \\ &= d_k \cdots d_n \\ &= d_{k-1}d_k \cdots d_n \end{aligned}$$

as $d_{k-1} = 1$. The $r_{k-1} \in T$, $s_{k-1} \notin T$ case can be settled in a similar manner.

If $r_{k-1} \in T$, $s_{k-1} \in T$, then $a_k = a_{k-1} - \alpha(r_{k-1}, s_{k-1})b_{k-1}$, $b_k = b_{k-1}$, $d_{k-1} = 1$. Let g_k be a greatest common divisor of a_k, b_k and let g_{k-1} be a greatest common divisor of a_{k-1}, b_{k-1}. From $g_{k-1} \mid b_{k-1}$ it follows that $g_{k-1} \mid b_k$ and from $g_{k-1} \mid a_{k-1}$, $g_{k-1} \mid b_{k-1}$ it follows that

$$g_{k-1} \mid \underbrace{a_{k-1} - \alpha(s_{k-1}, r_{k-1})b_{k-1}}_{a_k}.$$

Hence $g_{k-1} \mid b_k$, $g_{k-1} \mid a_k$ implies $g_{k-1} \mid g_k$. Conversely from $g_k \mid b_k$ we get $g_k \mid b_{k-1}$. From $g_k \mid a_k$, $g_k \mid b_{k-1}$ it follows that

$$g_k \mid \underbrace{a_k + \alpha(r_{k-1}, s_{k-1})b_{k-1}}_{a_{k-1}}.$$

Therefore $g_k \mid a_{k-1}$, $g_k \mid b_{k-1}$ implies $g_k \mid g_{k-1}$. Finally

$$\begin{aligned}\gcd(a_{k-1}, b_{k-1}) &= \gcd(a_k, b_k)\\ &= d_k \cdots d_n\\ &= d_{k-1} d_k \cdots d_n\end{aligned}$$

as $d_{k-1} = 1$.

(iii) Suppose the forward phase is terminated with $k = n$. If $d_n = a_n$, then as $u_n = 1$, $v_n = 0$, the equation $d_n = u_n a_n + v_n b_n$ plainly holds. The $d_n = b_n$ case is similar. We assume that

$$d_k \cdots d_n = u_k a_k + v_k b_k \tag{5}$$

holds and by inspecting all the arising cases we verify that

$$d_{k-1} \cdots d_n = u_{k-1} a_{k-1} + v_{k-1} b_{k-1} \tag{6}$$

also holds.

If $C_{k-1} = \text{swap}$, then $a_k = b_{k-1}$, $b_k = a_{k-1}$, $u_{k-1} = v_k$, $v_{k-1} = u_k$, $d_{k-1} = 1$. Substituting these into (5) we get (6). Assume that $C_{k-1} = (r_{k-1}, s_{k-1})$ and distinguish four cases.

If $r_{k-1} \notin T$, $s_{k-1} \notin T$, then from Table 1, $a_k = a_{k-1}/p$, $b_k = b_{k-1}/p$, $d_{k-1} = p$ and from Table 2, $u_k = u_{k-1}$, $v_k = v_{k-1}$. Plugging these into (5) we get

$$d_k \cdots d_n = u_{k-1}(a_{k-1}/p) + v_{k-1}(b_{k-1}/p).$$

Multiplying both sides by p gives

$$p d_k \cdots d_n = u_{k-1} a_{k-1} + v_{k-1} b_{k-1}.$$

As $d_{k-1} = p$ we get (6).

Suppose that $r_{k-1} \notin T$, $s_{k-1} \in T$, then $a_k = a_{k-1}/p$, $b_k = b_{k-1}$. If $t_u \notin T$, then $u_{k-1} = u_k/p$, $v_{k-1} = v_k$. Substituting these into (5) we get (6). If $t_u \in T$, then

$$u_{k-1} = [u_k - \alpha(t_u, s_k) b_k]/p, \;\; v_{k-1} = v_k + \alpha(t_u, s_k) a_k.$$

Plugging these in to (5) we get

$$\begin{aligned}& d_k \cdots d_n\\ = \;& [p u_{k-1} + \alpha(t_u, s_k) b_k](a_{k-1}/p) + [v_{k-1} - \alpha(t_u, s_k) a_k] b_{k-1}\\ = \;& (p u_{k-1})(a_{k-1}/p) + \alpha(t_u, s_k) b_k \underbrace{(a_{k-1}/p)}_{a_k} + v_{k-1} b_{k-1} - \alpha(t_u, s_k) a_k \underbrace{b_{k-1}}_{b_k}\end{aligned}$$

$$= \quad u_{k-1}a_{k-1} + v_{k-1}b_{k-1}.$$

As $d_{k-1} = 1$, (6) follows. The $r_{k-1} \in T$, $s_{k-1} \notin T$ case can be settled in an analogous way.

If $r_{k-1} \in T$, $s_{k-1} \in T$, then

$$\begin{aligned} a_k &= a_{k-1} - \alpha(r_{k-1}, s_{k-1})b_{k-1}, \\ b_k &= b_{k-1}, \\ u_{k-1} &= u_k, \\ v_{k-1} &= v_k - \alpha(r_{k-1}, s_{k-1})u_k. \end{aligned}$$

A routine computation shows that (5) implies (6).

(iv) Let us watch the algorithm at a step when $C_k = (r_k, s_k)$. From the proof of (i) we can read of that $h_{k+1} = h_k/N(p^2)$ or $h_{k+1} = h_k/N(p)$ unless $r_k \in T$, $s_k \in T$. In order to simplify the situation we assume that $h_{k+1} = h_k/N(p)$ in these cases and $h_{k+1} = h_k$ when $r_k \in T$, $s_k \in T$. Note that if $r_k \in T$, $s_k \in T$, then after a "swap" step we end up at a step where the height is divided by $N(p)$ or the forward phase of the algorithm terminates. In short at least at every other (counted) steps the height is divided by $N(p)$. Suppose that the algorithm terminates in z steps not counting the "swap" steps. Then $[N(p)]^{z/2} \leq N(a)N(b)$. From this the claim follows. □

2. The ring of integers. One can check that when the ring R is the ring of integers, then we may choose $N(a)$ to be a^2. In the $p = 2$, $q = 2$, $\alpha = 1$ special case $U = \{0, 1\}$ is a complete set of residues and $T = \{1\}$ is a reduced residue system modulo 2. With the $\alpha(1, 1) = 1$ choice we get a gcd algorithm. Tables 1 and 2 are reduce to Tables 3 and 4 respectively.

This is the well-known binary gcd algorithm. (See J. Stein [4].) It appears as Algorithm B vol. 2 Sec. 4.5.2 in D. E Knuth [3]. Because of the backward phase the binary gcd algorithm can be used to compute u and v.

Table 3. The forward phase of the modulo 2 algorithm

	$b_k \equiv 0$	$b_k \equiv 1$
$a_k \equiv 0$	$a_{k+1} = a_k/2$ $b_{k+1} = b_k/2$ $d_k = 2$	$a_{k+1} = a_k/2$ $b_{k+1} = b_k$ $d_k = 1$
$a_k \equiv 1$	$a_{k+1} = a_k$ $b_{k+1} = b_k/2$ $d_k = 1$	$a_{k+1} = b_k$ $b_{k+1} = a_k - b_k$ $d_k = 1$

Table 4. The backward phase of the modulo 2 algorithm

	$b_{k-1} \equiv 0$	$b_{k-1} \equiv 1$

$a_{k-1} \equiv 0$	$u_{k-1} = u_k$
	$v_{k-1} = v_k$
		If $u_k \equiv 0$ then
		$u_{k-1} = u_k/2$
		$v_{k-1} = v_k$
		If $u_k \equiv 1$ then
		$u_{k-1} = (u_k - b_k)/2$
		$v_{k-1} = v_k + a_k$

$a_{k-1} \equiv 1$	If $v_k \equiv 0$ then
	$u_{k-1} = u_k$
	$v_{k-1} = v_k/2$
	If $v_k \equiv 1$ then
	$u_{k-1} = u_k + b_k$
	$v_{k-1} = (v_k - a_k)/2$
		$u_{k-1} = v_k$
		$v_{k-1} = u_k - v_k$

Table 5. A modulo 2 example with $a = 1000$, $b = 133$

k	a_k	b_k	d_k	C_k	u_k	v_k
1	1000	133	1	$(0,1)$	-106	797
2	500	133	1	$(0,1)$	-79	297
3	250	133	1	$(0,1)$	-25	47
4	125	133	1	swap	83	-78
5	133	125	1	$(1,1)$	-78	83
6	8	125	1	swap	-78	5
7	125	8	1	$(1,0)$	5	-78
8	125	4	1	$(1,0)$	1	-31
9	125	2	1	$(1,0)$	1	-62
10	125	1	1	back	0	1

We illustrated the algorithm with the instance when $a = 1000$, $b = 133$. Table 5 contains the details.

Interestingly the algorithm does not contain multiplications or divisions except multiplying and dividing by 2. One can carry out the computations being able to test if a given integer is even or odd and able to compare magnitudes of integers.

In the $p = 3$, $q = 3$, $\alpha = 1$ case $U = \{0, 1, 2\}$, $U' = \{-1, 0, 1\}$ are complete sets of representatives and $T = \{1, 2\}$, $T' = \{-1, 1\}$ are reduced systems of representatives modulo 3. Set

$$\begin{aligned} \alpha(1,1) &= 1, & \alpha(1,2) &= -1, \\ \alpha(2,1) &= -1, & \alpha(2,2) &= 1 \end{aligned}$$

Now $N(\alpha(r,s)) = 1$. Consequently $4 = 2[N(\alpha(r,s)) + 1] \leq N(3) = 9$ and (2) holds. By Theorem 1 there is a modulo 3 gcd algorithm in Z. (People whose computer uses the base 3 number system will be particularly pleased.) We replace Tables 1 and 2 by Tables 6 and 7 to get the details of the algorithm.

Table 6. The forward phase of the modulo 3 algorithm

	$b_k \equiv 0$	$b_k \equiv 1$	$b_k \equiv 2$
$a_k \equiv 0$	$a_{k+1} = a_k/3$ $b_{k+1} = b_k/3$ $d_k = 3$	$a_{k+1} = a_k/3$ $b_{k+1} = b_k$ $d_k = 1$	$a_{k+1} = a_k/3$ $b_{k+1} = b_k$ $d_k = 1$
$a_k \equiv 1$	$a_{k+1} = a_k$ $b_{k+1} = b_k/3$ $d_k = 1$	$a_{k+1} = a_k - b_k$ $b_{k+1} = b_k$ $d_k = 1$	$a_{k+1} = a_k + b_k$ $b_{k+1} = b_k$ $d_k = 1$
$a_k \equiv 2$	$a_{k+1} = a_k$ $b_{k+1} = b_k/3$ $d_k = 1$	$a_{k+1} = a_k + b_k$ $b_{k+1} = b_k$ $d_k = 1$	$a_{k+1} = a_k - b_k$ $b_{k+1} = b_k$ $d_k = 1$

Table 8 depicts a test run with the $a = 1000$, $b = 133$ choices.

It should be clear by now that for each prime p, there is a gcd algorithm. We state this more formally as a theorem.

Theorem 2. *For each prime p there is a modulo p gcd algorithm in Z.*

P r o o f. For $p \leq 3$ the theorem has already been proved and so we assume that $p \geq 5$. Clearly

$$\begin{aligned} U &= \{0, 1, \ldots, p-1\}, \\ U' &= \{-(p-1)/2, \ldots, (p-1)/2\} \end{aligned}$$

are complete sets of representatives modulo p. Further $T = U \setminus \{0\}$, $T' = U' \setminus \{0\}$ are reduced residue systems modulo p. For each $r, s \in T$ there is an $\alpha(r,s) \in T'$ satisfying (1). Note that $N(\alpha(r,s)) \leq [(p-1)/2]^2$ and so

$$\begin{aligned} p^2 &= N(p) \\ &\geq 2[N(\alpha(r,s)) + 1] \\ &\geq 2[(p-1)/2]^2 + 2 \\ &= (p-1)^2/2 + 2. \end{aligned}$$

So (2) holds and by Theorem 1 there is a modulo p gcd algorithm in Z. □

When $p = 2$, $q = 4$, $\alpha = 2$, then $U = \{0, 1, 2, 3\}$ is a complete set of representatives and $T = \{1, 3\}$, $T' = \{-1, 1\}$ are reduced residue systems modulo

Table 7. The backward phase of the modulo 3 algorithm

	$b_{k-1} \equiv 0$	$b_{k-1} \equiv 1$	$b_{k-1} \equiv 2$

$a_{k-1} \equiv 0$	$u_{k-1} = u_k$
	$v_{k-1} = v_k$
		If $u_k \equiv 0$ then
		$u_{k-1} = u_k/3$
		$v_{k-1} = v_k$
		If $u_k \equiv 1$ then
		$u_{k-1} = (u_k - b_k)/3$
		$v_{k-1} = v_k + a_k$
		If $u_k \equiv 2$ then
		$u_{k-1} = (u_k + b_k)/3$
		$v_{k-1} = v_k - a_k$
			If $u_k \equiv 0$ then
			$u_{k-1} = u_k/3$
			$v_{k-1} = v_k$
			If $u_k \equiv 1$ then
			$u_{k-1} = (u_k + b_k)/3$
			$v_{k-1} = v_k - a_k$
			If $u_k \equiv 2$ then
			$u_{k-1} = (u_k - b_k)/3$
			$v_{k-1} = v_k + a_k$

$a_{k-1} \equiv 1$	If $v_k \equiv 0$ then
	$u_{k-1} = u_k$
	$v_{k-1} = v_k/3$
	If $v_k \equiv 1$ then
	$u_{k-1} = u_k + b_k$
	$v_{k-1} = (v_k - a_k)/3$
	If $v_k \equiv 2$ then
	$u_{k-1} = u_k - b_k$
	$v_{k-1} = (v_k + a_k)/3$
		$u_{k-1} = u_k$
		$v_{k-1} = v_k - u_k$
			$u_{k-1} = u_k$
			$v_{k-1} = v_k + u_k$

$a_{k-1} \equiv 2$	If $v_k \equiv 0$ then
	$u_{k-1} = u_k$
	$v_{k-1} = v_k/3$
	If $v_k \equiv 1$ then
	$u_{k-1} = u_k - b_k$
	$v_{k-1} = (v_k + a_k)/3$
	If $v_k \equiv 2$ then
	$u_{k-1} = u_k + b_k$
	$v_{k-1} = (v_k - a_k)/3$
		$u_{k-1} = u_k$
		$v_{k-1} = v_k + u_k$
			$u_{k-1} = u_k$
			$v_{k-1} = v_k - u_k$

4. With the

$$\begin{aligned} \alpha(1,1) &= 1, & \alpha(1,3) &= -1, \\ \alpha(3,1) &= -1, & \alpha(3,3) &= 1 \end{aligned}$$

choices we get a gcd algorithm. So the prime power case of the construction is not vacuous.

3. The Gaussian integers. The ring of algebraic integers of the field $Q(i)$ is $Z[i]$ and is called the ring of Gaussian integers. The function N here can be taken to be the norm, that is if $a = a_1 + a_2 i$, then $N(a) = a_1^2 + a_2^2$ since it has the required properties (i), (ii) and (iii). The factorization $2 = (1+i)(1-i)$ shows that the number 2 is not irreducible in $Z[i]$. As $2i = (1+i)(1+i)$ is,

Table 8. A modulo 3 example with $a = 1000$, $b = 133$

k	a_k	b_k	d_k	C_k	u_k	v_k
1	1000	133	1	$(1,1)$	27	-203
2	867	133	1	$(0,1)$	27	-176
3	289	133	1	$(1,1)$	-52	113
4	156	133	1	$(0,1)$	-52	61
5	52	133	1	swap	-23	9
6	133	52	1	$(1,1)$	9	-23
7	81	52	1	$(0,1)$	9	-14
8	27	52	1	swap	-25	13
9	52	27	1	$(1,0)$	13	-25
10	52	9	1	$(1,0)$	4	-23
11	52	3	1	$(1,0)$	1	-17
12	52	1	1	back	0	1

the prime $1 + i$ is going to play the role of p and $2i$ is going to play the role of q. Consider the ideal $I = \langle 2 \rangle$. The elements of I are linear combinations of 2 and $2i$ with coefficients from Z. In other words 2 and $2i$ form an integer basis for I. Therefore $U = \{0, 1, i, 1 + i\}$ is complete set of representatives modulo I and $T = \{1, i\}$ is a reduced residue system modulo I. Let us choose the numbers $\alpha(r, s)$ to be

$$\begin{array}{rclcrcl} \alpha(1,1) & = & 1, & & \alpha(1,i) & = & -i, \\ \alpha(i,1) & = & i, & & \alpha(i,i) & = & 1. \end{array}$$

Here (2) holds for each $r, s \in T$ and by Theorem 1 there is a gcd algorithm in $Z[i]$. This algorithm is known. (See A. Weilert [5].)

The 3 is a prime in $Z[i]$. Consider the ideal $I = \langle 3 \rangle$. As 3 and $3i$ form an integer basis for I, it follows that

$$\begin{array}{rcl} U & = & \{0, 1, 2, i, 1 + i, 2 + i, 2i, 1 + 2i, 2 + 2i\}, \\ U' & = & \{-1, 0, 1, -1 + i, i, 1 + i, -1 - i, -i, 1 - i\} \end{array}$$

are complete sets of representatives modulo I and $T = U \setminus \{0\}$, $T' = U' \setminus \{0\}$ are reduced residue systems modulo I. It is a straightforward matter to define the $\alpha(r, s)$ numbers for which $\alpha(r, s) \in T'$ and satisfies (1). Using $N(\alpha(r, s)) \leq 2$ one can check that $6 = 2[N(\alpha(r, s)) + 1] \leq N(3) = 9$ holds for each $r, s \in T$ and so there is a modulo I gcd algorithm in $Z[i]$.

Theorem 3. *If $p \in Z$ is a prime in Z such that $p \equiv 3 \pmod 4$, then p is prime in $Z[i]$ and there is a modulo p gcd algorithm in $Z[i]$.*

P r o o f. It is a well-known fact that if $p \in Z$ is a prime in Z such that $p \equiv 3 \pmod 4$, then p is prime in $Z[i]$. The elements p, pi form an integer basis for the ideal $I = \langle p \rangle$. So

$$\begin{aligned} U &= \{kp + lpi : 0 \le k, l \le p-1\}, \\ U' &= \{kp + lpi : -(p-1)/2 \le k, l \le (p-1)/2\} \end{aligned}$$

are complete sets of representatives modulo I and $T = U \setminus \{0\}$, $T' = U' \setminus \{0\}$ are reduced residue systems modulo I. For each $r, s \in T$, there is an $\alpha(r,s) \in T'$ satisfying (1). Now

$$\begin{aligned} N(\alpha(r,s)) &\le [(p-1)/2]^2 + [(p-1)/2]^2 \\ &= (p-1)^2/2. \end{aligned}$$

It follows that

$$\begin{aligned} p^2 &= N(p) \\ &\ge 2[N(\alpha(r,s)) + 1] \\ &= (p-1)^2 + 2. \end{aligned}$$

Therefore (2) holds and by Theorem 1 there is a modulo p gcd algorithm in $Z[i]$. □

The factorization $5 = (2+i)(2-i)$ shows that 5 is not a prime in $Z[i]$. But $2+i$ is a prime in $Z[i]$ and we will choose it to be p. Consider the ideal $I = \langle 2+i \rangle$. Note that 5, $2+i$ is an integer basis for I and consequently $U = \{0,1,2,3,4\}$ is a complete set of representatives modulo I and $T = \{1,2,3,4\}$ is a reduced residue system modulo I. A routine computation shows that $1i, 2i, 3i, 4i$ are congruent to $3, 1, 4, 2$ modulo I, respectively. It follows that $1, 2, 3, 4$ are congruent to i^0, i^3, i^1, i^2 modulo I, respectively. Therefore $T' = \{1, i, i^2, i^3\}$ is a reduced residue system modulo I. We may choose the $\alpha(r,s) \in T'$, $r, s \in T$ numbers as given in Table 9. As $4 = 2[N(\alpha(r,s)) + 1] \le N(p) = 5$ there is a further gcd algorithm in $Z[i]$.

Table 9. The $\alpha(r,s)$ numbers in the $p = 2 + i$ case in $Z[i]$

	1	2	3	4
1	1	i^1	i^3	i^2
2	i^3	1	i^2	i^1
3	i^1	i^2	1	i^3
4	i^2	i	i^1	1

4. The Eisenstein integers. The ring of the algebraic integers of the field $Q(\sqrt{-3})$ is equal to $Z[\vartheta]$, where $\vartheta = (1+\sqrt{-3})/2$. The function N is taken to be the norm too.

The rational integer 2 is a prime in $Z[\vartheta]$ and we choose it to be p. Consider the ideal $I = \langle 2\rangle$. Here 2, 2ϑ is an integer basis for I and so $U = \{0, 1, \vartheta, 1+\vartheta\}$ is a complete set of representatives modulo I and $T = \{1, \vartheta, 1+\vartheta\}$ is a reduced residue system modulo I. Note that $(1+\vartheta)\vartheta \equiv 1 \pmod{I}$ from which it follows that $1+\vartheta \equiv \vartheta^5 \pmod{I}$. We can see that $T' = \{1, \vartheta, \vartheta^5\}$ is a reduced residue system modulo I. Using this we choose $\alpha(r,s) \in T'$, $r, s \in T$ as depicted in Table 10 As each $\alpha(r,s)$ is a unit in $Z[\vartheta]$ we have $4 = 2[N(\alpha(r,s)) + 1] \leq N(p) = 4$, that is we are provided with a gcd algorithm in $Z[\vartheta]$. A number of possible applications of a gcd algorithm in $Z[\vartheta]$ is discussed in [1].

Table 10. The $\alpha(r,s)$ numbers in the $p = 2$ case in $Z[\vartheta]$

	1	ϑ	$1+\vartheta$
1	1	ϑ^5	ϑ
ϑ	ϑ	1	ϑ^2
$1+\vartheta$	ϑ^5	ϑ^4	1

The equation $7 = (2+\vartheta)(2+\overline{\vartheta})$ shows that 7 is not a prime in $Z[\vartheta]$. We choose p to be $2+\vartheta$, α to be 1. Let us consider the ideal $I = \langle 2+\vartheta\rangle$ for which 7, $2+\vartheta$ is an integer basis. Now $U = \{0, 1, 2, 3, 4, 5, 6\}$ is a complete set of representatives modulo I and $T = \{1, 2, 3, 4, 5, 6\}$ is a reduced residue system modulo I. One can verify that $1\vartheta, 2\vartheta, 3\vartheta, 4\vartheta, 5\vartheta, 6\vartheta$ are congruent to $5, 3, 1, 6, 4, 2$ modulo I respectively. From this it follows that $\vartheta, \vartheta^2, \vartheta^3, \vartheta^4, \vartheta^5, \vartheta^6$ are congruent to $5, 4, 6, 2, 3, 1$ respectively. Thus $T' = \{\vartheta, \vartheta^2, \vartheta^3, \vartheta^4, \vartheta^5, \vartheta^6\}$ is a reduced residue system modulo I. Consequently the numbers $\alpha(r,s) \in T'$, $r, s \in T$ can be chosen such that all of them are units. We have $4 = 2[N(\alpha(r,s)) + 1] \leq N(p) = 7$ and so there is a gcd algorithm in $Z[\vartheta]$.

Theorem 4. *Let $p \in Z$ be an odd prime in Z such that $p \equiv 2 \pmod{3}$. Then p is a prime in $Z[\vartheta]$ and there is a modulo p gcd algorithm in $Z[\vartheta]$.*

P r o o f. It is a well-known fact that if $p \in Z$ is a prime in Z such that $p \equiv 2 \pmod{3}$, then p is a prime in $Z[\vartheta]$. The ideal $I = \langle p\rangle$ has the integer basis p, $p\vartheta$. So

$$\begin{aligned} U &= \{kp + lp\vartheta : 0 \leq k, l \leq p-1\}, \\ U' &= \{kp + lp\vartheta : -(p-1)/2 \leq k, l \leq (p-1)/2\} \end{aligned}$$

are complete sets of representatives modulo I. Plotting the elements of U' on the complex plane we can see that they are in a parallelogram. Both sides of

the parallelogram have length p. There is a complete residue system U'' whose elements are in a rectangle. One side of the rectangle is of length p and the length of the other one is $(\sqrt{3}/2)p$. Of course $T = U \setminus \{0\}$, $T'' = U'' \setminus \{0\}$ are reduced residue systems modulo I. For each $r, s \in T$ there is an $\alpha(r, s) \in T''$ for which (1) holds. Now

$$\begin{aligned} N(\alpha(r,s)) &\leq [(p-1)/2]^2 + [(p-1)\sqrt{3}/4]^2 \\ &= (7/16)(p-1)^2. \end{aligned}$$

It follows that

$$\begin{aligned} p^2 &= N(p) \\ &\geq 2[N(\alpha(r,s)) + 1] \\ &= (7/8)(p-1)^2 + 2. \end{aligned}$$

Thus (2) holds and by Theorem 1 there is a modulo p gcd algorithm in $Z[\vartheta]$. □

REFERENCES

[1] I. B. Damgård, G. S. Frandsen. Efficient algorithm for gcd and cubic residuosity in the ring of Eisenstein integers. Fundamentals of computation theory, 109–117, Lecture Notes in Comput. Sci. vol. **2751**. Berlin, Springer, 2003.

[2] D. E. Knuth. An imaginary number system. *Comm. ACM* **3** (1960), 245–247.

[3] D. E. Knuth. The Art of Computer Programming, vol. 2, Seminumerical Algorithms, 3rd ed.. Upper Saddle River, NJ, Addison-Wesley, 2001.

[4] J. Stein. Computational problems associated with Racah algebra, *Journal of Comput. Phys.* **1**, *3* (1967), 397–405, DOI: 10.1016/0021-9991(67)90047-2.

[5] A. Weilert. $(1 + i)$-ary GCD computation in $Z[i]$ as an analogue to the binary GCD algorithm, *J. Symbolic Comput.* **30**, *5* (2000), 605–617.

Institute of Mathematics and Informatics
University of Pécs
Ifjúság u. 6
7624 Pécs, Hungary
e-mail: sszabo7@hotmail.com

Received June 24, 2020