An Efficient Barrett Reduction Algorithm for Gaussian Integer Moduli

Malek Safieh, Andreas Furch, Fabrizio De Santis
Siemens AG, Technology, Munich, Germany
{malek.safieh, andreas.furch, fabrizio.desantis}@siemens.com

***Abstract*—Gaussian integers are a subset of complex numbers that have integer numbers in both their real and imaginary parts. Similar to ordinary integer numbers, they can be equipped with modulo operations, which creates Gaussian integer rings and fields. Depending on the chosen modulus, these structures can be isomorphic to corresponding algebraic structures over integer numbers. However, computing modulo reduction for Gaussian integers can be computationally expensive, especially when the modulus itself is a Gaussian integer.**

In this work, we present a novel and efficient reduction algorithm for Gaussian integer moduli of arbitrary form based on the ideas of Barrett reduction for integer numbers. We show that the computational complexity of our proposed reduction algorithm is equivalent to previously known Montgomery reduction over Gaussian integers. However, unlike Montgomery's approach, our algorithm does not require domain transformations and can be more advantageous in various circumstances.

***Keywords*-Gaussian Integers, Modular Arithmetic, Modulo Reduction, Barrett Reduction.**

I. Introduction

The set of Gaussian integers, denoted by $\mathbb{Z}[i]$, is a subset of complex numbers that comprises all elements of the form $a + b\mathrm{i}$, where a and b are integers and i is the imaginary unit, i.e., $\mathrm{i} = \sqrt{-1}$. In other words, $\mathbb{Z}[\mathrm{i}]$ is defined as the set of all $a + b\mathrm{i} \in \mathbb{C} \mid a, b \in \mathbb{Z}$. The arithmetic properties of Gaussian integers are analogous to those of integer numbers, enabling the construction of finite rings and fields through Gaussian integer modulo reduction. Depending on the chosen Gaussian integer modulus, a finite Gaussian integer ring or Gaussian integer field can be constructed.

Gaussian integers find applications in the field of algebraic coding, as discussed in [8], while more recent proposals can be found in [6], [13], [16]. In the field of cryptography, Gaussian integers have been explored to create variants of RSA cryptosystem [4], [10], [11], [12], Rabin cryptosystem [1], and McEliece cryptosystem [7]. Fast arithmetic over Gaussian integers has been proposed in [18], [19] to speed up scalar multiplication on some types of elliptic curves.

While basic arithmetic operations on Gaussian integers can be implemented efficiently, Gaussian integer modulo reduction is typically rather inefficient when implemented straightforwardly. This is because it involves a Gaussian integer division, which in turn requires two integer divisions and rounding operations on the real and imaginary parts of a Gaussian rational number. As a result, more efficient reduction mechanisms for Gaussian integers are required to enable fast computations for Gaussian integer rings or fields.

In recent years, fast reduction algorithms for general Gaussian integers based on Montgomery modular arithmetic have been proposed, cf. [11], [12], [18], [19]. Furthermore, efficient reduction algorithms for Gaussian integer moduli of special forms have been presented in [20].

In general, efficient reduction algorithms employ two steps. The first step involves calculating a congruent solution using elementary arithmetic operations such as additions, subtractions, and multiplications, as well as digit operations like shifts and truncations. In the second step, a final reduction is performed to obtain the fully reduced result. While the final reduction step for integer numbers is typically straightforward and involves integer subtractions only, it is not as simple for Gaussian integers. This is because it requires the computation and comparison of multiple norm values, as shown in [18].

In this work, we apply the ideas of Barrett reduction for integer numbers to derive an efficient modulo reduction algorithm for Gaussian integer moduli of arbitrary form. In particular, we adapt Barrett's modulo reduction from [3], [5], [9], and take the advantage of the final reduction for Gaussian integers from [18], in order to derive a modulo reduction algorithm suitable for arbitrary Gaussian integer moduli. The proposed reduction algorithm is an alternative to the Montgomery reduction algorithm presented in [19] with equivalent computational complexity. However, unlike Montgomery's approach, our algorithm does not necessitate domain transformations to obtain the final reduced result. This allows for saving the costs of one multiplication and a complete Montgomery reduction for the transformation into the Montgomery domain, as well as the costs for one extra Montgomery reduction for the backward transformation from the Montgomery domain. Consequently, compared with the Montgomery reduction, the computational efficiency can be significantly increased using the proposed reduction, especially when only a few intermediate results have to be calculated until the final result is required.

Organization

The article is organized as follows: in Section II, we provide background information on Gaussian integer arithmetic and illustrate the Barrett reduction algorithm for integer numbers. In Section III, we first discuss the issues of directly applying this reduction algorithm to Gaussian integers, and then derive a

new reduction algorithm suitable for Gaussian integer moduli of arbitrary form. In Section IV, we discuss implementation aspects of the proposed reduction algorithm, presenting the results of complexity analysis, and comparing them to Montgomery reduction for Gaussian integers. Finally, conclusions are provided in Section V.

II. Preliminaries

This section provides a brief overview of Gaussian integers and the Barrett modulo reduction algorithm for integer numbers.

A. Gaussian Integers

Gaussian integers are a subset of complex numbers where both the real and imaginary parts are integers, denoted as $\mathbb{Z}[\mathrm{i}] = \{a + b\mathrm{i} \in \mathbb{C} : a, b \in \mathbb{Z}, \mathrm{i} = \sqrt{-1}\}$. When the complex operations of addition and multiplication are performed on Gaussian integers, modulo π, a finite ring or field is formed, depending on the particular choice of the modulus π.

Suppose $p = \pi\pi^*$ is an integer prime number such that $p \equiv 1 \bmod 4$, where π^* is the complex conjugate of π. In this case, the set of Gaussian integers taken modulo π together with the complex operations of addition and multiplication modulo π forms a Gaussian integer field denoted by $\mathcal{G}_p$:

$$\mathcal{G}_p = \{z \bmod \pi : z, \pi \in \mathbb{Z}[\mathrm{i}]\}. \tag{1}$$

This field is isomorphic to the integer prime field $\mathbb{Z}_p$, as demonstrated in [8].

The isomorphism $\phi : \mathbb{Z}_p \to \mathcal{G}_p$ maps an integer $s \in \mathbb{Z}_p$ to a Gaussian integer $z \in \mathcal{G}_p$ via Gaussian integer modulo reduction $z = s \bmod \pi$. This mapping is bijective, meaning that each element in $\mathbb{Z}_p$ is uniquely mapped to an element in $\mathcal{G}_p$, and vice versa. The inverse mapping $\phi' : \mathcal{G}_p \to \mathbb{Z}_p$ bijectively maps a Gaussian integer $z \in \mathcal{G}_p$ back to an element $s \in \mathbb{Z}_p$ as follows:

$$s = (zu\pi^* + z^*v\pi) \bmod p, \tag{2}$$

where z^* is the complex conjugate of the Gaussian integer z, and (u, v) are defined as $1 = v\pi + u\pi^*$. These coefficients can be obtained using the extended Euclidean algorithm, as described in [8].

Suppose $n = pq = \pi\pi^*$ is the product of two prime integer numbers p and q such that $p \equiv q \equiv 1 \bmod 4$ with $p \neq q$ and $\pi, \pi^* \in \mathbb{Z}[\mathrm{i}]$. In this case, the set $\mathcal{G}_n = z \bmod \pi, z \in \mathbb{Z}[\mathrm{i}]$, together with the operations of addition and multiplication modulo π, forms a Gaussian integer finite ring. This ring is isomorphic to the integer modular ring $\mathbb{Z}_n$, as shown in [6].

In this paper, we use $\mathrm{Re}\{z\}$ and $\mathrm{Im}\{z\}$ to denote the real and imaginary parts of any Gaussian integer $z \in \mathbb{Z}[\mathrm{i}]$, respectively. Additionally, we utilize the absolute value $|z| = \sqrt{zz^*} = \sqrt{a^2 + b^2}$ to measure the norm of a Gaussian integer $z = a + b\mathrm{i}$.

The addition of two Gaussian integers $x = a + b\mathrm{i}$ and $y = c + d\mathrm{i}$ can be computed using just 2 integer additions, as follows: $x+y = (a+c)+(b+d)\mathrm{i}$. Similarly, the multiplication of two Gaussian integers $x = a + b\mathrm{i}$ and $y = c + d\mathrm{i}$ can be computed using 3 integer multiplications, 2 integer additions, and 3 integer subtractions. Specifically, we have $xy = (v_2 - v_3) + (v_1 - v_2 - v_3)\mathrm{i}$, where $v_1 = (a + b)(c + d)$, $v_2 = ac$, and $v_3 = bd$.

Let $z, \pi \in \mathbb{Z}[\mathrm{i}]$ be Gaussian integers with $\pi \neq 0$. The naïve way of computing a Gaussian integer modulo reduction $z \bmod \pi$ consists of 3 complex multiplications[†], 1 complex subtraction, 1 complex division, and 1 complex rounding operation[‡] as shown in [8]:

$$z \bmod \pi = z - \left[\frac{z\pi^*}{\pi\pi^*}\right]\pi. \tag{3}$$

A more efficient reduction algorithm for Gaussian integers, based on the idea of Montgomery reduction for integer numbers, has been introduced in [18] and is illustrated in Appendix VI-B for the convenience of the reader."

B. Barrett Reduction

Given two integer numbers z and m, Barrett's reduction algorithm computes the remainder $r = z \bmod m$, where $z = qm + r$ in an efficient way. The main idea of Barrett's reduction algorithm is to find an approximation for the quotient q using elementary operations, and arrive at the final result r by subtractions of the modulo m. Barrett reduction was introduced in [2] and is illustrated in Alg. 1. This algorithm exploits a precomputed quantity $\mu = \lfloor \beta^{2k}/m \rfloor$ to reduce the complexity of the modulo reduction, where k denotes the bit-length of m. The radix is typically chosen to match the word-size of the underlying processor, in order to speed up arithmetic operations by employing only digit shifts and truncations [14].

Algorithm 1 Barrett Reduction Alg. [14, Alg. 14.42].

Input: Two positive integer numbers z and m, $\mu = \lfloor \beta^{2k}/m \rfloor$, $\beta > 3$
Output: Integer number $r = z \bmod m$

1: $q_1 \leftarrow \lfloor z/\beta^{k-1} \rfloor$
2: $q_2 \leftarrow q_1\mu$
3: $q_3 \leftarrow \lfloor q_2/\beta^{k+1} \rfloor$
4: $r_1 \leftarrow z \bmod \beta^{k+1}$
5: $r_2 \leftarrow q_3 m \bmod \beta^{k+1}$
6: $r' \leftarrow r_1 - r_2$
7: **if** $(r' < 0)$ **then**
8: $\quad r' \leftarrow r' + \beta^{k+1}$
9: **end if**
10: **while** $(r' \geq m)$ **do**
11: $\quad r' \leftarrow r' - m$
12: **end while**
13: $r \leftarrow r'$
14: **return** r

[†]If $\pi\pi^*$ can be precomputed, then only 2 complex multiplications are required.

[‡]The brackets $[\cdot]$ denote rounding to the closest Gaussian integer, i.e., $[a + b\mathrm{i}] = [a] + [b]\,\mathrm{i}$.

In particular, lines 1-9 of Alg. 1 efficiently reduce the input z to a congruent solution r'. Successively, the congruent r' is possibly further reduced in line 10 to 12 to obtain the final result r. We refer to this latter procedure as the final reduction. According to [14, Note 14.44], the final reduction is repeated at most twice, which can be simply implemented using comparisons and integer subtractions for integer numbers.

An improved version of the Barrett reduction, which reduces the number of iterations for the final reduction to at most one was presented by [3], [5], [9]. This improved version replaces the approximation of the quotient q_3 calculated on lines 1-3 of Alg. 1

$$q_3 = \left\lfloor \frac{\left\lfloor \frac{z}{\beta^{k-1}} \right\rfloor \left\lfloor \frac{\beta^{2k}}{m} \right\rfloor}{\beta^{k+1}} \right\rfloor, \tag{4}$$

with the following approximated quotient:

$$q_3 = \left\lfloor \frac{\left\lfloor \frac{z}{\beta^{k+\delta}} \right\rfloor \left\lfloor \frac{\beta^{k+\gamma}}{m} \right\rfloor}{\beta^{\gamma-\delta}} \right\rfloor, \tag{5}$$

where the optimal values for γ and δ can be determined according to the word size of the underlying processor [3], [5], [9].

III. Barrett Modulo Reduction for Gaussian Integers

In this section, we present a novel modulo reduction algorithm for Gaussian integers, which draws inspiration from the Barrett reduction technique (cf. Alg. 1).

Before starting, we highlight the major differences when working with Gaussian integers: the final reduction for Gaussian integers [17], [18], [19], [20] is considerably more expensive. Hence, it is imperative to identify a congruent solution that is as small as possible, in order to mitigate the high costs associated with the final reduction step. This can be done by taking inspiration from the improved version of the Barrett reduction [3], [5], [9] which allows for minimal costs on integer numbers, and adapting it to support Gaussian integers. This adaption firstly requires, replacing the floor divisions (cf. μ, lines 1 and 3 in Alg. 1) with appropriate rounding functions suitable for Gaussian integers to obtain to the smallest congruent. Secondly, changes to the associated values for γ and δ are necessary in order to guarantee the correctness of the reduction algorithm for Gaussian integers. Third, there is no need to correct negative values back to positive domain (cf. lines 7-8 of Alg. 1), as Gaussian integers already consist of positive and negative integers (for real and imaginary parts), hence no further correction is required in the final reduction.

We proceed as follows: first, we describe a variant of Alg. 1 that supports Gaussian integers. Next, we demonstrate that this algorithm always obtains a congruent solution r'. Then, we show how to obtain the final result r from r' by bounding the absolute values. Afterwards, we consider an efficient method to perform the final reduction for the proposed algorithm based on $|r'|$, and demonstrate that it always leads to the desired final result that would be obtained using Eq. (3). Finally, we provide an example (cf. Example 2) to exhibit higher efficiency for the final reduction using the signs of the real and imaginary parts of the resulting congruent.

The proposed algorithm that supports Gaussian integers is illustrated in Alg. 2. In line 6 of this algorithm, we subtract a multiple of π from z. Obviously, this always results in the congruent

$$r' = r_1 - r_2 = z - q_3\pi \equiv z - \left\lceil \frac{z}{\pi} \right\rfloor \pi = r, \tag{6}$$

since q_3, z, r', and π are Gaussian integers and $r = z - \lceil z/\pi \rfloor \pi$ is the final result, cf. Eq. (3).

In order to determine a congruent solution r' that is as small as possible, we should efficiently calculate the quotient q_3, which is as near as possible to $[z/\pi]$ from Eq. (3). Since Gaussian integers include signed integers for real and imaginary parts, we define the function fdiv as an efficient Gaussian integer division with rounding towards zero, i.e., for any Gaussian integers x and $y \neq 0$, we have

$$\begin{aligned} x \operatorname{fdiv} y &= s(\mathrm{Re}\left\{\frac{x}{y}\right\}) \left\lfloor \left| \mathrm{Re}\left\{\frac{x}{y}\right\} \right| \right\rfloor \\ &+ s(\mathrm{Im}\left\{\frac{x}{y}\right\}) \left\lfloor \left| \mathrm{Im}\left\{\frac{x}{y}\right\} \right| \right\rfloor \mathrm{i}, \end{aligned} \tag{7}$$

where $s(\cdot)$ denotes the sign function. Similarly, we define the function cdiv as an efficient Gaussian integer division with rounding away from zero as follows:

$$\begin{aligned} x \operatorname{cdiv} y &= s(\mathrm{Re}\left\{\frac{x}{y}\right\}) \left\lceil \left| \mathrm{Re}\left\{\frac{x}{y}\right\} \right| \right\rceil \\ &+ s(\mathrm{Im}\left\{\frac{x}{y}\right\}) \left\lceil \left| \mathrm{Im}\left\{\frac{x}{y}\right\} \right| \right\rceil \mathrm{i}. \end{aligned} \tag{8}$$

Note that Eq. (7) can be implemented using digit shifts, e.g., if y is a power of the basis β. Similarly, Eq. (8) can be implemented using digit shifts and addition of the constant value 1.

The following example illustrates the rounding functions defined in Eq. (3), Eq. (7), and Eq. (8), respectively.

Example 1. *Let* $x = 5+4\mathrm{i}$ *and* $y = 3$*, we have* $x/y \approx 1.66+1.33\mathrm{i}$*,* $z \operatorname{fdiv} y = 1+\mathrm{i}$*,* $z \operatorname{cdiv} y = 2+2\mathrm{i}$*, and* $[z/y] = 2+\mathrm{i}$*. Similarly, for* $x = -4-5\mathrm{i}$*, we have* $x/y \approx -1.33-1.66\mathrm{i}$*,* $z \operatorname{fdiv} y = -1-\mathrm{i}$*,* $z \operatorname{cdiv} y = -2-2\mathrm{i}$*, and* $[z/y] = -1-2\mathrm{i}$*.*

Next, we demonstrate that Alg. 2 always obtains the final result as Eq. (3). We proceed with the following lemma which provides basic bounds.

Algorithm 2 Barrett reduction for any Gaussian integer z and the modulus $\pi \neq 0$ with $|z| \leq \left|\pi^2\right|$, where $|\mathrm{Re}\{\pi\}| < \beta^k$, $|\mathrm{Im}\{\pi\}| < \beta^k$, and β is the chosen basis. The Gaussian integer $\mu = \beta^{k+\gamma} \operatorname{cdiv} \pi$ can be precomputed and stored, where $\gamma \geq k+3$ and $\delta \leq -3$.

input: Gaussian integers z, μ, π, integer numbers β, γ, δ
output: Gaussian integer $r = z \bmod \pi$

1: $q_1 \leftarrow z \operatorname{cdiv} \beta^{k+\delta}$
2: $q_2 \leftarrow q_1 \mu$
3: $q_3 \leftarrow q_2 \operatorname{fdiv} \beta^{\gamma-\delta}$
4: $r_1 \leftarrow z \bmod \beta^{\gamma-\delta}$
5: $r_2 \leftarrow q_3 \pi \bmod \beta^{\gamma-\delta}$
6: $r' \leftarrow r_1 - r_2$
7: **if** $(|r'| < |\pi|\,(\sqrt{2}-1)/\sqrt{2})$ **then**
8: $\quad \alpha \leftarrow 0$
9: **else if** $(|r'| < |\pi|\,/\sqrt{2})$ **then**
10: $\quad \alpha \leftarrow \operatorname{argmin}_{\hat{\alpha} \in \{0, \pm 1, \pm \mathrm{i}\}} |r' - \hat{\alpha}\pi|$
11: **else**
12: $\quad \alpha \leftarrow \operatorname{argmin}_{\hat{\alpha} \in \{\pm 1, \pm \mathrm{i}, \pm 1 \pm \mathrm{i}\}} |r' - \hat{\alpha}\pi|$
13: **end if**
14: $r \leftarrow r' - \alpha\pi$
15: **return** r

Lemma 1. *For any two Gaussian integers x, y, and a quotient $Q = [x/y]$, where $y \neq 0$, it holds*

$$|x \operatorname{fdiv} y| \leq \left|\frac{x}{y}\right| \leq |x \operatorname{cdiv} y|\,, \tag{9}$$

$$|Q| - \sqrt{2} \leq |x \operatorname{fdiv} y|\,, \tag{10}$$

$$|x \operatorname{cdiv} y| < \left|\frac{x}{y}\right| + \sqrt{2}, \tag{11}$$

$$\left|\frac{x}{y}\right| \leq |Q| + \frac{1}{\sqrt{2}}. \tag{12}$$

Proof: Eq. (9) *directly follows from the definition of the* fdiv *function (rounding towards zero), and the definition of the* cdiv *function (rounding away from zero). Let $x \operatorname{fdiv} y - Q = c + d\mathrm{i}$. Due to the difference between the rounding to the nearest Gaussian integer and the rounding towards zero function, cf. Eq.* (7)*, we have $0 \leq |c| \leq 1$, $0 \leq |d| \leq 1$. This results in*

$$0 \leq |c + d\mathrm{i}| \leq \sqrt{2},$$

and the upper bound in Eq. (10) *holds. Similarly, for $x/y - x \operatorname{cdiv} y = e + f\mathrm{i}$ we have $0 \leq |e| < 1$, $0 \leq |f| < 1$, due to the difference between the regular division and the rounding away from zero function, cf. Eq.* (8)*. This results in*

$$0 \leq |e + f\mathrm{i}| < \sqrt{2},$$

and the upper bound in Eq. (11) *holds. Finally, let $x/y - Q = g + h\mathrm{i}$. Due to the rounding to the nearest Gaussian integer function, we have $0 \leq |g| \leq 1/2$, $0 \leq |h| \leq 1/2$. This results in*

$$0 \leq |g + h\mathrm{i}| \leq \frac{1}{\sqrt{2}},$$

and the upper bound in Eq. (12) *holds.* ∎

In line 3 of Alg. 2, we observe that

$$\begin{aligned} q_3 &= q_2 \operatorname{fdiv} \beta^{\gamma-\delta} = q_1 \mu \operatorname{fdiv} \beta^{\gamma-\delta} \\ &= (z \operatorname{cdiv} \beta^{k+\delta}) \cdot (\beta^{k+\gamma} \operatorname{cdiv} \pi) \operatorname{fdiv} \beta^{\gamma-\delta}. \end{aligned}$$

Next, we bound the absolute value of the quotient q_3 using Eq. (9):

$$\begin{aligned} & \left|(\frac{z}{\beta^{k+\delta}}) \cdot (\frac{\beta^{k+\gamma}}{\pi}) \operatorname{fdiv} \beta^{\gamma-\delta}\right| \\ \leq\ & \left|(z \operatorname{cdiv} \beta^{k+\delta}) \cdot (\beta^{k+\gamma} \operatorname{cdiv} \pi) \operatorname{fdiv} \beta^{\gamma-\delta}\right| = |q_3| \\ \leq\ & \left|\frac{(z \operatorname{cdiv} \beta^{k+\delta}) \cdot (\beta^{k+\gamma} \operatorname{cdiv} \pi)}{\beta^{\gamma-\delta}}\right|. \end{aligned} \tag{13}$$

In the following, we denote the quotient from Eq. (3) as $Q = [z/\pi]$. Now, considering the lower bound of $|q_3|$ from Eq. (13), we have that

$$|q_3| \geq \left|\frac{z}{\pi} \cdot \beta^{\gamma-\delta}\right| \operatorname{fdiv} \beta^{\gamma-\delta} = |z \operatorname{fdiv} \pi| \geq |Q| - \sqrt{2}, \tag{14}$$

where we used the lower bound from Eq. (10). Consequently, using Eq. (14), we can bound the difference between the absolute values of Q and q_3 as follows:

$$|Q| - |q_3| \leq \sqrt{2}. \tag{15}$$

According to Eq. (11), we obtain the upper bound of $|q_3|$ from Eq. (13) as follows:

$$|q_3| < \frac{\left(\left|\frac{z}{\beta^{k+\delta}}\right| + \sqrt{2}\right)\left(\left|\frac{\beta^{k+\gamma}}{\pi}\right| + \sqrt{2}\right)}{|\beta^{\gamma-\delta}|},$$

$$|q_3| < \frac{\left|\frac{z}{\beta^{k+\delta}}\right|\left|\frac{\beta^{k+\gamma}}{\pi}\right| + \sqrt{2}\left(\left|\frac{z}{\beta^{k+\delta}}\right| + \left|\frac{\beta^{k+\gamma}}{\pi}\right| + \sqrt{2}\right)}{|\beta^{\gamma-\delta}|},$$

$$|q_3| < \left|\frac{z}{\pi}\right| + \sqrt{2}\left(\left|\frac{z}{\beta^{k+\gamma}}\right| + \left|\frac{\beta^{k+\delta}}{\pi}\right| + \left|\frac{\sqrt{2}}{\beta^{\gamma-\delta}}\right|\right).$$

Now, using the upper bound of Eq. (12) from Lemma 1, we can estimate the difference between the absolute values of the quotients as

$$|q_3| - |Q| < \sqrt{2}\left(\left|\frac{z}{\beta^{k+\gamma}}\right| + \left|\frac{\beta^{k+\delta}}{\pi}\right| + \left|\frac{\sqrt{2}}{\beta^{\gamma-\delta}}\right|\right) + \frac{1}{\sqrt{2}}. \tag{16}$$

Without loss of generality, we assume a modulus $\pi \in \mathbb{Z}[i]$ with $\mathrm{Re}\{\pi\} > \mathrm{Im}\{\pi\} \geq 0$, where $\mathrm{Re}\{\pi\}$ has k digits and the most significant digit is different from zero:

$$\begin{aligned} \beta^{k-1} &< |\mathrm{Re}\{\pi\}| < \beta^k, \\ 0 &< |\mathrm{Im}\{\pi\}|\,. \end{aligned}$$

Assuming two Gaussian integers x and y with at most k digits in their real and imaginary parts, we can bound the real and imaginary parts of their product $z = xy$ as follows:

$$0 \leq |\mathrm{Re}\{z\}|\,, |\mathrm{Im}\{z\}| < \beta^{2k}. \tag{17}$$

Hence, we can bound the absolute values of π and z as

$$|\pi| > \sqrt{0 + (\beta^{k-1})^2} = \beta^{k-1}, \quad (18)$$

$$|z| < \sqrt{(\beta^{2k})^2 + (\beta^{2k})^2} = \sqrt{2}\beta^{2k}. \quad (19)$$

Substituting both Eq. (18) and Eq. (19) in Eq. (16) results in

$$|q_3| - |Q| < \sqrt{2}\left(\left|\frac{\sqrt{2}\beta^{2k}}{\beta^{k+\gamma}}\right| + \left|\frac{\beta^{k+\delta}}{\beta^{k-1}}\right| + \left|\frac{\sqrt{2}}{\beta^{\gamma-\delta}}\right|\right) + \frac{1}{\sqrt{2}}, \quad (20)$$

$$|q_3| - |Q| < 2\left|\beta^{k-\gamma}\right| + \sqrt{2}\left|\beta^{\delta+1}\right| + 2\left|\beta^{\delta-\gamma}\right| + \frac{1}{\sqrt{2}}. \quad (21)$$

Please note that basis β is typically a power of two that matches the word size of the processor being considered. Suppose, for example, a binary basis, i.e., $\beta = 2$, we obtain

$$|q_3| - |Q| < 2 \cdot 2^{k-\gamma} + \sqrt{2} \cdot 2^{\delta+1} + 2 \cdot 2^{\delta-\gamma} + \frac{1}{\sqrt{2}}. \quad (22)$$

Now, let $\epsilon = 2 \cdot 2^{k-\gamma} + \sqrt{2} \cdot 2^{\delta+1} + 2 \cdot 2^{\delta-\gamma} + \frac{1}{\sqrt{2}}$, with $\gamma \geq k+3$, $\delta \leq -3$, we get

$$0 < 2 \cdot 2^{k-\gamma} + 2 \cdot 2^{\delta-\gamma} \leq \frac{1}{4} + \frac{2}{2^{k+6}}, \quad (23)$$

$$0 < \sqrt{2} \cdot 2^{\delta+1} \leq \frac{\sqrt{2}}{4}, \quad (24)$$

$$|q_3| - |Q| < \epsilon \leq \frac{1+\sqrt{2}}{4} + \frac{2}{2^{k+6}} + \frac{1}{\sqrt{2}} < 1.33 < \sqrt{2}, \quad (25)$$

where we assumed an extreme case with $k = 1$ to obtain a maximum for ϵ, i.e., the difference between $|Q|$ and $|q_3|$, cf. Eq. (25). In general, using $\gamma \geq k+3$ and $\delta \leq -3$, the resulting upper bound will decrease when increasing the base β, e.g., consider the case for $k = 1$, we have $\epsilon < 0.83$ for $\beta = 4$ and $\epsilon < 0.71$ for $\beta = 32$.

In conclusion, with Eq. (15) and Eq. (25), we have proved that the upper bound for the difference between the absolute value of q_3 (resulting from Alg. 2) and of Q (for the final result, cf. Eq. (3)) is at most $\sqrt{2}$. Next, we demonstrate how to obtain the final result of Eq. (3) after line 6 of Alg. 2 using the following lemma.

Lemma 2. *Let $\mathcal{G}_n$ be any Gaussian integer ring, with $n = \pi\pi^*$. For any $z \notin \mathcal{G}_n$ and $r = z \bmod \pi \in \mathcal{G}_n$, we have*

$$|r| < |z|. \quad (26)$$

We refer to [18, Lemma 1] for the derivation.

We aim to compensate the difference between Q and q_3 using a corresponding offset α, i.e., we obtain the final result after line 6 of Alg. 2 as

$$r = r' - \alpha\pi. \quad (27)$$

Since the distance between $|Q|$ and $|q_3|$ is bounded by $\sqrt{2}$ (cf. Eq. (15) and Eq. (25)), we consider all possible offset candidates for α to identify the correct offset candidate. The offset α must be a Gaussian integer, since r, r', and π are Gaussian integers. Consequently, using the upper bound $\sqrt{2} = |\pm 1 \pm \mathrm{i}|$ from Eq. (15) and (25), we may restrict the possible offset candidates based on the absolute value to

$$\alpha \in 0, \pm 1, \pm\mathrm{i}, \pm 1 \pm \mathrm{i}. \quad (28)$$

Consequently, substituting all possible candidates for α in Eq (27) results in nine different congruent solutions.

However, according to Lemma 2, the final result can be obtained by minimizing the absolute value over all possible offset candidates from Eq. (28) as

$$r = r' - \alpha\pi, \quad (29)$$

$$\alpha = \underset{\hat{\alpha} \in \{0, \pm 1, \pm\mathrm{i}, \pm 1 \pm \mathrm{i}\}}{\operatorname{argmin}} |r' - \hat{\alpha}\pi|. \quad (30)$$

Minimizing the absolute value over nine different offsets to obtain the final result r may be costly for some applications. Hence, we demonstrate in the following that the number of possible offset candidates can be reduced based on $|r'|$.

Lemma 3. *For any r that is an element of the Gaussian integer ring $\mathcal{G}_n$, $n = \pi\pi^*$, we have*

$$|r| < \frac{|\pi|}{\sqrt{2}}, \quad (31)$$

cf. [18, Lemma 1] for the derivation.

To efficiently determine a final result, we use Lemma 3 and the following bounds to restrict the required number of comparisons to obtain the final result.

For $|r'| < \frac{\sqrt{2}-1}{\sqrt{2}}|\pi|$, we already have the unique result, i.e., $r = r'$. For $\frac{\sqrt{2}-1}{\sqrt{2}}|\pi| \leq |r'| < \frac{|\pi|}{\sqrt{2}}$, minimizing the absolute value over the following offset candidates $\hat{\alpha}$ is sufficient to determine the correct offset

$$\alpha = \underset{\hat{\alpha} \in \{0, \pm 1, \pm\mathrm{i}\}}{\operatorname{argmin}} |r' - \hat{\alpha}\pi|. \quad (32)$$

Otherwise, we have

$$\alpha = \underset{\hat{\alpha} \in \{\pm 1, \pm\mathrm{i}, \pm 1 \pm \mathrm{i}\}}{\operatorname{argmin}} |r' - \hat{\alpha}\pi|. \quad (33)$$

According to Eq. (32) and Eq. (33), up to nine offset candidates $\hat{\alpha}$ may be needed. Nonetheless, not all potential offsets have to be considered. These potential offsets may be further reduced based on the sign of $\mathrm{Re}\{r'\}$ and $\mathrm{Im}\{r'\}$, i.e., because we can restrict the quadrant of interest. This procedure is illustrated in Example 2.

Example 2. *Let $\pi = 8+3\mathrm{i}$, we have $p = \pi\pi^* = 73$ and the set $\mathcal{G}_p$ is a Gaussian integer field isomorphic to the field $\mathbb{Z}_p$ over ordinary integers. Figure 1 demonstrates all possible elements from $\mathcal{G}_{73}$ with blue points and the origin. Consider for example the multiplication of the two field elements $x = 3 + 2\mathrm{i}$ and $y = 2+2\mathrm{i}$, we get $z = xy = 2+10\mathrm{i}$. Now, apply the reduction according to Alg. 2 for example with $\beta = 2$, $\gamma = k + 3$, $\delta = -3$, and hence $\mu = 56 - 21\mathrm{i}$. Line 3 of this algorithm results in $q_3 = \mathrm{i}$, and line 6 in $r' = 5 + 2\mathrm{i}$. For the absolute value we get $2.50 \approx \frac{\sqrt{2}-1}{\sqrt{2}}|\pi| < |r'| = \sqrt{29} \approx 5.39 < \frac{|\pi|}{\sqrt{2}} \approx 6.04$. Hence, we determine the correct representative using the offsets in line 10 of Alg. 2. Since r' is in the first quadrant, we*

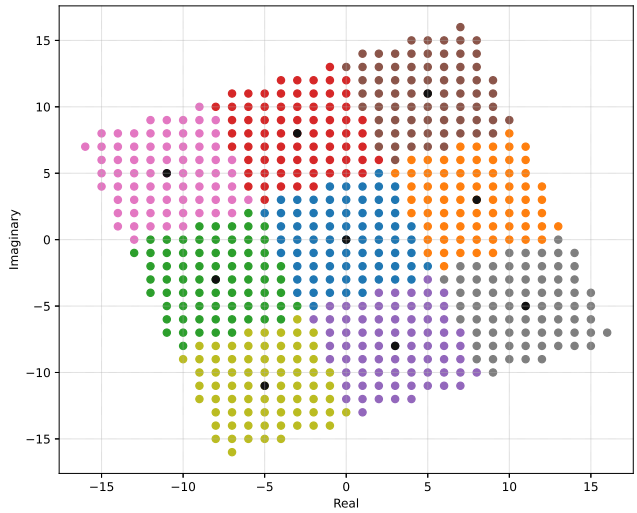

Fig. 1. Example for $p = 73$ and $\pi = 8 + 3\mathrm{i}$. Elements of the Gaussian integer field $\mathcal{G}_{73}$ are the origin and the blue points around the origin. Possible offsets according to Eq. (28) are depicted in black points. All points with other colors illustrate the sum of field elements with an offset from Eq. (28).

get only three possible offset candidates, i.e., $\hat{\alpha} \in \{0, 1, -\mathrm{i}\}$. *Consequently, computing* $|r' - 0| \approx 5.39$, $|r' - \pi| \approx 3.16$, *and* $|r' - \mathrm{i}\pi| \approx 10.0$ *leads to* $\alpha = -1$. *Finally, we obtain the final result in line 14 of Alg. 2 as* $r = r' - \alpha\pi = -3 - \mathrm{i}$.

IV. Discussion

In this section, we discuss implementation aspects related to Alg. 2. First, we show how that the Manhattan weight can be used for fast reduction to obtain smaller congruent solutions. Then, we perform a complexity analysis showing that the proposed method is more efficient than the naive reduction approach and has equivalent complexity compared with Montgomery reduction [18], but does not require additional operations for domain transformations.

A. Fast Reduction with Manhattan Weight

In previous sections, we used the absolute value of a Gaussian integer to measure its norm for offset comparisons in the final reduction (cf. Eq. (32) and Eq. (33)). However, calculating the absolute value of a Gaussian integer and performing comparison can be computationally expensive and inefficient.

To address these issues, one could adopt the approach presented in [18] for fast reduction, which replaces the absolute value by the Manhattan weight of a Gaussian integer. This approach reduces computational complexity and improves efficiency. The Manhattan weight of a Gaussian integer $x = a + b\mathrm{i}$ is defined as $\|x\| = |a| + |b|$. This weight is calculated by adding the magnitudes of the real and imaginary parts of the Gaussian integer $x = a + b\mathrm{i}$, while ignoring their signs of a and b.

Note that $|x| \leq \|x\|$ holds for any Gaussian integer x. This can be seen by squaring both sides of the inequality, we refer to [18] for further derivations. Alg. 3 illustrates the final reduction using Manhattan weight, which may be used to replace lines 7-15 of Alg. 2. The number of possible offset candidates $\hat{\alpha}\pi$ in line 1 of Alg. 3 may be further reduced based on the signs of $\mathrm{Re}\{r'\}$ and $\mathrm{Im}\{r'\}$, as illustrated in Example 2. An alternative version of Alg. 2 using Manhattan weight for fast reduction is reported in the Appendix VI-C for the convenience of the reader.

It is worth mentioning that the final reduction step using the Manhattan weight method may not always result in a unique solution. In fact, there are at most two possible solutions that are congruent [18]. Despite this, the Manhattan weight method remains a useful tool for applications that require obtaining a smaller congruent solution for intermediate results. The final result can still be determined using the final reduction step outlined in Alg. 2.

Algorithm 3 Fast Reduction using Manhattan Weight.

Input: Gaussian integers r', π s.t. $r \equiv r' \bmod \pi$ and $|r'| \leq \sqrt{2}\,|\pi|$
Output: $r \equiv r' \bmod \pi$ and $|r| \leq |r'|$

1: $\alpha \leftarrow \mathrm{argmin}_{\hat{\alpha} \in \{0, \pm 1, \pm \mathrm{i}, \pm i \pm \mathrm{i}\}} \|r' - \hat{\alpha}\pi\|$
2: $r \leftarrow r' - \alpha\pi$
3: **return** r

B. Complexity Analysis

In the following, we compare the computational complexity of our proposed reduction algorithms with that of existing reduction methods, i.e., the naive reduction of Eq. (3) and Montgomery reduction algorithm for Gaussian integers [18]. We have not included a comparison with efficient reduction algorithm for Gaussian integers proposed in [20], as this latter is optimized for specific Gaussian integer moduli of special forms, hence not suitable for general usage. Also note that we can safely neglect the complexity requirements of the final reduction, as all the considered algorithms use the same procedure for final reduction.

When considering Alg. 2, it is possible to neglect the costs associated with calculating μ, as it can be precomputed once and stored, particularly if the modulus π is fixed. Lines 1-6 of this algorithm require digit operations, two multiplications, and one subtraction for Gaussian integers. It is worth noting that the fdiv function utilized to compute q_3 can be implemented using simple digit shifts. On the other hand, the cdiv function used to compute q_1 and μ may require an additional step of adding a constant 1 to the real and imaginary parts, respectively. Similarly, the modulo functions to determine r_1 and r_2 can be implemented using truncations on the real and imaginary parts. Additionally, if the Gaussian integer modulus π is fixed, then the costs for the two multiplications are reduced to two multiplications by a constant, i.e., μ and π.

The complexity of a fast reduction using the Manhattan weight depends on the signs of $\mathrm{Re}\{r'\}$ and $\mathrm{Im}\{r'\}$. In worst

case, it accounts for 9 complex subtractions, since all nine $\hat{\alpha}\pi$ values may be precomputed, cf. Eq. (32) and Eq. (33). Additionally, the Manhattan weight must be computed 9 times using 9 integer additions, i.e., additions of the real and imaginary parts (cf. Sec. IV-A). Please note that in some applications, it is sufficient to determine a congruent $r' \equiv z \bmod \pi$ and neglect the final reduction for all intermediate results, i.e., until the final result is required. This holds for both Barrett and Montgomery reduction algorithms on Gaussian integers.

The complexity of all considered algorithms is summarized in Table I, reporting the required number of complex arithmetic operations. Truncations and digit shifts are not included, since they can be implemented efficiently. Table I shows that a naive reduction technique requires one complex division (i.e., two integer divisions for real and imaginary parts), and hence is not suitable for efficient implementations.

The required complex arithmetic operations for the Montgomery approach from [18] include the costs for Montgomery reduction algorithm, as well as for both the forward and backward domain transformations. The costs for the forward transformation can be approximated by a complex multiplication as well as the costs for a complete Montgomery reduction algorithm from [18]. Similarly, 1 extra complete Montgomery reduction is employed for the backward transformation (cf. Appendix VI-B). We estimate the costs for the Montgomery reduction with 2 complex multiplications and 1 complex addition (without costs for the final reduction). Hence, these transformations for the Montgomery domain using Gaussian integers can be estimated with at least 5 complex multiplications and 2 complex additions.

These domain transformations, and hence the corresponding costs are not required for the proposed reduction algorithm, which increases the computational efficiency, especially if the final result is required after a few number of intermediate results. For example, the multiplication of two Gaussian integers with a modulo reduction would employ 8 complex multiplications and 3 complex additions, for the forward transformation, the calculation in the Montgomery domain including modulo reduction, and transforming the result back into the ordinary domain. Using the proposed algorithm only 3 complex multiplications and 1 complex addition are needed, since the calculation is done in the ordinary domain.

TABLE I
NUMBER OF REQUIRED COMPLEX ARITHMETIC OPERATIONS. COSTS FOR MONTGOMERY DOMAIN TRANSFORMATIONS ARE ACCOUNTED IN THE TABLE, BUT THE COSTS FOR FINAL REDUCTION ARE LEFT OUT FOR BOTH MONTGOMERY AND BARRETT (ALG. 2) SINCE THEY ARE JUST THE SAME.

	Addition / Subtraction	Multiplication by a constant	Division
Barrett Reduction Alg. 2	1	2	-
Montgomery Reduction [18]	1	2	-
Montgomery Transformations [18]	2	5	-
Naive Reduction Eq. (3)	1	2	1

V. CONCLUSION

In this paper, we presented an efficient reduction algorithm for arbitrary Gaussian integer moduli that builds on the ideas of Barrett reduction for integer numbers. Our proposed algorithm involves only basic arithmetic operations such as integer multiplications, additions, subtractions, and simple digit operations. It has a computational complexity that is comparable to the Montgomery reduction algorithm on Gaussian integer moduli. The key advantage of our proposed method is that it does not entail the costs associated with the Montgomery domain transformation, making it more advantageous in many practical cases.

REFERENCES

[1] Y. Awad, A. N. El-Kassar, and T. Kadri, *Rabin Public-Key Cryptosystem in the Domain of Gaussian Integers*, In: International Conference on Computer and Applications (ICCA). Aug. 2018, pp. 336–340.

[2] P. Barrett, *Implementing the Rivest Shamir and Adleman Public Key Encryption Algorithm on a Standard Digital Signal Processor*, Advances in Cryptology – CRYPTO' 86, pp.311-326, 1986.

[3] J.-F. Dhem, *Modified Version of the Barrett Algorithm,*. technical report, 1994.

[4] H. Elkamchouchi, K. Elshenawy, and H. Shaban, *Extended RSA cryptosystem and digital signature schemes in the domain of Gaussian integers*, In: The 8th International Conference on Communication Systems (ICCS). Vol. 1. Nov. 2002, 91–95 vol.1.

[5] N. Emmart, F. Zheng and C. Weems, *A New Variant of the Barrett Algorithm Applied to Quotient Selection,*. In: 2018 IEEE 25th Symposium on Computer Arithmetic (ARITH), 2018, pp. 138-144.

[6] J. Freudenberger, F. Ghaboussi, and S. Shavgulidze, *New Coding Techniques for Codes over Gaussian Integers*, In: IEEE Transactions on Communications 61.8, Aug. 2013, pp. 3114–3124. ISSN: 0090-6778.

[7] J. Freudenberger and J. P. Thiers, *A New Class of Q-Ary Codes for the McEliece Cryptosystem*, In: Cryptography 5.1, 2021. ISSN: 2410-387X. doi: 10.3390/cryptography5010011. URL:https://www.mdpi.com/2410-387X/5/1/11.

[8] K. Huber, *Codes over Gaussian integers*, In: IEEE Transactions on Information Theory, 1994, pp. 207–216.

[9] M. Knezevic, F. Vercauteren and I. Verbauwhede, *Faster Interleaved Modular Multiplication Based on Barrett and Montgomery Reduction Methods,*. In: IEEE Transactions on Computers, vol. 59, no. 12, pp. 1715-1721, Dec. 2010.

[10] A. Koval and B. S. Verkhovsky, *Analysis of RSA over Gaussian Integers Algorithm*. In: Fifth International Conference on Information Technology: New Generations (ITNG), Apr. 2008, pp. 101–105. doi: 10.1109/ ITNG.2008.44.

[11] A. Koval, *Security systems based on Gaussian integers: Analysis of basic operations and time complexity of secret transformations*. Dissertation, New Jersey Institute of Technology, 2011.

[12] A. Koval, *Algorithm for Gaussian Integer Exponentiation*. In: Information Technology: New Generations. Springer International Publishing, 2016, pp. 1075–1085.

[13] C. Martinez, R. Beivide, and E. Gabidulin, *Perfect Codes for Metrics Induced by Circulant Graphs*. In: IEEE Transactions on Information Theory 53.9, Sept. 2007, pp. 3042–3052.

[14] A. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 2001. ISBN: 0-8493-8523-7.

[15] P. L. Montgomery, *Modular multiplication without trial division*, In: Mathematics of computation, 1985, pp. 519–521.

[16] D. Rohweder, J. Freudenberger, and S. Shavgulidze, *Low-Density Parity-Check Codes over Finite Gaussian Integer Fields*. In: 2018 IEEE International Symposium on Information Theory (ISIT), June 2018, pp. 481–485.

[17] M. Safieh and J. Freudenberger, *Montgomery Modular Arithmetic over Gaussian Integers*. In: 24th International Information Technology Conference (IT), Zabljak, Montenegro, Feb. 2020.

[18] M. Safieh and J. Freudenberger, *Montgomery Reduction for Gaussian Integers*. In: Cryptography, Jan. 2021.

[19] M. Safieh, J. Thiers, and J. Freudenberger, *A Compact Coprocessor for the Elliptic Curve Point Multiplication over Gaussian Integers*. In: Electronics, 2020.
[20] M. Safieh and F. De Santis, *Efficient Reduction Algorithms for Special Gaussian Integer Moduli*. In: 29th IEEE Symposium on Computer Arithmetic, ARITH 2022, Lyon, France, Sept. 2022.

VI. Appendix

A. Montgomery Integer Reduction

The Montgomery reduction for integer numbers is described in Algorithm 4. It replaces the calculation of the modulo operation $\bmod m$ by $\bmod R$, where $R > m$ is a power of two, hence replacing the modulo operation with truncation. For using such a reduction, integer numbers have to be mapped to the Montgomery domain. Given an integer input x, this can be mapped to Montgomery domain as follows:

$$X = xR \bmod m. \tag{34}$$

The reduction function $\mu(X)$ determines the backward transformation from the Montgomery domain, because $\mu(X) = xRR^{-1} \bmod m = x \bmod m$.

Algorithm 4 Montgomery Reduction Algorithm for Integer Numbers [15].

input: Z, with $0 \leq Z < mR$, $m' = -m^{-1} \bmod R$, and $R = 2^l \geq m$
output: $M = \mu(Z) = ZR^{-1} \bmod m$

$t \leftarrow Zm' \bmod R$
$q \leftarrow (Z + tm) \operatorname{div} R$
if $(q \geq m)$ **then**
  $M \leftarrow q - m$
else
  $M \leftarrow q$
end if
return M

Typically, all variables are mapped at the beginning of the calculation into the Montgomery domain using this reduction function as

$$X = \mu(xR^2) = xR^2R^{-1} \bmod m = xR \bmod m, \tag{35}$$

since only 1 precomputed value $R^2 \bmod m$ is required. Hence, a multiplication by R^2 and a full Montgomery reduction are required for the forward transformation to Montgomery domain, while a full Montgomery reduction is needed for the backward transformation from the Montgomery domain.

B. Montgomery Reduction for Gaussian Integers

The following algorithm describes Montgomery reduction for Gaussian integers according to [18]. Please note that Montgomery domain transformations for Gaussian integers are computed the same way as integer numbers.

Algorithm 5 Montgomery Reduction Algorithm for Gaussian Integers [18].

input: $Z = XY$, $\pi' = -\pi^{-1} \bmod R, R = 2^l > |\pi|/\sqrt{2}$
output: $M = \mu(Z) = ZR^{-1} \bmod \pi$

$t \leftarrow Z\pi' \bmod R$
$r' \leftarrow (Z + t\pi) \operatorname{fdiv} R$
if $(|r'| < |\pi|\,(\sqrt{2}-1)/\sqrt{2})$ **then**
  $\alpha \leftarrow 0$
else if $(|r'| < |\pi|\,/\sqrt{2})$ **then**
  $\alpha \leftarrow \operatorname{argmin}_{\hat{\alpha} \in \{0, \pm 1, \pm \mathrm{i}\}} |r' - \hat{\alpha}\pi|$
else
  $\alpha \leftarrow \operatorname{argmin}_{\hat{\alpha} \in \{\pm 1, \pm \mathrm{i}, \pm 1 \pm \mathrm{i}\}} |r' - \hat{\alpha}\pi|$
end if
$M \leftarrow r' - \alpha\pi$
return M

C. Barrett Reduction for Gaussian Integers using Manhattan Weight

The following algorithm is a modified version of Alg. 2 using Manhattan weight as described in Section IV-A.

Algorithm 6 Barrett reduction for any Gaussian integer z and the modulus $\pi \neq 0$ with $|z| \leq \left|\pi^2\right|$, where $|\mathrm{Re}\{\pi\}| < \beta^k$, $|\mathrm{Im}\{\pi\}| < \beta^k$, and β is the chosen basis. The Gaussian integer $\mu = \beta^{k+\gamma} \operatorname{cdiv} \pi$ can be precomputed and stored, where $\gamma \geq k + 3$ and $\delta \leq -3$.

input: Gaussian integers z, μ, π, integer numbers β, γ, δ
output: $r = z \bmod \pi$

$q_1 \leftarrow z \operatorname{cdiv} \beta^{k+\delta}$
$q_2 \leftarrow q_1\mu$
$q_3 \leftarrow q_2 \operatorname{fdiv} \beta^{\gamma-\delta}$
$r_1 \leftarrow z \bmod \beta^{\gamma-\delta}$
$r_2 \leftarrow q_3\pi \bmod \beta^{\gamma-\delta}$
$r' \leftarrow r_1 - r_2$
$\alpha \leftarrow \operatorname{argmin}_{\hat{\alpha} \in \{0, \pm 1, \pm \mathrm{i}, \pm i \pm \mathrm{i}\}} \|r' - \hat{\alpha}\pi\|$
$r \leftarrow r' - \alpha\pi$
return r