Infrastructure: Structure Inside the Class Group of a Real Quadratic Field

M. J. Jacobson Jr. and R. Scheidler

Cows and Fields

Suppose that you are a wise ancient Greek, and that you have been given the task of counting the Sun God's cattle on the island of Sicily. They are too numerous to count manually, and your only clue is a puzzle that relates the number of cows and bulls of a particular color to those of another. How can you determine the size of your herd from such cryptic information?

A precise version of this numerical puzzle is the well-known cattle problem of Archimedes [2]. A number of accounts of its solution appear in the literature [31, 21, 19]. Unfortunately for the hypothetical Greek sage, this problem does not have a small and completely elementary solution—it would take more than 2,000 years before the first solution was discovered by Amthor [1]. Had a solution been available, the Greek would likely have been shocked to find that the Sun God had somehow managed to pack a herd of more than 10^{206544} animals onto Sicily!

One of the main ingredients in Amthor's solution was solving the Diophantine equation

$$x^2 - 4729414y^2 = 1$$

for integers x and y. This equation is a specific instance of the (incorrectly named) Pell equation

$$x^2 - Dy^2 = 1 \ , \tag{0.1}$$

where D is assumed to be a positive nonsquare integer. Lenstra [21] and Williams [31] have written excellent expository articles on the history and modern methods for solving this equation; see also Jacobson and Williams [19] for a more exhaustive treatment. The infrastructure plays an important role in this story, so we begin our discussion here.

One of the simplest methods for solving an instance of the Pell equation is based on continued fraction expansions of quadratic irrationals. Following [19, Ch. 3, p. 57], define $\delta = (q + \sqrt{D})/r$, where

$$r = \begin{cases} 2 & \text{if } D \equiv 0 \text{ or } 1 \pmod 4, \\ 1 & \text{otherwise} \end{cases}$$

and

$$q = \begin{cases} 1 & \text{if } D \equiv 1 \pmod 4, \\ 0 & \text{otherwise}. \end{cases}$$

Put $P_0 = q, Q_0 = r, q_0 = \lfloor \delta \rfloor, G_{-1} = r, G_0 = rq_0 - q$, $B_{-1} = 0$, $B_0 = 1$, and apply the recurrences

$$\begin{aligned} P_{i+1} &= q_i Q_i - P_i \,, \\ Q_{i+1} &= \frac{D - P_{i+1}^2}{Q_i} \,, \\ q_{i+1} &= \left\lfloor \frac{P_{i+1} + \sqrt{D}}{Q_{i+1}} \right\rfloor , \\ G_{i+1} &= q_{i+1} G_i + G_{i-1} \,, \\ B_{i+1} &= q_{i+1} B_i + B_{i-1} \,, \end{aligned}$$

for $i = 0, 1, 2, \ldots$ until we find the least positive index p for which $Q_p = r$.

For example, consider $D = 193$. Applying the formulas from above, the continued fraction expansion of $(q+\sqrt{D})/r = (1+\sqrt{193})/2$ is obtained as follows:

M. J. Jacobson Jr. is associate professor of computer science at the University of Calgary. His email address is `jacobs@cpsc.ucalgary.ca`.

R. Scheidler is professor of mathematics and computer science at the University of Calgary. Her email address is `rscheidl@ucalgary.ca`.

DOI: http://dx.doi.org/10.1090/noti1064

n	P_n	Q_n	q_n	G_n	B_n
-1				2	0
0	1	2	7	13	1
1	13	12	2	28	2
2	11	6	4	125	9
3	13	4	6	778	56
4	11	18	1	903	65
5	7	8	2	2584	186
6	9	14	1	3487	251
7	5	12	1	6071	437
8	7	12	1	9558	688
9	5	14	1	15629	1125
10	9	8	2	40816	2938
11	7	18	1	56445	4063
12	11	4	6	379486	27316
13	13	6	4	1574389	113327
14	11	12	2	3528264	253970
15	13	2			

We see that $Q_{15} = 2$. Set $p = 15$, and notice that r divides both G_{p-1} and B_{p-1}. We find that

$$(G_{p-1}/r)^2 - D(B_{p-1}/r)^2 = 1764132^2 - 193 \cdot 126985^2 = -1.$$

Hence, $(1764132, 126985)$ is a solution of the negative Pell equation $x^2 - Dy^2 = -1$ for $D = 193$. Continuing the continued fraction expansion of $(1 + \sqrt{193})/2$ yields a repetition of the cycle of (P_n, Q_n) pairs. Stopping at the second occurrence of $Q_n = 2$ yields

n	P_n	Q_n	q_n	G_n	B_n
29	11	12	2	12448646853698	896073208080
30	13	2			

Setting $p = 30$, we find again that r divides both G_{p-1} and B_{p-1}. This time, we obtain the identity

$$(G_{p-1}/r)^2 - D(B_{p-1}/r)^2$$
$$= 6224323426849^2 - 193 \cdot 448036604040^2 = 1.$$

Thus, $(t, u) = (6224323426849, 448036604040)$ is a solution of $x^2 - 193y^2 = 1$. It can be proved that the first solution obtained in this manner is the fundamental (i.e., minimal) solution of (0.1) [19, Ch. 3], and all solutions (T, U) can be obtained by computing $T + U\sqrt{D} = (t + u\sqrt{D})^n$ for integers $n \geq 1$ [19, Corollary 1.10].

In addition to the agricultural connection to fields through Archimedes' cattle problem, further examination of the continued fraction method for solving Pell equations reveals another connection to fields, specifically real quadratic number fields. Consider the quantity

$$\epsilon_D = \frac{G_{p-1} + B_{p-1}\sqrt{D}}{r} = \frac{3528264 + 253970\sqrt{193}}{2}$$

obtained after the first period in the continued fraction expansion from the example. It belongs to the real quadratic field $\mathbb{Q}(\sqrt{D}) = \{a + b\sqrt{D} \mid a, b \in \mathbb{Q}\}$. The *norm* of an element $\alpha = a + b\sqrt{D} \in \mathbb{Q}(\sqrt{D})$ is the rational number $\mathcal{N}(\alpha) = \alpha\overline{\alpha} = a^2 - b^2D$, where $\overline{\alpha} = a - b\sqrt{D}$ denotes the *conjugate* of α. For the quantity ϵ_D from our example, we obtain

$$\mathcal{N}(\epsilon_D) = \frac{G_{p-1} + B_{p-1}\sqrt{D}}{2} \frac{G_{p-1} - B_{p-1}\sqrt{D}}{2} = -1 \ .$$

Because $\mathcal{N}(\epsilon_D)$ is an integer, ε_D is an algebraic integer in $\mathbb{Q}(\sqrt{D})$, and because it divides 1, ε_D is a nontrivial unit in the subring $\mathcal{O}_D = \{a + b\delta \mid a, b \in \mathbb{Z}\}$ of $\mathbb{Q}(\sqrt{D})$. In fact, ε_D is the smallest unit exceeding 1 in $\mathcal{O}_D$ and is called the *fundamental unit* of $\mathcal{O}_D$, as all units in this ring can be expressed[1] as $\pm\varepsilon_D^k$ for some $k \in \mathbb{Z}$.

The fundamental solution to the Pell equation is the smallest power of ε_D with integer coefficients and norm equal to 1. In the example, one can verify that

$$\varepsilon_{193}^2 = 6224323426849 + 448036604040\sqrt{193}$$

and, as shown above,

$$\mathcal{N}(\varepsilon_{193}^2)$$
$$= 6224323426849^2 - 193 \cdot 448036604040^2 = 1,$$

so $(6224323426849, 448036604040)$ is the fundamental solution of the Pell equation for $D = 193$. In general, ε_D^k for some $k \in \{1, 2, 3, 6\}$ yields the fundamental solution of the Pell equation. Thus computing the fundamental unit of $\mathcal{O}_D$ also yields all solutions to (0.1).

There are further connections to $\mathbb{Q}(\sqrt{D})$. Consider the quantities $\theta_j = (G_{j-2} + B_{j-2}\sqrt{D})/2$ for $j = 1, 2, \ldots, p-1$. Like ε_D, they all belong to $\mathcal{O}_D$. The set of principal $\mathcal{O}_D$-ideals $(\theta_j) = \theta_j\mathcal{O}_D$ forms the *infrastructure* of $\mathbb{Q}(\sqrt{D})$. Its remarkable group-like properties led to faster algorithms for solving the Pell equation, and other applications. These properties, and additional applications of the infrastructure, are the focus of this paper.

We begin our discussion of the infrastructure with a brief study of ideals in the ring of integers of a real quadratic field. The infrastructure is formally introduced, and its group-like properties explained, in the section "What Is the Infrastructure?". Further uses (and nonuses) of the infrastructure are discussed in the section "What Else Can (and Can't!) You Do with Infrastructure?", and practical implementation issues arising in these applications are the subject of the section "Practical Issues". We offer a glimpse beyond infrastructure in real quadratic fields in the section "Beyond Infrastructure in Real Quadratic Fields" and into relevant open problems in the section "Open Problems", concluding with suggested sources for further reading in the section "Further Reading".

[1] *This is a special case of Dirichlet's unit theorem for an arbitrary algebraic number field.*

What Is the Infrastructure?

The infrastructure is a collection of certain $\mathcal{O}_D$-ideals of a real quadratic field $\mathbb{Q}(\sqrt{D})$. For simplicity, assume henceforth that D is squarefree, and $q = r - 1$ (with q and r as defined in the section "Cows and Fields"). Then $\mathcal{O}_D$ is the *maximal order* or *ring of integers* of $\mathbb{Q}(\sqrt{D})$, i.e., the integral closure of $\mathbb{Z}$ in $\mathbb{Q}(\sqrt{D})$. As seen earlier, $\mathcal{O}_D$ is precisely the two-dimensional $\mathbb{Z}$-module in $\mathbb{Q}(\sqrt{D})$ with basis $\{1, (r-1+\sqrt{D})/r\}$. The nonzero ideals of $\mathcal{O}_D$ are exactly the two-dimensional $\mathbb{Z}$-submodules of $\mathcal{O}_D$ with bases of the form $\{SQ/r, S(P+\sqrt{D})/r\}$, where $S, Q, P \in \mathbb{Z}$, r divides Q, and rQ divides $D - P^2$. An $\mathcal{O}_D$-ideal $\mathfrak{a}$ is *primitive* if it cannot be written as an integer multiple of another $\mathcal{O}_D$-ideal; we can take $S = 1$ in this case, and write $\mathfrak{a} = [Q, P]$. The conjugate of an ideal $\mathfrak{a}$ is the ideal $\overline{\mathfrak{a}}$ containing the conjugates of the elements in $\mathfrak{a}$. If $\mathfrak{a} = [Q, P]$, then $\overline{\mathfrak{a}} = [Q, -P]$.

The product of two $\mathcal{O}_D$-ideals $\mathfrak{a}, \mathfrak{b}$ is defined to be the collection of all finite sums of products $\alpha\beta$ with $\alpha \in \mathfrak{a}$ and $\beta \in \mathfrak{b}$. This is easily seen to be an $\mathcal{O}_D$-ideal, so the set of all $\mathcal{O}_D$-ideals forms an infinite monoid under multiplication with identity $\mathcal{O}_D$. Efficient formulas for computing a $\mathbb{Z}$-basis of the product of two ideals given in $\mathbb{Z}$-basis representation exist, based on the composition formulas of Gauß for binary quadratic forms [19, Sec. 5.4]. A *principal* ideal consists of all the $\mathcal{O}_D$-multiples of some fixed element $\theta \in \mathcal{O}_D$, called a *generator* of the ideal, and is denoted by (θ). The principal ideals form an infinite submonoid of the ideals.

The *(ideal) class group* Cl_D, discovered by Gauß in the context of binary quadratic forms [14], is an algebraic object of great interest in its own right. It is defined as the group of nonzero ideal equivalence classes under multiplication, where $\mathfrak{a}$ and $\mathfrak{b}$ are *equivalent* if there exist nonzero $\alpha, \beta \in \mathcal{O}_D$ such that $(\alpha)\mathfrak{a} = (\beta)\mathfrak{b}$. The class group Cl_D is a finite abelian group whose order is called the *(ideal) class number* and is denoted by h_D. One way to prove that Cl_D is finite is through the definition of reduced ideals. A primitive ideal $\mathfrak{a} = [Q, P]$ is said to be *reduced* if Q is a minimum in $\mathfrak{a}$, i.e., if there exists no nonzero $\alpha \in \mathfrak{a}$ such that $|\alpha| < Q$ and $|\overline{\alpha}| < Q$. Given any primitive ideal $\mathfrak{a} = [Q, P]$ of $\mathcal{O}_D$, an equivalent reduced ideal can be found by expanding the continued fraction of $(P + \sqrt{D})/Q$ [19, Ch. 5]. Furthermore, it can be shown that if $\mathfrak{a}$ is reduced, then the coefficients Q and P are bounded (roughly) by $\sqrt{D}$ [19, Sec. 5.1], implying that there are only finitely many reduced ideals in $\mathbb{Q}(\sqrt{D})$. Combining this with the fact that every ideal is equivalent to a reduced ideal yields the finiteness of the ideal class group.

Ideal reduction also leads to a method for computing in the class group. The idea is to use a reduced ideal to represent its entire equivalence class. Multiplying elements in the class group then consists of multiplying a reduced representative from each class and reducing the product. This operation is efficient, with bit complexity polynomial in $\log D$ [19, Sec. 5.4].

The main problem with this method is testing whether two ideal classes are equal. Although the number of reduced representatives of an equivalence class is finite, it could still be too large for an exhaustive comparison to be feasible. In particular, as we will see shortly, the *regulator* $R_D = \log \varepsilon_D$ provides an estimate of the number of reduced ideals in an equivalence class. It is known that $h_D R_D \approx \sqrt{D}$ [19, Eqn. 7.1]. So if h_D is small, one would have to exhaustively test $O(\sqrt{D})$ reduced equivalent ideals.

In order to speed up this process, it would be helpful if elements of the class group had their own internal structure. This is precisely what Shanks discovered in 1972 and termed *infrastructure* [26]. Although each ideal equivalence class has its own infrastructure, we confine our discussion to the class of principal ideals—the identity class of Cl_D—and refer to "the" infrastructure of $\mathbb{Q}(\sqrt{D})$ as the set of all reduced principal $\mathcal{O}_D$-ideals.

The ideal $\mathcal{O}_D = [r, r-1]$ is principal and reduced. Computing the continued fraction expansion of $(P_0 + \sqrt{D})/Q_0$ with $Q_0 = r$ and $P_0 = r - 1$, using the formulas from the previous section, yields a sequence of ideals $\mathfrak{a}_j = [Q_{j-1}, P_{j-1}]$ for $j = 1, 2, \ldots$. One can show that all these ideals are reduced and equivalent, satisfying

$$\mathfrak{a}_{j+1} = (\psi_j)\mathfrak{a}_j \tag{0.2}$$

where $\psi_j = (P_j + \sqrt{D})/Q_{j-1}$ [19, Sec. 5.3].

Notice the similarity of this process to that of the previous section for solving the Pell equation using continued fractions. In fact, the sequence of reduced principal ideals obtained in this fashion consists precisely of the ideals (θ_j) with $\theta_j = (G_{j-2} + B_{j-2}\sqrt{D})/r$; i.e., $\mathfrak{a}_j = (\theta_j)$. Thus

$$\theta_j = \prod_{k=1}^{j-1} \psi_k$$

which can be derived from (0.2).

One way to describe the structure of the infrastructure is to draw an analogy to a more familiar algebraic object, namely, a finite multiplicative cyclic group $G = \langle g \rangle$ of order n. To generate all the elements in this group, one begins with the identity element $1 = g^0$ and successively multiplies by g until finding $g^n = 1$. The elements of G can be visualized on a circle of circumference n, as depicted in Figure 1. Notice that the elements of G are regularly spaced around the circle. The *distance* of an element g^i is defined to be i and measures how far around the circle g^i is from 1

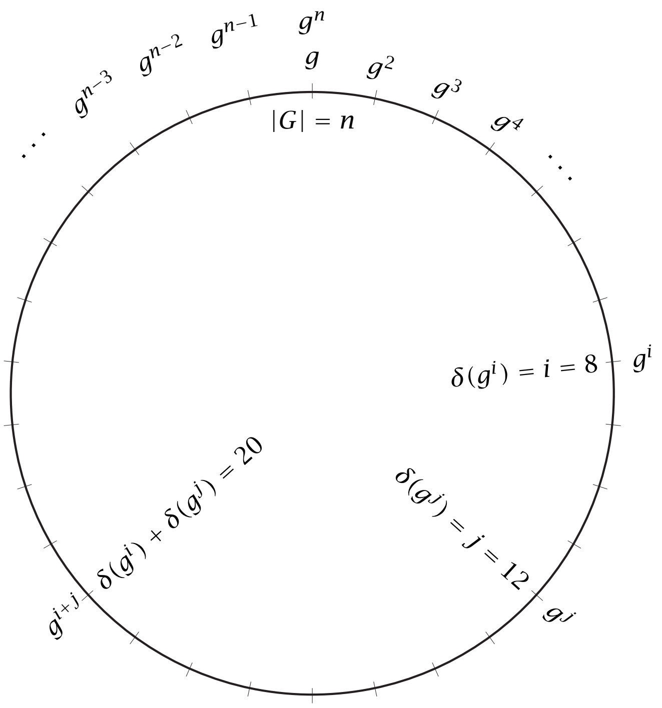

Figure 1. Cyclic group G of order n generated by g, depicted on a circle. Note that consecutive elements (powers of g) are distance 1 apart on the circle. The circumference of the circle is equal to n, the order of G.

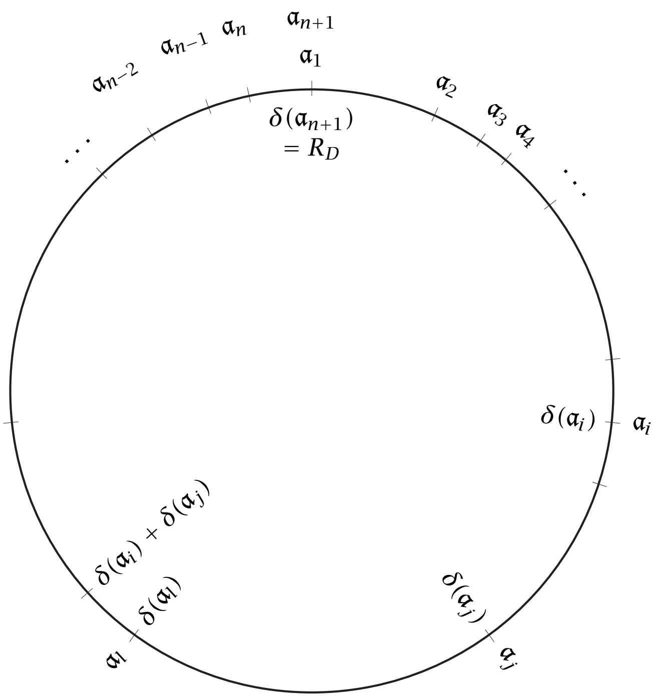

Figure 2. The infrastructure of the principal class of $\mathcal{O}_D$, depicted on a circle. Note that the distance between consecutive elements varies, but is generally close to 1. The circumference of the circle is equal to the regulator R_D.

(in clockwise direction). Using this definition, the distance between two consecutive group elements on the circle is 1, and the distance around the entire circle is n, the order of G.

The infrastructure, as we have seen, also describes a finite cyclic structure. To generate all the elements in the infrastructure, one starts with the ideal $\mathfrak{a}_1 = \mathcal{O}_D = [Q_0, P_0]$ (recall that $Q_0 = r$ and $P_0 = r - 1$) and computes the ideals $\mathfrak{a}_i = [Q_i, P_i]$, $i = 2, 3, \ldots$, by computing the continued fraction expansion of $(P_0 + \sqrt{D})/Q_0$, as described above, until arriving at $\mathcal{O}_D$ again. Thus one step in this continued fraction expansion plays a similar role to multiplication by g in G. The ideals $\mathfrak{a}_i$ can also be visualized on a circle, as depicted in Figure 2 (see [19, Fig. 7.1]; figure used with kind permission of Springer Science+Business Media). Here the *distance* of $\mathfrak{a}_i = (\theta_i)$ is defined to be $\delta(\mathfrak{a}_i) = \log \theta_i$. This notion of distance measures how far $\mathfrak{a}_i$ is around the circle from $\mathcal{O}_D$. Using this definition, the distance between two consecutive ideals $\mathfrak{a}_{i+1}$ and $\mathfrak{a}_i$ is $\delta(\mathfrak{a}_{i+1}, \mathfrak{a}_i) = \log(\theta_{i+1}/\theta_i) = \log \psi_i$. Although the distance between consecutive infrastructure ideals is therefore not constant, the Khintchine-Lévy Theorem implies that $\log \psi_i$ is on average about 1.18 [19, Theorem 3.17]. In other words, even though the ideals are not evenly spaced around the circle—unlike the cyclic group scenario—the distance between any two neighbors is, nevertheless, generally close to one. The distance around the entire circle, analogous to the order of G, is $R = \log \varepsilon_D$, the regulator of $\mathbb{Q}(\sqrt{D})$.

As an example, the infrastructure of $\mathbb{Q}(\sqrt{193})$ is listed in Table 1 (reproduced from [19, Table 7.1], table used with kind permission of Springer Science+Business Media). The coefficients of the ideals are taken from the continued fraction expansion of $(1 + \sqrt{193})/2$ from the previous section. The distances are computed using the corresponding G_j and B_j values. Notice that $\mathfrak{a}_{16} = \mathfrak{a}_1$, so $\delta(\mathfrak{a}_{16}) \approx 15.07631652$ is the regulator of $\mathbb{Q}(\sqrt{193})$. Notice also that $\delta(\mathfrak{a}_j) \approx j$ in all cases, as one would expect from the Khintchine-Lévy Theorem.

The analogy comparing the infrastructure to a cyclic group shows that the idea of computing the regulator[2] of $\mathcal{O}_D$ using the continued fraction expansion of $\sqrt{D}$ is similar in spirit to computing the order of a cyclic group by enumerating the entire group. The complexity of the latter is exactly n operations in G, and, similarly, the Khintchine-Lévy law implies that the complexity of the former is $O(R)$, which is $O(D^{1/2+\epsilon})$ in

[2]*Here, "computing" the regulator R_D means computing a sufficiently accurate floating point approximation.*

Table 1. Principal Cycle for $D = 193$

j	$\mathfrak{a}_j = [Q_{j-1}, P_{j-1}]$	$\delta(\mathfrak{a}_j)$
1	$[2,1]$	0
2	$[12,13]$	2.59869817
3	$[6,11]$	3.32835583
4	$[4,13]$	4.82844171
5	$[18,11]$	6.65671165
6	$[8,7]$	6.80572746
7	$[14,9]$	7.85709282
8	$[12,5]$	8.15679754
9	$[12,7]$	8.71127845
10	$[14,5]$	9.16513386
11	$[8,9]$	9.65688343
12	$[18,7]$	10.61682945
13	$[4,11]$	10.94102199
14	$[6,13]$	12.84657298
15	$[12,11]$	14.26937782
16	$[2,13]$	15.07631652

the worst case (when $h_D = 1$). However, Shanks's discovery of the infrastructure revealed even more structural similarities to G, which led to significant improvements in regulator computation.

In the cyclic group setting, one well-known improvement to computing the group order is to apply a time-memory tradeoff and use the *baby-step giant-step strategy.* The idea is that, in addition to moving through the group using successive multiplications by g (baby steps), one can take larger steps through the group by multiplying by g^i for some positive integer i (giant steps). As depicted in Figure 1, the distance between two successive baby steps is only 1, but each giant step advances by a distance of i, and in particular the distance of $g^j g^i$ is precisely $j + i$, the sum of the distances of the inputs. To compute the order n of G, one begins by computing a list of baby steps $\mathcal{L} = \{1, g, g^2, \dots, g^l\}$ where $l \approx \sqrt{H}$ for a suitable bound $H \geq n$. Then the giant steps $g^{2l}, g^{3l}, \dots$ are computed until an element is found in the list $\mathcal{L}$ of baby step elements, say $g^{il} = g^j$. This yields $g^{il} g^{-j} = 1$, and we have $n = il - j$ if H is a reasonably tight bound on n. The complexity of this algorithm is $O(\sqrt{H}) = O(\sqrt{n})$ operations in G. Modifications exist that achieve complexity $O(\sqrt{n})$ even without knowing a bound on n [5, 29].

Shanks's breakthrough was the discovery that there exists a similar notion of a giant step in the infrastructure. Consider the effect of multiplying an infrastructure ideal $\mathfrak{a}_j$ by $\mathfrak{a}_i$. The resulting infrastructure ideal $\mathfrak{a}$ is obtained by multiplying $\mathfrak{a}_j$ and $\mathfrak{a}_i$ and reducing their product, yielding $(\gamma)\mathfrak{a} = \mathfrak{a}_j\mathfrak{a}_i$ for some $\gamma \in \mathbb{Q}(\sqrt{D})$. The principal ideal factor (γ) arises from the fact that, in general, the reduced ideal $\mathfrak{a}$ is not equal to $\mathfrak{a}_j\mathfrak{a}_i$ but merely equivalent to it.

Denote by $\mathfrak{a}_j * \mathfrak{a}_i$ the first reduced ideal $\mathfrak{a} = (\gamma^{-1})\mathfrak{a}_j\mathfrak{a}_i$ obtained by applying reduction to the product ideal $\mathfrak{a}_j\mathfrak{a}_i$. As $\mathfrak{a}_j$ and $\mathfrak{a}_i$ are both principal, the ideal $\mathfrak{a}_j * \mathfrak{a}_i$ is also principal, and as it is reduced, it lies somewhere on the infrastructure. The question is, what is its distance? As $\mathfrak{a}_j = (\theta_j)$ and $\mathfrak{a}_i = (\theta_i)$, we obtain

$$\begin{aligned}\delta(\mathfrak{a}_j * \mathfrak{a}_i) &= \log(\theta_j\theta_i/\gamma)\\ &= \log\theta_j + \log\theta_i - \log\gamma\\ &= \delta(\mathfrak{a}_j) + \delta(\mathfrak{a}_i) - \log(\gamma).\end{aligned}$$

It can be shown that $-\log 2 \leq \log\gamma \leq \log\Delta$ [19, Eqn. 5.38], and hence $\delta(\mathfrak{a}_j*\mathfrak{a}_i) \approx \delta(\mathfrak{a}_j)+\delta(\mathfrak{a}_i)$ as the "error" $\log\gamma$ in distance is small in comparison to the circumference $R_D \in O(\sqrt{D})$ and, in general, the ideal distances $\delta(\mathfrak{a}_j)$ and $\delta(\mathfrak{a}_i)$. Thus, as illustrated in Figure 2, multiplication by $\mathfrak{a}_i$ advances the distance by approximately $\delta(\mathfrak{a}_i)$ and has a similar effect as a giant step in the cyclic group setting. The main difference is that in G a giant step using g^i advances the distance by exactly i, i.e., distances in G are exactly additive, whereas in the infrastructure case the advance in distance by applying a giant step with ideal $\mathfrak{a}_i$ is only approximately i.

Shanks's discovery imposes a fascinating structure on the cycle of reduced principal ideals that gives the infrastructure its name and is tantalizingly close to a cyclic group structure. Interestingly, among all the requirements on an abelian group, the infrastructure violates merely associativity—and only just barely. Closure holds (the result of the multiplication/reduction of two infrastructure ideals is another infrastructure ideal), and the operation is easily seen to be commutative. There exists an identity element (namely $\mathcal{O}_D$, as $\mathfrak{a} * \mathcal{O}_D = \mathfrak{a}\mathcal{O}_D = \mathfrak{a}$), and it is possible to define inverses (roughly, the inverse of the infrastructure ideal $\mathfrak{a} = [Q, P]$ is the conjugate ideal $\overline{\mathfrak{a}}$ of distance $R_D - \delta(\mathfrak{a}) + \log\mathcal{N}(\mathfrak{a})$ with $\mathcal{N}(\mathfrak{a}) = Q/r$). The fact that distances are not exactly additive in the infrastructure prevents it from being associative: the infrastructure ideals $(\mathfrak{a} * \mathfrak{b}) * \mathfrak{c}$ and $\mathfrak{a} * (\mathfrak{b} * \mathfrak{c})$ both have distance close to $\delta(\mathfrak{a}) + \delta(\mathfrak{b}) + \delta(\mathfrak{c})$ but need not be equal.

Nevertheless, using this "giant step" operation in the infrastructure leads to a straightforward adaptation of baby-step giant-step to compute the regulator R_D. The baby-step list consists of the first l ideals obtained by applying baby steps (using the continued fraction algorithm), and the giant steps consist of successively multiplying by an ideal $\mathfrak{a}_l$ with distance close to l and reducing the product. After some adjustments to account for the fact that distances are not exactly additive [19, Sec. 5.3], $R_D = \delta(\mathfrak{a}_i) - \delta(\mathfrak{a}_j)$ is obtained once a giant-step ideal $\mathfrak{a}_i$ is found that is equal to one of the baby-step ideals $\mathfrak{a}_j$. Using the bound of $O(D^{1/2+\epsilon})$ for R_D, this algorithm has complexity $O(D^{1/4+\epsilon})$. As with order computation, there exists a variation that does not depend on an upper bound on R_D and requires only $O(\sqrt{R_D})$ infrastructure operations [7, 8]. The overall bit complexity of this algorithm is $O(\sqrt{R_D}D^{\epsilon})$, as each of the infrastructure operations has complexity polynomial in $\log D$. This is currently the fastest deterministic algorithm that unconditionally computes R_D.

To see how this algorithm works in practice, consider the previous example of $D = 193$ ([19, Example 7.15]). By expanding the continued fraction of $(1+\sqrt{193})/2$, essentially enumerating the entire infrastructure by baby steps, we found that $\varepsilon_{193} = 1764132 + 126985\sqrt{D}$, and thus $R_{193} = \log \varepsilon_{193} \approx 15.07631652$. Instead, suppose that the first six infrastructure ideals, taken from Table 1, form our baby-step list,

$$\mathcal{L} = \{\mathfrak{a}_1, \mathfrak{a}_2, \ldots, \mathfrak{a}_6\},$$

and that the ideal $\mathfrak{a}_4 = [4, 13]$ with distance $\delta(\mathfrak{a}_4) \approx 4.82844171$ is used for the giant steps. Successively multiply $\mathfrak{a}_4$ by itself and apply subsequent reduction, thereby obtaining the sequence of giant-step ideals:

$$\begin{aligned}
\mathfrak{b}_2 &= \mathfrak{a}_4 * \mathfrak{a}_4 = [8, 9] = \mathfrak{a}_{11}, & \delta(\mathfrak{b}_2) &\approx 9.65688343\ , \\
\mathfrak{b}_3 &= \mathfrak{b}_2 * \mathfrak{a}_4 = [12, 11] = \mathfrak{a}_{15}, & \delta(\mathfrak{b}_3) &\approx 14.26937782\ , \\
\mathfrak{b}_4 &= \mathfrak{b}_3 * \mathfrak{a}_4 = [6, 11] = \mathfrak{a}_{3}, & \delta(\mathfrak{b}_4) &\approx 18.40467235\ .
\end{aligned}$$

We find that $\mathfrak{b}_4 \in \mathcal{L}$ with $\mathfrak{b}_4 = \mathfrak{a}_3$, so

$$\begin{aligned}
&R_{193} \\
&= \delta(\mathfrak{b}_4) - \delta(\mathfrak{a}_3) \approx 18.40467235 - 3.32835583 \\
&= 15.07631652,
\end{aligned}$$

as required.

What Else Can (and Can't!) You Do with Infrastructure?

The analogy between a finite cyclic group and the infrastructure can be extended further, in the sense that a number of additional useful algorithms and techniques from the group setting can be adapted to solving problems in the infrastructure. For example, a more general version of the problem of computing the order of g is the *discrete logarithm problem*, computing the smallest power of g equal to a given group element a. The result represents the "distance" from the identity element to a. The infrastructure analogy is the *distance problem*: given an infrastructure ideal $\mathfrak{a}$, find its distance $\delta(\mathfrak{a})$ from $\mathcal{O}_D$. Via compact representations, which are discussed in more detail at the end of this section, this problem can be shown to be polynomially equivalent to the *principal ideal problem*: finding a generator of $\mathfrak{a}$. As in the cyclic group case, the distance problem can be solved using the baby-step giant-step algorithm in $O(\sqrt{\delta(\mathfrak{a})})$ operations in the infrastructure. Testing whether two ideals $\mathfrak{a}$ and $\mathfrak{b}$ are equivalent can also be done with this method by testing whether $\mathfrak{a}\bar{\mathfrak{b}}$ is principal.

A fundamental algorithm in the cyclic group setting is fast exponentiation, by which g^n, the group element of "distance" n from 1, can be computed rapidly (in $O(\log n)$ group operations) by combining squarings and multiplications by g according to the binary expansion of n. In the infrastructure, the analogue of this algorithm allows one to compute efficiently the ideal whose distance is closest to n. By combining giant steps (in this case, ideal squaring followed by reduction) with baby steps, such an ideal can be computed in $O(\log n)$ infrastructure operations.

As in the cyclic group setting, the basic operation of finding an element with distance close to n enables several additional applications and algorithms. One such application, common to both settings, is public-key cryptography. For example, in a cyclic group, Diffie and Hellman [11] showed how two parties can agree on a shared secret over an insecure communication channel by exchanging g^a and g^b (where a and b are generated randomly by the respective parties and kept secret) and each independently computing the common group element $g^{ab} = (g^a)^b = (g^b)^a$. An analogous infrastructure key exchange protocol was proposed in 1989 by Buchmann and Williams [9] using the computation of closest ideals in the infrastructure in place of exponentiation. Roughly speaking, the protocol participants exchange infrastructure ideals close to a and b and independently compute the infrastructure ideal close to ab. The result is a protocol whose security is related to the principal ideal problem in a similar way to how the Diffie-Hellman scheme is related to the discrete logarithm problem. This protocol is noteworthy in that it was the first public-key cryptosystem proposed whose security was based on a structure that is not a group. Other protocols, as well as improvements to the original cryptosystem, have also been proposed; a survey can be found in [19, Ch. 14].

The ability to compute closest ideals also leads to a simple method for testing whether a given real number S is a good numerical approximation of an integer multiple of R_D. Once again, the idea stems from the cyclic group setting, where S is a multiple of the order of a generator g if and only if g^S is the group identity. Similarly, in the infrastructure scenario, if S is a multiple of the regulator R_D, then the closest ideal to S has to be $\mathcal{O}_D$, because R_D is the distance around the entire infrastructure, i.e., from $\mathcal{O}_D$ to itself. Thus, one simply computes the ideal closest to S and checks whether it is $\mathcal{O}_D$. This observation also enables accurate approximations of R_D to be computed efficiently given only a coarse approximation: one simply computes the ideal closest to the approximation and computes the resulting distance to the required precision.

An application of testing for multiples of the regulator using closest ideals is an improvement, due to Lenstra [20], of the baby-step giant-step algorithm for computing the regulator R_D. After computing an approximation H to $h_D R_D$, which can be done analytically [19, Sec. 10.2], the ideal $\mathfrak{a}$ with distance closest to H is computed. Baby-step giant-step is then used to search for the ideal closest to $\mathfrak{a}$ whose distance is a multiple of the regulator; the number of infrastructure operations required is thus asymptotic to $\sqrt{|H - h_D R_D|}$, the square root of the error in the approximation. Given this multiple of the regulator, a combination of baby-step giant-step (to check whether the regulator is larger than a given bound) and closest ideal computations (to find smaller multiples of R_D) is used to find R_D. By balancing the time required for the various parts of the algorithm, an overall complexity of $O(D^{1/5+\epsilon})$ is achieved. This algorithm is unconditionally correct, but the runtime applies only if the Extended Riemann Hypothesis (ERH) holds, as it is required to guarantee the accuracy of the approximation H of $h_D R_D$.

There are some techniques from the cyclic group setting that unfortunately do not translate well to the infrastructure. One such technique is computing the order of the group given a multiple of the group order. This can be done in polynomial time given the prime factorization of the order multiple m by finding, for each prime divisor p, the largest integer i for which m/p^i is also a multiple of the order. This technique does not generalize to the infrastructure, as there is no known way to "factor" multiples of the regulator, which are real numbers.

The best-known method to solve the problem of finding R_D given an integer multiple S of R_D is the technique used in Lenstra's $O(D^{1/5+\epsilon})$ algorithm, namely, using baby-step giant-step to check whether the regulator is larger than a certain bound and checking whether S/p is also a multiple of R_D by testing whether the ideal closest to S/p is equal to $\mathcal{O}_D$ for all primes p less than another suitable bound. When used in the context of Lenstra's algorithm, where a regulator multiple must also be computed, optimization results in the $O(D^{1/5+\epsilon})$ runtime. When the regulator multiple S is given as input, reoptimization results in an algorithm with complexity $O(S^{1/3}D^{\epsilon})$ for unconditionally computing R_D from S.

An interesting consequence of this algorithm is that it leads to an efficient practical algorithm for unconditionally computing R_D. Buchmann's index-calculus algorithm [4] computes R_D in time subexponential in $\log D$, but both the runtime and correctness of this algorithm depend on the ERH. Using the previous algorithm to verify unconditionally that this output is in fact the regulator results in an overall algorithm [15, 10] that computes R_D unconditionally in expected time $O(D^{1/6+\varepsilon})$, as the index-calculus algorithm computes R_D in expected time $O(D^{\epsilon})$ and $R_D \in O(D^{1/2+\epsilon})$. Although this algorithm is not deterministic, due to its dependence on the index-calculus algorithm to produce a multiple of R_D, it is the fastest known algorithm that computes an unconditionally correct approximation of R_D.

Two more techniques from the cyclic group setting that do not work well in the infrastructure are the Pollard-rho algorithm [24] for computing the order n of a group element g and the Pohlig-Hellman algorithm [23] for discrete logarithms. Pollard's algorithm uses a random walk in the group and a method to detect cycles in this walk. The result is a probabilistic algorithm with expected runtime $O(\sqrt{n})$ that requires only a constant amount of storage, unlike baby-step giant-step, which requires $O(\sqrt{n})$ storage. The random walk and cycle finding algorithm can certainly be applied in the infrastructure. The problem is that, by virtue of the probabilistic nature of the algorithm, the results produced are not minimal; that is, the algorithm produces an integer multiple of R_D which need not be equal to R_D. In the case of a cyclic group, the output can be factored and minimized as described above. However, as there is no appropriate notion of factorization for distances (which are real numbers), one would again be forced to apply a minimization strategy similar to that described above for multiples of the regulator. In addition, as the distances of the ideals involved will grow to be much larger than R_D, the precision requirements to ensure numerical accuracy are almost certainly too unwieldy to make this approach practical.

The Pohlig-Hellman algorithm solves the discrete logarithm problem by reducing it to discrete logarithm problems in order p subgroups for all primes p that divide the order n of G. Thus this algorithm is very effective when n is smooth, i.e., has only small prime factors. Adapting this algorithm to the infrastructure seems fruitless, as there is no corresponding notion of smoothness for the regulator.

One additional computational issue that arises in the infrastructure is the representation of the fundamental unit itself, as well as principal ideal generators. When computing only the regulator or ideal distances, the sizes of operands encountered are generally manageable, although, as discussed in the next section, the numerical accuracy or precision of their floating point approximations becomes problematic. The alternative, explicitly working with the principal ideal generators as opposed to their logarithms, is tempting in that problems with approximations are eliminated, but the size of the operands is too large. For example, the size of the fundamental unit (i.e., the regulator) is $O(D^{1/2+\epsilon})$, so it cannot even be written in time polynomial in $\log D$.

Fortunately, this problem can also be solved using arithmetic in the infrastructure. The idea is to use a type of binary exponentiation to come up with a representation of principal ideal generators, including the fundamental unit, in the form of a power-product of elements of $\mathbb{Q}(\sqrt{D})$. An expression of this form is called a *compact representation* [6], [19, Ch. 12]. The compact representation of an element $\theta \in \mathbb{Q}(\sqrt{D})$ can be computed in time polynomial in $\log\log|\theta| \log D$ given an approximation of $\log|\theta|$. The basic arithmetic operations (including testing equality, multiplication, division, norm, computing $x \mod m$ and $y \mod m$ where $\theta = x + y\sqrt{D}$) can be performed on compact representations in polynomial time [18]. Most importantly, compact representations have *size* polynomial in $\log\log|\theta| \log D$, which motivates their name. The result is that the algorithms for computing R_D mentioned above can also compute a compact representation of the fundamental unit, as well as the fundamental solution of Pell's equation, without changing the asymptotic runtime. In addition, computations requiring explicit manipulation of fundamental units and solutions of Pell's equation, for example, as part of the resolution of other Diophantine equations [18], can be performed even for large values of D.

All of the applications mentioned in this section can, in principle, be implemented relatively easily. However, there are a number of practical concerns which are described in the next section.

Practical Issues

One of the difficulties arising in infrastructure arithmetic stems from the fact that distances are real numbers and thus need to be approximated to sufficient accuracy in any computer implementation. To that end, any (potentially unknown) infrastructure ideal is represented by a known infrastructure ideal and an approximation of the relative distance between the two ideals. More explicitly, let $p \in \mathbb{N}$ be a prespecified precision parameter. For any $f \in \mathbb{N}$ with $f < 2^p$, a *near reduced (f,p)-representation* of a reduced ideal $\mathfrak{a}$ is a triple $(\mathfrak{b}, d, k)$ consisting of another reduced ideal $\mathfrak{b}$ and two integers $d > 0$ and $k < 0$ [16]. Here $\mathfrak{b} = (\theta)\mathfrak{a}$ for some unknown relative generator $\theta \in \mathbb{Q}(\sqrt{D})$ that is of magnitude about 2^k, and d is a $(p+1)$-bit integer that approximates the most significant $p+1$ bits of θ with relative error $2^{-p}f$. In practice, it is possible to keep the relative errors significantly below one. Moreover, $|k|$ and θ tend to be small, and, in fact, the ideals in any two near-reduced (f,p)-representations of the same reduced ideal can be shown to be at most five baby steps apart in the infrastructure.

Ideal arithmetic is now conducted on (f,p)-representations. If $\mathfrak{a}$ is taken to be $\mathcal{O}_D$, then each (f,p)-representation consists of an infrastructure ideal and an approximation of its principal ideal generator. Arithmetic in the infrastructure can then be performed using arithmetic on these representations. However, such arithmetic introduces error propagation, so care must be taken in choosing the precision p large enough to keep the relative errors sufficiently small. For example, in the key agreement protocol mentioned in the section "What Else Can (and Can't!) You Do with Infrastructure?", suppose that the two communicants use secret exponents of bit size n. In order to guarantee that the two parties arrive at the same key ideal after execution of the protocol, i.e., after their respective double exponentiations, one needs to ensure that $p \geq 2n + \log_2(50n)$ [16].

The computation of giant steps presents another computational nuisance. As mentioned earlier, the standard giant-step algorithm first multiplies two reduced ideals and subsequently reduces the primitive part of the product ideal. This technique has two disadvantages. First, in the reduction procedure, it is very costly to compute all the basis coefficients Q_j, P_j of the (unnecessary) intermediate ideals via the continued fraction algorithm. Second, multiplying two reduced ideals results in an ideal whose basis coefficients are roughly twice as large as those of the two input ideals; the reduction process gradually shrinks the coefficients back down to original size.

In the 1980s Shanks began to compute class numbers of imaginary quadratic fields with his programmable hand-held calculator. Frustrated by the fact that the large size of the intermediate results arising in ideal multiplication exceeded the display capacity of his calculator, he developed a significantly more efficient ideal multiplication and reduction algorithm to which he assigned the Fortran designator NUCOMP (short for "new composition")[3] [27].

NUCOMP, in essence, stops the multiplication process before completion and applies a type of intermediate reduction before generating the reduced ideal product. Recall that traditional giant-step computation applies the continued-fraction algorithm to the quadratic irrational

[3]*Shanks described his work using the language of quadratic forms, in which composition of forms is the analogue of ideal multiplication.*

$(P+\sqrt{D})/Q$ arising from the product ideal $\mathfrak{a} = [Q, P]$ whose coefficients are of order D until a reduced ideal is obtained. Shanks's idea was to avoid computing the nonreduced product ideal $\mathfrak{a}$ altogether and replace the quadratic irrational $\alpha = (P+\sqrt{D})/Q$ by a suitable *rational* approximation of α whose numerator and denominator are of magnitude $\sqrt{D}$, i.e., significantly smaller. Then the continued-fraction algorithm is replaced by the more efficient extended Euclidean algorithm, and the partial quotients obtained in both methods will be identical, up to a certain point in the continued fraction expansion. Since the basis coefficients of any of the intermediate ideals can be recovered through simple formulas involving the convergents of the rational approximation, the costly computation of all the intermediate ideals is avoided.

But at which point in the Euclidean algorithm is the reduced target ideal—or at least an "almost" reduced ideal—reached? In collaboration with Atkin, Shanks answered this question as well. Once the remainders have decreased to approximately $\sqrt[4]{D}$, the corresponding ideal is recovered. This ideal is frequently reduced, and is at most two continued fraction steps away from being reduced. Moreover, this computation only involves integers of magnitude $\sqrt{D}$ or less (with a few exceptions that may be as large as $D^{3/4}$).

Shanks originally formulated his NUCOMP method for imaginary quadratic fields, but it was extended to real quadratic fields by van der Poorten [30]. Numerical experiments [17], [16] showed that NUCOMP can be up to 50 percent faster than ordinary ideal multiplication and reduction.

Beyond Infrastructure in Real Quadratic Fields

While the ideal class group of an imaginary quadratic field is a very large finite abelian group, its counterpart in real quadratic fields tends to be a rather dull object, namely, a very small group that is frequently trivial. Instead, surprisingly, the principal class—which is, after all, merely the identity element of this group—exhibits an interesting, almost grouplike structure. As described above, the cycle of reduced principal ideals constitutes the infrastructure. This raises the natural question of whether infrastructures occur in other settings. Indeed, this is the case.

The closest analogue to real quadratic number fields is real quadratic (or *hyperelliptic*) *function fields*. Informally speaking, to obtain such a field, $\mathbb{Q}$ is simply replaced by a rational function field $\mathbb{F}_q(x)$ of odd characteristic.[4] Then the square root of a polynomial $D(x) \in \mathbb{F}_q[x]$ of even degree and square leading coefficient is adjoined to $\mathbb{F}_q(x)$. Ideals are now represented by a pair of polynomials in $\mathbb{F}_q[x]$ instead of a pair of integers, and for reduced ideals, these polynomials have degree less than $\deg(D)/2$. Arithmetic on ideals is completely analogous to real quadratic field arithmetic, and, once again, every ideal class has an associated infrastructure [28]. The main difference between real quadratic function fields and number fields is that, in the former, distances are *integers*. Loosely speaking, distances are degrees of quadratic irrational functions, as opposed to logarithms of quadratic irrational numbers. In particular, ideals are discretely and relatively evenly spaced around the infrastructure, and a baby step always produces an advance in distance of at least one (and almost always exactly one for large base fields $\mathbb{F}_q$). The result is a cleaner and simpler ideal arithmetic, avoiding the need for numerical approximations as discussed in the previous section. Moreover, interestingly, integer distances make it possible to provide an explicit and effective embedding of the principal infrastructure into a finite cyclic group whose order is the regulator, as was discovered independently by Fontein [12] and Mireles Morales [22]. Such an embedding into a finite cyclic group does not exist in the number field setting, although embedding the infrastructure into an *infinite* cyclic group is possible (see [20]).

Beyond quadratic extensions, any *global field*, i.e., any number field or function field over a finite field, of unit rank one exhibits an infrastructure analogous to the one described above. Number fields of unit rank one can have degree at most four over $\mathbb{Q}$; specifically, they include real quadratic, complex cubic, and totally complex quartic fields. In contrast, function fields of unit rank one can have arbitrarily high extension degree over $\mathbb{F}_q(x)$. Moreover, unit groups of higher rank result in infrastructures that are essentially higher-dimensional tori. In his 1987 *Habilitationsschrift* [3], Buchmann established that in fact every number field has an infrastructure and presented the first generic algorithm for computing the class group and regulator of an arbitrary number field. Twenty-one years later, Schoof in his excellent 2008 treatise [25] provided a modern treatment of the general number field setting, using *Arakelov divisor* theory—the number field analogue to divisor theory of function fields—as a natural setting for the infrastructure phenomenon and Buchmann's algorithm. Most recently, the number field and function field scenarios were finally combined by Fontein [13], who gave a unified and completely

[4]*Hyperelliptic function fields can also be defined over finite fields of characteristic 2, but as their representation is somewhat more complicated, we will not consider them here.*

general description of the infrastructure of any global field, including baby steps and giant steps, the relationship of infrastructure arithmetic to arithmetic in the divisor class group, and (for the function field setting) an algorithm for computing a system of fundamental units and the regulator of the field.

Open Problems

Although Shanks's discovery of the infrastructure of a real quadratic field has already led to new applications and improved algorithms in that setting, research into this rich field is by no means exhausted. There are still a number of interesting open problems.

A noticeable gap persists between what is possible for computing the regulator R_D of a quadratic number field unconditionally and deterministically and what seems possible in practice. The fastest deterministic, unconditional algorithm has complexity $O(\sqrt{R_D}D^\epsilon)$, or $O(D^{1/4+\epsilon})$ using the fact that $R_D \in O(D^{1/2+\epsilon})$. Assuming the ERH only for the complexity analysis reduces this to $O(D^{1/5+\epsilon})$. Allowing nondeterminism (i.e., randomization) yields $O(D^{1/6+\epsilon})$. Finally, allowing nondeterminism and the assumption of the ERH for the correctness of the output reduces the complexity to subexponential time $O(\exp(\sqrt{\log D \log\log D}))$. Are further improvements possible?

Work continues to improve the performance of arithmetic in the infrastructure. The adaptation of Shanks's NUCOMP technique from imaginary to real quadratic fields represents notable progress in this direction, and further improvements are certainly possible. The development of (f,p)-representations to maintain accurate approximations of distances has helped by reducing the precision requirements; perhaps this can be improved further.

Finally, arithmetic in general infrastructures is still in its infancy. In particular, very little work—algorithmic and especially computational—has been done for infrastructures of global fields with degree larger than 3. The works of Schoof and Fontein provide two different settings with which to conceptualize these infrastructures. Which one will turn out to be more advantageous in terms of concrete algorithms and implementations?

It is the authors' hope that this article has given the reader a glimpse into the infrastructure and its many interesting applications and generalizations. There is still much ongoing work in this area and even more left to be done. Readers are encouraged to join the research into this interesting area.

Further Reading

As this article was written in an expository manner, many details and proofs were deliberately omitted. For a thorough treatment of infrastructure in real quadratic fields, including many of the associated algorithms and applications, we recommend [19]. The book of Buchmann and Vollmer [8] treats much of this material in the language of binary quadratic forms and is also highly recommended. For more information about infrastructure in functions fields and higher-dimensional analogues, we refer the reader to the references in the section "Beyond Infrastructure in Real Quadratic Fields".

Acknowledgment

The authors thank Sarah Chisholm, Monireh Rezai Rad and Peter Zvengrowski for their careful proofreading and suggestions for improvement of this article.

References

1. A. Amthor, Das Problema Bovinum des Archimedes, *Zeitschrift für Math. u. Physik* (Hist. Litt. Abtheilung) **25** (1880), 153–171.
2. Archimedes, *The Cattle Problem*, in English verse by S. J. P. Hillion and H. W. Lenstra Jr., Mercator, Santpoort, 1999.
3. J. Buchmann, *Zur Komplexität der Berechnung von Einheiten und Klassenzahlen algebraischer Zahlkörper*, Habilitationsschrift, Düsseldorf, 1987.
4. ______, A subexponential algorithm for the determination of class groups and regulators of algebraic number fields, *Séminaire de Théorie des Nombres* (Paris), 1988–89, pp. 27–41, Prazr. Math., vol. 91, Birkhäuser Boston, 1990.
5. J. Buchmann, M. J. Jacobson Jr., and E. Teske, On some computational problems in finite abelian groups, *Mathematics of Computation* **66** (1997), no. 220, 1663–1687.
6. J. Buchmann, C. Thiel, and H. C. Williams, Short representation of quadratic integers, *Computational Algebra and Number Theory, Mathematics and its Applications* vol. 325, Kluwer, Dordrecht, 1995, pp. 159–185.
7. J. Buchmann and U. Vollmer, A Terr algorithm for computations in the infrastructure of real-quadratic number fields, *Journal de Théorie des Nombres de Bordeaux* **18** (2006), no. 3, 559–572.
8. ______, *Binary Quadratic Forms: An Algorithmic Approach*, Algorithms and Computation in Mathematics, vol. 20, Springer-Verlag, Berlin, 2007.
9. J. Buchmann and H. C. Williams, A key-exchange system based on real quadratic fields, CRYPTO '89, Lecture Notes in Computer Science, vol. 435, Springer, New York, 1989, pp. 335–343.
10. R. de Haan, M. J. Jacobson Jr., and H. C. Williams, A fast, rigorous technique for computing the regulator of a real quadratic field, *Mathematics of Computation* **76** (2007), no. 260, 2139–2160.
11. W. Diffie and M. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory* **22** (1976), 472–492.
12. F. Fontein, Groups from cyclic infrastructures and Pohlig-Hellman in certain infrastructures, *Advances in*

Mathematics of Communications **2** (2008), no. 3, 293–307.

13. ______, The infrastructure of a global field of arbitrary unit rank, *Mathematics of Computation* **80** (2011), no. 276, 2325–2357.
14. C. F. Gauss, *Disquisitiones Arithmeticae*, Springer Verlag, 1986, English ed.: translated by A. A. Clark.
15. M. J. Jacobson Jr., Á. Pintér, and P. G. Walsh, A computational approach for solving $y^2 = 1^k + 2^k + \cdots + x^k$, *Mathematics of Computation* **72** (2003), no. 244, 2099–2110.
16. M. J. Jacobson Jr., R. Scheidler, and H. C. Williams, An improved real quadratic field based key exchange procedure, *Journal of Cryptology* **19** (2006), 211–239.
17. M. J. Jacobson Jr. and A. J. van der Poorten, *Computational aspects of NUCOMP*, Algorithmic Number Theory—ANTS-V (Sydney, Australia), Lecture Notes in Computer Science, vol. 2369, Springer-Verlag, Berlin, 2002, pp. 120–133.
18. M. J. Jacobson Jr. and H. C. Williams, Modular arithmetic on elements of small norm in quadratic fields, *Designs, Codes and Cryptography* **27** (2002), no. 1–2, 93–110.
19. ______, *Solving the Pell Equation*, CMS Books in Mathematics, Springer, New York, 2009.
20. H. W. Lenstra Jr., On the calculation of regulators and class numbers of quadratic fields, London Mathematical Society Lecture Note Series, vol. 56, Cambridge University Press, 1982, pp. 123–150.
21. ______, Solving the Pell equation, *Notices of the AMS* **49** (2002), no. 2, 182–192.
22. D. J. Mireles Morales, *An analysis of the infrastructure in real function fields*, E-print archive no. 2008/299, 2008.
23. S. C. Pohlig and M. E. Hellman, An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance, *IEEE Transactions on Information Theory* **24** (1978), 106–110.
24. J. M. Pollard, Theorems on factorization and primality testing, *Proceedings of the Cambridge Philosophical Society* **76** (1974), 521–528.
25. R. J. Schoof, Computing Arakelov class groups. *Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography*, Math. Sci. Res. Inst. Publ., vol. 44, Cambridge Univ. Press, Cambridge, 2008, pp. 447–495.
26. D. Shanks, The infrastructure of real quadratic fields and its applications, *Proceedings of the 1972 Number Theory Conference*, Univ. Colorado, Boulder, Colorado, 1972, pp. 217–224.
27. ______, On Gauss and composition. I, II, *Proceedings of the NATO ASI on Number Theory and Applications* (R. A. Mollin, ed.), Kluwer Academic Press, 1989, pp. 163–179.
28. A. Stein, Explicit infrastructure for real quadratic function fields and real hyperelliptic curves, *Glasnik Matematicki* **44** (2009), no. 1, 89–126.
29. D. C. Terr, A modification of Shanks' baby-step giant-step algorithm, *Mathematics of Computation* **69** (2000), no. 230, 767–773.
30. A. J. van der Poorten, A note on NUCOMP, *Mathematics of Computation* **72** (2003), no. 244, 1935–1946.
31. H. C. Williams, Solving the Pell equation, *Proceedings of the Millennial Conference on Number Theory*, A K Peters, Natick, MA, 2002, pp. 397–435.