INFRASTRUCTURE, ARITHMETIC, AND CLASS NUMBER COMPUTATIONS IN PURELY CUBIC FUNCTION FIELDS OF CHARACTERISTIC AT LEAST 5

BY

ERIC LANDQUIST

B.S., Virginia Tech, 1998
M.S., Virginia Tech, 2000

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mathematics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2009

Urbana, Illinois

Doctoral Committee:

Professor Alexandru Zaharesu, Chair
Professor Renate Scheidler, Director of Research, University of Calgary
Associate Professor Iwan Duursma
Emeritus Professor Leon McCulloh
Emeritus Professor Stephen Ullom

Abstract

One of the more difficult and central problems in computational algebraic number theory is the computation of certain invariants of a field and its maximal order. In this thesis, we consider this problem where the field in question is a purely cubic function field, $K/\mathbb{F}_q(x)$, with $\mathrm{char}(K) \geq 5$. In addition, we will give a divisor-theoretic treatment of the infrastructures of K, including a description of its arithmetic, and develop arithmetic on the ideals of the maximal order, $\mathcal{O}$, of K.

Historically, the infrastructure, $\mathcal{R}_{\mathbf{C}}$, of an ideal class, $\mathbf{C} \in Cl(\mathcal{O})$ has been defined as a set of reduced ideals in $\mathbf{C}$. However, we extend work of Paulus and Rück [PR99] and Jacobson, Scheidler, and Stein [JSS07b] to define $\mathcal{R}_{\mathbf{C}}$ as a certain subset of the divisor class group, $\mathcal{J}_K$, of a cubic function field, K, specifically, the subset of *distinguished* divisors whose classes map to $\mathbf{C}$ via $\mathcal{J}_K \to Cl(\mathcal{O})$. Our definition of distinguished generalizes the same notion by Bauer for purely cubic function fields of unit rank 0 [Bau04] to those of unit rank 1 and 2 as well. Further, we prove a bijection between $\mathcal{R}_{\mathbf{C}}$, as a set of distinguished divisors, and the infrastructure of $\mathbf{C}$ defined by "reduced" ideals, as in [Sch00, SS00, Sch01, LSY03, Sch04]. We describe the arithmetic on $\mathcal{R}_{\mathbf{C}}$, providing new results on the baby step and giant step operations and generalizing notions of the inverse of a divisor in $\mathcal{R}_{[\mathcal{O}]}$ from quadratic infrastructures in [JSS07b] to cubic infrastructures. We also give algorithms to compute the various operations.

For the infrastructure arithmetic, as well as for computing in $Cl(\mathcal{O})$, we derive ideal arithmetic for any purely cubic function field, K, with $\mathrm{char}(K) \neq 2$, generalizing work of Scheidler [Sch01] and Bauer [Bau04]. In addition, we show how to determine the unique distinguished ideal in a given ideal class in the case that K has unit rank 0, extending results of Bauer [Bau04] from cubic function fields defined by a non-singular curve to those defined by a singular curve as well. For the ideal arithmetic and reduction methods, we provide algorithms as well.

Finally, we describe methods to compute the divisor class number, h, of K, and in the case that $\mathcal{O}$ has unit rank 1 or 2, the regulator and ideal class number of $\mathcal{O}$ as well. A method of Scheidler and Stein [SS07, SS08] determines sharper upper and lower bounds on h, for a given cubic function field, than those given by the Hasse-Weil Theorem. We then employ Shanks' Baby Step-Giant Step algorithm [Sha71] and Pollard's Kangaroo method [Pol78], to search this interval and compute the desired invariants for purely cubic function fields of unit rank 0 and 1. The total complexity of the method to compute these invariants is $O\left(q^{(2g-1)/5+\varepsilon(g)}\right)$ ideal operations as $q \to \infty$, where $0 \leq \varepsilon(g) \leq 1/5$. With this approach, we computed the 28 decimal digit divisor class numbers of two purely cubic function fields of genus 3: one of unit rank 0 and one of unit rank 1. We also computed the 25 decimal digit divisor class numbers of two purely cubic function fields of genus 4: one of unit rank 0 and one of unit rank 1. In the unit rank 1 examples, we factored the divisor class numbers

into the ideal class numbers and the respective 26 and 24 decimal digit S-regulators. We believe that these are the largest divisor class numbers ever computed for a cubic function field of genus at least 4 and the largest regulators ever computed for any cubic function field, respectively.

To Bethany.

Acknowledgments

Many thanks are in order for all the people who have helped make this thesis possible and have supported me along the way. First and foremost, thanks to my advisor, Renate Scheidler, for all her time and effort she spent carefully reviewing this thesis and her numerous suggestions for improvement. She has certainly helped me become a better mathematician and writer. In addition, my thanks go to her and Mark Bauer for introducing me to the topics contained in this thesis. They have been fun problems to work on. Next, thanks go to Andreas Stein for introducing me to the researchers in the Centre for Information Security and Cryptography (CISaC) at the University of Calgary, including Professor Scheidler, and his constant support for this work. Much of the computational work of Chapter 6 was done under the guidance and support of Professor Stein during a stay at the University of Wyoming, and this thesis was completed with his support at Carl von Ossietzky Universität Oldenburg in Germany. I am thankful for the opportunity to continue this and related work as his Wissenschaftlicher Mitarbeiter in Oldenburg and am indebted for the time he has invested in me and the many opportunities he has provided that have helped me develop as a mathematician.

I thank the rest of my committee at the University of Illinois, Iwan Duursma, Leon McCulloh, Stephen Ullom, and Alexandru Zaharescu, for their review of my work, suggestions, and the time that they have invested in me to help me become a better mathematician. In addition, thanks to the rest of the faculty and staff at the University of Illinois for their support.

I also thank Hugh Williams and the members of the CISaC group for their support and hospitality during two stays in Calgary, as well as their continued friendship. The initial work of this thesis was supported during the latter of the two visits. Thanks go to my friends at CISaC for many discussions on the topics of function fields and infrastructure, especially, Mike Jacobson, along with the aforementioned professors, Felix Fontein, Pieter Rozenhart, Andrew Shallue, Adrian Tang, Jonathan Webster, and Qingquan Wu. Additional thanks go to Felix Fontein for making suggestions to Chapter 5.

Thanks go to Professors Bauer, Scheidler, and Stein, along with Hameeduz Zaman of the University of Calgary and Christopher Yarrish of Harrisburg Area Community College for providing some of the code and other related materials used for the computational results in Chapter 6. In addition, thanks go to Mark Wrubleski of the University of Calgary and Jonathan Manton and Tony Mullen of the University of Illinois for providing me with computational resources and support.

Thanks to Stephen Galbraith and David Mireles-Morales for answering questions about their work and discussions on infrastructures.

Thanks to my friends and family, including my new friends at Carl von Ossietzky Universität

Oldenburg, for their constant support, encouragement, and prayers. Most of all, extra special thanks go to my wife and soul-mate, Bethany, whose daily unconditional love, support, encouragement, and prayers is more than I could ever ask for or deserve.

Finally, greatest thanks and highest praise to my Savior, Yahushua (Jesus in Hebrew), without whom I am nothing, and who provides perfect and eternal security, more than any cryptosystem.

Table of Contents

List of Notation

k	A perfect field in Chapters 1 and 2.
$\mathbb{Z}$	The ring of integers.
$\mathbb{N}$	The set of positive integers.
$\mathbb{N}_0$	$\mathbb{N} \cup \{0\}$
$\mathbb{Q}$	The field of rational numbers.
$\mathbb{R}$	The field of real numbers.
$\mathbb{C}$	The field of complex numbers.
$\mathbb{F}_q$	The finite field of q elements.
x, Y	Indeterminants.
$k[x]$	The ring of polynomials with coefficients in k.
$k(x)$	$\{f(x)/g(x) \mid f(x), g(x) \in k[x],\ g(x) \neq 0\}$
K^*	The multiplicative group, $K \setminus \{0\}$, of a field K.
$[K : k(x)]$	The extension degree of K over $k(x)$.
$\overline{k}$	An algebraic closure of k.
$\mathrm{char}(K)$	The characteristic of a field, K.
$\mathcal{O} = \mathcal{O}_K$	The maximal order (ring of integers) of a field, K.
$\mathcal{O}^*$	The group of units of $\mathcal{O}$.
r	The rank of $\mathcal{O}^*$.
$Cl(\mathcal{O})$	The ideal class group of $\mathcal{O}$ (or of K).
h_x	The ideal class number, $\lvert Cl(\mathcal{O})\rvert$.
g	The genus of a function field K.
$\mathcal{D} = \mathcal{D}(K)$	The group of divisors of a function field K.
$\mathcal{D}_0 = \mathcal{D}_0(K)$	The subgroup of $\mathcal{D}$ of degree 0 divisors of K.
$\mathcal{P} = \mathcal{P}(K)$	The subgroup of $\mathcal{D}_0$ of principal divisors of K.
$\mathcal{J} = \mathcal{J}_K$	The divisor class group, $\mathcal{D}_0/\mathcal{P}$, of K.

h	The divisor class number, $\lvert\mathcal{J}_K\rvert$, of K.
S	The set of infinite places of K: $\{\infty_0, \ldots, \infty_r\}$.
D^S	The infinite part of a divisor, $D \in \mathcal{D}$: the maximal subsum of D with support contained in S.
D_S	The finite part of a divisor $D \in \mathcal{D}$: $D - D^S$.
$\mathcal{D}_0^S$	$\{D \in \mathcal{D}_0 \mid \operatorname{supp}(D) \subseteq S\}$
$\mathcal{P}^S$	$\mathcal{D}_0^S \cap \mathcal{P}$
R^S	The S-regulator of K: $\lvert\mathcal{D}_0^S/\mathcal{P}^S\rvert$.
R_x	The x-regulator of K: $\left(R^S \prod_{i=0}^{r} \deg(\infty_i)\right) / \gcd(\deg(\infty_0), \ldots, \deg(\infty_r))$.
$O(f(x))$	$g(x) = O(f(x))$ as $x \to \infty$ if and only if there exist positive $c, x_0 \in \mathbb{R}$ such that $0 \leq g(x) \leq cf(x)$ for all $x > x_0$.
$\lfloor n \rfloor$	$\max\{m \in \mathbb{Z} \mid m \leq n\}$
$\lceil n \rceil$	$\min\{m \in \mathbb{Z} \mid m \geq n\}$
$[n]$	$\lfloor n + 1/2 \rfloor$
$a \mid b$	a divides b.
$a \nmid b$	a does not divide b.
$a^n \Vert b$	$a^n \mid b$, but $a^{n+1} \nmid b$.

Chapter 1

Introduction

One of the more difficult and central problems in computational algebraic number theory is the computation of certain invariants of a field and its associated maximal order. This thesis considers this problem for the case of cubic function fields. In particular, we will give a description of the infrastructure of a cubic function field and its arithmetic, arithmetic on ideals, and the computation of the divisor class number and regulator. In this introduction, we will give a short explanation of these ideas. We will begin with a brief history of algebraic curves and other topics considered in the content of this thesis, including the motivation for our work on cubic function fields. We will then describe how this thesis fits into the body of published literature on this topic. Finally, we will give an outline of the thesis, noting the main results that we will prove.

1.1 Motivation and Background

The study of algebraic curves has a long and varied history. Conic sections (circles, ellipses, parabolas, and hyperbolas) are curves of the form $C : a_1Y^2 + (a_2x + a_3)Y + a_4x^2 + a_5x + a_6 = 0$, where $a_i \in k$, for some field k, and $1 \leq i \leq 6$, and were studied in ancient Greece, beginning with Menaechmus in the 4th century, BC. Though Menaechmus did not express these curves as an equation, he used parabolas and hyperbolas to solve the classical problem of doubling the cube. Further work in this area continued with Aristaeus (the Elder). Euclid followed shortly thereafter and is most famous for his *Elements*, which gives a rigorous development of classical geometry and is the most popular and influential textbook ever written. Archimedes (c. 287 BC-c. 212 BC), applied many concepts from conics to various engineering tasks for his home city of Syracuse and proved several theoretical results as well [Boy91].

During the Renaissance, conics were extensively studied for application to astronomy, optics, and gravitation. Key figures include Kepler (1571-1630), who formulated laws of planetary motion empirically, and Newton (1643-1727), whose *Philosophiae Naturalis Principia Mathematica* contains the statement of his laws of motion and gravitation, as well as a geometric derivation of Kepler's laws.

Further, Newton was the first to study *cubic plane curves*, which are of the form $C : a_1Y^3 + (a_2x + a_3)Y^2 + (a_4x^2 + a_5x + a_6)Y + a_7x^3 + a_8x^2 + a_9x + a_{10} = 0$, with $a_i \in k$, for $1 \leq i \leq 10$, giving a characterization of such curves in [Har10]. A special class of cubic curves are *elliptic curves*, which may be written in the *(long) Weierstrass form* $C : Y^2 + (a_1x + a_3)Y = x^3 + a_2x^2 + a_4x + a_6$, where a_1, a_2, a_3, a_4, $a_6 \in k$ and there is no point for which both partial derivatives simultaneously vanish. Elliptic curves arise from elliptic integrals, which are connected with the problem of computing the

arc length of an ellipse, hence the name.

The study of elliptic curves is extensive, but we note a few important developments and generalizations that relate to the historical background of this thesis. In 1936 Hasse showed that the number points, (a, b), with $a, b \in \mathbb{F}_q$, on an elliptic curve with coefficients in $k = \mathbb{F}_q$, is in the interval $\left(\left(\sqrt{q}-1\right)^2, \left(\sqrt{q}+1\right)^2\right)$ [Has36]. Furthermore, the set of such points, together with a point "at infinity," form a group via the familiar chord-tangent method.

A natural generalization of elliptic curves are *hyperelliptic curves*, which are of the form $C : Y^2 + h(x)Y = f(x)$, with $h(x), f(x) \in \mathbb{F}_q[x]$, $\deg(h) \leq g$, and $\deg(f) = 2g+1, 2g+2$, where $g \in \mathbb{N}_0$ is the *genus* of the curve. Though hyperelliptic curves do not possess the same group law as elliptic curves, arithmetic may be performed in the *divisor class group* (or the *Picard group*) of C, denoted $\mathcal{J}_C$. The famous theorem of Weil generalizes Hasse's Theorem and states that $\left(\sqrt{q}-1\right)^{2g} < |\mathcal{J}_C| < \left(\sqrt{q}+1\right)^{2g}$ [Wei48]. In addition, one can consider the *function field* of C. Specifically, if y is such that $y^2 + h(x)y = f(x)$, then the function field $K = \mathbb{F}_q(C) = \mathbb{F}_q(x, y)$ is a quadratic extension of $\mathbb{F}_q(x)$ and is called a *quadratic* or *hyperelliptic* function field. Given this relationship between a function field and its defining curve, we often write $\mathcal{J}_K = \mathcal{J}_C$. Arithmetic in $\mathcal{J}_K$, for any hyperelliptic function field K, was first derived by Cantor [Can87], though much faster arithmetic has since been developed by considering curves of a fixed genus (see [CF06] for several examples). Extensions of $\mathbb{F}_q(x)$ of higher degrees are possible via other types of curves. In particular, a *cubic* function field is an extension, $K/\mathbb{F}_q(x)$, of degree 3.

Current interest in function fields is fueled by applications to cryptography and coding theory. Though the first such application of function fields was to error-correcting codes by Goppa [Gop81] in 1981, this area of research intensified since the foundational work of Koblitz and Miller, applying elliptic curves defined over finite fields of large characteristic to public key cryptography [Mil86, Kob87]. The popularity of elliptic curve cryptography increased rapidly because of the fast arithmetic and strong security. Koblitz proposed further schemes using the divisor class group of a hyperelliptic curve in [Kob89]. If $g = 2$, then hyperelliptic cryptosystems are just as secure as elliptic cryptosystems and have smaller key sizes, but current implementations of the arithmetic are not quite as fast.[1] Additional cryptographic schemes have been proposed in what are called real hyperelliptic function fields by Scheidler, Stein, and Williams [SSW96] and Jacobson, Scheidler, and Stein [JSS07b].

Cryptography in cubic function fields, however, cannot compete with cryptography in quadratic function fields for several reasons. First, the arithmetic is slower. Second, there are vulnerabilities which make them less secure [Die06, DT08]. Finally, any cryptographic protocol based on function fields must use a divisor class group whose order, $h = h_K$, called the *divisor class number*, is known and divisible by a sufficiently large prime. However, this is a difficult problem if K is a cubic function field of large characteristic. The best results to date focus on function fields defined by Picard curves, which are of the form $C : Y^3 = f(x)$, where $f(x) \in \mathbb{F}_q[x]$, $\deg(f(x)) = 4$, and $f(x)$ is monic and square-free. In [BTW05], Bauer, Teske, and Weng computed the 55-digit divisor class number (with a 52-digit prime factor) of such a cubic function field, and Weng [Wen06] improved on their method to compute a prime 53-digit divisor class number. In contrast, very fast point-counting algorithms exist for elliptic curves of small characteristic [Sat00] and large characteristic [Sch95]. If K is a hyperelliptic function field, there are efficient methods to compute h if $\mathrm{char}(K)$

[1] Cryptosystems based on hyperelliptic curves of genus $g \geq 3$ are less secure due to an attack in [GTTD07].

is small [Ked01, Ked03, DV06b], or if $g = 2$ [GH00]. For function fields of sufficiently large genus, index calculus methods are effective [Thé03, Die06, GTTD07, DT08]. For other cases, the problem becomes very difficult. This leads us to our main motivation for studying cubic function fields.

To establish some more historical context, we will note some similarities between function fields and *number fields*, finite algebraic extensions of $\mathbb{Q}$; collectively, number fields and function fields are called *global* fields. If K is a global field, then its maximal order (or ring of integers) is denoted $\mathcal{O} = \mathcal{O}(K)$. If $\mathcal{I}(\mathcal{O})$ is the group of ideals of $\mathcal{O}$ and $\mathcal{P}(\mathcal{O})$ the subgroup of principal ideals, then $Cl(\mathcal{O}) = \mathcal{I}(\mathcal{O})/\mathcal{P}(\mathcal{O})$ (or $Cl(K)$) is the *(ideal) class group* of K and its order, denoted h_x (or $h(K)$), is called the *(ideal) class number*. If K is a function field, then for the cases we will be concerned with, $Cl(\mathcal{O})$ is isomorphic to a subgroup of $\mathcal{J}_K$, so $h_x \mid h$. The problem of computing the class number of a number field originated with Gauss, who discovered the ideal class group, describing it using the language of quadratic forms in his monumental work, *Disquisitiones Arithmeticae*, written in 1798. This problem remains today, as Cohen writes, "The determination of the structure of $Cl(K)$ and in particular of the class number $h(K)$ is one of the main problems in algorithmic algebraic number theory." ([Coh93], pg. 208) Though this statement was written in the context of number fields, it is certainly true for function fields as well, and is one of the main motivations of the research of this thesis.

At this point, it is important to note that we can classify global fields into two main cases, loosely called *real* and *imaginary*. Approaches to computing the class number, and other important related invariants, will differ in both situations. If K is any global field, then the rank of the unit group, $\mathcal{O}^*$, of $\mathcal{O} = \mathcal{O}_K$ is denoted r. K is called *imaginary* or *real* depending whether $r = 0$ or $r > 0$, respectively. These terms are motivated by quadratic number fields, such as $\mathbb{Q}\left(\sqrt{-2}\right)$ and $\mathbb{Q}\left(\sqrt{2}\right)$, which have unit rank 0 and 1, respectively. We first consider imaginary fields. In this case, h_x will be large. In particular, for imaginary function fields in the cases we will be concerned with, the ideal class group, $Cl(\mathcal{O}) \cong \mathcal{J}_K$ so that $h_x = h$. Moreover, Gauss showed that every ideal class of an imaginary quadratic number field contains a unique representative, called a *reduced* ideal. Analogous notions hold in imaginary function fields as well. In this way, it is easy to operate directly in $Cl(\mathcal{O})$, using methods such as Shanks' Baby Step-Giant Step method [Sha71] and Pollard's Rho [Pol75] and Kangaroo algorithms [Pol78] to search for h_x. Our approach to compute h for imaginary cubic function fields will use these methods. On the other hand, if $r > 0$, then (at least in every case that has been studied) h_x is generally very small, and another invariant called the *regulator*, which is defined in terms of any generating set of $\mathcal{O}^*$, is generally large. If K is a number field of positive unit rank, then its regulator, $R \in \mathbb{R}$. For the function fields under our consideration, we have $h = Rh_x$, so that $R \in \mathbb{N}$. (In contrast to the number field case, the regulator depends on the set, S, of the infinite places of K and is hence referred to as the S-regulator, R^S.) Instead of using $Cl(\mathcal{O})$, real global fields possess an additional structure that will aid us in the computation of R, and in the function field case, h as well.

In 1972, Shanks [Sha72] discovered that the principal ideal class (the identity element of $Cl(\mathcal{O})$) of a real quadratic number field does not contain a unique reduced ideal. Moreover, he found that the set of reduced principal ideals form a cycle under a "baby step" operation and that they possess a group-like structure (lacking associativity) under the "giant step," or composition, operation. He called this set the *infrastructure* of the field; the infrastructure of K is commonly denoted $\mathcal{R} = \mathcal{R}(K)$. By traversing $\mathcal{R}$, Shanks was able to compute an approximation of the regulator. Improvements to

Shanks' method and further descriptions of the infrastructure of a real quadratic number field were given by Lenstra [Len82], Schoof [Sch82], and Williams [Wil85]. In addition, Buchmann and Williams [BW88] showed that every number field of unit rank 1 exhibits an infrastructure. Real quadratic function fields were found to possess an infrastructure by Stein [Ste92], who then computed the regulator of several such function fields via their infrastructures. Regulators of purely cubic function fields of unit rank 1 were computed by Scheidler and Stein in [SS98, SS00, Sch01] and of unit rank 2 by Lee, Scheidler, and Yarrish in [LSY03], essentially by making baby steps in $\mathcal{R}(K)$, for the function field, K, under consideration.

In each of these cases we mentioned, $\mathcal{R}$ was described as a set of reduced ideals (or reduced fractional ideals). However, recent work by Paulus and Rück [PR99], Jacobson, Scheidler, and Stein [JSS07a, JSS07b], Schoof [Sch08], Galbraith, Harrison, and Mireles [GHM08], and Fontein [Fon09] has provided a different perspective on infrastructures by considering them in terms of reduced divisors. This description allows one to understand the structure of infrastructures more naturally and has the additional advantage of giving a very intuitive explanation for the effectiveness of various methods to compute regulators.

Relatively little research has been done on the theory of infrastructures, especially for global fields of unit rank greater than 1. Infrastructures are now known to exist in any global field of positive unit rank, and each ideal class, $\mathbf{C} \in Cl(\mathcal{O})$ has an infrastructure, $\mathcal{R}_{\mathbf{C}}$. Therefore, our desire to compute the class number and regulator of a cubic function field of positive unit rank compels us to learn more about the function field's infrastructures, particularly from a divisor-theoretic perspective.

1.2 How the Thesis Fits into the Literature

In this section, we consider the three main topics of this thesis, the infrastructure of a cubic function field and its arithmetic, the arithmetic of $\mathcal{I}(\mathcal{O})$, and the computation of divisor class numbers and regulators. For each topic, we show how our results compare with, extend, and generalize previous work.

As noted earlier, if K is a global field of positive unit rank, then $\mathcal{R} = \mathcal{R}(K)$ has traditionally been represented as a set of reduced principal fractional ideals, while more recent work has expressed $\mathcal{R}_{\mathbf{C}}$, for any ideal class $\mathbf{C} \in Cl(\mathcal{O})$, as a set of reduced divisors in the function field case [PR99, JSS07a, JSS07b, GHM08, Fon09] and as a set of Arakelov divisors in the number field case [Sch08, Fon09]. If K is a real quadratic function field, then every divisor class of K contains a unique reduced divisor of a particular form, so that $\mathcal{R}$ may be considered as a subset of $\mathcal{J}_K$. This does not hold for cubic function fields, however, so we found an appropriate, more restrictive classification of divisors with which to define $\mathcal{R}_{\mathbf{C}}$. In particular, our notions of i-distinguished and distinguished generalize the definition of a distinguished divisor by Bauer [Bau04] from totally ramified cubic function fields (and analogous notions for totally ramified superelliptic curves by Galbraith, Paulus, and Smart [GPS02]) to any cubic function field having an infinite place of degree 1; we prove that every divisor class in $\mathcal{J}_K$ contains a unique 0-distinguished divisor and that every distinguished divisor is 0-distinguished. We then define the infrastructure of a cubic function field as a set of distinguished divisors, so that $\mathcal{R}_{\mathbf{C}}$ may be considered as a subset of $\mathcal{J}_K$. We further show a correspondence between distinguished divisors and reduced fractional ideals, therefore unifying the divisor-theoretic views of the infrastructure of a cubic function field with the ideal-theoretic constructions as previously

defined in [Sch00, SS00, Sch01, LSY03, Sch04]. Various properties of the infrastructure, in particular the distance measure of an infrastructure, follow naturally from this correspondence and generalize notions proved in [PR99, JSS07b].

The infrastructure of any global field of positive unit rank has two main operations: the baby step and giant step operations. A baby step computes a distinguished divisor close to a given distinguished divisor, in terms of distance, while a giant step is similar to divisor addition. Since the sum of two distinguished divisors is not distinguished in general, the giant step operation requires us to reduce a divisor to an equivalent distinguished divisor close to the sum, in terms of distance. In the case of purely cubic function fields, the baby step operation was defined for unit rank 1 infrastructures in [Sch00, SS00, Sch01, Sch04] and for unit rank 2 infrastructures in [LSY03, Sch04], and the giant step operation, along with a reduction method, for unit rank 1 infrastructures in [Sch01]. Therefore, we generalize the giant step operation and reduction techniques to unit rank 2 infrastructures as well. Two other operations have been defined for hyperelliptic infrastructures: inverses and what is called the divisor below $y \in \mathbb{N}_0$. The divisor below y is the divisor whose distance, δ, is maximal such that $\delta \leq y$. These operations were first defined and used to compute the regulator of a real hyperelliptic function field in [SW99]. Faster algorithms to compute the divisor below y are found in [JSS07a]. We therefore generalize these operations to the infrastructure of a purely cubic function field of unit rank 1 for the purpose of computing R^S for such function fields.

With the exception of the baby step operation, each of the aforementioned infrastructure operations require methods to compute either in $\mathcal{J}_K$ or in $\mathcal{I}(\mathcal{O})$. Some methods to compute in $\mathcal{J}_K$ are described by Hess using Riemann-Roch spaces [Hes99, Hes02] and by Khuri-Makdisi using techniques from linear algebra [KM04, KM07]. In this thesis, however, we will consider arithmetic in $\mathcal{I}(\mathcal{O})$. In [Sch01], Scheidler derived ideal multiplication for the product of two ideals whose norms are relatively prime and for the square of an ideal if K is a purely cubic function fields with $\mathrm{char}(K) \neq 3$. In [Bau04], Bauer developed general ideal multiplication and inversion if $K = \mathbb{F}_q(C)$ is a purely cubic function field, with $\mathrm{char}(K) \neq 3$ and C non-singular, and improved the method of [GPS02] to reduce a given ideal to the unique equivalent distinguished ideal if $r = 0$. Thus, in Chapter 4, we generalize these results to describe ideal multiplication and inversion in any purely cubic function field, K, with $\mathrm{char}(K) \neq 3$, and extend ideal reduction techniques in the case that $r = 0$. That is, we do not require K to be defined by a non-singular curve. Further, we generalize the corresponding algorithms in [Bau04] to compute products, inverses, and distinguished ideals. We also note that the results of Chapter 4 were provided for a slightly different basis than the one we will use, though without proof, in an unpublished set of notes, [Bau05]; therefore, we supply a rigorous treatment to these results.

In Section 1.1, we noted that there are fast methods to compute the divisor class number of elliptic function fields [Sch95, Sat00] and hyperelliptic function fields of small characteristic [Ked01, Ked03, DV06b]. Pila [Pil90, Pil05] generalized the method of Schoof for counting points on elliptic curves over fields of large characteristic to Abelian varieties and Adleman and Huang [AH96, AH01] subsequently improved on Pila's method. A randomized algorithm of Huang and Ierardi [HI98] applies to plane curves. However, none of the generalizations of Schoof's algorithm have been implemented. Kedlaya's algorithm for counting points on hyperelliptic curves of small chacteristic was generalized to Artin-Schreier curves by Denef, Vercauteren, Lauder, and Wan [DV02, LW02, LW04], superelliptic curves by Gaudry, Gürel, and Lauder [GG01, Lau03], C_{ab} curves by Denef and

Vercauteren [DV06a], general nondegenerate curves by Castryck, Denef, and Vercauteren [CDV06], and hyperelliptic curves of medium characteristic by Gaudry and Gürel [GG03]. Each of these methods use p-adic methods.

We will apply the arithmetic of $Cl(\mathcal{O})$ and $\mathcal{R}$, for a purely cubic function field, K, of unit rank 0 and 1, respectively, to the problem of computing $h = h_K$ and R^S if $\mathrm{char}(K)$ is large. In contrast to the work we just cited, the method we will use is based on a method of Scheidler and Stein [SS07, SS08], which first finds an approximation, E, of h and an upper bound, U, on the error, $|h - E|$.[2] Using either Shanks' Baby Step-Giant Step algorithm or Pollard's Kangaroo method, we then search for h in the interval $(E - U, E + U)$. The technique evaluates a truncated Euler product form of the zeta function of K to find the estimates, and sets an upper bound on the infinite tail of this Euler product to obtain the error bounds. This technique was first used for quadratic number fields by Lenstra [Len82] and Schoof [Sch82], extending ideas of Shanks [Sha71, Sha72], and subsequently adapted to quadratic function fields by Stein and Williams [SW99]. Optimizations were made by Stein and Teske [ST02a] by observing that h tends to lie near the center of the interval $(E-U, E+U)$, and additional application via the parallelized Kangaroo method in [ST02b] yielded the 29-digit divisor class number (and regulator) of a real hyperelliptic function field of genus 3. For our computations, we will show how to optimize the Baby Step Giant Step and Kangaroo methods to the cubic function field setting, applying and generalizing results of Stein, Williams, and Teske [SW99, ST02b, ST05]. Further, we will give computational results, as in [ST02a], to show that h tends to lie near the center of the interval $(E - U, E + U)$ in cubic function fields as well, and will analyze computational results to optimize the expected running time of these computations over all purely cubic function fields over a fixed field and of a given genus.

1.3 Summary and Contributions of the Thesis

In Chapter 2, we will give more complete details of the theoretical background of function fields, divisors, and maximal orders. Chapter 3 then restricts to cubic function fields and develops a divisor-theoretic description of the infrastructures of a cubic function field of positive unit rank. We begin with standard notation and basic results on such function fields, including a characterization of cubic function fields of each possible signature. We introduce the notions of a distinguished and i-distinguished divisor and prove that every divisor class of a cubic function field with at least one infinite place, ∞_0, of degree 1 contains a unique 0-distinguished divisor. We show that distinguished divisors are 0-distinguished in this case and that 0-distinguished divisors are reduced. We also define similar notions for ideals and fractional ideals and furthermore prove that distinguished fractional ideals are precisely "reduced" fractional ideals, as defined in [Sch00, SS00, Sch01, LSY03]. Therefore, we define $\mathcal{R}_{\mathbf{C}}$, for any cubic function field with $r > 0$ as a particular set of distinguished divisors, and define a distance measure on the divisors. This divisor-theoretic approach to infrastructure has led to numerous results and insights, in particular improved bounds on the size of infrastructure divisors and on the size of an infrastructure itself, along with its structure.

In Chapter 4, we develop ideal arithmetic for purely cubic function fields. Specifically, we give a complete description of ideal multiplication and inversion for any purely cubic function field, K,

[2] In fact, Scheidler and Stein give two approximations, E_1 and E_2, of h, and two error bounds, U_2 and U_3, for $|h - E_2|$.

with $\mathrm{char}(K) \neq 3$, and provide algorithms for these computations. In addition, if $r = 0$, we show how to compute the unique distinguished ideal equivalent to a given ideal.

Chapter 5 then develops arithmetic in $\mathcal{R}_{\mathbf{C}}$, for any purely cubic function field with $\mathrm{char}(K) \geq 5$. This chapter relies heavily upon the correspondence between distinguished fractional ideals and distinguished divisors. We state previously known results on reduced bases for fractional ideals and how to use these bases to reduce a fractional ideal to an equivalent distinguished ideal. Results on ideal reduction in the unit rank 2 case are new, as well as bounds on the number of reduction steps that are required. The results on reduced bases are then used to define the baby step operation and the ideal reduction is used to define the giant step operation. Further, we prove sharp upper and lower bounds on the length of a baby step. In addition to the baby step and giant step operations, we define the inverse of an infrastructure divisor and the divisor below $y \in \mathbb{N}_0$ for unit rank 1 infrastructures. For each operation, we formalize their computation in an algorithm. Chapter 5 concludes with a set of applications of infrastructure arithmetic, the most important of which states previously known results on computing the S-regulator and a system of fundamental units of a purely cubic function field of positive unit rank. We also identify and prove results on the three-fold symmetry of unit rank 2 infrastructures.

Chapter 6 applies the arithmetic of $\mathcal{J}_K$ and $\mathcal{R}$ if K is a purely cubic function field to compute h and R^S for several such purely cubic function fields with $r = 0$ and $r = 1$. We begin by stating the algorithms from [SS00, LSY03, Sch04] to compute R^S and a system of fundamental units if $r > 0$. We then give the first analysis of the algorithms of [LSY03] for the case $r = 2$. We follow by describing how Shanks' Baby Step-Giant Step method and Pollard's Kangaroo algorithm may be adapted to improve the computation of h in the case $r = 0$ and of R^S if $r = 1$. Further, we give a slight improvement in the application of the Kangaroo method to the computation of R^S. We then describe the methods of Scheidler and Stein [SS07, SS08] to determine good estimates, E_1 and E_2, of h, and upper bounds, U_1, U_2, and U_3, on the error, $|h - E_i|$, for $i = 1, 2, 3$ ($E_2 = E_3$), so that $h \in (E_i - U_i, E_i + U_i)$. We implemented this method on purely cubic function fields of unit rank 0 and 1 and obtained data to optimize computations for genera $3 \leq g \leq 7$. We then used this data to compute the 28 decimal digit divisor class numbers of two genus 3 purely cubic function fields: one with $r = 0$ and one with $r = 1$, and the 25 decimal digit divisor class numbers of two genus 4 purely cubic function fields: one with $r = 0$ and one with $r = 1$. In the unit rank 1 examples, we determined the 26 and 24 decimal digit S-regulators for the respective genus 3 and genus 4 examples. We believe that these are the largest divisor class numbers ever computed for a cubic function field with $g \geq 4$ and the largest S-regulators ever computed for any cubic function field, respectively. We conclude the chapter by giving estimates of the expected time to compute h and R^S for the cases $r = 0$ and $r = 1$, respectively, for $g = 3$, 4 and a range of constant fields, $\mathbb{F}_q$.

Finally, Chapter 7 summarizes our results and poses a number of open questions that have resulted from this work and highlights areas of future work. In particular, we give some thoughts on how to extend the methods described in Chapter 6 to compute the regulator of a purely cubic function field of unit rank 2.

Chapter 2

Overview of Function Fields

This chapter presents an overview of function fields and establishes the notation that will be used throughout this thesis. Section 2.1 gives the definition of a function field and other related definitions and properties. Section 2.2 discusses divisors in terms of places of a function field K from an algebraic perspective and states the Riemann-Roch Theorem. Divisors of places are related to divisors of points on a curve C that defines a model of K in Section 2.3 to provide a geometric interpretation of divisors. Section 2.4 considers the infinite places, defining the notion of the signature of the function field K_x. Lastly, the connection between divisors of K and ideals of its maximal order, $\mathcal{O} = \mathcal{O}_x$, is made in Section 2.5, the units of $\mathcal{O}$ are defined, and various relevant properties of $\mathcal{O}$ will be examined. For a general introduction to function fields, we direct the reader to [Has80], [Sti93], or [Ros02]. In what follows we will highlight the material that will be of particular use to us.

2.1 Function Fields

Let k be a field, x a transcendental element over k, and let $k(x)$ denote the field of rational functions with coefficients in k. In the next chapter, we will consider the case where k is a finite field $\mathbb{F}_q$ of q elements, but in this chapter we will mainly present this background in its full generality. A(n) *(algebraic) function field* K over k is a finite algebraic extension of $k(x)$, and $n = [K : k(x)]$ is called the *extension degree*. Throughout this chapter, we will assume that $\mathrm{char}(k) \nmid n$ so that $K/k(x)$ is a separable and simple extension. Since K is a finite algebraic extension of $k(x)$, we may write K as $K = k(x, y)$, where y is a root of the equation $C : F(x, Y) = 0$, with $F(x, Y) \in k(x)[Y]$ an irreducible polynomial. If $F(x, Y)$ remains irreducible in $\overline{k}(x, Y)$, then $F(x, Y)$ is said to be *absolutely irreducible*, and we can in fact choose an absolutely irreducible polynomial $F(x, Y)$ such that $F(x, y) = 0$ and $K = k(x, y)$. We will also write $K = k(C)$. We will describe the following concept in more detail later in this chapter, but we note now that for a given function field K, there exist curves $C, C' \in k[x, y]$ such that $k(C) \cong k(C')$. The particular curve C that is chosen defines a *model* for the field K, and depends on the variable x, and we will commonly write K_x to identify K with this model. The algebraic closure of k in K is called the *constant field* of K, and will be denoted $\tilde{k}$. If $\tilde{k} = k$, then k is called the *full* constant field of K. We conclude this section with an important result that allows us to determine when k is the full constant field of K.

Proposition 2.1.1 (Corollary III.6.7 of [Sti93]) *If $K = k(x, y)$ is a function field and $F(x, Y) \in k(x)[Y]$ is the minimal polynomial of y over $k(x)$, then $F(x, Y)$ is absolutely irreducible if and only if k is the full constant field of K.*

2.2 Places

Let $K = K_x$ be a model of an arbitrary function field K over k. A *place*, P, of a function field K is the unique maximal ideal of some discrete valuation ring, $\mathcal{O}_P \subset K$, whose field of fractions is K. We will denote the set of places of K by $\mathbb{P}_K$. Associated with $\mathcal{O}_P$ is a normalized discrete valuation $v_P : K \to \mathbb{Z} \cup \{\infty\}$. Let $\mathcal{O}_P^* = \mathcal{O}_P \setminus P$ be the group of units of $\mathcal{O}_P$. If $\alpha \in \mathcal{O}_P$, then $v_P(\alpha) = m$, where m is maximal such that $\alpha \in P^m$. If $\alpha \in K^* \setminus \mathcal{O}_P$, then $v_P(\alpha) = -v_P(1/\alpha)$. By convention, $v_P(0) = \infty$ for all places P. In order to clarify this notion, we give an alternate and equivalent definition of v_P. It is well-known that $\mathcal{O}_P$ is a principal ideal domain so that $P = p\mathcal{O}_P$ for some element $p \in \mathcal{O}_P$. (See Theorem I.1.6 of [Sti93].) This element p is called a *prime element* of $\mathcal{O}_P$, and by extension, a prime element of K. Then for every $\alpha \in K^*$, $\alpha = p^m u$, where $u \in \mathcal{O}_P^*$, and we define $v_P(\alpha) = m$. Thus, for every place $P \in \mathbb{P}_K$, we have $\mathcal{O}_P = \{\alpha \in K \mid v_P(\alpha) \geq 0\}$, $\mathcal{O}_P^* = \{\alpha \in K \mid v_P(\alpha) = 0\}$, and $P = \{\alpha \in K \mid v_P(\alpha) > 0\}$ (Theorem I.1.12 of [Sti93]). The *degree* of a place P, denoted $\deg(P)$, is the degree of the field extension $\left[\mathcal{O}_P/P : \tilde{k}\right]$. For any $\alpha \in K$ and $P \in \mathbb{P}_K$, P is called a *zero* of α if $v_P(\alpha) > 0$, and a *pole* of α if $v_P(\alpha) < 0$. The next section classifies places into finite and infinite places, and will describe the places of $k(x)$ more explicitly. That description will be followed by the generalization to places of arbitrary function fields.

2.2.1 Finite and Infinite Places of $k(x)$

The set of places of $K = K_x$ are grouped into finite and infinite places. Before describing places in arbitrary function fields, we will first describe them in the case of the function field $k(x)$. The following results are stated and proved in Proposition I.2.1 and Theorem I.2.2 in [Sti93]. For the function field $k(x)$, the *finite* places of $k(x)$ correspond with the monic irreducible polynomials $p(x) \in k[x]$. Let P be the place corresponding with the irreducible polynomial $p(x)$. Then

$$P = \left\{ \frac{f(x)}{g(x)} \mid f(x),\, g(x) \in k[x],\ \ p(x) \mid f(x),\ \ p(x) \nmid g(x) \right\}$$

and

$$\mathcal{O}_P = \left\{ \frac{f(x)}{g(x)} \mid f(x),\, g(x) \in k[x],\ \ p(x) \nmid g(x) \right\} \ .$$

Therefore, $P = \langle p(x) \rangle \mathcal{O}_P$. If $\alpha \in k(x)^*$, then α can be written $\alpha = p^m(x)\frac{f(x)}{g(x)}$, where $p(x) \nmid f(x)$, $g(x)$, and $v_P(\alpha) = m$. We also have $\deg(P) = \deg(p(x))$. There is only one *infinite* place in $\mathbb{P}_{k(x)}$, denoted ∞, and this corresponds with the prime element $1/x$. For the infinite place, we have:

$$\infty = \left\{ \frac{f(x)}{g(x)} \mid f(x),\, g(x) \in k[x],\ \ \deg(f(x)) < \deg(g(x)) \right\}$$

and

$$\mathcal{O}_\infty = \left\{ \frac{f(x)}{g(x)} \mid f(x),\, g(x) \in k[x],\ \ \deg(f(x)) \leq \deg(g(x)) \right\} \ .$$

Therefore, $\infty = \left\langle \frac{1}{x} \right\rangle \mathcal{O}_\infty$. For any $0 \neq \alpha = \frac{f(x)}{g(x)} \in k(x)$, we then have $v_\infty(\alpha) = -\deg(\alpha) = \deg(g(x)) - \deg(f(x))$, and $\deg(\infty) = 1$.

Notice that each place of $k(x)$ corresponds with a prime element of $k(x)$. The same situation in fact holds for any extension of $k(x)$, but we will explain this more precisely later. We may therefore

use the terms *primes* and *places* interchangeably. We will now examine the behavior of places of general function fields.

2.2.2 Places in Extensions of $k(x)$

When we consider the places of an extension $K = K_x$ of $k(x)$, we consider the analogy of number fields. As with prime ideals in extensions of number fields, the places of the model of K each *lie over* a place of $k(x)$. A place, $\mathfrak{P} \in \mathbb{P}_K$, lies over $P \in \mathbb{P}_{k(x)}$ (and P *lies under* $\mathfrak{P}$) if $P \subseteq \mathfrak{P}$, and this is denoted by $\mathfrak{P} \mid P$. If $\mathfrak{P} \mid P$, then equivalently, we have $\mathcal{O}_P \subseteq \mathcal{O}_{\mathfrak{P}}$ and there exists some integer $e \geq 1$ such that $v_P(\alpha) = ev_{\mathfrak{P}}(\alpha)$ for all $\alpha \in K$. (See Proposition III.1.4 of [Sti93].) If P is a finite or infinite place of $k(x)$, and $\mathfrak{P} \in \mathbb{P}_K$ with $\mathfrak{P} \mid P$, then $\mathfrak{P}$ is said to be *finite* or *infinite*, respectively. We will typically denote the infinite place of $k(x)$ by ∞ and infinite places above ∞ by ∞_i, $i \in \{0, 1, \ldots, r\}$, where $r+1$ is the number of infinite places of our model of K. The valuation corresponding to ∞_i will be denoted v_i. We will show later that the model of K is characterized by how ∞ *splits* into the infinite places above it in K. Next we will consider sums of places of a field K.

2.2.3 Divisors

A *divisor* of K is a formal sum $D = \sum_{P \in \mathbb{P}_K} m_P P$, where $m_P = 0$ for all but a finite number of places. Let $v_P(D) = m_P$ (often denoted $\mathrm{ord}_P(D)$) be the *order* of D at P, and let $\mathrm{supp}(D) = \{P \in \mathbb{P}_K \mid v_P(D) \neq 0\}$ be the *support* of D. If $P \in \mathrm{supp}(D)$ and $v_P(D) > 0$, then P is called a *zero* of D, and if $v_P(D) < 0$, then P is called a *pole* of D. Note that we may define a partial ordering on divisors. If D_1 and D_2 are two divisors, then we write $D_1 \geq D_2$ if $v_P(D_1) \geq v_P(D_2)$ for every place P. The *degree* of the divisor is defined to be $\deg(D) = \sum_{P \in \mathbb{P}_K} m_P \deg(P)$. It is clear that the set of divisors of K forms an abelian group under addition, which we denote by $\mathcal{D} = \mathcal{D}(K)$. The set of divisors of degree 0 is a proper subgroup of $\mathcal{D}$, and is denoted $\mathcal{D}_0 = \mathcal{D}_0(K)$. For any $\alpha \in K^*$, the *divisor of* α is defined to be the divisor $\mathrm{div}(\alpha) = \sum_{P \in \mathbb{P}_K} v_P(\alpha) P$. It is a well-known result that $\mathrm{div}(\alpha) \in \mathcal{D}_0$. See Proposition 5.1 of [Ros02], for example. A *principal* divisor is a divisor D such that $D = \mathrm{div}(\alpha)$ for some $\alpha \in K^*$. Furthermore, $\mathcal{P} = \mathcal{P}(K)$ is defined as the subgroup of $\mathcal{D}_0$ of principal divisors of K.

An important group associated with K is the *(degree zero) divisor class group*, which is often called the *Jacobian* of K, and is defined $\mathcal{J} = \mathcal{J}_K = \mathcal{D}_0/\mathcal{P}$. Therefore, two (degree 0) divisors, D_1 and D_2, are (linearly) equivalent, written $D_1 \sim D_2$, if $D_1 - D_2 = \mathrm{div}(\alpha)$ for some $\alpha \in K^*$. It is a well-known result that $\mathcal{J}$ is a finite abelian group under addition (see Lemma 5.6 of [Ros02]), and its order, $h = h_K = |\mathcal{J}|$, is called the *divisor class number* of K. The divisor class group, and hence h, is independent of the model of K and is therefore an invariant of the function field K.

2.2.4 Infinite Places

Here we focus on the infinite places of K and introduce some notation that will be useful later, especially when we connect divisors to ideals of the maximal order of K. For the given model of K, let $S = S_x = \{\infty_0, \ldots, \infty_r\}$ denote the set of infinite places of K and let $f = \gcd(\deg(\infty_0), \ldots, \deg(\infty_r))$. We may write any divisor $D \in \mathcal{D}$ as a sum of two divisors, a *finite* and an *infinite* divisor,

$D = D_S + D^S$, where $\text{supp}(D^S) \subseteq S$ and $\text{supp}(D_S) \subseteq \mathbb{P}_K \setminus S$. Let $\mathcal{D}^S = \{D \in \mathcal{D} \mid \text{supp}(D) \subseteq S\}$ and let $\mathcal{D}_S = \{D \in \mathcal{D} \mid \text{supp}(D) \subset \mathbb{P}_K \setminus S\}$. It is clear that each of these sets are proper subgroups of $\mathcal{D}$. Now let $\mathcal{D}_0^S = \mathcal{D}^S \cap \mathcal{D}_0$, $\mathcal{P}^S = \mathcal{P} \cap \mathcal{D}^S$, and $\mathcal{P}_S = \mathcal{P} \cap \mathcal{D}_S$. Notice that $\mathcal{P}^S$ is a subgroup of $\mathcal{D}_0^S$. The *S-regulator* of K is defined to be the order of the corresponding quotient group: $R^S = R_K^S = \left[\mathcal{D}_0^S : \mathcal{P}^S\right]$. The definitions of the rings $\mathcal{O}_P$ may be extended to the *ring of S-integers*: $\mathcal{O}_S = \{\alpha \in K \mid v_P(\alpha) \geq 0 \text{ for all } P \notin S\}$, and the group of *S-units* is defined to be the subset $\mathcal{O}_S^* = \{\alpha \in \mathcal{O}_S \mid v_P(\alpha) = 0 \text{ for all } P \notin S\}$.

2.2.5 The Riemann-Roch Theorem

In this section, we will briefly state a central theorem of algebraic geometry that identifies an important invariant of a function field. This invariant will allow us to give bounds on the divisor class number for function fields over $\mathbb{F}_q(x)$. For a divisor $D \in \mathcal{D}$, let $L(D) = \{\alpha \in K^* \mid \text{div}(\alpha) \geq -D\} \cup \{0\}$. $L(D)$ is a finite dimensional vector space over k whose dimension is commonly denoted $l(D)$. The Riemann-Roch Theorem gives another important invariant of the field K.

Theorem 2.2.1 (Riemann-Roch) *There is an integer $g \geq 0$ and a divisor class $\mathcal{C}$ such that for $W \in \mathcal{C}$ and $A \in \mathcal{D}$:*

$$l(A) = \deg(A) - g + 1 + l(W - A) \ .$$

The invariant g is called the *genus* of the function field K, the divisor class $\mathcal{C}$ is called the *canonical class*, and any divisor $W \in \mathcal{C}$ is called a *canonical divisor*.

If $k = \mathbb{F}_q$, then the Hasse-Weil Theorem gives bounds for the size of the divisor class number of a function field K over $\mathbb{F}_q(x)$.

$$(\sqrt{q} - 1)^{2g} \leq h \leq (\sqrt{q} + 1)^{2g} \ .$$

2.3 The Geometry of Places

Thus far, we have given a largely algebraic description of function fields. We will now give a geometric description of function fields, relating places with points on a curve C. Much of this section is addressed in Appendix B of [Sti93], and that source gives more information and background. We will make note of topics that are essential for this discussion here. In particular, projective space will be defined, and it will be shown that function fields can be written as the function field of an irreducible non-singular curve. We will begin by working over an algebraically closed field, $\overline{k}$, and will later restrict ourselves to working over a non-algebraically closed field k.

2.3.1 Affine Space

The familiar notion of Cartesian n-space may be generalized to *affine n-space*, that is, the set

$$\mathbf{A}^n = \mathbf{A}^n\left(\overline{k}\right) = \left\{(a_1, \ldots, a_n) \mid a_i \in \overline{k}\right\} \ .$$

A subset $V \subseteq \mathbf{A}^n$ is called an *(affine) algebraic set* if there is some set of polynomials $M \subseteq \overline{k}[x_0, \ldots, x_{n-1}]$ such that $V = \{P \in \mathbf{A}^n \mid F(P) = 0 \text{ for all } F \in M\}$. An algebraic set V is

called a(n) *(affine) variety* if V is *irreducible*, that is, if there do not exist any non-empty algebraic subsets V_1 and V_2 such that $V_1 \cup V_2 = V$. For any algebraic set V, the *ideal of* V is the set $I(V) = \left\{F \in \overline{k}[x_0, \ldots, x_{n-1}] \mid F(P) = 0 \text{ for all } P \in V\right\}$. $I(V)$ is an ideal of $\overline{k}[x_0, \ldots, x_{n-1}]$ and is a prime ideal if and only if V is a variety.

The *coordinate ring* of any variety V is the set $\mathcal{O}(V) = \overline{k}[x_0, \ldots, x_{n-1}]/I(V)$, which is an integral domain, since $I(V)$ is prime. It is also common to write $\mathcal{O}(V)$ as $\overline{k}[V]$. The quotient field of $\mathcal{O}(V)$, $\overline{k}(V)$, is called the *function field* of V. The *dimension* of V is the transcendence degree of $\overline{k}(V)$ over $\overline{k}$. For any point $P \in V$, the local ring $\mathcal{O}_P(V) = \left\{f \in \overline{k}(V) \mid f = g/h,\ g, h \in \mathcal{O}(V),\ h(P) \neq 0\right\}$, and the corresponding maximal ideal is denoted

$$\mathfrak{m}_P(V) = \left\{f \in \overline{k}(V) \mid f = g/h,\ g, h \in \mathcal{O}(V),\ h(P) \neq 0,\ g(P) = 0\right\} \ .$$

Note the similarities between these definitions and our earlier development of places. For certain varieties, it will be shown that the two notions are indeed identical.

An *affine planar curve* is a variety $V \subseteq \mathbf{P}^2$ of dimension 1. In this case, $I(V)$ is generated by some irreducible $F(x, Y) \in \overline{k}[x, Y] \setminus \overline{k}$, and therefore, $\overline{k}(V)$ is a function field in the same sense that we described in Section 2.1, except here the function field is defined over the algebraic closure of k. Given an irreducible polynomial $F(x, Y) \in \overline{k}[x, Y] \setminus \overline{k}$, we may find the associated variety $V = \{P \in \mathbf{A}^2 \mid F(P) = 0\}$. Thus, for simplicity, we will refer to both a polynomial equation $C : F(x, Y) = 0$ and its variety as curves, and will often write the function field of this variety as $\overline{k}(C)$ to underscore the connection of the field with the generating curve. A point $P \in V$ is called *non-singular* if $\mathcal{O}_P(V)$ is a discrete valuation ring, and *singular* otherwise. If every $P \in V$ is non-singular, then the curve V is called *non-singular* or *smooth*, and is called *singular* otherwise. A curve $C : F(x, Y) = 0$ is said to be *singular* or *non-singular* if its associated variety is singular or non-singular, respectively. Equivalently, C is non-singular, if there is no point $P \in \mathbf{A}^2$ that simultaneously satisfies

$$F(P) = 0, \quad \frac{\partial F(P)}{\partial x} = 0, \quad \text{and} \quad \frac{\partial F(P)}{\partial Y} = 0 \ ,$$

and is singular otherwise. We must show now that every function field can be written as the function field of a smooth curve. To show this, we must consider the related topics of projective space and curves.

2.3.2 Projective Space

In $\mathbf{A}^{n+1} \setminus \{0, \ldots, 0\}$, there is an equivalence relation $(a_0, a_1, \ldots, a_n) \sim (b_0, b_1, \ldots, b_n)$ if there is some $\lambda \in \overline{k}^*$ such that $a_i = \lambda b_i$ for all i. This equivalence class is denoted by $(a_0 : a_1 : \ldots : a_n)$, and the set of such classes is called *projective n-space*. We denote this space by $\mathbf{P}^n = \mathbf{P}^n\left(\overline{k}\right) = \left\{(a_0 : a_1 : \ldots : a_n) \mid a_i \in \overline{k}\right\}$. Elements of $\mathbf{P}^n$ are *(projective) points* and the coordinates of a point are called *homogeneous coordinates*. A *homogeneous polynomial* is a polynomial $F(x_0, \ldots, x_n) \in \overline{k}[x_0, \ldots, x_n]$ in which each term has the same degree. A *projective planar curve*, then, is the set of solutions $(a_0 : a_1 : a_2) \in \mathbf{P}^2$ to a homogeneous polynomial $F(x_0, x_1, x_2) \in \overline{k}[x_0, x_1, x_2]$. We may obtain a homogeneous polynomial $\overline{F}(x_0, \ldots, x_n)$ from an affine polynomial $F(x_0, \ldots, x_{n-1})$ by $\overline{F} = x_n^d F(x_0/x_n, \ldots, x_{n-1}/x_n)$, where $d = \deg(F)$. We may reverse this by simply substituting

$x_n = 1$.

In projective n-space, we have analogous notions of varieties and ideals of varieties. The function field of a projective variety, V, however, is defined as

$$\overline{k}(V) = \{g/h \mid g,\, h \in \mathcal{O}(V),\ \ g,\, h \text{ are homogeneous},\ \ \deg(g) = \deg(h),\ \ h \neq 0\} \quad .$$

We have analogous definitions of $\mathcal{O}_P(V)$ and $\mathfrak{m}_P(V)$ as well.

Given an affine variety $V \subseteq \mathbf{A}^n$, the *projective closure* of V is defined to be the projective variety $\overline{V} = \left\{P \in \mathbf{P}^n \mid \overline{F}(P) = 0 \text{ for all } F \in I(V)\right\}$. There is a natural isomorphism $\overline{k}(V) \cong \overline{k}\left(\overline{V}\right)$.

Given a function field K, we wish to find a smooth projective variety V, and hence an affine polynomial $F(x,\, y)$, such that $K \cong \overline{k}(V)$. To accomplish this, the notion of morphisms between varieties will be defined. Given any two varieties $V_1 \in \mathbf{P}^m$ and $V_2 \in \mathbf{P}^n$, there is a map $\phi : V_1 \to V_2$, $\phi = (F_0,\, \ldots,\, F_n)$ in which (a) $F_i \in \overline{k}[x_0,\, \ldots,\, x_m]$ are homogeneous of the same degree, (b) not all F_i are in $I(V_1)$, and (c) for all $H \in I(V_2)$, $H(F_0,\, \ldots,\, F_n) \in I(V_1)$. If $G_0,\, \ldots,\, G_n \in \overline{k}[x_0,\, \ldots,\, x_m]$ also satisfy conditions (a), (b), and (c), then $(F_0,\, \ldots,\, F_n)$ and $(G_0,\, \ldots,\, G_n)$ are said to be *equivalent* if $F_i G_j \equiv F_j G_i \pmod{I(V_1)}$ for all $0 \leq i,\, j \leq n$. The equivalence class of $(F_0,\, \ldots,\, F_n)$ is called a *rational map* from V_1 to V_2 and is denoted by $\phi = (F_0 : \ldots : F_n)$, separating the F_i by colons to identify the map with the equivalence class. Two varieties, V_1 and V_2, are said to be *birationally equivalent* if there are rational maps $\phi_1 : V_1 \to V_2$ and $\phi_2 : V_2 \to V_1$ such that $\phi_1 \circ \phi_2$ and $\phi_2 \circ \phi_1$ are the identity maps on V_2 and V_1, respectively. Two varieties V_1 and V_2 are birationally equivalent if and only if $\overline{k}(V_1)$ is $\overline{k}$-isomorphic to $\overline{k}(V_2)$. A rational map $\phi = (F_0 : \ldots : F_n)$ is *regular* (or *defined*) at a point $P \in V$ if there exist polynomials $G_0,\, \ldots,\, G_n \in \overline{k}[x_0,\, \ldots,\, x_n]$ such that $\phi = (G_0 : \ldots : G_n)$ and $G_i(P) \neq 0$ for at least one i. A rational map $\phi : V_1 \to V_2$ that is regular at every point $P \in V$ is called a *morphism*. If there is a morphism $\phi_2 : V_2 \to V_1$ such that $\phi \circ \phi_2$ and $\phi_2 \circ \phi$ are the identity on V_2 and V_1, respectively, then ϕ is an *isomorphism*. If two varieties are isomorphic, then they are birationally equivalent.

Restricting ourselves now to curves, given a projective curve C', there exists a non-singular projective curve C, and a birational morphism $\phi : C \to C'$. This curve is unique up to isomorphism and is called the *non-singular model* of C. Therefore, for any algebraic function field K, there is a non-singular projective curve C such that $K \cong \overline{k}(C)$. To construct this curve, we first choose generating elements $x,\, y \in K$ such that $K = \overline{k}(x,\, y)$. (Such elements exist by Proposition III.9.2 of [Sti93].) Let $F(x,\, Y) \in \overline{k}[x,\, Y]$ be an irreducible (affine) polynomial in which $F(x,\, y) = 0$, let $C' = \left\{P \in \mathbf{A}^2 \mid F(P) = 0\right\}$, and find the projective closure of C', $\overline{C'} \subseteq \mathbf{P}^2$. If C is the non-singular model of $\overline{C'}$, then $K \cong \overline{k}(C)$. Now that we have connected function fields to non-singular varieties, we make the geometric connection between points and places.

2.3.3 Divisors

If V is a non-singular projective curve, $K = \overline{k}(V)$ is its function field, and $P \in V$, then there is a one-to-one correspondence $P \to \mathfrak{m}_P(V)$ between the points on a variety and the places of its function field. Therefore, we may think of divisors of V, and therefore of K, as a formal sum of points. We let $\mathcal{D}(V) = \left\{\sum_{P \in V} m_P P \mid m_P \in \mathbb{Z},\ \ m_P = 0 \text{ for almost all } P\right\}$ denote the group of divisors of a variety V, so there is a one-to-one correspondence between divisors of $\mathcal{D}(V)$ and divisors of $\mathcal{D}(K) = \mathcal{D}\left(\overline{k}(V)\right)$. All the related notions of principal divisors, degree of a divisor, divisor class

groups, and so on, carry over as well. We highlight the principal divisors in this case. If $\alpha \in \overline{k}(V)^*$, then its divisor is of the form $\text{div}(\alpha) = \sum_{P \in V} v_P(\alpha)P$, where $v_P(\alpha)$ (often denoted $\text{ord}_P(\alpha)$) is the order of vanishing of α at the point P. One key difference to note, though, is that the degree of any $P \in V$ is 1. Since we will be working over a finite field, specifically k, rather than its algebraic closure, we will now consider what happens over a non-algebraically closed field k.

2.3.4 Varieties and Curves over k

Now we adapt the previous notions to the setting of varieties over a field k that is not necessarily algebraically closed. There are a few details that we will need to highlight. An affine variety V is said to be *defined over* k if $I(V)$ is generated by $F_1, \ldots, F_m \in k[x_0, \ldots, x_{n-1}]$, and $V(k) = V \cap \mathbf{A}^n(k)$ is called the set of *k-rational points of* V. The ideal of the variety is denoted $I(V/k) = I(V) \cap k[x_0, \ldots, x_{n-1}]$, the residue class ring is $\mathcal{O}(V/k) = k[x_0, \ldots, x_{n-1}]/I(V/k)$, and the function field of V over k is the quotient field of $\mathcal{O}(V/k)$, which is denoted $k(V)$. A (planar) curve over k, then, is simply a variety $V \subset \mathbf{A}^2(k)$ of dimension 1 and is generated by a single polynomial $F(x, Y) \in k[x, Y]$. As before, we will often write $C : F(x, Y) = 0$ to denote this curve and will write its function field as $k(C)$.

Similarly, a projective variety, V is *defined over* k if $I(V)$ is generated by homogeneous polynomials $F_1, \ldots, F_m \in k[x_0, \ldots, x_n]$, and $V(k) = V \cap \mathbf{P}^n(k)$ is called the set of *k-rational points of* V. The ideal of the variety, residue class ring, and function field are all defined in the same manner for projective varieties as they are for affine varieties.

Let $\text{Gal}\left(\overline{k}/k\right)$ be the Galois group of $\overline{k}$ over k. In the case $k = \mathbb{F}_q$, then $\text{Gal}\left(\overline{\mathbb{F}}_q/\mathbb{F}_q\right)$ is generated by the Frobenius automorphism σ, where $\sigma(a) = a^q$ for any $a \in \overline{\mathbb{F}}_q$. For a point $P = (a_0 : \ldots : a_n) \in V \subseteq \mathbf{P}^n$ and for an arbitrary $\tau \in \text{Gal}\left(\overline{k}/k\right)$, we write $\tau(P) = (\tau(a_0) : \ldots : \tau(a_n))$, and similarly for affine points. For any projective variety, V, we then have $V(k) = \left\{P \in V \mid \tau(P) = P \text{ for all } \tau \in \text{Gal}\left(\overline{k}/k\right)\right\}$ and $k(V) = \left\{f \in \overline{k}(V) \mid \tau(f) = f \text{ for all } \tau \in \text{Gal}\left(\overline{k}/k\right)\right\}$. In particular, we will consider projective curves defined over k, which are described analogously to affine curves defined over k.

Therefore, any function field $K/k(x)$ may be written in the form $K \cong k(C)$, where $C : F(x, Y) = 0$ and $F(x, Y) \in k[x, Y]$ is an irreducible non-singular (affine) polynomial. If V is the projective closure of the variety of $F(x, Y)$ and W is another projective curve that is birationally equivalent, but not isomorphic, to V, with $G(x, Y) \in k[x, Y]$ the (possibly singular) polynomial that generates $I(W)$, then we saw that $K \cong k(V) \cong k(W)$. This is the basis for determining different models of a function field. Different models of K are thus obtained if we apply a birational transformation to a curve and its defining polynomial. We will give an example of this shortly. Lastly, we will consider divisors of a variety over k.

2.3.5 Divisors over k

We showed how an automorphism, $\tau \in \text{Gal}\left(\overline{k}/k\right)$, acts on points. We note that τ extends linearly to act on a divisor. Thus, a divisor $D = \sum_{P \in V} m_P P \in \mathcal{D}(V)$ is *defined over* k if $\tau(D) = D$. The group of divisors of V defined over k is denoted $\mathcal{D}(V/k)$. A *prime divisor* is a divisor $0 < D \in \mathcal{D}(V/k)$ that cannot be written $D = D_1 + D_2$, where $0 < D_1, D_2 \in \mathcal{D}(V/k)$. Prime divisors in $\mathcal{D}(V/k)$ are in one-to-one correspondence with the places of the function field $k(V)$. In this case the prime divisors

of degree one are the k-rational points of V and correspond with the places of $k(V)$ of degree 1. In general, if a place $P \in \mathbb{P}_{k(V)}$ has degree d, then the corresponding point $P \in V/\overline{k}$ is defined over a degree d extension of k.

As in the case of divisors over $\overline{k}$, we may also have divisors of degree 0 and principal divisors defined over k. Let $\mathcal{D}_0(V/k)$ and $\mathcal{P}(V/k)$ denote the groups of degree 0 and principal divisors defined over k, respectively. We form the quotient group $\mathcal{J} = \mathcal{J}(V/k) = \mathcal{D}_0(V/k)/\mathcal{P}(V/k)$, the divisor class group over k. We noted earlier that $\mathcal{J}$ is finite and that the divisor class number, h, is its order in the case that V is a planar curve, but this result holds for any variety. If g is the genus of the variety V/k and $k = \mathbb{F}_q$, then the Hasse-Weil Theorem holds as well. We will mainly be concerned with applying these notions and results to varieties arising from planar curves. Since such curves are associated with a function field over k, we may say that $\mathcal{J} = \mathcal{J}_K$ and $h = h_K$ are the divisor class group and the divisor class number of K, respectively.

We have now made the connection between the algebra and the geometry of function fields. In particular, we have given an explicit description of the different models of a function field and to some extent, how to change models of a function field. In the next section, we will examine the place at infinity in particular. We will characterize a model by the places at infinity and in general will extend the familiar notions of prime splitting from algebraic number fields to algebraic function fields.

2.4 Signatures

We will resume discussing divisors in terms of places, but will keep in mind that for a given function field, K, we are working with a specific model of K. Also recall from our earlier discussion of places of K that places of K lie above places of $k(x)$. Here we will describe the behavior of these places in more depth. Let K be a finite algebraic extension of $k(x)$ such that $\mathrm{char}(K) \nmid [K : k(x)]$, let $P \in \mathbb{P}_{k(x)}$ and let $\mathfrak{P}_1, \ldots, \mathfrak{P}_m \in \mathbb{P}_K$ be the set of places that lie over P. As with prime ideals in extensions of number fields, P *splits* into the places $\mathfrak{P}_1, \ldots, \mathfrak{P}_m$ of K. For any $\mathfrak{P} \mid P$, the integer $e(\mathfrak{P} \mid P) = e$ such that $v_P(\alpha) = e v_{\mathfrak{P}}(\alpha)$ for every $\alpha \in K$ is called the *ramification index* of $\mathfrak{P}$ over P. The *inertia degree* of $\mathfrak{P} \mid P$ is $f(\mathfrak{P} \mid P) = [\mathcal{O}_{\mathfrak{P}}/\mathfrak{P} : \mathcal{O}_P/P]$, and is finite if the extension $K/k(x)$ is finite. For the places $\mathfrak{P}_i$, $1 \leq i \leq m$, above P, let $e_i = e(\mathfrak{P}_i \mid P)$ and $f_i = f(\mathfrak{P}_i \mid P)$. We have a result that is equivalent to the familiar property of primes in extensions of number fields: $\sum_{i=1}^{m} e_i f_i = [K : k(x)] = n$. (See Theorem III.1.11 of [Sti93].) If $e(\mathfrak{P} \mid P) > 1$, then $\mathfrak{P} \mid P$ is *ramified*, otherwise $\mathfrak{P}$ is said to be *unramified*. If $m = 1$ and $e_1 = n$, then $P = \mathfrak{P}^n$ is said to be *totally* ramified. If $m = 1$ and $e_1 = 1$, then $P = \mathfrak{P}$ is said to be *inert*. If $m = n$, then $P = \mathfrak{P}_1 \cdots \mathfrak{P}_n$ is said to *split completely*. If $\mathrm{char}(k) \nmid e(\mathfrak{P} \mid P)$ for all $\mathfrak{P} \mid P$, then P is said to be *tamely ramified*.

The *signature* of a place $P \in \mathbb{P}_{k(x)}$ is defined as the $2m$-tuple

$$\mathrm{sig}_K(P) = (e_1,\, f_1;\, e_2,\, f_2;\, \ldots;\, e_m,\, f_m) \ .$$

The signature of ∞ is called the *signature* of K, and is denoted by

$$\mathrm{sig}(K_x) = (e(\infty_0|\infty),\, f(\infty_0|\infty);\, \ldots;\, e(\infty_r|\infty),\, f(\infty_r|\infty)) \ .$$

There is an interesting difference here between number fields and function fields. A function field may have any possible signature at infinity, as long as $\sum_{i=1}^{m} e_i f_i = n$, whereas the signature of a number field always satisfies $e_i f_i = 1$ if v_i is a real embedding, and $e_i f_i = 2$ if v_i is a complex embedding.

We will continue to focus on the places at infinity and will consider the completion of K with respect to its infinite places. Let $k\left\langle x^{-1/e}\right\rangle$ denote the field of *Puiseux series* in $x^{1/e}$, where $e \in \mathbb{N}$. If $0 \neq \alpha \in k\left\langle x^{-1/e}\right\rangle$, then α may be written uniquely as $\alpha = \sum_{i=-m}^{\infty} a_{-i} x^{-i/e}$, where $m \in \mathbb{Z}$, $a_i \in k$ for $i \leq m$, and $a_m \neq 0$. If $\alpha = \sum_{i=-m}^{\infty} a_{-i} x^{-i/e} \in k\left\langle x^{-1/e}\right\rangle$, then in the context of Puiseux series, $\operatorname{sgn}(\alpha) = a_m$ is called the *sign* of α, $\deg(\alpha) = m$ is the *degree* (in $x^{-1/e}$) of α, and $|\alpha| = c^m = c^{\deg \alpha}$ is the *absolute value* of α for some fixed constant $c > 1$. If $k = \mathbb{F}_q$, then typically $c = q$, otherwise $c = \mathbf{e}$, where $\mathbf{e}$ is the base of the natural logarithm. The *floor* of α is defined to be the polynomial part of α, that is, $\lfloor \alpha \rfloor = \sum_{i=0}^{m} a_i x^{i/e} \in k\left[x^{1/e}\right]$. For $\alpha = 0$, we have $\operatorname{sgn}(0) = 0$, $|0| = 0$, and $\lfloor 0 \rfloor = 0$ so that $\deg(0) = -\infty$. The completion of $k(x)$ with respect to ∞ is the field of *Laurent series* $k\left\langle x^{-1}\right\rangle$. (A Laurent series is a Puiseux series with $e = 1$.) Likewise, the completion of $\mathbb{F}_q(x)$ with respect to any finite place P is the field of Laurent series $\mathbb{F}_{q^{\deg(P)}}\langle P\rangle$ in P over $\mathbb{F}_{q^{\deg(P)}}$.

We now restrict ourselves to the case $k = \mathbb{F}_q$. Let $K/\mathbb{F}_q(x)$ be a function field, and let $K = K_x$ be a specific model of the field. If the signature of K is $(e_1,\, f_1;\, e_2,\, f_2;\, \ldots;\, e_m,\, f_m)$ and ∞ is tamely ramified, then the completion of K with respect to the infinite place ∞_i is $K_{\infty_i} = \mathbb{F}_{q^{f_i}}\left\langle x^{-1/e_i}\right\rangle$. For $\alpha \in K_{\infty_i}$, the i-absolute value of α is defined to be $|\alpha|_i = q^{-v_i(\alpha)}$.

For each infinite place of K, ∞_i, for $0 \leq i \leq r$, there is a corresponding unique embedding $\sigma_i : K \hookrightarrow K_{\infty_i}$ that preserves the valuation v_i. Therefore, if $e = \operatorname{lcm}(e_0, \ldots, e_r)$, then there are $r+1$ distinct embeddings of the function field K into $\overline{k}\left\langle x^{-1/e}\right\rangle$. We may therefore write $|\alpha|_i = |\sigma_i(\alpha)|$. This allows us to write any element $\alpha \in K$ as a Puiseux series, and will be used to express these elements in an algorithm, though in a truncated form of course.

2.4.1 Determining Signatures

In order to identify the signature of a place, the first tool to use is Kummer's Theorem. Though this works in almost every case, unfortunately we cannot always determine the signature using this method, but we can at least glean some information.

Theorem 2.4.1 (Kummer's Theorem, Theorem III.3.7 of [Sti93]) *Let $K = \mathbb{F}_q(x, y)$ be a function field, where $F(x, y) = 0$, $F(Y) \in \mathcal{O}_P[Y]$ is an absolutely irreducible monic polynomial, and $P \in \mathbb{P}_{\mathbb{F}_q(x)}$. If $F(Y) \equiv \prod_{i=1}^{r} F_i(Y)^{d_i} \pmod{P}$ is the factorization of $F(Y)$ into irreducible polynomials modulo P, then*

1. *the number of distinct places $\mathfrak{P}_i \mid P$ of K is at least r and for each $i \in \{1, \ldots r\}$, there exists a place $\mathfrak{P}_i$ such that $f(\mathfrak{P}_i \mid P) \geq \deg F_i$; and*

2. *if $d_i = 1$ for all i, then there are exactly r distinct places $\mathfrak{P}_i \mid P$, with $e(\mathfrak{P}_i \mid P) = 1$ and $f(\mathfrak{P}_i \mid P) = \deg F_i$.*

In Part 2 of Kummer's Theorem, the condition "$d_i = 1$ for all i" can be replaced with the stronger condition, "$\left\{1, y, \ldots, y^{n-1}\right\}$ is an $\mathbb{F}_q[x]$-basis of the integral closure $\overline{\mathcal{O}_P}$ of $\mathcal{O}_P$ in K as a module of rank n over $\mathcal{O}_P$"; the former condition implies the latter condition. We note that this theorem

holds for infinite places as well, and we will demonstrate this in an example below. In order to gain more complete information on the splitting of a particular finite place, we must instead find an appropriate minimal field of Puiseux series in which the roots of $F(Y) \pmod P$ live. More precisely, we have the following result in [LRS+08].

Theorem 2.4.2 (Theorem 3.1 of [LRS+08]) *Let $K = \mathbb{F}_q(x, y)$ be an algebraic function field, where $F(x, y) = 0$ and $F(Y) \in \mathbb{F}_q[x][Y]$ is a monic polynomial that is irreducible over $\mathbb{F}_q(x)$. Let P be any finite place of $\mathbb{F}_q(x)$, $\mathfrak{P}_1, \mathfrak{P}_2, \ldots, \mathfrak{P}_r$ the places of K lying above P, and write $e_i = e(\mathfrak{P}_i \mid P)$, $f_i = f(\mathfrak{P}_i \mid P)$, and $n_i = e_i f_i$ for $1 \leq i \leq r$. Then there exists an enumeration of the roots $y^{(0)}, y^{(1)}, \ldots, y^{(n-1)}$ of $F(Y)$ as*

$$\left(y^{(0)}, y^{(1)}, \ldots, y^{(n-1)}\right) = (y_{1,1}, \ldots, y_{1,n_1}, y_{2,1}, \ldots y_{2,n_2}, \ldots, y_{r,1}, \ldots, y_{y,n_r})$$

so that $y_{i,j}$ lies in an extension E of $\overline{\mathbb{F}}_q \langle P \rangle$ of degree e_i, but in no subfield of E, for $1 \leq j \leq n_i$ and $1 \leq i \leq r$. If $e_i = 1$ for some $i \in \{1, 2, \ldots, r\}$, then $y_{i,j} \in \mathbb{F} \langle P \rangle$ for $1 \leq j \leq n_i$, where $\mathbb{F}$ is an extension of $\mathbb{F}_{q^{\deg(P)}}$ of degree at most f_i.

We noted earlier that if $C : F(x, Y) = 0$ is an affine curve, with $F(x, Y) \in k[x, Y]$, then finite places of $k(C)$ of degree d correspond with the points on the affine variety of $F(x, Y)$ over a degree d extension of k. The situation for determining the infinite places of $k(C)$, and in general, how ∞ splits in K, is more complicated. The affine place at infinity of $k(x)$ corresponds to the projective point $[0 : 1 : 0]$. If $\overline{C}$ is the projective curve defined over k corresponding with C, then the infinite places of $k(C)$ correspond with the infinite projective points of C: $\left\{[a : b : 0] \in \overline{C}/k\right\}$. In order to determine the splitting of ∞, we may apply Kummer's Theorem if $\left\{1, y, \ldots, y^{n-1}\right\}$ is an $\mathbb{F}_q[x]$-basis of $\overline{\mathcal{O}_\infty}$ over $\mathcal{O}_\infty$, where $y \in K$ and $F(x, y) = 0$. In general, it is easier to check the weaker condition of Kummer's Theorem. In other words, if ∞ does not ramify, then we may apply Kummer's Theorem as well. Otherwise we must determine the appropriate Puiseux series in which the roots of $F(Y)$ lie modulo the prime element $1/x$.

In order to determine how $F(x, Y) \in \mathbb{F}_q[x, Y]$ splits modulo $1/x$, we may consider a suitable transformation of the curve. Let $C : F(x, Y) = 0$, and choose integers l and m minimal so that after dividing C by x^m and substituting $v = Y/x^l$, the coefficients of v are in $\mathbb{F}_q[1/x]$. Substitute $u = 1/x$ and define the curve $C' : G(u, v) = 0$ with $G(u, v) \in k[u, v]$. We may now obtain information about $\text{sig}(\infty)$ from the factorization of $G(u, v)$ modulo u. Also note that C and C' are birationally equivalent, so that $\mathbb{F}_q(C) \cong \mathbb{F}_q(C')$. Therefore, F and G are two models for the same curve. The following example illustrates this procedure.

Example 2.4.3 *Let $k = \mathbb{F}_7$, $C : Y^3 = x^6 + 2x^5 + x^2$, and $K = K_x = \mathbb{F}_7(C)$. Divide C by x^6 to obtain $C : \frac{Y^3}{x^6} = 1 + \frac{2}{x} + \frac{1}{x^4}$. Substitute $v = Y/x^2$ and $u = 1/x$ so that $C' : v^3 = 1 + 2u + u^4$. Then $K \cong K_u = \mathbb{F}_7(C')$. Now $v^3 \equiv 1 \pmod u$, and $v^3 - 1 \equiv (v-1)(v-2)(v-4) \pmod u$, so by Kummer's Theorem the signature of K_x (or of $\infty = 1/x$) is $\text{sig}(K_x) = (1, 1; 1, 1; 1, 1)$. On the projective closure of C, these places correspond with the projective points $[1 : 1 : 0]$, $[1 : 2 : 0]$, and $[1 : 4 : 0]$. Using this same process, we cannot determine the signature of K_u (or of $\infty = 1/u$), since $Y^3 \equiv 0 \pmod x$, and there is a triple root. We will see later that $\text{sig}(K_u) = (3, 1)$. As noted earlier, $\mathbb{F}_7(C) \cong \mathbb{F}_7(C')$, and C and C' are two different models for the same function field; K_x has three*

infinite places, and K_u has only one. In this case, C' is the non-singular model of the function field $K = \mathbb{F}_7(C)$.

We will shift now to looking at a particular subring of K_x, the maximal order, and its ideals. We will establish a connection between divisors and these ideals.

2.5 The Maximal Order of K

We again assume a specific model, $K_x = k(x, y)$, for our function field $K/k(x)$, and let $C : F(x, Y) = 0$ be the defining curve. If $F(x, Y) = (Y - y)(Y - y_1)\cdots(Y - y_{n-1})$ in the splitting field of F over $k(x)$, then we write $y_1 = y'$, $y_2 = y''$, $\ldots$, $y_{n-1} = y^{(n-1)}$. If $\alpha \in K$, then $\alpha = a_0 + a_1 y + a_2 y^2 + \ldots + a_{n-1} y^{n-1}$, and the n *conjugates* of α are defined to be $\alpha^{(i)} = a_0 + a_1 y^{(i)} + a_2 \left(y^{(i)}\right)^2 + \ldots + a_{n-1}\left(y^{(i)}\right)^{n-1}$ for $0 \leq i \leq n-1$. The *maximal order* or *ring of integers* of K is the integral closure of $k[x]$ in K, and is denoted $\mathcal{O} = \mathcal{O}_x = \mathcal{O}(K_x) = \overline{k[x]}$. Equivalently, $\mathcal{O}$ is isomorphic to the coordinate ring of K, $k[C] \cong k[x, Y]/\langle F(x, Y)\rangle$, and is isomorphic to the ring $\mathcal{O}_S$, of elements $\alpha \in K$ with nonnegative valuation at each finite place. $\mathcal{O}$ is an integral domain and a $k[x]$-module of rank n, and therefore has a $k[x]$-basis, called an *integral basis*, which is written $\{\beta_1 = 1, \beta_2, \ldots, \beta_n\}$. The *norm* of $\alpha \in K$ is defined to be $N(\alpha) = \alpha\alpha' \cdots \alpha^{(n-1)} \in k(x)$. If an element $\alpha \in K$ belongs to $\mathcal{O}$, then $N(\alpha) \in k[x]$. If $N(\alpha) \in k^*$, then α is called a *unit* of $\mathcal{O}$. The set of units of $\mathcal{O}$ forms a group under multiplication and is written $\mathcal{O}^*$. The *discriminant* of $\mathcal{O}$ (or of K) is the square of the determinant:

$$\Delta = \Delta(1, \beta_2, \ldots, \beta_n) = \begin{vmatrix} 1 & 1 & \ldots & 1 \\ \beta_2 & \beta_2' & \ldots & \beta_2^{(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_n & \beta_n' & \ldots & \beta_n^{(n-1)} \end{vmatrix}^2 .$$

The discriminant of $\mathcal{O}$ is a polynomial in $k[x]$ and is independent of the basis of $\mathcal{O}$ up to a factor that is a square in k^*. The *discriminant* of $F(x, Y)$ is $\Delta(F) = \prod_{0 \leq i < j \leq n-1}(y_i - y_j)^2 \in k[x]$, and there is some $I_F \in k[x]$, called the *index* (or *conductor*) of $F(x, Y)$ such that $\Delta(F) = I_F^2 \Delta$. The index of F is unique up to square factors in $\mathbb{F}_q^*$, however, we will abuse notation and speak of "the" index of F and "the" discriminant of K, bearing in mind this square factor. The curve $C : F(x, y) = 0$ is non-singular if and only if $I_F \in k^*$.

With the definition of the discriminant of K, we now have a convenient formula with which to express the genus of K. This theorem, proved in [Sti93], is rewritten with notation more familiar to this thesis in Theorem 1.2.15 of [Wu07].

Theorem 2.5.1 (Hurwitz Genus Formula, Theorem III.4.12 of [Sti93])
Let $K = K_x$ be a function field with full constant field $\tilde{k}$ in which ∞ is tamely ramified, let $n = [K : k(x)]$, and let g be the genus of K. Then

$$g = \frac{\deg(\Delta) + \delta_\infty - 2n}{2\left[\tilde{k} : k\right]} + 1 \ ,$$

where $\delta_\infty = \sum_{i=0}^{r}(e(\infty_i \mid \infty) - 1) f(\infty_i \mid \infty)$ and $\infty_0, \ldots, \infty_r$ are all the places lying above ∞.

Recall that if $F(x, y)$ is absolutely irreducible, then $\tilde{k} = k$. We will express g explicitly in terms of $F(x, y)$ for cubic function fields in the next chapter. Since $\mathcal{O}$ is a ring, it is a natural question to then consider its ideals.

2.5.1 Ideals

An *(integral) ideal*, $\mathfrak{a}$, of $\mathcal{O} = \mathcal{O}_x$ is an $\mathcal{O}$-submodule of $\mathcal{O}$. Moreover, $\mathfrak{a}$ is a free $k[x]$-submodule of $\mathcal{O}$ of rank n. As such, it has a $k[x]$-basis $\{\lambda_1, \ldots, \lambda_n\}$, and we will write $\mathfrak{a} = [\lambda_1, \ldots, \lambda_n]$. (We will use this square bracket notation to denote an ideal in terms of the integral basis, and will use angle brackets to denote a generating set.) In terms of the basis of $\mathcal{O}$, there are polynomials $l_{i,j} \in k[x]$ such that $\lambda_i = l_{i,1} + l_{i,2}\beta_2 + \ldots + l_{i,n}\beta_n$ for $1 \leq i \leq n$. It will be helpful when we develop ideal multiplication to have another kind of basis. In Chapter 20, Section 3 (page 380), of [Has80], Hasse notes that the basis of an ideal in an algebraic number field can be written in a special form. He notes in the proof of the Strong Approximation Theorem on page 399 that a similar basis can be found for an ideal of a function field. Specifically, we have the following statement.

Proposition 2.5.2 *Every integral ideal, $\mathfrak{a}$, of $\mathcal{O}$ has a $k[x]$-basis of the form $\{\lambda_1, \ldots, \lambda_n\}$, where $\lambda_i = \sum_{j=1}^{i} m_{i,j}\beta_j$, $m_{i,j} \in k[x]$ is monic, and $m_{i,i} \neq 0$ for all $1 \leq i \leq n$.*

Such a basis is called a *triangular* basis. Using the notation of this proposition, we denote $L(\mathfrak{a}) = m_{1,1}$.

A *fractional ideal* of $\mathcal{O}$ is a set of the form $\mathfrak{f} = \mathfrak{a}/d$, where $0 \neq d \in k[x]$ and $\mathfrak{a}$ is an integral ideal of $\mathcal{O}$. (We will typically write $\mathfrak{a}$ when referring specifically to an integral ideal, and $\mathfrak{f}$ when referring to a fractional ideal.) The unique monic polynomial of minimal degree $d = d(\mathfrak{f})$ such that $d\mathfrak{f}$ is an integral ideal is called the *denominator* of $\mathfrak{f}$. If $\{\lambda_1, \ldots, \lambda_n\}$ is a $k[x]$-basis of $\mathfrak{a}$, then $\mathfrak{f} = \mathfrak{a}/d = [\lambda_1/d, \ldots, \lambda_n/d]$. Like integral ideals, $\mathfrak{f}$ has a $k[x]$-basis of the form $\{\lambda_1, \ldots, \lambda_n\}$, but $\lambda_i = (l_{i,1} + l_{i,2}\beta_2 + \ldots + l_{i,n}\beta_n)/d$, with $l_{i,1}, \ldots, l_{i,n}, d \in k[x]$ collectively (though not necessarily pair-wise) coprime, for $1 \leq i \leq n$. Furthermore, if $\{L(\mathfrak{a}), \lambda_2, \ldots, \lambda_n\}$ is a triangular basis of $\mathfrak{a}$, then $\{1, \lambda_2/L(\mathfrak{a}), \ldots, \lambda_n/L(\mathfrak{a})\}$ is a $k[x]$-basis of $\mathfrak{f} = \mathfrak{a}/L(\mathfrak{a})$ such that $d(\mathfrak{f}) = L(\mathfrak{a})$. Notice that an integral ideal is a fractional ideal for which $d = 1$. Henceforth, we will assume that all fractional and integral ideals are nonzero.

The *norm* of a fractional ideal $\mathfrak{f}$ is the determinant

$$N(\mathfrak{f}) = a\frac{1}{d^n}\begin{vmatrix} l_{1,1} & l_{1,2} & \cdots & l_{1,n} \\ l_{2,1} & l_{2,2} & \cdots & l_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{n,1} & l_{n,2} & \cdots & l_{n,n} \end{vmatrix},$$

where $a \in k$ is chosen so that $N(\mathfrak{f})$ is monic. The norm of an integral ideal $\mathfrak{a}$ is defined in the same manner, except $d = 1$. For fractional ideals we have $N(\mathfrak{f}) \in k(x)$, and for integral ideals we have $N(\mathfrak{a}) \in k[x]$. Using the notation of Proposition 2.5.2, notice that from this definition of the norm of $\mathfrak{a}$, we have $N(\mathfrak{a}) = \prod_{i=1}^{n} m_{i,i}$, so $L(\mathfrak{a}) \mid N(\mathfrak{a})$.

The *discriminant* of an ideal $\mathfrak{f}$ (fractional or integral) is defined

$$\Delta(\mathfrak{f}) = a \begin{vmatrix} \lambda_1 & \lambda_1' & \cdots & \lambda_1^{(n-1)} \\ \lambda_2 & \lambda_2' & \cdots & \lambda_2^{(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_n & \lambda_n' & \cdots & \lambda_n^{(n-1)} \end{vmatrix}^2 ,$$

where $a \in k^*$ is chosen so that $\Delta(\mathfrak{f})$ is monic. Like the discriminant of $\mathcal{O}$, the discriminant of $\mathfrak{f}$ is independent of the basis of $\mathfrak{f}$. As with the norms, $\Delta(\mathfrak{f}) \in k(x)$ and $\Delta(\mathfrak{a}) = \Delta(d\mathfrak{f}) \in k[x]$. More specifically, we have the relation $\Delta(\mathfrak{f}) = a^2 N(\mathfrak{f})^2 \Delta$, for some $a \in k^*$. For an ideal $\mathfrak{a} \subseteq \mathcal{O}$, the *degree* of $\mathfrak{a}$ is defined to be $\deg(\mathfrak{a}) = \deg(N(\mathfrak{a}))$.

2.5.2 The Ideal Class Group

Given two fractional ideals, $\mathfrak{f}$ and $\mathfrak{g}$, of $\mathcal{O} = \mathcal{O}_x$, the product $\mathfrak{fg} = \{\sum_i a_i b_i \mid a_i \in \mathfrak{f},\, b_i \in \mathfrak{g}\}$, where the sum is finite. This is of course another fractional ideal, so the set of fractional ideals forms a group under multiplication, with identity $\mathcal{O}$. For any two fractional ideals $\mathfrak{f}$ and $\mathfrak{g}$, we have $N(\mathfrak{fg}) = N(\mathfrak{f})N(\mathfrak{g})$. An ideal is *principal* if it is generated by a single element, $a \in K$, and we write $\mathfrak{f} = \langle a \rangle$. Let $\mathcal{I}(\mathcal{O})$ denote the fractional ideals of $\mathcal{O}$ and let $\mathcal{P}(\mathcal{O})$ denote the subgroup of principal fractional ideals of $\mathcal{O}$. The *ideal class group* is the quotient group $Cl(\mathcal{O}) = \mathcal{I}(\mathcal{O})/\mathcal{P}(\mathcal{O})$. We will show in a moment that $Cl(\mathcal{O})$ is finite; its order, $h_x = |Cl(\mathcal{O})|$, is called the *(ideal) class number* of $\mathcal{O}$ (or of K_x). We will also see that h_x may differ depending on the curve defining the function field K.

We have seen the connection between places, points, and their divisors. Now we show the connection between divisors and fractional ideals. Recall the previous notation for divisors developed in Section 2.2. A result originally proved by Schmidt, [Sch31], and also given in Proposition 14.1 of [Ros02], gives two useful exact sequences that will form the basis for much of this thesis and the connection between divisors and ideals in particular. For a fixed model $K = K_x$, we have the following.

$$(0) \to k^* \to \mathcal{O}_S^* \to \mathcal{P}^S \to (0) \tag{2.1}$$

$$(0) \to \mathcal{D}_0^S/\mathcal{P}^S \to \mathcal{J}_K \to \mathcal{D}_S/\mathcal{P}_S \to \mathbb{Z}/f\mathbb{Z} \to (0) \tag{2.2}$$

With the following theorem we may relate many of these objects to ideals.

Theorem 2.5.3 (Theorem 14.5 of [Ros02]) *Let $K = K_x$ be a function field with constant field k and S the set of infinite places of K.*

1. *There exists an element $x \in K$ such that the poles of x are precisely the places of S.*
2. *For this element x, $\overline{k[x]} = \mathcal{O}_x = \mathcal{O}_S$.*
3. *$\mathcal{O}_x$ is a Dedekind domain.*
4. *There is a one-to-one correspondence between the non-zero prime ideals of $\mathcal{O}_x$ and prime divisors of K not contained in S.*

5. $\mathcal{D}_S/\mathcal{P}_S \cong Cl(\mathcal{O}_x)$.

Part 1 therefore justifies our notation $\mathcal{O} = \mathcal{O}_x$ and $K = K_x$ and shows that this model of K is characterized by its signature. Part 2 shows that in particular, the units, $\mathcal{O}^*$, of $\mathcal{O}$ are equal to the S-units $\mathcal{O}_S^*$, so that (2.1) may be rewritten

$$(0) \to k^* \to \mathcal{O}^* \to \mathcal{P}^S \to (0) \ . \tag{2.3}$$

Extending Part 4, there is an isomorphism between the divisors $\mathcal{D}_S$ and the nonzero fractional ideals of $\mathcal{O}$. This is the *Fundamental theorem of ideal theory in an algebraic function field*, Chapter 24, page 401 of [Has80].

$$\Phi : \mathcal{D}_S \to \mathcal{I}(\mathcal{O}), \ \ D_S \mapsto \{\alpha \in K^* \mid \mathrm{div}(\alpha)_S \geq D_S\} \tag{2.4}$$

The inverse of Φ is given by

$$\Phi^{-1} : \mathcal{I}(\mathcal{O}) \to \mathcal{D}_S, \ \ \mathfrak{f} \mapsto \sum_{\mathfrak{p} \notin S} m_{\mathfrak{p}} \mathfrak{p}, \quad \text{where } m_{\mathfrak{p}} = \min\{v_{\mathfrak{p}}(\alpha) \mid \alpha \in \mathfrak{f} \setminus \{0\}\} \ . \tag{2.5}$$

Furthermore, Part 5 shows that Φ extends to an isomorphism $\Phi : \mathcal{D}_S/\mathcal{P}_S \to Cl(\mathcal{O})$. Applying this to (2.2), we have

$$(0) \to \mathcal{D}_0^S/\mathcal{P}^S \to \mathcal{J}_K \to Cl(\mathcal{O}) \to \mathbb{Z}/f\mathbb{Z} \to (0) \ . \tag{2.6}$$

From this it follows that $fh = R^S h_x$. Since h is finite, the S-regulator and ideal class number of $\mathcal{O}$ are also finite. For the cases we will be concerned with, f will be trivial to compute, and either R^S or h_x will be small as well. Therefore, h can be found by computing R^S and h_x, and vice-versa.

2.5.3 Units

In this section, we will focus on the unit group $\mathcal{O}^* = \mathcal{O}_x^*$ and show its relationship with the group $\mathcal{P}^S$. As with number fields, Dirichlet's Unit Theorem applies to function fields.

Theorem 2.5.4 (Proposition 14.2 of [Ros02], Dirichlet's Unit Theorem) *Let $K = K_x$ be a function field with constant field k and $\mathcal{O} = \mathcal{O}_x$ its maximal order. If $|S| = r + 1$, then $\mathcal{P}^S \cong \mathbb{Z}^r$, and in particular, $\mathcal{O}^* \cong k^* \times \mathbb{Z}^r$.*

A set of generators of the free part of $\mathcal{O}^*$ is called a *system of fundamental units* of $\mathcal{O}$ and we write $\{\epsilon_1, \ldots, \epsilon_r\}$ for a given system of fundamental units. Throughout this thesis, r will denote the *unit rank* of $\mathcal{O} = \mathcal{O}_x$. If $r = 1$, then $\epsilon = \epsilon_1$ is called *the fundamental unit* of $\mathcal{O}$, and is unique up to inversion and multiplication by a constant in k^*. We can make this choice of ϵ unique up to a constant multiple, therefore, if we require $v_0(\epsilon) < 0$, for a fixed ordering of the infinite places of K_x.

Since $\mathcal{O}$ is a subring of a specific model of K that depends on the places at infinity, we may alternatively refer to the fundamental units and unit rank of K if we are working within the context of a specific model. K_x is called *totally imaginary* if $r = 0$, and *totally real* if $r = n - 1$. In general, there are more possibilities for the unit rank and the splitting of ∞, of course, but in the interest of simplicity, other models are typically characterized by $\mathrm{sig}(K_x)$. The motivation for these definitions comes from algebraic number fields. An imaginary quadratic field has only one place at infinity and its maximal order, therefore, has unit rank 0. Likewise if all of the n complex roots of a number field

are real, then the maximal order of the field has unit rank $n-1$. Recall our comment in Section 2.4 about the signature of ∞ for function fields versus number fields. It follows that the unit rank of a given model of a function field of degree n may take on any integer value between 0 and $n-1$, whereas for number fields of a fixed degree n, the limited possibilities for the signature allow only certain unit ranks to occur.

Recall that for any unit $\epsilon \in \mathcal{O}^*$, we have $N(\epsilon) \in k^*$. Therefore, $v_P(\epsilon) = 0$ for any finite place $P \in \mathbb{P}_K$, so $\mathrm{supp}(\mathrm{div}(\epsilon)) \subseteq S$ and $\mathrm{div}(\epsilon) \in \mathcal{P}^S$. By (2.3), we have $\mathcal{O}^* \cong k^* \times \mathcal{P}^S$. Therefore, if $\{\epsilon_1, \ldots, \epsilon_r\}$ is a system of fundamental units of $\mathcal{O}$, then $\mathrm{div}(\epsilon_i) = \sum_{j=0}^{r} v_j(\epsilon_i)\infty_j \in \mathcal{P}^S$ for each $1 \leq i \leq r$ and $\mathcal{P}^S = \langle \mathrm{div}(\epsilon_1), \ldots, \mathrm{div}(\epsilon_r) \rangle$.

Let $\{\epsilon_1, \ldots, \epsilon_r\}$ be a system of fundamental units of $\mathcal{O}$ and $f_i = \deg(\infty_i)$ for $0 \leq i \leq r$. We define the $r \times (r+1)$ integer matrix

$$M = M(\epsilon_1, \ldots, \epsilon_r) = \begin{pmatrix} -f_0 v_0(\epsilon_1) & -f_1 v_1(\epsilon_1) & \cdots & -f_r v_r(\epsilon_1) \\ -f_0 v_0(\epsilon_2) & -f_1 v_1(\epsilon_2) & \cdots & -f_r v_r(\epsilon_2) \\ \vdots & \vdots & \ddots & \vdots \\ -f_0 v_0(\epsilon_r) & -f_1 v_1(\epsilon_r) & \cdots & -f_r v_r(\epsilon_r) \end{pmatrix} .$$

Since $0 = \deg(\mathrm{div}(\epsilon_i)) = \sum_{j=0}^{r} -f_j v_j(\epsilon_i)$ for each $i \leq i \leq r$, the sum of the entries in each row of M is 0, so each $r \times r$ minor of M has the same absolute value. Furthermore, $\{\eta_1, \ldots, \eta_r\}$ is another system of fundamental units of $\mathcal{O}$ if and only if $(\epsilon_i)_{1 \leq i \leq r} A = (\eta_i)_{1 \leq i \leq r}$ for some $r \times r$ integer matrix A of determinant ± 1. Therefore, the absolute value of any $r \times r$ minor of M is also independent of the choice of system of fundamental units of $\mathcal{O}$. The absolute value of any $r \times r$ minor of M is called the *(x-)regulator* of $\mathcal{O}$, and is denoted $R_x = R_x(\mathcal{O})$.

Recall the S-regulator $R^S = \left[\mathcal{D}_0^S : \mathcal{P}^S\right]$. By Lemma 14.3 of [Ros02], we have

$$R^S = \frac{f R_x}{\prod_{i=0}^{r} f_i} \ , \tag{2.7}$$

where, as before, $f = \gcd(f_0, f_1, \ldots, f_r)$. Combining this with $fh = R^S h_x$, we have

$$\left(\prod_{i=0}^{r} f_i \right) h = R_x h_x \ . \tag{2.8}$$

Since any system of fundamental units of $\mathcal{O}$ is connected with the regulator, we will see that similar approaches can be used to compute both. However, it is much easier to compute R_x than it is to find a system of fundamental units. We gave the Hasse-Weil bounds for h earlier and also noted that one of R_x and h_x will tend to be small. Since $\prod_i f_i \leq n$ will be small, we have $R_x h_x \approx q^g$, so generally either R_x or h_x will be on the order of q^g. On the other hand, if $\{1, \beta_1, \ldots, \beta_n\}$ is an $\mathbb{F}_q[x]$-basis of $\mathcal{O}$, then the fundamental units of a field can be expressed in the form $\epsilon_i = a_{i,0} + a_{i,1}\beta_1 + \ldots + a_{i,n}\beta_n$, where $a_{i,j} \in k[x]$, for $0 \leq j \leq n$ and $1 \leq i \leq r$, and $\deg(a_{i,j})$ can be as large as R_x. So while the space required to represent R_x is $O(g \log(q))$, an integral basis representation of a fundamental unit requires space $O(q^g \log(q))$, and is therefore impossible to compute for fields of even moderate size and genus. We also note that R^S is the definition of the regulator given by Schmidt in [Sch31], while R_x is the definition given by Rosen in [Ros02].

In the next section, we will need a specific generating set of $\mathcal{P}^S$, so we conclude this section by describing how such a set may be determined uniquely. We will use the notion of the Hermite normal form of a matrix. The following definition may be found in Section 2.4.2 of [Coh93]. An $r \times r$ matrix, $A = (a_{i,j})$, with integer coefficients, is said to be in *Hermite normal form* if

- A is an upper triangular matrix, that is, $a_{i,j} = 0$ if $i > j$;
- $a_{i,i} > 0$ for every i; and
- $0 \leq a_{i,j} < a_{i,i}$ for every $j > i$.

If the set of column vectors of a matrix B is a basis of a $\mathbb{Z}$-submodule, Λ, of $\mathbb{Z}^r$, then we may determine a unique generating set of Λ by putting B into Hermite normal form.

Theorem 2.5.5 (Theorem 2.4.3 of [Coh93]) *If B is an $r \times r$ matrix with integer coefficients, then there is a unique $r \times r$ matrix $A = BU$ such that A is in Hermite normal form and $U \in GL_r(\mathbb{Z})$.*

Methods to determine the Hermite normal form of a matrix are given by Algorithms 2.4.4 and 2.4.5 in [Coh93].

One major application of this theorem is that every r-dimensional submodule, Λ, of $\mathbb{Z}^r$ has a unique (ordered) basis such that the matrix whose column vectors are the basis vectors of Λ is in Hermite normal form. If $\{\epsilon_1, \ldots, \epsilon_r\}$ is any system of fundamental units of $\mathcal{O}$, then the column vectors $\mathbf{e}_i = (v_j(\epsilon_i))_{1 \leq j \leq r}^{\top}$, for $1 \leq i \leq r$, span an r-dimensional submodule, Λ, of $\mathbb{Z}^r$. (If $\epsilon \in \mathcal{O}^*$, then $v_0(\epsilon)$ is uniquely determined by the other infinite valuations of ϵ, so we may omit this valuation.) There exists a unique ordered basis of Λ such that the matrix, $A = (a_{ij})$, whose columns are the basis vectors, is in Hermite normal form. Hence, there exists a transformation matrix $U = (u_{i,j}) \in GL_r(\mathbb{Z})$ such that $A = BU$, where B is the matrix whose columns are the vectors $\mathbf{e}_i$, for $1 \leq i \leq r$. The matrix A, in turn, defines a unique set of units, $\mathcal{U} = \mathcal{U}(\mathcal{O}) = \{\eta_1, \ldots, \eta_r\}$, via $\eta_i = \prod_{j=1}^{r} \epsilon_j^{u_{j,i}}$. Since $U \in GL_r(\mathbb{Z})$, $\mathcal{U}$ is a system of fundamental units of $\mathcal{O}$. Define $\mathcal{E} = \mathcal{E}(\mathcal{O}) = \{\operatorname{div}(\eta_i) \mid \eta_i \in \mathcal{U}\}$. Then $\mathcal{E}$ is a basis of $\mathcal{P}^S$. Topologically, $\mathbb{Z}^r/\Lambda$ is an r-dimensional torus. In particular, it is a circle if $r = 1$ and a torus if $r = 2$.

We noted earlier that if $r = 1$, then the choice of fundamental unit, $\epsilon \in \mathcal{O}^*$, such that $v_0(\epsilon) < 0$ is unique up to constant multiples. In this case, $\{\operatorname{div}(\epsilon)\}$ is a uniquely determined basis of $\mathcal{P}^S$. Since $v_1(\epsilon) > 0$, the 1×1 matrix $(v_1(\epsilon))$ is already in Hermite normal form, so this method is consistent with our earlier note. For arbitrary unit rank, we have $v_i(\eta_j) \geq 0$ for all $1 \leq i, j \leq r$, so $v_0(\eta_j) < 0$ for all $1 \leq j \leq r$.

With this result, we may choose a unique representative of each class in $\mathcal{D}_0^S/\mathcal{P}^S$, determined "modulo $\mathcal{E}$." More exactly, every class in $\mathcal{D}_0^S/\mathcal{P}^S$ contains a unique divisor, D_∞, such that $0 \leq v_i(D_\infty) < v_i(\eta_i) = a_{i,i}$, for each $1 \leq i \leq r$. Given a divisor $E_\infty \in \mathcal{D}_0^S$, we may obtain this representative $D_\infty \in [E_\infty]$ by applying the Euclidean algorithm. We recursively define the sequence of degree 0 divisors: $D_{r+1} = E_\infty$, D_r, D_{r-1}, $\ldots$, $D_1 = D_\infty$, where $D_i = D_{i+1} - q_i \operatorname{div}(\eta_i)$ and q_i is defined via $v_i(D_{i+1}) = q_i a_{i,i} + n_i$, with $0 \leq n_i < a_{i,i}$. In this way, the coordinate vector of D_∞, with respect to $\infty_1, \ldots, \infty_r$, lies in the fundamental domain of Λ, that is, the parallelepiped spanned by the coordinate vectors of the η_i, for $1 \leq i \leq r$, with respect to $\infty_1, \ldots, \infty_r$. We will call the divisor D_∞ the *minimal* representative of the class $[E_\infty]$, and will also write $D_\infty \equiv E_\infty \pmod{\mathcal{E}}$.

2.5.4 Relating Ideals and Divisors

In this section, we will highlight some important relationships between ideals and divisors to extend the notions that were developed in Section 2.5.2. The isomorphism Φ, defined in (2.4), established the correspondence of finite divisors of K and fractional ideals of $\mathcal{O} = \mathcal{O}_x$. Let $D \in \mathcal{D} = \mathcal{D}(K_x)$. D is said to be *effective* if $D \geq 0$, and its effective part is denoted by D^+. That is, if $D = \sum_{\mathfrak{P} \in \mathbb{P}_K} m_{\mathfrak{P}} \mathfrak{P}$, then $D^+ = \sum_{\mathfrak{P} \in \mathbb{P}_K, m_{\mathfrak{P}} > 0} m_{\mathfrak{P}} \mathfrak{P}$. D is said to be *finitely effective* if its finite part, D_S, is effective. A finitely effective divisor $D \in \mathcal{D}$ is called *semi-reduced* if there does not exist a non-empty sub-sum of D_S of the form $\operatorname{div}(a(x))_S$ for some $a(x) \in k[x] \setminus k$.

Because $\mathcal{O} = \mathcal{O}_x$ is a Dedekind domain, an ideal, $\mathfrak{a}$, of $\mathcal{O}$ factors uniquely into a product of prime ideals of $\mathcal{O}$, and the product of the norms of these primes is $N(\mathfrak{a})$. Under Φ, a place $\mathfrak{P} \in \mathbb{P}_K$ maps to a prime ideal $\mathfrak{p}$ of $\mathcal{O}$ of degree $\deg(\mathfrak{P})$. Notice that a divisor $D \in \mathcal{D}_S$ is finitely effective if and only if $\Phi(D)$ is an integral ideal of $\mathcal{O}$. An integral ideal $\mathfrak{a}$ is called *primitive* if and only if $\Phi^{-1}(\mathfrak{a})$ is a (finite) semi-reduced divisor. Equivalently, $\mathfrak{a}$ is primitive if and only if there is no polynomial $a(x) \in k[x]$ such that $\langle a(x) \rangle \mid \mathfrak{a}$. (See Definition 2.2 and Lemma 2.6 of [Bau04].) We also note the following observation, which is Lemma 2.7 of [Bau04]. We note that this result, proved for purely cubic function fields of signature sig(3, 1) holds for general function fields.

Lemma 2.5.6 (Lemma 2.7 of [Bau04]) *If $D \in \mathcal{D}_0$ is finitely effective and $\mathfrak{a} = \Phi(D_S)$, then $\mathfrak{a}$ is an integral ideal and* $\deg(D_S) = \deg(\mathfrak{a})$.

As another observation, notice that if D, $E \in \mathcal{D}$, $\mathfrak{f} = \Phi(D_S)$, and $\mathfrak{g} = \Phi(E_S)$, then $\Phi((D+E)_S) = \Phi(D_S + E_S) = \Phi(D_S)\Phi(E_S) = \mathfrak{fg}$. If $\alpha \in K^*$, then $\Phi(\operatorname{div}(\alpha)_S) = \langle \alpha \rangle$.

We conclude this section with an important result on how to invert an ideal in the ideal class group. We will also use this result to define the notion of a reduced ideal.

Lemma 2.5.7 *If $\mathfrak{a}$ is an ideal of $\mathcal{O} = \mathcal{O}_x$, then there is a primitive ideal, which we denote $\overline{\mathfrak{a}}$, such that $\mathfrak{a}\overline{\mathfrak{a}} = \langle L(\mathfrak{a}) \rangle$.*

Proof: Since $L(\mathfrak{a}) \in \mathfrak{a}$, we have $\langle L(\mathfrak{a}) \rangle \subseteq \mathfrak{a}$, so there is an integral ideal, $\mathfrak{b}$, such that $\mathfrak{ab} = \langle L(\mathfrak{a}) \rangle$. Thus, $\Phi^{-1}(\mathfrak{a}) + \Phi^{-1}(\mathfrak{b}) = \operatorname{div}(L(\mathfrak{a}))_S$. $L(\mathfrak{a}) \in k[x]$ is the polynomial of smallest degree in $\mathfrak{a}$, so $\Phi^{-1}(\mathfrak{a}) \leq \operatorname{div}(L(\mathfrak{a}))_S$, and $L(\mathfrak{a})$ has the smallest possible degree such that $\Phi^{-1}(\mathfrak{a}) - \operatorname{div}(L(\mathfrak{a}))_S \leq 0$. Suppose that $\Phi^{-1}(\mathfrak{b})$ is not semi-reduced. Then there is some polynomial, $a(x) \in k[x] \setminus k$, such that $\Phi^{-1}(\mathfrak{b}) - \operatorname{div}(a(x))_S \geq 0$. Then $\Phi^{-1}(\mathfrak{a}) - \operatorname{div}(L(\mathfrak{a}))_S \leq -\operatorname{div}(a(x))_S$, so $\Phi^{-1}(\mathfrak{a}) \leq \operatorname{div}(L(\mathfrak{a})/a(x))_S$. Thus, $L(\mathfrak{a})/a(x) \in \mathfrak{a}$, but the norm of any element of $\mathfrak{a}$ must be in $k[x]$, so $a(x) \mid L(\mathfrak{a})$. This, however, contradicts the minimality of the degree of $L(\mathfrak{a})$. Therefore, $\Phi^{-1}(\mathfrak{b})$ is semi-reduced, so $\mathfrak{b}$ is primitive. □

Since $\mathfrak{a}\overline{\mathfrak{a}}$ is principal, the classes $[\mathfrak{a}]$ and $[\overline{\mathfrak{a}}]$ are inverses in $Cl(\mathcal{O})$.

In this chapter, we provided a general overview of function fields, divisors, ideals, class groups, and units. We turn now to applying these concepts to one of our main objects of interest, the infrastructure of a cubic function field defined over a finite field.

Chapter 3

Divisors, Ideals, and Infrastructure of Cubic Function Fields

In Chapter 2, we presented an overview of function fields and described divisors and ideals in a relatively general context. In this chapter, we will focus our attention on cubic function fields over a finite field, and will concentrate in particular on infrastructures of purely cubic function fields of characteristic at least 5. In the first section, we will present notation specific to cubic function fields, and in Section 3.2, we will list each possible signature of a cubic function field, giving properties of each signature. The notions of reduced and distinguished divisors and ideals will be defined in Section 3.3. A key result of this section establishes the existence of a unique divisor class representative for all but one signature. Finally, we will use this notion of a distinguished divisor to define the infrastructure of an ideal class in Section 3.4, along with a distance measure on the infrastructure divisors. Therefore, the treatment of infrastructure we give in this chapter differs from the more traditional ideal-theoretic approach.

The original description of the infrastructure of unit rank 1 cubic function fields is given in [SS00, Sch01], and the infrastructure of unit rank 2 cubic function fields was used to compute the system of fundamental units and regulator in [LSY03]. The new results of this chapter therefore extend the work in those sources to describe the infrastructure of unit rank 2 cubic function fields in more complete detail. Moreover, the divisor theoretic description of infrastructure has led to improved results on the size of infrastructure elements and provides one with a more intuitive idea of the behavior of infrastructure. A description of the arithmetic of infrastructure, along with further improvements will be given in Chapter 5.

3.1 Cubic Function Fields

For the remainder of this thesis, we will restrict ourselves to the case that $k = \mathbb{F}_q$ and $K/\mathbb{F}_q(x)$ is a separable extension of degree 3. Such a function field is called a *cubic* function field. Let $F(x, Y) \in \mathbb{F}_q[x, Y]$ be an absolutely irreducible polynomial such that $K \cong \mathbb{F}_q(C)$, with $C : F(x, Y) = 0$. At this point, we note that while the previous concepts that we have developed hold for any characteristic, we will assume that $\mathrm{char}(\mathbb{F}_q) \geq 5$ in this chapter. If we apply a suitable translation $Y \mapsto Y - a(x)$, where $a(x) \in \mathbb{F}_q[x]$, we may write C in the form $C : F(x, Y) = Y^3 - AY + B = 0$, where $A, B \in \mathbb{F}_q[x]$, and $F(x, Y)$ is absolutely irreducible. By Proposition 2.1.1, $F(x, Y)$ is irreducible if and only if F is irreducible over $\mathbb{F}_q(x)$ and $\mathbb{F}_q$ is the full constant field of K; this is not a heavy restriction. In this case, the discriminant of $F(x, Y)$ is $\Delta(F) = 4A^3 - 27B^2$. If there does not exist a polynomial $Q \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$ such that $Q^2 \mid A$ and $Q^3 \mid B$, then $F(x, Y)$ is said to be in *standard form*. We will assume henceforth that $F(x, Y)$ is such a polynomial. From Section 2.5, we have $\Delta(F) = I_F^2 \Delta$,

where Δ is the discriminant of K and I_F is the index of F.

3.1.1 Purely Cubic Function Fields

We will now only consider the situation where $A = 0$. We call such a function field $K = \mathbb{F}_q(C)$ a *purely* cubic function field. In this case, we will write $C : Y^3 = GH^2$, where $G, H \in \mathbb{F}_q[x]$, $\gcd(G, H) = 1$, and both G and H are square-free. Because of these restrictions, there is no nonconstant polynomial $Q \in \mathbb{F}_q[x]$ such that $Q^3 \mid GH^2$, so $Y^3 = GH^2$ is in standard form. Since K is a purely cubic function field, we may write $K = \mathbb{F}_q(x, \rho)$, where $\rho \in K$ satisfies $\rho^3 = GH^2$. For any such ρ, if $\omega = \rho^2/H$, then $\{1, \rho, \omega\}$ is an $\mathbb{F}_q(x)$-basis of $K/\mathbb{F}_q(x)$ and is independent of the particular cube root of ρ chosen. (We will choose a specific basis in the next section.) Moreover, from Corollary 3.2 of [Sch04] or Theorem 6.4 of [LRS$^+$08], $\{1, \rho, \omega\}$ is an $\mathbb{F}_q[x]$-basis of $\mathcal{O} = \mathcal{O}_x$. The cube roots of GH^2 in $\overline{\mathbb{F}}_q\langle x\rangle$ are ρ, $\iota\rho$, and $\iota^2\rho$, where ι is a primitive cube root of unity in $\mathbb{F}_{q^d}$, $d = 1$ if $q \equiv 1 \pmod 3$, and $d = 2$ if $q \equiv 2 \pmod 3$. Therefore, if $\alpha = a + b\rho + c\omega \in K$, then $\alpha' = a + b\iota\rho + c\iota^2\omega$ and $\alpha'' = a + b\iota^2\rho + c\iota\omega$ are the conjugates of α. From this, we have

$$N(\alpha) = \alpha\alpha'\alpha'' = a^3 + b^3GH^2 + c^3G^2H - 3abcGH \in \mathbb{F}_q(x) \tag{3.1}$$

and $\Delta = \Delta(K) = -27(GH)^2$. Notice that $\Delta(F) = -27G^2H^4$, so that $I_F = aH$, for some $a \in \left(\mathbb{F}_q^*\right)^2$. Lastly, if $\mathfrak{a}$ is an ideal, then $\mathfrak{a}^{(i)} = \left\{\alpha^{(i)} \mid \alpha \in \mathfrak{a}\right\}$, for $i = 1, 2$, and $\mathfrak{a}'$ and $\mathfrak{a}''$ are called the *conjugates* of $\mathfrak{a}$. For any ideal $\mathfrak{a}$, we have $\langle N(\mathfrak{a})\rangle = \mathfrak{a}\mathfrak{a}'\mathfrak{a}''$ and $N\left(\mathfrak{a}^{(i)}\right) = N(\mathfrak{a})$, for $i = 0, 1, 2$.

We continue by considering the properties of cubic function fields of each possible signature.

3.2 Possible Signatures

As opposed to cubic number fields, which have either 2 or 3 infinite places, cubic function fields may have 1, 2, or 3 infinite places, and there are five possible ways that ∞ splits in K: $\infty = \infty_0$, $\infty = \infty^3$, $\infty = \infty_0\infty_1$, $\infty = \infty_0\infty_1^2$, and $\infty = \infty_0\infty_1\infty_2$. In this section, we will gather information about the class groups, completions of K, unit group, and the regulator of K_x in each of these five cases. We first present criteria for determining the splitting of ∞ for a given curve C.

Theorem 3.2.1 (Theorem 8.1 of [LRS$^+$08]) *Let $K = \mathbb{F}_q(C)$ be a cubic function field such that $C : F(x, Y) = Y^3 - AY + B = 0$ is in standard form, $D = \Delta(F)$ is the discriminant of $F(x, Y)$, and $\iota \in \overline{\mathbb{F}}_q$ is a primitive cube root of unity. If $\deg(D)$ is even and $\operatorname{sgn}(D) \in \mathbb{F}_q^2$, set $s = -3\operatorname{sgn}\left(9B + \sqrt{-3D}\right)/2 \in \mathbb{F}_q(\iota)$. Then $\operatorname{sig}(K)$ is*

- $(1, 1; 1, 1; 1, 1)$ *if*
  - $3\deg(A) > 2\deg(B)$, $2 \mid \deg(A)$, *and* $\operatorname{sgn}(A) \in \mathbb{F}_q^2$; *or*
  - $3\deg(A) < 2\deg(B)$, $3 \mid \deg(B)$, $\operatorname{sgn}(B) \in \mathbb{F}_q^3$, *and* $q \equiv 1 \pmod 3$; *or*
  - $3\deg(A) = 2\deg(B)$, $2 \mid \deg(D)$, $\operatorname{sgn}(D) \in \mathbb{F}_q^2$, *and* $s \in \mathbb{F}_q(\iota)^3$.
- $(1, 1; 1, 2)$ *if*
  - $3\deg(A) > 2\deg(B)$, $2 \mid \deg(A)$, *and* $\operatorname{sgn}(A) \notin \mathbb{F}_q^2$; *or*

- $3\deg(A) < 2\deg(B)$, $3 \mid \deg(B)$, *and* $q \equiv 2 \pmod 3$; *or*
- $3\deg(A) = 2\deg(B)$, $2 \mid \deg(D)$, *and* $\operatorname{sgn}(D) \notin \mathbb{F}_q^2$.

- $(1,3)$ *if*
  - $3\deg(A) < 2\deg(B)$, $3 \mid \deg(B)$, *and* $\operatorname{sgn}(B) \notin \mathbb{F}_q^3$; *or*
  - $3\deg(A) = 2\deg(B)$, $2 \mid \deg(D)$, $\operatorname{sgn}(D) \in \mathbb{F}_q^2$, *and* $s \notin \mathbb{F}_q(\iota)^3$.

- $(1,1;2,1)$ *if*
  - $3\deg(A) > 2\deg(B)$ *and* $2 \nmid \deg(A)$; *or*
  - $3\deg(A) = 2\deg(B)$ *and* $2 \nmid \deg(D)$.

- $(3,1)$ *if* $3\deg(A) < 2\deg(B)$ *and* $3 \nmid \deg(B)$.

Notice that the signature $(1,1;2,1)$ is impossible if $K = \mathbb{F}_q(C)$ is a purely cubic function field. If, in addition, $C : Y^3 = GH^2$ and GH^2 is monic, then the signature $(1,3)$ is impossible as well. One advantage of this theorem is that the signature of a cubic function field, determined by a given curve in standard form, may be determined quickly. Since the primary focus of this thesis is on purely cubic function fields, we give the following corollary of Theorem 3.2.1.

Corollary 3.2.2 *If $K = \mathbb{F}_q(C)$ is a purely cubic function field such that $C : Y^3 = F(x)$, then* $\operatorname{sig}(K)$ *is*

- $(1,1;1,1;1,1)$ *if* $3 \mid \deg(F)$, $\operatorname{sgn}(F) \in \mathbb{F}_q^3$, *and* $q \equiv 1 \pmod 3$.
- $(1,1;1,2)$ *if* $3 \mid \deg(F)$ *and* $q \equiv 2 \pmod 3$.
- $(1,3)$ *if* $3 \mid \deg(F)$ *and* $\operatorname{sgn}(F) \notin \mathbb{F}_q^3$.
- $(3,1)$ *if* $3 \nmid \deg(F)$.

We note that whenever K has an infinite place of degree 1, we can replace the curve $C : Y^3 = F$ by an isomorphic curve, $C_1 : Y^3 = F_1$, where F_1 is monic, without changing the signature. Obviously, if $\operatorname{sgn}(F) \in \mathbb{F}_q^3$, then we simply divide the curve by $\operatorname{sgn}(F)$ and replace Y by $Y/\operatorname{sgn}(F)^{1/3}$. This does not change the signature. Note that if $q \equiv 2 \pmod 3$, then this is always possible because every element in $\mathbb{F}_q$ is a cube, as $a^{(2q-1)/3}$ is the unique cube root of $a \in \mathbb{F}_q$. Now suppose that $q \equiv 1 \pmod 3$ and $\operatorname{sgn}(F) \notin \mathbb{F}_q^3$. Then $3 \nmid \deg(F)$, since otherwise $\operatorname{sig}(K) = (1,3)$, contradicting the assumption that K has an infinite place of degree 1. We then switch to an isomorphic curve as follows. If $\deg(F) \equiv 2 \pmod 3$, then we may write $\deg(F) = 3n+2$ and $C : Y^3 = ax^{3n+2} + f_{3n+1}x^{3n+1} + \cdots + f_0$, with $a, f_i \in \mathbb{F}_q$, for $0 \le i \le 3n+1$. Dividing C by a^{3n+3}, we have $C : \left(Y/a^{n+1}\right)^3 = (x/a)^{3n+2} + \left(f_{3n+1}/a^2\right)(x/a)^{3n+1} + \cdots + f_0$. Replacing Y by Y/a^{n+1} and x by x/a, we obtain $C_1 : Y^3 = x^{3n+2} + \left(f_{3n+1}/a^2\right)x^{3n+1} + \cdots + f_0 = F_1$, with F_1 monic. Next, if $\deg(F) \equiv 1 \pmod 3$, then we may write $\deg(F) = 3n+1$ and $C : Y^3 = ax^{3n+1} + f_{3n}x^{3n} + \cdots + f_0$, with $a, f_i \in \mathbb{F}_q$, for $0 \le i \le 3n$. Dividing C by a^{6n+3}, we have $C : \left(Y/a^{2n+1}\right)^3 = \left(x/a^2\right)^{3n+1} + \left(f_{3n}/a^3\right)(x/a^2)^{3n} + \cdots + f_0$. Replacing Y by Y/a^{2n+1} and x by x/a^2, we obtain $C_1 : Y^3 = x^{3n+1} + \left(f_{3n}/a^3\right)x^{3n} + \cdots + f_0 = F_1$, with F_1 monic. None of these substitutions changes the signature. For the case in which $q \equiv 1 \pmod 3$,

$3 \mid \deg(F)$, and $\operatorname{sgn}(F) \notin \mathbb{F}_q^3$, no such substitution is possible. In this case, the place of K at infinity is inert and does not have degree 1.

Using the signatures of Theorem 3.2.1 and Corollary 3.2.2, we apply the Hurwitz Genus Formula (Theorem 2.5.1) to obtain a convenient formula the genus of K. (See (3.2) and Section 8 of [LRS+08].)

$$g = \frac{1}{2}(\deg(\Delta) + \epsilon_\infty(K)) - 2 \ , \tag{3.2}$$

where

$$\epsilon_\infty(K) = \begin{cases} 2 & \text{if } 3\deg(A) < 2\deg(B) \text{ and } 3 \nmid \deg(B) \\ 1 & \text{if } 3\deg(A) \geq 2\deg(B) \text{ and } 3 \nmid \deg(\Delta(F)) \\ 0 & \text{otherwise .} \end{cases}$$

If K is a purely cubic function field, with defining curve $C : Y^3 = GH^2$, then (3.2) simplifies to

$$g = \begin{cases} \deg(GH) - 2 & \text{if } 3 \mid \deg(GH^2) \\ \deg(GH) - 1 & \text{if } 3 \nmid \deg(GH^2) \ . \end{cases} \tag{3.3}$$

We note that if K is a cubic function field, then there may exist birationally equivalent curves, C_1 and C_2, such that $K \cong \mathbb{F}_q(C_1) \cong \mathbb{F}_q(C_2)$, and $\operatorname{sig}(\mathbb{F}_q(C_1)) \neq \operatorname{sig}(\mathbb{F}_q(C_2))$. For example, let $F_1(x, Y) = Y^3 - \left(x^4 + x + 1\right) \in \mathbb{F}_7[x, Y]$. We first show that $F_1(x, Y)$ is absolutely irreducible in $\mathbb{F}_7[x, Y]$. If not, then $Y - a(x)$ would be a factor of $F_1(x, Y) = 0$, for some $a(x) \in \overline{\mathbb{F}}_7[x]$. Then $a^3(x) - \left(x^4 + x + 1\right) = 0$, so $3\deg(a) = \deg\left(a^3\right) = \deg\left(x^4 + x + 1\right) = 4$ and $\deg(a) = 4/3$, which is impossible. Now let $C_1 : F_1(x, Y) = 0$ and $K_x = \mathbb{F}_7(C_1)$. By the criteria in Theorem 3.2.1, we have $\operatorname{sig}(K_x) = (3, 1)$. We then make a birational transformation from C_1 to another curve C_2. Substituting $x = 1/u$ and $y = v/u^2$ into C_1, we obtain $v^3/u^6 = 1/u^4 + 1/u + 1$. Multiplying this through by u^6, we have $v^3 = u^6 + u^5 + u^2 = \left(u^4 + u^3 + 1\right) u^2$. Let $C_2 : v^3 = \left(u^4 + u^3 + 1\right) u^2$ and $K_u = \mathbb{F}_7(C_2)$. Then $K_x \cong K_u$, but $\operatorname{sig}(K_u) = (1, 1; 1, 1; 1, 1)$.

We will now consider various characteristics of the different splitting behaviors. After this, we will focus our attention on unramified unit rank 1 and unit rank 2 cubic function fields. A similar characterization for quadratic function fields is found in Sections 3 and 4 of [JSS07b].

3.2.1 Totally Inert Fields: $\operatorname{sig}(\mathbb{F}_q(C)) = (1, 3)$

In this case, the place at infinity of $\mathbb{F}_q(x)$ is inert in the function field $K = K_x = \mathbb{F}_q(C)$, that is, $\infty = \infty_0$. Such fields are also called *unusual imaginary* fields. We have $S = \{\infty_0\}$, so $r = 0$ and $\mathcal{O}_x^* \cong \mathbb{F}_q^*$. The groups $\mathcal{D}_0^S$ and $\mathcal{P}^S$ are trivial so that $R^S = \left[\mathcal{D}_0^S : \mathcal{P}^S\right] = 1$. Since $\deg(\infty_0) = 3$, we have $f = f_0 = 3$, so $R_x = R^S = 1$. By (2.8), we have $3h = h_x$, so the ideal and divisor class groups of K are the objects of interest. If $D \in \mathcal{D}_0$, then $D = D_S - (\deg(D_S)/3)\infty_0$, so $3 \mid \deg(D_S)$. Since D is uniquely determined by D_S, under the given constraint, the isomorphism Φ, defined in (2.4), extends naturally to an isomorphism $\Psi : \mathcal{D}_0 \to \mathcal{I}_3$, where $\mathcal{I}_3$ is the group of fractional ideals of $\mathcal{O}_x$ whose norms have degree divisible by 3. Likewise, Φ^{-1} extends to the isomorphism Ψ^{-1}.

In this case, there is only one embedding of K into its completion, $K_{\infty_0} \cong \mathbb{F}_{q^3}\left\langle x^{-1}\right\rangle$, and for a function $\alpha \in K_{\infty_0}$, we have $\deg(\alpha) = -3v_0(\alpha)$. However, if we consider C over $\mathbb{F}_{q^3}$, then $K_x' = K/\mathbb{F}_{q^3}(x)$ will have three embeddings into $\mathbb{F}_q\left\langle x^{-1}\right\rangle$, so that $\operatorname{sig}\left(K_x'\right) = (1, 1; 1, 1; 1, 1)$, hence one reason for calling such models unusual. There is also no number field analogue to this case. We

will not consider these models much further since they do not occur for purely cubic function fields in which GH^2 is monic.

3.2.2 Totally Ramified Fields: $\mathrm{sig}(\mathbb{F}_q(C)) = (3, 1)$

In this case, the place at infinity of $\mathbb{F}_q(x)$ totally ramifies in the function field $K = K_x = \mathbb{F}_q(C)$, that is, $\infty = \infty_0^3$. Recall that such fields are called totally imaginary. Again we have $S = \{\infty_0\}$, $r = 0$, $\mathcal{O}_x^* \cong \mathbb{F}_q^*$, and $R^S = R_x = 1$, but now $f = f_0 = 1$ so that $h = h_x$, and the completion $K_{\infty_0} = \mathbb{F}_q\left\langle x^{-1/3}\right\rangle$. For a function $\alpha \in K_{\infty_0}$, we have $\deg(\alpha) = -v_0(\alpha)$. Since $\mathcal{D}_0^S/\mathcal{P}^S$ is trivial, the ideal and divisor class groups of K are again the objects of interest, and by (2.6), we in fact have $\mathcal{J}_K \cong Cl(K)$. If $D \in \mathcal{D}_0$, then $D = D_S - \deg(D_S)\infty_0$. Since D is uniquely determined by D_S, Φ extends naturally to an isomorphism $\Psi : \mathcal{D}_0 \to \mathcal{I}$, and hence, to the isomorphism $\Psi : \mathcal{J}_K \to Cl(K)$. Likewise, Φ^{-1} extends to the isomorphism Ψ^{-1}. Therefore, by operating in the ideal class group, we simultaneously work in the divisor class group. In [Bau04], Bauer developed ideal arithmetic to compute in $Cl(K)$ for fields of this signature, with C nonsingular. In Chapter 4, we will generalize this arithmetic to nonsingular models as well, and in Chapter 6, we will present results on computing h for fields of this signature. In Section 3.3 in this chapter, we will prove the existence of a unique representative of each ideal (and hence, divisor) class. We will call such ideals and divisors *distinguished*.

3.2.3 Partially Ramified Fields: $\mathrm{sig}(\mathbb{F}_q(C)) = (1, 1; 2, 1)$

In this case, the place at infinity of $\mathbb{F}_q(x)$ splits partially and ramifies partially in $K = K_x = \mathbb{F}_q(C)$, that is, $\infty = \infty_0\infty_1^2$. Here, we have $S = \{\infty_0, \infty_1\}$ and $r = 1$, so $\mathcal{O}_x^* \cong \mathbb{F}_q^* \times \mathbb{Z}$, and there is a unique fundamental unit, ϵ, such that $\deg(\epsilon) > 0$. Now, $e_0 = 1$, $e_1 = 2$, and $f = f_0 = f_1 = 1$, so the completions of K are $K_{\infty_0} = \mathbb{F}_q\left\langle x^{-1}\right\rangle$ and $K_{\infty_1} = \mathbb{F}_q\left\langle x^{-1/2}\right\rangle$. For a function $\alpha \in K_{\infty_0}$, we have $\deg(\alpha) = -v_0(\alpha)$. We also have $R_x = R^S$. Since $|S| = 2$, the group $\mathcal{D}_0^S$ is nontrivial, and we have $\mathcal{D}_0^S = \langle \infty_1 - \infty_0 \rangle$. Therefore, R_x is the order of the divisor class $[\infty_1 - \infty_0]$ in $\mathcal{P}^S$ and $\mathcal{P}^S = \langle R_x(\infty_1 - \infty_0)\rangle$. The exact sequence (2.3) states that $\mathcal{O}_x$ maps onto $\mathcal{P}^S$ with kernel $\mathbb{F}_q^*$ so that $\mathcal{P}^S \cong \mathbb{Z}$. Since $R_x(\infty_1 - \infty_0)$ generates $\mathcal{P}^S$ and ϵ generates the free part of $\mathcal{O}^*$, we have $\mathrm{div}(\epsilon) = R_x(\infty_1 - \infty_0)$. Thus, $R_x = \deg(\epsilon)$ is the regulator of K and $h = R_x h_x$. As with every other field of positive unit rank that has been studied, we expect h_x to be small in general, but there is no evidence, experimental or theoretical, to support that claim for fields of this signature. Nevertheless, we believe it is a fair assumption, so that, aside from h, R_x is the main quantity of interest for fields of this signature.

Given a divisor $D \in \mathcal{D}_0$, we may write D uniquely as $D = D_S - \deg(D_S)\infty_0 + v_1(D)(\infty_1 - \infty_0)$. Since any divisor in $\mathcal{D}_0$ is uniquely determined by D_S and $v_1(D)$, we may extend Φ in this case to an isomorphism $\Psi : \{D \in \mathcal{D}_0 \mid v_1(D) = 0\} \to \mathcal{I}$. Likewise, we may extend Φ^{-1} to a similar isomorphism Ψ^{-1}. Since cubic function fields of this signature are not purely cubic, we will not consider this case beyond Section 3.3.

3.2.4 Unramified, Partially Split Fields: $\mathrm{sig}(\mathbb{F}_q(C)) = (1, 1; 1, 2)$

In this case, the infinite place of $\mathbb{F}_q(x)$ splits partially, but does not ramify in $K = K_x = \mathbb{F}_q(C)$. In other words, $\infty = \infty_0\infty_1$. Like the previous case, we have $S = \{\infty_0, \infty_1\}$ and $r = 1$, so $\mathcal{O}_x^* \cong \mathbb{F}_q^* \times \mathbb{Z}$, and there is a unique fundamental unit, ϵ, such that $\deg(\epsilon) > 0$. For both infinite places, we have $e_i = 1$, but $f_0 = 1$ and $f_1 = 2$, so the completions of K at the places in S are $K_{\infty_0} = \mathbb{F}_q\left\langle x^{-1}\right\rangle$ and $K_{\infty_1} = \mathbb{F}_{q^2}\left\langle x^{-1}\right\rangle$. For a function $\alpha \in K_{\infty_0}$, we have $\deg(\alpha) = -v_0(\alpha)$. Here we have $\mathcal{D}_0^S = \langle \infty_1 - 2\infty_0\rangle$ and $\mathcal{P}^S = \left\langle R^S(\infty_1 - 2\infty_0)\right\rangle$, so $\mathrm{div}(\epsilon) = R^S(\infty_1 - 2\infty_0)$. By (2.7) and (2.8), we have $2R^S = R_x$ and $2h = R_x h_x$, so the regulator $R_x = \deg(\epsilon) = 2v_1(\epsilon)$. Based on the numerical results in Section 9 and Table 1 of [SS00] as well as our computations discussed in Section 6.4.6, we again expect h_x to be small in general, but this has not been proved theoretically. However, we again make this assumption, so that h and R_x are the quantities of interest. In Chapter 6, we will give results on computing these quantities for function fields of this signature.

In the case of signature $(1, 1; 1, 2)$, we may write $D \in \mathcal{D}_0$ as $D = D_S - \deg(D_S)\infty_0 + v_1(D)(\infty_1 - 2\infty_0)$. Then D is uniquely determined by D_S and $v_1(D)$, so Φ extends to an isomorphism, Ψ, from the subgroup of $\mathcal{D}_0$ with $v_1(D) = 0$ onto $\mathcal{I}$, and Φ^{-1} extends to Ψ^{-1}.

We also note that by the "Prolongation" (Extension) Theorem, Chapter 12, Page 183 of [Has80], v_1 extends to $\mathbb{F}_q(\iota)\left\langle x^{-1}\right\rangle$. Therefore, if $\alpha \in K$, then $\deg(\alpha') = \deg(\alpha'\alpha'')/2$ and $v_1(\alpha) = -2\deg(\alpha')$.

This case, along with the unit rank 2 case, will represent the two main cases of interest in this chapter. In particular, we will define and describe the infrastructure of each ideal class for fields of these two signatures.

3.2.5 (Totally) Split Fields: $\mathrm{sig}(\mathbb{F}_q(C)) = (1, 1; 1, 1; 1, 1)$

In this setting, the infinite place of $\mathbb{F}_q(x)$ splits completely in $K = K_x = \mathbb{F}_q(C)$: $\infty = \infty_0\infty_1\infty_2$. Recall that such fields K are called totally real. Here, we have $S = \{\infty_0, \infty_1, \infty_2\}$, $r = 2$, and $\mathcal{O}_x^* \cong \mathbb{F}_q^* \times \mathbb{Z}^2$, so $\mathcal{O} = \mathcal{O}_x$ has a system of fundamental units $\{\epsilon_1, \epsilon_2\}$ that generates the free part of the unit group. We will assume that $\mathcal{U} = \mathcal{U}(\mathcal{O}) = \{\epsilon_1, \epsilon_2\}$ is the system determined by the Hermite normal form procedure described in Section 2.5.3. Notice that $e_i = f_i = 1$ for $i \in \{0, 1, 2\}$, so $f = 1$, $R^S = R_x$, and $h = R_x h_x$. We also have three embeddings of K into $\mathbb{F}_q\left\langle x^{-1}\right\rangle$, which is the completion of K with respect to each $\infty_i \in S$. For an element $\alpha \in \mathbb{F}_q\left\langle x^{-1}\right\rangle$, we have $\deg(\alpha) = -v_0(\alpha)$.

Considering the exact sequence (2.3), we have $\mathcal{P}^S = \langle \mathrm{div}(\epsilon_1), \mathrm{div}(\epsilon_2)\rangle$. By definition, we have $\mathcal{D}_0^S = \{a\infty_0 + b\infty_1 - (a + b)\infty_2 \mid a, b \in \mathbb{Z}\}$, so $\mathcal{D}_0^S$ has two generators, and we may express $\mathcal{D}_0^S$ equivalently in several ways. To be consistent with earlier and later notation, we will write

$$\mathcal{D}_0^S = \langle \infty_1 - \infty_0, \infty_2 - \infty_0\rangle \ .$$

Then,

$$\mathrm{div}(\epsilon_i) = -\deg(\epsilon_i')(\infty_1 - \infty_0) - \deg(\epsilon_i'')(\infty_2 - \infty_0) \ ,$$

for $i = 1, 2$, and $\mathcal{E} = \{\mathrm{div}(\epsilon_1), \mathrm{div}(\epsilon_2)\}$. The regulator, R_x, of K is the absolute value of any 2×2 minor of the matrix

$$\begin{pmatrix} \deg(\epsilon_1) & \deg(\epsilon_1') & \deg(\epsilon_1'') \\ \deg(\epsilon_2) & \deg(\epsilon_2') & \deg(\epsilon_2'') \end{pmatrix} \ .$$

As with the fields of unit rank 1, we expect h_x to be small, though this has not been proved

theoretically. Thus, h and R_x are the quantities of interest.

If $D \in \mathcal{D}_0$, then we can write D as

$$D = D_S - \deg(D_S)\infty_0 + v_1(D)(\infty_1 - \infty_0) + v_2(D)(\infty_2 - \infty_0) \ . \tag{3.4}$$

Any divisor of $\mathcal{D}_0$ is uniquely determined by D_S, $v_1(D)$, and $v_2(D)$. Thus, Φ extends to an isomorphism, Ψ, from the subgroup of $\mathcal{D}_0$ of divisors D, with $v_1(D) = v_2(D) = 0$ onto $\mathcal{I}$, and Φ^{-1} extends to the corresponding isomorphism Ψ^{-1}.

In the cases that we are concerned with, namely function fields of signatures $(1, 1; 1, 2)$ and $(1, 1; 1, 1; 1, 1)$, we may choose $G(x)$ and $H(x)$ to be monic, and we will henceforth assume that they are. Likewise, we will choose the basis element $\rho \in K$ such that $\rho^3 = GH^2$ and $\mathrm{sgn}(\rho) = 1$, when considered as an element in K_{∞_0}. As before, we choose $\omega = \rho^2/H$ so that $\mathrm{sgn}(\omega) = 1$ in the completion, K_{∞_0}, as well, so that $\{1, \rho, \omega\}$ is a basis of $K/\mathbb{F}_q(x)$.

In the first two sections, we covered general properties of cubic function fields, including attributes specific to each signature. In particular, we noted properties surrounding the divisor class group. In the next section, we will focus in particular on the problem of identifying unique divisor class representatives.

3.3 Reduced and Distinguished Divisors and Ideals

In order to compute in the ideal class group of a totally imaginary function field efficiently, we will need to identify a unique representative of each class. For computational purposes, it will be convenient for these representatives to be small in a particular sense. In this section, we will define two important notions, the concepts of reduced and distinguished, for divisors, fractional ideals, and integral ideals, and will give various properties of such divisors and ideals. Of particular importance for the last section of this chapter, we will use distinguished divisors to define and describe the infrastructure of an ideal class for fields of positive unit rank. Much of this section will state and generalize equivalent notions in [Sch01], [GPS02], [Bau04], and [JSS07b].

3.3.1 Reduced Divisors and Ideals

In this section, we define the notion of reduced for divisors and ideals and will establish a hierarchy of divisors: reduced implies semi-reduced, which, in turn, implies finitely effective. Further, we will show that every divisor class contains a reduced divisor. This notion of reduced gives a practical restriction on the degree of the effective part of a divisor.

Let $D \in \mathcal{D} = \mathcal{D}(K)$. If D is semi-reduced and $\deg(D^+) \leq g$, then we say that D is *reduced*. We say that a primitive ideal, $\mathfrak{a}$, is *reduced* if and only if $\Phi^{-1}(\mathfrak{a})$ is a reduced divisor. We have the following observation, which is a slight generalization of Lemma 2.8 of [Bau04].

Lemma 3.3.1 (Lemma 2.8 of [Bau04]) *Suppose $K = K_x$ is a cubic function field with a unique place at infinity, ∞_0. If $D \in \mathcal{D}_0$, $\deg(D_S) \geq 0$, and $\mathfrak{a} = \Phi(D_S)$, then $\mathfrak{a}$ is reduced if and only if $\mathfrak{a}$ is primitive and $\deg(\mathfrak{a}) \leq g$. If K has positive unit rank and $\mathfrak{a}$ is reduced, then $\deg(\mathfrak{a}) \leq g$.*

Proof: If K has unit rank 0, then, by definition, $\mathfrak{a}$ is reduced if and only if $D = \Phi^{-1}(\mathfrak{a}) - (\deg(\mathfrak{a})/\deg(\infty_0))\infty_0$ is reduced, which in turn holds if and only if D is semi-reduced and $\deg(D^+) \leq$

g. Since $r = 0$, we have $D = D_S - (\deg(D_S)/\deg(\infty_0))\infty_0$, so $D_S = D^+$, and hence, $\deg(\mathfrak{a}) = \deg(D^+)$. Thus, $\mathfrak{a}$ is reduced if and only if $\mathfrak{a}$ is primitive and $\deg(\mathfrak{a}) \leq g$.

If K has positive unit rank, then $\deg(\infty_0) = 1$. If $\mathfrak{a}$ is reduced, then $D = \Phi^{-1}(\mathfrak{a})$ is reduced. Thus, $\deg(\mathfrak{a}) = \deg(D) = \deg(D_S) = \deg(D^+) \leq g$. □

For the remainder of this section, we will assume that $K = K_x$ is a cubic function field, with maximal order $\mathcal{O} = \mathcal{O}_x$, such that $\deg(\infty_0) = 1$; only totally inert cubic function fields do not apply to the rest of this discussion. Before we show that every divisor class contains a reduced divisor, we first note results stating that there exist finitely effective and semi-reduced divisors in each class. The following two lemmas were proved for the case that $\text{sig}(K_x) = (3, 1)$ in Lemmas 1.6 and 1.8 in [Bau04], but the results extend to all cubic function fields. We therefore omit the proofs, as they may be found in the given source.

Lemma 3.3.2 (Lemma 1.6 of [Bau04]) *Every divisor $D \in \mathcal{D}_0$ is equivalent to a finitely effective divisor.*

Lemma 3.3.3 (Lemma 1.8 of [Bau04]) *Every divisor $D \in \mathcal{D}_0$ is equivalent to a semi-reduced divisor.*

We now show that every divisor class contains a reduced divisor. This result was proved for totally ramified superelliptic function fields in Lemma 1 of [GPS02], and in Lemma 1.10 of [Bau04] for totally ramified purely cubic function fields. The proof given here was adapted from the proofs given in [GPS02] and [Bau04], and uses a technique found in Section 4 of [PR99].

Lemma 3.3.4 *If $K = K_x$ is a cubic function field such that $\deg(\infty_0) = 1$, then every divisor $D \in \mathcal{D}_0$ is equivalent to a reduced divisor of the form $E = E^+ - \deg(E^+)\infty_0$.*

Proof: By Lemma 3.3.2, it suffices to show that a finitely effective divisor $D \in \mathcal{D}_0$ is equivalent to a reduced divisor. We first show that there exists an effective (and hence finitely effective) divisor E_1 such that $E_1 - g\infty_0 \sim D$. Let W be a canonical divisor of K. By the Riemann-Roch Theorem, we have

$$l(D + g\infty_0) = \deg(D + g\infty_0) - g + 1 + l(W - D - g\infty_0) \ .$$

Since $\deg(D + g\infty_0) = g$ and $l(W - D - g\infty_0) \geq 0$, we have $l(D + g\infty_0) \geq 1$. Thus, there exists some function $\alpha \in K^*$ such that $\text{div}(\alpha) \geq -D - g\infty_0$. Let $E_1 = \text{div}(\alpha) + (D + g\infty_0) \geq 0$. Thus, E_1 is effective and $E_1 - g\infty_0 = D + \text{div}(\alpha)$, that is, $E_1 - g\infty_0 \sim D$. If $E = E_1 - g\infty_0$, then E is finitely effective and $\deg(E^+) \leq \deg(E_1) = g$. (The inequality is strict if $\infty_0 \in \text{supp}(E_1)$.) If E is semi-reduced, then E is a reduced divisor of the desired form.

If $E_1 - g\infty_0$ is not semi-reduced, then by Lemma 3.3.3, there is a semi-reduced divisor equivalent to it. (Recall that the infinite places of K are $S = \{\infty_0, \ldots, \infty_r\}$, and $e_i = e(\infty_i \mid \infty)$ are the respective ramification indices.) Specifically, there is a polynomial $a(x) \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$, with

$\mathrm{div}(a(x))_S \leq E_1$, and effective divisors E_2 and E_3 such that

$$\begin{aligned} D &\sim E_1 - g\infty_0 - \mathrm{div}(a(x)) \\ &= E_1 - g\infty_0 - \left(\mathrm{div}(a(x))_S - \frac{\deg(a)}{n}(\infty_0 + e_1\infty_1 + \cdots + e_r\infty_r)\right) \\ &= E_2 + \frac{\deg(a)}{n}(e_1\infty_1 + \cdots + e_r\infty_r) - \left(g - \frac{\deg(a)}{n}\right)\infty_0 \\ &= E_3 - \deg(E_3)\infty_0 = E \end{aligned}$$

is semi-reduced. Here, $E_2 = E_1 - \mathrm{div}(a(x))_S$ and $E_3 = E_2 + (\deg(a)/n)(e_1\infty_1 + \cdots + e_r\infty_r)$. We see that E_2 is effective by construction, and E_3 is effective because $E_2 \geq 0$ and $(\deg(a)/n)(e_1\infty_1 + \cdots + e_r\infty_r) \geq 0$. Notice that $0 \leq \deg(E_3) = \deg(E^+) = g - \deg(a(x))/n < g$, so that E is a reduced divisor of the desired form. □

For hyperelliptic function fields in which ∞ ramifies, each divisor class contains a unique reduced divisor, and if ∞ splits, then each divisor class contains a unique reduced divisor of the form $D = D_S - \deg(D_S)\infty_1 + v_0(D)(\infty_0 - \infty_1)$, where $v_0(D) \geq 0$ and $0 \leq \deg(D_S) + v_0(D) \leq g$.[1] In cubic function fields, however, uniqueness does not always hold, so we need a more restrictive definition in order to identify a unique representative in each divisor class.[2] In the next section, we will present two such notions, *i-distinguished* and *distinguished*.

3.3.2 Minima and Distinguished Divisors

In this section, we will define the notions of a minimum of a fractional ideal and distinguished and i-distinguished divisors and ideals. We will establish the relationship between these concepts and then proceed to extend the hierarchy of divisors. By definition, a reduced divisor is semi-reduced, and hence finitely effective. Here, we will give a series of results to show that distinguished divisors are 0-distinguished and that i-distinguished divisors are reduced. Furthermore, we will prove that if $\mathrm{sig}(K_x) \neq (1, 3)$, then every divisor class of K_x has a unique 0-distinguished divisor. In addition, we will give analogous results for integral and fractional ideals. We will conclude the section with an important result that will be used to understand the structure of the infrastructures of a cubic function field. The definition of a distinguished divisor is related to the notion of a minimum of a fractional ideal, which we define next.

Let K be a cubic function field of positive unit rank. For a fractional ideal, $\mathfrak{f}$, of $\mathcal{O}$ and an element $\theta \in \mathfrak{f}$ the *normed body* of θ in $\mathfrak{f}$ is defined as

$$\mathcal{N}_{\mathfrak{f}}(\theta) = \left\{\alpha \in \mathfrak{f} \setminus \{0\} \mid \mathrm{div}(\alpha)^S \geq \mathrm{div}(\theta)^S\right\} \cup \{0\} = \left\{\alpha \in \mathfrak{f} \mid \deg\left(\alpha^{(i)}\right) \leq \deg\left(\theta^{(i)}\right),\ 0 \leq i \leq r\right\}. \tag{3.5}$$

If $\mathcal{N}_{\mathfrak{f}}(\theta) = \mathbb{F}_q\theta$, then θ is called a *minimum* in $\mathfrak{f}$. Equivalently, if $\mathfrak{a} = d(\mathfrak{f})\mathfrak{f}$ and θ is a minimum in $\mathfrak{f}$, then $d(\mathfrak{f})\theta$ is a minimum in $\mathfrak{a}$. Shortly, we will show that 1 is a minimum in $\mathcal{O}$ (see also Theorem 4.1 of [SS00] or Lemma 3.2.1 of [LSY03]), so that minima indeed exist.

[1] See Propositions 3.1 and 4.1 of [PR99] or Sections 4.1 and 4.3 of [JSS07b] for these respective situations.

[2] For an example of two equivalent reduced ideals in a totally ramified cubic function field, see Example 5.6 of [Bau04].

The following definition generalizes the definition of distinguished in Definition 1.11 of [Bau04] and the concept underlying Theorem 2 of [GPS02], for fields of positive unit rank. In the totally ramified case, the definition of distinguished is the same as the one in [Bau04]. Distinguished divisors are finitely effective divisors of the form $\Psi^{-1}(\mathfrak{a})$, whose finite part has minimal degree in its equivalence class. However, they will require an additional restriction that we will show is related to the definition of a minimum of a fractional ideal.

Definition 3.3.5 *Let $K = K_x$ be a cubic function field such that $\deg(\infty_0) = 1$. We call a finitely effective divisor $D \in \mathcal{D}_0$* distinguished *if*

- *D is of the form $D = D_S - \deg(D_S)\infty_0$, and*
- *for any other finitely effective divisor E equivalent to D, such that $\deg(E_S) \leq \deg(D_S)$ and $E^S \geq D^S$, we have $D = E$.*

We say that an ideal, $\mathfrak{a}$, of $\mathcal{O}$ is distinguished *if $\Psi^{-1}(\mathfrak{a})$ is distinguished. We call a fractional ideal, $\mathfrak{f}$, of $\mathcal{O}$,* distinguished *if $\mathfrak{f} = \mathfrak{a}^{-1}$ for some distinguished ideal, $\mathfrak{a}$.*

In particular, if a divisor, D, is distinguished, then $D^+ = D_S$, so $v_i(D) = 0$ for all $1 \leq i \leq r$, and $-\Psi$ establishes a one-to-one correspondence between distinguished divisors of K and distinguished fractional ideals of $\mathcal{O}$. The following result gives an equivalent characterization of a distinguished ideal if K is totally ramified. The proof may be found in [Bau04], but the result follows from the definition of distinguished and the assumption that $\text{sig}(K) = (3, 1)$.

Lemma 3.3.6 (Lemma 2.9 of [Bau04]) *Let K be a totally ramified cubic function field and $\mathcal{O}$ its maximal order. An integral ideal, $\mathfrak{a}$, of $\mathcal{O}$, is distinguished if and only if for any other integral ideal $\mathfrak{b} \sim \mathfrak{a}$, with $\deg(\mathfrak{b}) \leq \deg(\mathfrak{a})$, we have $\mathfrak{a} = \mathfrak{b}$.*

Next, we will formally establish the relationship between the notion of minima and the notion of distinguished. In papers by Scheidler et al. [Sch00, SS00, Sch01, LSY03], fractional ideals in which 1 is a minimum are called *reduced*. With this next result, we will offer new terminology for such ideals; we will show that such ideals are distinguished.

Theorem 3.3.7 *Let $K = K_x$ be a cubic function field such that $\deg(\infty_0) = 1$. A fractional ideal, $\mathfrak{f}$, of $\mathcal{O} = \mathcal{O}_x$ is distinguished if and only if 1 is a minimum in $\mathfrak{f}$.*

Proof: Let $D = -\Psi^{-1}(\mathfrak{f})$. Suppose first that $\mathfrak{f}$, and hence D, is distinguished. Let $\alpha \in \mathfrak{f}$ be a non-zero element such that $\text{div}(\alpha)^S \geq 0$, that is, $\alpha \in \mathcal{N}_{\mathfrak{f}}(1)$. We will show that $\alpha \in \mathbb{F}_q^*$. Let $E = D + \text{div}(\alpha)$. Then $E^S = D^S + \text{div}(\alpha)^S \geq D^S$. Also, $\deg(\text{div}(\alpha)) = 0$ and $\text{div}(\alpha)^S \geq 0$ imply $\deg(\text{div}(\alpha)_S) \leq 0$, so $\deg(E_S) \leq \deg(D_S)$. Since D is distinguished, we have $E = D$, so $\text{div}(\alpha) = 0$. Hence, $\alpha \in \mathbb{F}_q^*$, so 1 is a minimum in $\mathfrak{f}$.

Conversely, suppose that $\mathfrak{f}$ is a fractional ideal such that 1 is a minimum in $\mathfrak{f}$. Let E be any finitely effective divisor equivalent to D such that $\deg(E_S) \leq \deg(D_S)$ and $E^S \geq D^S$. We will show that $E = D$. Since $E \sim D$, we have $E = D + \text{div}(\alpha)$, for some $\alpha \in K^*$. Thus, $\text{div}(\alpha)^S = E^S - D^S \geq 0$ and $0 \leq E_S = D_S + \text{div}(\alpha)_S$. It follows that $\text{div}(\alpha)_S \geq -D_S = \Phi^{-1}(\mathfrak{f})$, so $\alpha \in \mathfrak{f}$, and $\alpha \in \mathcal{N}_{\mathfrak{f}}(1)$. Since 1 is a minimum in $\mathfrak{f}$, we have $\alpha \in \mathbb{F}_q^*$, so $\text{div}(\alpha) = 0$. Thus, $E = D$, so D, and hence $\mathfrak{f}$, is distinguished. □

In [Sch00, SS00, Sch01, LSY03], integral ideals, $\mathfrak{a}$, were called reduced if $\mathfrak{a}/L(\mathfrak{a})$ was a reduced fractional ideal, or, equivalently, if $L(\mathfrak{a})$ was a minimum in $\mathfrak{a}$. Here, however, this is not the case, so that $L(\mathfrak{a})$ is not a minimum of a distinguished ideal $\mathfrak{a}$ in general. It is more convenient to define distinguished ideals in the manner given since integral ideals will be used to compute in the ideal class group of totally imaginary fields, the case in which minima will not be used.

In previous work on cubic infrastructure noted above, the infrastructure of a function field was defined in terms of fractional ideals in which 1 is a minimum. Therefore, Theorem 3.3.7 unifies the ideal notation of infrastructure with the divisor theoretic concepts of [Bau04] and [JSS07b].

In the context of infrastructure, we will be working directly from $\mathcal{O}$ as a fractional ideal most frequently. Thus, we note that as an immediate consequence of the definition of distinguished and Theorem 3.3.7, $\mathcal{O}$ is a distinguished (fractional) ideal of $\mathcal{O}$ and 1 is a minimum in $\mathcal{O}$.

The following result gives another important relationship between minima and distinguished fractional ideals and will be applied in Chapter 5 to define the baby step operation on infrastructure. The proof given here is essentially the same as the one in the given sources, except it has been rewritten in the language of divisors.

Lemma 3.3.8 (Proposition 5.2 of [SS00], Lemma 3.1.1 of [LSY03]) *An element $\theta \in K = K_x$ is a minimum in a fractional ideal, $\mathfrak{f}$, of $\mathcal{O} = \mathcal{O}_x$ if and only if the fractional ideal $\langle \theta^{-1} \rangle \mathfrak{f}$ is distinguished.*

Proof: By definition, θ is a minimum in $\mathfrak{f}$ if and only if $\mathcal{N}_{\mathfrak{f}}(\theta) = \left\{\alpha \in \mathfrak{f} \setminus \{0\} \mid \mathrm{div}(\alpha)^S \geq \mathrm{div}(\theta)^S\right\} \cup \{0\} = \mathbb{F}_q\theta$. Let $0 \neq \alpha \in \mathcal{N}_{\langle\theta^{-1}\rangle\mathfrak{f}}(1)$, so that, by definition, $\mathrm{div}(\alpha)^S \geq 0$. Adding the infinite part of the divisor of θ, we have $\mathrm{div}(\alpha)^S + \mathrm{div}(\theta)^S = \mathrm{div}(\alpha\theta)^S \geq \mathrm{div}(\theta)^S$. So $\mathcal{N}_{\mathfrak{f}}(\theta) = \mathbb{F}_q\theta$ if and only if $\alpha\theta \in \mathbb{F}_q^*\theta$, which holds if and only if $\alpha \in \mathbb{F}_q^*$, which in turn is true if and only if 1 is a minimum in $\langle \theta^{-1} \rangle \mathfrak{f}$, that is, if and only if $\langle \theta^{-1} \rangle \mathfrak{f}$ is distinguished. □

As an immediate consequence, we have the following statement, which applies the previous lemma to distinguished principal fractional ideals.

Corollary 3.3.9 *θ is a minimum in $\mathcal{O}$ if and only if the principal fractional ideal $\langle \theta^{-1} \rangle$ is distinguished.*

In the following discussion, we will define what it means for a divisor to be i-distinguished, a notion related to that of a distinguished divisor. We will show that distinguished divisors are 0-distinguished, and will proceed to prove that (i-)distinguished divisors are semi-reduced and then reduced in Theorem 3.3.15.

For the definition of i-distinguished, we need the following map, which generalizes the isomorphism, Ψ, defined in Section 3.2 for every signature except for the totally inert case. This map will serve as a convenient means to map an ideal to a divisor. If $\deg(\infty_i) = 1$, for some $\infty_i \in S$, we define $\Psi_i : \{D \in \mathcal{D}_0 \mid v_j(D) = 0 \text{ for all } 0 \leq j \leq r \text{ with } j \neq i\} \to \mathcal{I}(\mathcal{O}_x)$ by

$$\Psi_i(D_S - \deg(D_S)\infty_i) = \Phi(D_S) \ . \tag{3.6}$$

Thus, Ψ_i is an isomorphism with inverse Ψ_i^{-1}, given by $\Psi_i^{-1}(\mathfrak{f}) = \Phi^{-1}(\mathfrak{f}) - \deg(\mathfrak{f})\infty_i$. If $i = 0$, then $\Psi = \Psi_0$.

Definition 3.3.10 *Let $K = K_x$ be a cubic function field such that $\deg(\infty_i) = 1$, for some $\infty_i \in S$. We call a finitely effective divisor $D \in \mathcal{D}_0$ i-*distinguished *if*

- *D is of the form $D = D^+ - \deg(D^+)\infty_i$, and*
- *for any other divisor $E = E^+ - \deg(E^+)\infty_i \in \mathcal{D}_0$ equivalent to D, such that $\deg(E^+) \leq \deg(D^+)$, we have $D = E$.*

*We say that an ideal, $\mathfrak{a}$, of $\mathcal{O}$ is i-*distinguished *if $\Psi_i^{-1}(\mathfrak{a})$ is i-distinguished. We call a fractional ideal, $\mathfrak{f}$, of $\mathcal{O}$, i-*distinguished *if $\mathfrak{f} = \mathfrak{a}^{-1}$ for some i-distinguished ideal, $\mathfrak{a}$.*

As with integral ideals, the notion of a(n) (i-)distinguished fractional ideal may be expressed in terms of the isomorphism Ψ.

Lemma 3.3.11 *A fractional ideal, $\mathfrak{f}$ is (i-)distinguished if and only if the divisor $-\Psi_i^{-1}(\mathfrak{f})$ is (i-)distinguished.*

Proof: A fractional ideal $\mathfrak{f}$ is (i-)distinguished if and only if $\mathfrak{f} = \mathfrak{a}^{-1}$ and $\mathfrak{a}$ is a(n) (i-)distinguished ideal, which is true if and only if $\mathfrak{f} = \mathfrak{a}^{-1}$ and $\Psi_i(\mathfrak{a})$ is a(n) (i-)distinguished divisor, which is true if and only if $\Psi_i\left(\mathfrak{f}^{-1}\right)$ is a(n) (i-)distinguished divisor, which is true if and only if $-\Psi_i(\mathfrak{f})$ is a(n) (i-)distinguished divisor. □

Notice that if $r = 0$, then for any finitely effective divisor, $D \in \mathcal{D}_0$, we have $D_S = D^+$, and for any other finitely effective divisor, $E \in \mathcal{D}_0$, $\deg(E_S) \leq \deg(D_S)$ if and only if $E^S \geq D^S$. Thus, for any totally ramified cubic function field, the definitions of 0-distinguished and distinguished are equivalent. In other cases, it is not obvious that distinguished implies 0-distinguished. The following lemma, however, proves this is so.

Lemma 3.3.12 *If K is a cubic function field of positive unit rank and $D \in \mathcal{D}_0$ is a distinguished divisor, then D is 0-distinguished.*

Proof: Let D be a distinguished divisor and $E = E^+ - \deg(E^+)\infty_0$ a divisor equivalent to D such that $\deg(E^+) \leq \deg(D^+) = \deg(D_S)$. It suffices to show that $E = D$. Obviously, $v_{\mathfrak{P}}(E) \geq 0$ for all places, $\mathfrak{P} \in \mathbb{P}_K \setminus \{\infty_0\}$, so E is finitely effective. Also, $v_i(E) \geq 0$, for all $1 \leq i \leq r$. It follows that $\deg(E_S) = \deg(E^+) - \sum_{i=1}^r v_i(E)\deg(\infty_i) \leq \deg(E^+)$, so $\deg(E_S) \leq \deg(E^+) \leq \deg(D_S)$. Since $E \sim D$, we have $E = D + \mathrm{div}(\alpha)$ for some function $\alpha \in K^*$. Thus, $\mathrm{div}(\alpha) = E^+ - \deg(E^+)\infty_0 - D_S + \deg(D_S)\infty_0$. Since $\deg(E^+) \leq \deg(D_S)$, we have $v_0(E) \geq v_0(D)$, so $v_0(\alpha) = v_0(E) - v_0(D) \geq 0$. Finally, since $v_i(E) \geq v_i(D) = 0$, for all $1 \leq i \leq r$, we have $v_i(\alpha) = v_i(E) - v_i(D) \geq 0$, for all $1 \leq i \leq r$. Thus, $\mathrm{div}(\alpha)^S \geq 0$, so $E^S = D^S + \mathrm{div}(\alpha)^S \geq D^S$. Since D is distinguished, we have $D = E$, so D is 0-distinguished. □

We will use the following to show that a(n) (i-)distinguished divisor is semi-reduced.

Lemma 3.3.13 *If $a(x) \in \mathbb{F}_q[x]$, then $n \mid \deg(\mathrm{div}(a)_S)$, where $n = [K : \mathbb{F}_q(x)]$.*

Proof: If $a(x) \in \mathbb{F}_q[x]$, then $a(x)$ factors uniquely as $a(x) = c\prod_i p_i(x)^{b_i}$, where $c \in \mathbb{F}_q^*$, $b_i \in \mathbb{N}$, and each $p_i(x)$ is a monic irreducible polynomial of $\mathbb{F}_q[x]$. For each $p_i(x)$, we have $\langle p_i(x)\rangle = \prod_j \mathfrak{p}_{i,j}^{e_{i,j}}$, where $\sum_j e_{i,j}f_{i,j} = n$ and $f_{i,j} = \deg(\mathfrak{p}_{i,j})$. Thus, $\deg(\mathrm{div}(p_i(x))_S) = n$ for each i. It follows that $n \mid \deg(\mathrm{div}(a(x))_S)$. □

We now show that (i-)distinguished divisors are semi-reduced and reduced. Since distinguished divisors are 0-distinguished, which is a special case of being i-distinguished, we will henceforth omit the parentheses, and show that, by implication, every distinguished divisor is semi-reduced and reduced. Theorem 3.3.15 generalizes Lemma 1.12 of [Bau04] to all non-inert cubic function fields.

Lemma 3.3.14 *If D is an i-distinguished divisor, then D is semi-reduced.*

Proof: Let D be an i-distinguished divisor. If D were not semi-reduced, then there would be some polynomial $a(x) \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$, such that $\mathrm{div}(a(x))_S \leq D_S \leq D^+$. By Lemma 3.3.13, we have $3 \mid \deg(\mathrm{div}(a)_S)$. Then,

$$\begin{aligned} D - \mathrm{div}(a(x)) &= D^+ - \deg\left(D^+\right)\infty_i - \left(\mathrm{div}(a(x))_S - \frac{\deg(\mathrm{div}(a)_S)}{3}\sum_{j=0}^{r} e_j\infty_j\right) \\ &= D^+ - \mathrm{div}(a(x))_S + \frac{\deg(\mathrm{div}(a)_S)}{3}\left(\sum_{j=0}^{r} e_j\infty_j - \infty_i\right) \\ &\quad - \left(\deg\left(D^+\right) - \frac{\deg(\mathrm{div}(a)_S)}{3}\right)\infty_i \\ &=: E \ . \end{aligned}$$

Notice that E is finitely effective, $E = E^+ - \deg\left(E^+\right)\infty_i$, and $\deg(E^+) < \deg(D^+)$, which contradicts the assumption that D is i-distinguished. Thus, D is semi-reduced. □

Theorem 3.3.15 *If $K = K_x$ is a cubic function field such that $\deg(\infty_i) = 1$, for some infinite place $\infty_i \in S$, and D is an i-distinguished divisor, then D is reduced.*

Proof: Suppose that D is an i-distinguished divisor. By Lemma 3.3.4, there is a reduced divisor $E = E^+ - \deg(E^+)\infty_i$ equivalent to D, that is, $\deg(E^+) \leq g$. Since D is i-distinguished, we have $\deg(D^+) \leq \deg(E^+) \leq g$. Furthermore, D is semi-reduced, by Lemma 3.3.14, so D is reduced. □

This theorem and the preceding discussion establish a hierarchy of degree 0 divisors:

$$\text{distinguished} \implies \text{0-distinguished} \implies \text{reduced} \implies \text{semi-reduced} \implies \text{finitely effective.}$$

In the case that K is totally ramified, we have

$$\text{distinguished} \iff \text{0-distinguished.}$$

We continue by showing that each divisor class of K contains a unique i-distinguished divisor, provided that $\deg(\infty_i) = 1$, in Theorem 3.3.16. The uniqueness result does not always hold for distinguished divisors, but a certain collection of distinguished divisors will comprise the infrastructure of a function field of positive unit rank. We will also use this theorem in the unit rank 0 case to find a unique distinguished representative of each ideal class. This important result generalizes Theorem 1 of [GPS02] and Corollary 5.3 of [Bau04]. The proof has been adapted slightly from the proof in [GPS02].

Theorem 3.3.16 *If $K = K_x$ is a cubic function field such that $\deg(\infty_i) = 1$, for some $\infty_i \in S$, then every divisor class of $\mathcal{D}_0$ contains a unique i-distinguished divisor.*

Proof: By Lemma 3.3.4, each divisor class contains a reduced divisor, $E = E^+ - \deg(E^+)\infty_i$. We will find an i-distinguished divisor $D \sim E$. If E is a principal divisor, then we choose $D = 0 \sim E$, and we are done. Assume, then, that E is not principal. Recall from Section 2.2.5, that for a divisor D, $L(D) = \{\alpha \in K^* \mid \text{div}(\alpha) \geq -D\} \cup \{0\}$ is a vector space of dimension $l(D)$ over $\mathbb{F}_q$. Suppose that $l(E) > 0$. Then there is some function $\alpha \in K^*$ such that $\text{div}(\alpha) \geq -E$. Since $\deg(\text{div}(\alpha)) = \deg(E) = 0$, it follows that $\text{div}(\alpha) = -E$, which contradicts the assumption that E is not principal. Thus, $l(E) = 0$.

Now let $E_m = E + m\infty_i$ for any integer $m > 0$. Then $\deg(E_m) = m$ and $L(E_m) \subseteq L(E_{m+1})$, so that considering their dimensions, we have $l(E_m) \leq l(E_{m+1})$. Similarly, $l(W - E_{m+1}) \leq l(W - E_m)$, where W is a canonical divisor. By the Riemann-Roch Theorem, we have $l(E_m) = \deg(E_m) - g + 1 + l(W - E_m) = m - g + 1 + l(W - E_m)$ and $l(E_{m+1}) = m - g + 2 + l(W - E_{m+1})$, so $0 \leq l(E_{m+1}) - l(E_m) = l(W - E_{m+1}) - l(W - E_m) + 1 \leq 1$. It follows that either $l(E_{m+1}) = l(E_m)$ or $l(E_{m+1}) = l(E_m) + 1$.

Suppose that $l(E_m) = 0$ for all integers $m \geq 0$. Since $E = E^+ - \deg(E^+)\infty_i$ is reduced, we have $\deg(E^+) \leq g$, thus, $-E - g\infty_i = -E^+ - (g - \deg(E^+))\infty_i \leq 0 = \text{div}(\alpha)$, where $\alpha \in \mathbb{F}_q^*$. So $\alpha \in \mathbb{F}_q \subseteq L(E_g)$, and $l(E_g) \geq 1$, which contradicts our assumption that $l(E_m) = 0$ for all integers $m \geq 0$. Therefore, there exists some $m \leq g$ such that $l(E_m) > 0$.

Since the values of $l(E_m)$ increase by at most 1 with increasing m, there is a minimal positive integer m such that $0 = l(E_{m-1}) < l(E_m) = 1$. Then $L(E_m) = \mathbb{F}_q\alpha$, for some function $\alpha \in K^*$, such that $\text{div}(\alpha) \geq -E - m\infty_i$. Let $D_1 = \text{div}(\alpha) + E + m\infty_i \geq 0$ and $D = D_1 - m\infty_i = E + \text{div}(\alpha)$. Since D_1 is effective, we have $D_1 = D^+$, and since $D \sim E$, we have $m = \deg(D^+)$. We claim that D is i-distinguished.

Let $D_2 = D_2^+ - \deg\left(D_2^+\right)\infty_i$ be a divisor such that $D_2 \sim D$ and $\deg\left(D_2^+\right) \leq \deg(D^+)$. Then $D_2 \sim E$, so there is some function $\alpha_2 \in K^*$ such that $D_2^+ = E + \deg\left(D_2^+\right)\infty_i + \text{div}(\alpha_2) \geq 0$. Therefore, $\alpha_2 \in L(E + \deg\left(D_2^+\right)\infty_i)$. If $\deg\left(D_2^+\right) < \deg(D^+) = m$, then for some $0 \leq l < m$, we have $\alpha_2 \in L(E_l) = \{0\}$, contradicting the assumption that $\alpha_2 \neq 0$. If, on the other hand, $\deg\left(D_2^+\right) = \deg(D^+)$, then $\alpha_2 \in L(E_m) = \mathbb{F}_q\alpha$, so that $\alpha_2 = a\alpha$, for some constant $a \in \mathbb{F}_q^*$. Therefore, $D_2 = D$, so D is i-distinguished.

To establish uniqueness, let $D = D^+ - \deg(D^+)\infty_0$ and $E = E^+ - \deg(E^+)\infty_0$ be equivalent i-distinguished divisors. Without loss of generality, assume $\deg(E^+) \leq \deg(D^+)$. By the definition of i-distinguished, we have $D = E$. □

From this theorem, along with the observation that the definitions of 0-distinguished and distinguished are equivalent and the fact that $Cl(\mathcal{O}) \cong \mathcal{J}_K$ in totally ramified cubic function fields, we have the following consequence, which was proved in [Bau04]. The corresponding result for totally ramified superelliptic function fields is Theorem 2 of [GPS02]. This result is especially important for performing arithmetic in $Cl(\mathcal{O})$ for totally imaginary fields, since we will need to identify a unique ideal class representative.

Corollary 3.3.17 (Corollaries 5.2 and 5.3 of [Bau04]) *If $K = K_x$ is a totally ramified cubic function field, then every divisor class contains a unique distinguished divisor and every ideal class contains a unique distinguished ideal.*

Recall from Section 2.5.3 that any system of fundamental units of $\mathcal{O}$ gives rise to an r-dimensional submodule, Λ, of $\mathbb{Z}^r$ by considering the coefficient vectors of the divisors of the fundamental

units. Furthermore, there is a unique system, $\mathcal{U} = \{\eta_1, \ldots, \eta_r\}$, and corresponding divisors, $\mathcal{E} = \{\text{div}(\eta_1), \ldots, \text{div}(\eta_r)\}$, such that the matrix formed by the coefficient vectors of the divisors of $\mathcal{E}$ is in Hermite normal form. We also showed that for every divisor $E_\infty \in \mathcal{D}_0^S/\mathcal{P}^S$, there is a unique divisor $D_\infty \in [E_\infty]$ such that $0 \leq v_i(D_\infty) < v_i(\eta_i)$, for each $1 \leq i \leq r$. Recall that D_∞ is called the *minimal representative* of $[E_\infty]$, and we write $D_\infty \equiv E_\infty \pmod{\mathcal{E}}$. The next result also follows partly from Theorem 3.3.16 and is a key result that will be used in the next section to understand the structure of infrastructure and prove several of its arithmetic properties.

Theorem 3.3.18 *Let $K = K_x$ be a cubic function field such that $\deg(\infty_i) = 1$, for some infinite place, $\infty_i \in S$. For each ideal class $\mathbf{C} \in Cl(\mathcal{O})$, there are R^S distinct i-distinguished divisors, $D_1, \ldots, D_{R^S}$, such that $\Phi((D_j)_S) \in \mathbf{C}$, for $1 \leq j \leq R^S$. Moreover, there is a one-to-one correspondence between the divisors $D_1, \ldots, D_{R^S}$ and the R^S distinct minimal representatives of the elements of $\mathcal{D}_0^S/\mathcal{P}^S$. Finally, let D, E be i-distinguished divisors such that $\mathfrak{a} = \Phi(D_S)$, $\mathfrak{b} = \Phi(E_S) \in \mathbf{C}$, with $\mathfrak{b} = \langle\alpha\rangle\,\mathfrak{a}$, for some $\alpha \in K^*$. If D_∞ and E_∞ are the minimal representatives of $[D_\infty]$ and $[E_\infty]$ corresponding to D and E, respectively, and $\text{div}(\alpha) = \text{div}(\alpha)_S - \deg(N(\alpha))\infty_i + A_\infty$, then $A_\infty \equiv E_\infty - D_\infty \pmod{\mathcal{E}}$.*

Proof: Since $f = 1$, the exact sequence, (2.6), simplifies to: $(0) \to \mathcal{D}_0^S/\mathcal{P}^S \to \mathcal{J}_K \to Cl(\mathcal{O}) \to (0)$, so the map $\mathcal{J}_K \twoheadrightarrow Cl(\mathcal{O})$ is an R^S-to-1 map. Thus, there are R^S distinct divisor classes that map to a given ideal class $\mathbf{C} \in Cl(\mathcal{O})$, say $\mathbf{D}_1, \ldots, \mathbf{D}_{R^S}$. By Theorem 3.3.16, each $\mathbf{D}_j$, for $1 \leq j \leq R^S$ contains a unique i-distinguished divisor D_j. From the proof of Proposition 14.1 of [Ros02], the homomorphism taking $\mathcal{J}_K \to Cl(\mathcal{O})$ is the homomorphism induced by Φ, so $\Phi((D_j)_S) \in \mathbf{C}$, for each j.

From (2.6), we have $\mathcal{J}_K/(D_0^S/\mathcal{P}^S) \cong Cl(\mathcal{O})$, so under the isomorphism, each coset $\mathbf{D}_j + (D_0^S/\mathcal{P}^S) \mapsto \mathbf{C}$. Since $Cl(\mathcal{O}) \cong \mathcal{D}_S/\mathcal{P}_S$, $\mathbf{C} \mapsto \mathbf{D}$, for some divisor class $\mathbf{D} \in \mathcal{D}_S/\mathcal{P}_S \subseteq \mathcal{J}_K$. Thus, for each $\mathbf{D}_j$, there is a unique infinite divisor class, $[E_{\infty,j}] \in \mathcal{D}_0^S/\mathcal{P}^S$, such that $\mathbf{D}_j + [E_{\infty,j}] = \mathbf{D}$. This establishes a correspondence between the i-distinguished divisor D_j and the minimal representative, $D_{\infty,j}$, of $[E_{\infty,j}]$, for each j.

For the last part, let $[A] = [E] - [D]$. Suppose that $[A] \neq [\text{div}(\alpha)_S - \deg(N(\alpha))\infty_i]$. Then $\mathfrak{b} = \Phi(E_S) \neq \Phi(\text{div}(\alpha)_S + D_S) = \Phi(\text{div}(\alpha)_S)\Phi(D_S) = \langle\alpha\rangle\,\mathfrak{a}$, which is a contradiction. Since $[0] = [A] + [A_\infty]$ and $[D] + [D_\infty] = [E] + [E_\infty] = \mathbf{D}$, we have $[E_\infty] - [D_\infty] = [D] - [E] = -[A] = [A_\infty]$, and the result follows. □

We note that many of the definitions and results of this section generalize to function fields of higher degree. First, the definition of the normed body of a fractional ideal generalizes to any function field. The definitions of reduced and distinguished extend to any function field in which there is an infinite place, ∞_0, of degree 1. Likewise, the definition of i-distinguished extends to any function field in which $\deg(\infty_i) = 1$, and we can show that distinguished $\implies$ 0-distinguished $\implies$ reduced $\implies$ semi-reduced in those settings. For any totally ramified function field, we have distinguished $\iff$ 0-distinguished. Therefore, Theorems 3.3.7, 3.3.15, 3.3.16, and 3.3.18, Lemmas 3.3.8, 3.3.12, and 3.3.14, and Corollary 3.3.9 generalize to any function field, K, with at least one infinite place, $\infty_i \in \mathbb{P}_K$, of degree 1. Lastly, Corollary 3.3.17 holds in any totally ramified function field, generalizing Theorem 2 of [GPS02].

In the next section, we will continue to discuss the last two cases of Section 3.2, $\text{sig}(K_x) = (1, 1; 1, 2)$ and $\text{sig}(K_x) = (1, 1; 1, 1; 1, 1)$, together, since these include our two scenarios of interest;

namely, K_x is a purely cubic function field of positive unit rank. In particular, we will define the infrastructure of a purely cubic function field as a certain set of distinguished divisors and will define a distance measure on these divisors. Thus, for the remainder of this thesis, $K = K_x$ will be a purely cubic function field of positive unit rank that is unramified at ∞, so that $\deg(\infty_0) = 1$, and if $r = 2$, then $\deg(\infty_1) = \deg(\infty_2) = 1$ as well.

3.4 Cubic Infrastructure

Recall from Section 1.1 that the set of reduced principal ideals in the maximal order of a real quadratic global field has traditionally been called the (principal) infrastructure of the field, and that the ideals in this set form a cycle under the baby step operation. In fact, any ideal class of any global field of positive unit rank exhibits an infrastructure. The first description of the infrastructure of a cubic function field was given by Scheidler and Stein in [SS00] for unit rank 1 function fields, and was given in terms of distinguished fractional ideals. The definition given here, however, extends the divisor theoretic definition of quadratic infrastructures in Section 4 of [JSS07b] to cubic infrastructures. Shanks [Sha72] was the first to discover the infrastructure of a real quadratic number field, naming it so because of the unique structure he found within the principal ideal class. In what follows, we will show the analogous structure in the cubic function field setting.

In this section, we will define the infrastructure of a real cubic function field as a set of distinguished divisors, generalizing the description and results of [PR99] and [JSS07b]. In addition, each infrastructure divisor will correspond with a distinguished ideal, hence we replaced the earlier terminology of "reduced" with "distinguished" in this context. Describing the infrastructure in divisor-theoretic terminology not only clarifies the underlying mathematics, but also ultimately improves on current methods to compute the system of fundamental units and regulator of a cubic function field. Moreover, this divisor theoretic approach has led to a number of insights and new theoretical results. We also note that a similar approach to describing the infrastructure of a number field, using analogous notions of Arakelov divisors, is given by Schoof [Sch08].

After we define the infrastructure of an ideal class, we will discuss alternate representations of the infrastructure that will be used to develop the theory of infrastructure. We will proceed to define a distance measure on the infrastructure divisors and prove upper bounds on the size of both the infrastructure and the degree of the finite part of the divisors therein. We will then use Theorem 3.3.18 to make some observations on the structure of the infrastructures of a function field in both positive unit rank cases, and conclude the section with a discussion defining the conjugates of an infrastructure divisor and their various properties, especially as they relate to distance and the regulator of a cubic function field.

3.4.1 Infrastructure

In this section, we define the infrastructure of an ideal class and describe three additional ways of viewing infrastructures that will be useful in certain contexts.

Definition 3.4.1 *Let $K = K_x$ be a cubic function field of positive unit rank and $\mathcal{O} = \mathcal{O}_x$ its ring*

of integers. If $\mathbf{C} \in Cl(\mathcal{O})$*, then we call*

$$\mathcal{R}_{\mathbf{C}} = \left\{ \Psi^{-1}(\mathfrak{a}) \mid \mathfrak{a} \in \mathbf{C} \text{ is a distinguished ideal} \right\}$$

the infrastructure *of the ideal class* $\mathbf{C}$*. If* $\mathbf{C} = [\mathcal{O}]$*, then we call* $\mathcal{R}_{\mathbf{C}}$ *the* (principal) *infrastructure of* $\mathcal{O}$ *(or of* K*) and write* $\mathcal{R} = \mathcal{R}(K) = \mathcal{R}_{\mathbf{C}}$*.*

From the classification of signatures in Section 3.2, we have $\deg(\infty_0) = 1$ for every cubic function field of positive unit rank, so distinguished divisors and ideals exist in each positive unit rank case, and $\mathcal{R}_{\mathbf{C}}$ is well-defined. Furthermore, by Lemma 3.3.12 and Theorem 3.3.16, each divisor $D \in \mathcal{R}_{\mathbf{C}}$ lies in a unique divisor class. We will pay particular attention to the infrastructure of the principal ideal class because it is most useful for the applications that we will be concerned with.

We note that the infrastructure of $\mathbf{C}$ has two main arithmetic operations, baby steps and giant steps. These operations and related concepts will be discussed in depth in Chapter 5.

Though we have defined $\mathcal{R}_{\mathbf{C}}$ in terms of divisors, there are three different ways to represent infrastructure elements, with a fourth for the principal infrastructure. The first description of infrastructure is via the definition; that is, distinguished divisors in $\mathcal{R}_{\mathbf{C}}$. The second description is via the distinguished (integral) ideals in an ideal class $\mathbf{C}$, that is, the set $\{\mathfrak{a} = \Psi(D) \mid D \in \mathcal{R}_{\mathbf{C}}\}$. The third description is via the distinguished fractional ideals of an ideal class $\mathbf{C}$, that is the set $\{\mathfrak{f} = \Psi(-D) \mid D \in \mathcal{R}_{\mathbf{C}}\}$. If $\mathbf{C} = [\mathcal{O}]$, then the fourth characterization of infrastructure is the set of minima in $\mathcal{O}$. If $D \in \mathcal{R}$, then $\mathfrak{f} = \Psi(-D)$ is a distinguished principal fractional ideal, so there is an element $\alpha \in \mathcal{O}$ such that $\mathfrak{f} = \langle \alpha^{-1} \rangle$. By Corollary 3.3.9, α is a minimum in $\mathcal{O}$. Therefore, we may express the principal infrastructure as $\mathcal{R} = \left\{ \Psi^{-1}(\langle \theta \rangle) \mid \theta \text{ is a minimum in } \mathcal{O} \right\}$.

Furthermore, if $\mathbf{C} = \mathcal{O}$, then we can make the correspondence of Theorem 3.3.18 explicit. Let $D \in \mathcal{R}$ and $\langle \theta \rangle = \mathfrak{a} = \Psi(D)$, for some $\theta \in \mathcal{O}$. Then there is an associate of θ, $\alpha = \theta\eta$, for some $\eta \in \mathcal{O}^*$, such that $\operatorname{div}(\alpha) = D + D_\infty$, where D_∞ is the minimal representative of $[D_\infty] \in \mathcal{D}_0^S / \mathcal{P}^S$. Equivalently, if $\operatorname{div}(\theta) = D + E_\infty$, for some $E_\infty \in \mathcal{D}_0^S$, then $D_\infty \in [E_\infty]$ and can be obtained from E_∞ via the Euclidean algorithm described in Section 2.5.3. Thus, D_∞ corresponds with the (0-)distinguished divisor D.

While we will mainly use the divisor representation of the infrastructure, various properties and arithmetic operations on $\mathcal{R}_{\mathbf{C}}$ will be derived using the other characterizations. These properties and operations have been developed using these other characterizations in earlier literature, so we will be stating and generalizing these results. We have already used $\mathcal{R}_{\mathbf{C}}$ directly to establish the fact that each infrastructure divisor lies in a unique divisor class, generalizing Theorem 1 of [GPS02] and Corollary 5.3 of [Bau04]. In Section 5.3.3, distinguished ideals will be used in practice to perform the giant step operation; that is, divisor addition followed by reduction. This will extend the analogous notions for cubic function fields of unit rank 1 in Chapter 7 of [Sch01]. Distinguished fractional ideals will be used to develop the baby step operation in Section 5.3.2 and to actually perform baby steps in implementations, stating the results previously given in Chapter 6 of [Sch01] and Chapter 3 of [LSY03], for example. Lastly, the infinite divisors, D_∞, associated to a principal distinguished divisor, $D \in \mathcal{R}$, will be obtained from the minima of $\mathcal{O}$ as we just described, and will be used to define the notion of the distance of a divisor, generalizing the definition given in Chapter 7 of [Sch01] for the infrastructure of unit rank 1 cubic function fields to general infrastructures and also the unit rank 2 case. We describe this concept of distance in the next section.

3.4.2 Distance

In this section, we will define a distance measure on 0-distinguished divisors, and on divisors in $\mathcal{R}_{\mathbf{C}}$ in particular. The definition of this measure is derived from the minimal infinite divisors associated to these divisors.

Let $\mathbf{C} \in Cl(\mathcal{O})$. We fix a distinguished divisor $E \in \mathcal{R}_{\mathbf{C}}$ and let $\mathfrak{b} = \Psi(E) \in \mathbf{C}$. If $\mathbf{C} = [\mathcal{O}]$, then we fix $E = 0$ and $\mathfrak{b} = \mathcal{O}$. If D is any 0-distinguished divisor and $\mathfrak{a} = \Psi(D)$, then there is a unique function, $\alpha \in K^*$, such that $\mathfrak{a} = \langle \alpha \rangle \, \mathfrak{b}$ and $\mathrm{div}(\alpha) = A + D_\infty = A_S - \deg(A_S)\infty_0 + D_\infty$, where D_∞ is the minimal representative of $[D_\infty] \in \mathcal{D}_0^S/\mathcal{P}^S$ and $A = D - E$. In the unit rank 1 case, we simply choose α such that $0 \leq \deg(\alpha) < \deg(\epsilon) = R_x$. We define the *distance* of D as

$$\delta_E(D) := \begin{cases} \delta_0(D) := \deg(\alpha) & \text{if } r = 1 \\ (\delta_0(D),\, \delta_1(D),\, \delta_2(D)) := (\deg(\alpha),\, \deg(\alpha'),\, \deg(\alpha'')) & \text{if } r = 2 \ , \end{cases}$$

and will call $\delta_i(D)$ the *i-component* of $\delta(D)$. Likewise, if D_1 and D_2 are 0-distinguished divisors such that $\Psi(D_1)$, $\Psi(D_2) \in \mathbf{C}$, then we call $\delta_{D_1}(D_2) = \delta_E(D_2) - \delta_E(D_1)$ the *relative distance* between D_1 and D_2. If $\mathbf{C} = [\mathcal{O}]$, or if the fixed divisor E is understood, then we will drop the subscript on δ. Since divisors $D \in \mathcal{R}_{\mathbf{C}}$ are 0-distinguished, the distance function restricts to a measure on $\mathcal{R}_{\mathbf{C}}$, for any ideal class $\mathbf{C} \in Cl(\mathcal{O})$. Furthermore, we note that these definitions are well-defined by Theorem 3.3.18.

We give some further remarks on distance. This definition of distance is the same as the definition given for real quadratic function fields (see Section 4 of [JSS07b], for example) and for unit rank 1 cubic function fields in Section 7 of [Sch01]. Equivalent to the definition, we have $\delta_0(D) = \deg(\alpha) = -v_0(A) - v_0(D_\infty) = \deg(A_S) - v_0(D_\infty)$ and $\delta_i(D) = \deg\left(\alpha^{(i)}\right) = -v_i(A) - v_i(D_\infty) = -v_i(D_\infty)$, for $i = 1, 2$. In the unit rank 1 case, we will also write $\delta_1(D) = 2\deg(\alpha') = \deg(\alpha'\alpha'')$. In the unit rank 2 case, if $\mathcal{E}$, or the divisors of any system of fundamental units, are unknown, then it will be impossible to determine the distance of a given divisor as defined here. In practice, therefore, we will implicitly use a similar distance measure, but redefined in terms of an unknown system of fundamental units. Nevertheless, the results on distance still hold.

The following results follow immediately from the definition of distance.

Lemma 3.4.2 (Equation 7.1 of [Sch01]) *Let $E \in \mathcal{R}_{\mathbf{C}}$ be a fixed divisor, D_1, $D_2 \in \mathcal{R}_{\mathbf{C}}$, and $\Psi(D_2) = \langle \alpha \rangle \, \Psi(D_1)$.*

1. *If $r = 1$, then*
   - *$\delta(0) = \delta_0(0) = 0$, where the subscript is the divisor 0;*
   - *$\delta_{D_1}(D_2) = \deg(\alpha)$; and*
   - *$\delta_E(D_2) \equiv \delta_E(D_1) + \deg(\alpha) \pmod{R_x}$.*
2. *If $r = 2$, then*
   - *$\delta(0) = \delta_0(0) = (0,\, 0,\, 0)$, where the subscript is the divisor 0;*
   - *$\delta_{D_1}(D_2) = (\deg(\alpha),\, \deg(\alpha'),\, \deg(\alpha''))$; and*

- $\delta_E(D_2) \equiv \delta_E(D_1) + (\deg(\alpha), \deg(\alpha'), \deg(\alpha''))\pmod{\Lambda}$, *where addition is componentwise, and* Λ *is the set of coordinate vectors of* $\mathcal{E} = \mathcal{E}(\mathcal{O}) = \{\mathrm{div}(\eta_1), \mathrm{div}(\eta_2)\}$, *with* $\mathcal{U} = \{\eta_1, \eta_2\}$ *determined as in Section 2.5.3.*

3.4.3 Size Properties of Infrastructures

In this section, we prove some statements about the infrastructure, namely the size of divisors in $\mathcal{R}_{\mathbf{C}}$ and an upper bound on the size of $\mathcal{R}_{\mathbf{C}}$ itself, for any ideal class $\mathbf{C} \in Cl(\mathcal{O})$.

Lemma 3.3.1 and Theorem 3.3.15 allow us to improve several bounds related to the divisors in $\mathcal{R}_{\mathbf{C}}$. Some immediate conclusions improve on Theorem 4.5 and Corollary 4.6 of [SS00], Lemma 6.2 of [Sch01], and Lemma 3.1.3 of [LSY03], which give bounds on degree of the norms of ideals associated with infrastructure divisors. While the result for defining curves $C : Y^3 = GH^2$, with $3 \nmid \deg(GH^2)$ is not new, this result offers an improvement for the case in which $3 \mid GH^2$, replacing $g + 1$ with g in the given sources.

Proposition 3.4.3 *Let* $K = K_x$ *be a purely cubic function field and* $\mathbf{C} \in Cl(\mathcal{O}_x)$. *If* $D \in \mathcal{R}_{\mathbf{C}}$, $\mathfrak{a} = \Psi(D)$, *and* $\mathfrak{f} = -\Psi(D)$, *then* $\deg(D_S) = \deg(\mathfrak{a}) \leq g$, $-g \leq \deg(N(\mathfrak{f}))$, *and* $\deg(d(\mathfrak{f})) = \deg(L(\mathfrak{a})) \leq g$. *If* $\mathbf{C} = [\mathcal{O}]$ *and* $\mathfrak{a} = \langle\alpha\rangle$, *with* α *a minimum of* $\mathcal{O}_x$, *then* $0 \leq \deg(N(\alpha)) \leq g$.

Proof: Since $D \in \mathcal{R}_{\mathbf{C}}$, D, $\mathfrak{f}$, and $\mathfrak{a}$ are distinguished. By Lemma 3.3.1 and Theorem 3.3.15, D and $\mathfrak{a}$ are reduced and $\deg(D^+) = \deg(D_S) = \deg(\mathfrak{a}) \leq g$. Since $\deg(N(\mathfrak{f})) = -\deg(D_S)$, we have $-g \leq \deg(N(\mathfrak{f}))$. Lastly, since $d(f) = L(\mathfrak{a})$ and $L(\mathfrak{a}) \mid N(\mathfrak{a})$, we have $\deg(d(\mathfrak{f})) = \deg(L(\mathfrak{a})) \leq \deg(N(\mathfrak{a})) = \deg(\mathfrak{a}) \leq g$. Finally, in the principal case, we have $\deg(N(\alpha)) = \deg(\mathfrak{a})$, so the given bounds on $\deg(N(\alpha))$ hold. □

In actual implementations, most ideals corresponding with infrastructure divisors are of degree g, so the bounds in Proposition 3.4.3 are sharp.

Next, we have a convenient upper bound on the size of $\mathcal{R}_{\mathbf{C}}$.

Proposition 3.4.4 *If* $K = K_x$ *is a purely cubic function field and* $\mathcal{O} = \mathcal{O}_x$ *is its maximal order, then* $|\mathcal{R}_{\mathbf{C}}| \leq R^S$ *for any ideal class* $\mathbf{C} \in Cl(\mathcal{O})$.

Proof: From Theorem 3.3.18 there are $\left[\mathcal{D}_0^S : \mathcal{P}^S\right] = R^S$ distinct 0-distinguished divisors, D, such that $\Psi(D) \in \mathbf{C}$. Since distinguished divisors are 0-distinguished, there are at most R^S distinguished divisors that map to $\mathbf{C}$ via Ψ. By the definition of infrastructure, then, we have $|\mathcal{R}_{\mathbf{C}}| \leq R^S$. □

This result improves on Theorem 6.5 of [SS00] for the case $r = 1$ by a factor of 2. Since a 0-distinguished divisor is not necessarily distinguished, we can expect the inequality in Proposition 3.4.4 to be strict.

The results in this section established bounds on the size of $\mathcal{R}_{\mathbf{C}}$ and its divisors. In the next section, we discuss how the divisors of $\mathcal{R}_{\mathbf{C}}$ are structured.

3.4.4 Structure of Infrastructures

In this section, we will discuss the geometry of $\mathcal{R}_{\mathbf{C}}$; that is, how the distance function structures the divisors of $\mathcal{R}_{\mathbf{C}}$. Specifically, we will show that each infrastructure, $\mathcal{R}_{\mathbf{C}}$, has the structure of discrete

points on a circle in the unit rank 1 case and discrete points on a torus in the unit rank 2 case. In addition, we will explain the presence of "holes" in $\mathcal{R}_{\mathbf{C}}$.

Let $\mathbf{C} \in Cl(\mathcal{O})$, fix a distinguished divisor, $E \in \mathcal{R}_{\mathbf{C}}$, to which all distances are relative, and define the set $\mathcal{Q}_{\mathbf{C}} = \{D \in \mathcal{D}_0 \mid D \text{ is 0-distinguished and } \Psi(D) \in \mathbf{C}\}$. Now $|\mathcal{Q}_{\mathbf{C}}| = R^S$ and there are R^S distinct minimal representatives obtained from the classes of $\mathcal{D}_0^S/\mathcal{P}^S$. We denote these representatives by $\left\{D_{n,\infty} \mid 0 \leq n < R^S\right\}$. In the unit rank 1 case, we have $D_{n,\infty} = n(\infty_1 - 2\infty_0)$, for $0 \leq n < R^S$, and in the unit rank 2 case, these infinite divisors are determined uniquely modulo $\mathcal{E}$. By Theorem 3.3.18, there is a one-to-one correspondence between the divisors in $\mathcal{Q}_{\mathbf{C}}$ and the minimal representatives of the classes in $\mathcal{D}_0^S/\mathcal{P}^S$.

We will make some brief remarks about $\mathcal{Q} = \mathcal{Q}_{[\mathcal{O}]}$ in particular, and show that it possesses a group structure. Each divisor $D \in \mathcal{Q}$ may be identified with the divisor class $[D] \in \mathcal{J}_K$. Since $\Psi(D) \in [\mathcal{O}]$ for each $D \in \mathcal{Q}$, we see that the kernel of the surjection $\mathcal{J}_K \twoheadrightarrow Cl(\mathcal{O})$ in (2.6) is $\{[D] \mid D \in \mathcal{Q}\} \cong \mathcal{D}_0^S/\mathcal{P}^S$. For D_1, $D_2 \in \mathcal{Q}$, we define an operation, $\oplus$, on $\mathcal{Q}$ by $D_1 \oplus D_2 = D$, where D is the unique 0-distinguished divisor in $[D] \in \mathcal{J}_K$ and $[D_1] + [D_2] = [D]$ is the addition of divisor classes in $\mathcal{J}_K$. In this way, $\mathcal{Q}$ is a group under the operation $\oplus$. Moreover, we have $\mathcal{Q} \cong \mathcal{D}_0^S/\mathcal{P}^S$. This group law is defined explicitly on the analogous set of ideals in Theorem 4.2 of [PR99].

To describe the general structure of $\mathcal{Q}_{\mathbf{C}}$ and the corresponding structure of $\mathcal{R}_{\mathbf{C}}$, we consider the unit rank 1 case first. Under the one-to-one correspondence between $\mathcal{Q}_{\mathbf{C}}$ and $\mathcal{D}_0^S/\mathcal{P}^S$, there is a unique 0-distinguished divisor $D_n \in \mathcal{Q}_{\mathbf{C}}$ associated with $D_{n,\infty} = n(\infty_1 - 2\infty_0)$, for each $0 \leq n < R^S$. Thus, $\delta_1(D_n) = n$ and $2n \leq \delta(D_n) < \delta(D_{n+1}) < R_x$. In this way, the distance function, δ, maps each 0-distinguished divisor onto a unique point in the cyclic group $\mathbb{Z}/R_x\mathbb{Z}$ and establishes an ordering of the divisors in $\mathcal{Q}_{\mathbf{C}}$. Since $\mathcal{R}_{\mathbf{C}} \subseteq \mathcal{Q}_{\mathbf{C}}$, this structure exists among the divisors in $\mathcal{R}_{\mathbf{C}}$. Specifically, if $\mathcal{R}_{\mathbf{C}} = \{D_0, D_1, \ldots, D_{l_{\mathbf{C}}}\}$, where $l_{\mathbf{C}} = |\mathcal{R}_{\mathbf{C}}|$, then $2n \leq \delta(D_n) < \delta(D_{n+1}) < R_x$ for all integers $0 \leq n < l_{\mathbf{C}}$.

In the unit rank 2 case, δ maps each 0-distinguished divisor, $D \in \mathcal{Q}_{\mathbf{C}}$, to a unique point on the torus determined modulo Λ, creating a 2-dimensional, periodic structure of the 0-distinguished divisors. Again, since $\mathcal{R}_{\mathbf{C}} \subseteq \mathcal{Q}_{\mathbf{C}}$, the structure on $\mathcal{Q}_{\mathbf{C}}$ carries to the structure on the distinguished divisors in $\mathcal{R}_{\mathbf{C}}$.

For a given ideal class, $\mathbf{C}$, some, perhaps most of the 0-distinguished divisors in $\mathcal{Q}_{\mathbf{C}}$ will be distinguished; in other words, elements of $\mathcal{R}_{\mathbf{C}}$. As such, there will be some minimal representatives that do not correspond with infrastructure divisors, so there will be "holes" in $\mathcal{R}_{\mathbf{C}}$ for most, if not all infrastructures. The conditions under which these holes exist will be discussed in Section 5.3.2. In the particular case that $\mathbf{C} = [\mathcal{O}]$, we note that $\mathcal{R}$ is not a group under the giant step operation, which we mentioned in Section 3.4.1, failing to be associative. The presence of these holes gives us a sense for why this is the case. Methods are given in [PR99, GHM08, Fon08b, Fon08c, Mir08] to fill in these holes and work instead with the group $\mathcal{Q}$.

In this section, we showed that $\mathcal{R}_{\mathbf{C}}$ has a cyclic structure in unit rank 1 and a toroidal structure in unit rank 2 by describing the similar structure of the superset $\mathcal{Q}_{\mathbf{C}}$. We noted that this leads to holes in the infrastructure of an ideal class and helps to explain why it is not a group. In the next section, we will describe the relationship between the conjugates of a distinguished ideal, if they exist, and the infrastructures of a cubic function field.

3.4.5 Conjugates of Infrastructure Divisors

In this section, we will define and describe the behavior of the conjugates of a distinguished divisor and give sufficient conditions for when they are again distinguished divisors in the same or in a different infrastructure. We will use these results to describe the relationship between the distance of a divisor and its conjugates and the fundamental unit(s) of a purely cubic function field of positive unit rank. These results will be used in Chapter 5 to describe further structure of the principal infrastructure in the unit rank 2 setting.

Lemma 3.4.5 *Let $K/\mathbb{F}_q(x)$ be a cubic Galois extension, $\mathfrak{a} \in \mathbf{C} \in Cl(\mathcal{O})$ a distinguished ideal with conjugates $\mathfrak{a}^{(i)}$, and $D^{(i)} := \Psi^{-1}\left(\mathfrak{a}^{(i)}\right)$, for $i = 0, 1, 2$. Then $D^{(i)} \in \mathcal{R}_{\mathbf{C}^{(i)}}$, and if $\mathbf{C} = [\mathcal{O}]$, then $D^{(i)} \in \mathcal{R}$ for each $0 \leq i \leq 2$. If $\mathfrak{a} = \mathcal{O}$ or $\mathfrak{a}$ is the product of totally ramified prime ideals, then $D = D' = D''$.*

Proof: If $\mathfrak{a}$ is a nontrivial distinguished ideal of $\mathcal{O}$ and $K/\mathbb{F}_q(x)$ is Galois, then for each element $\theta \in \mathfrak{a}$, we have θ', $\theta'' \in \mathcal{O}$. Thus, $\mathfrak{a}'$ and $\mathfrak{a}''$ are ideals of $\mathcal{O}$. For each $i = 0, 1, 2$, let $\mathfrak{f}^{(i)} = \left(\mathfrak{a}^{(i)}\right)^{-1}$ so that $\mathfrak{f}$ is a distinguished fractional ideal whose conjugates are fractional ideals of $\mathcal{O}$. By definition, $\theta \in \mathcal{N}_{\mathfrak{f}^{(i)}}(1)$ if and only if $\mathrm{div}(\theta)^S \geq 0$, which is true if and only if $\mathrm{div}\left(\theta^{(i)}\right)^S \geq 0$ for all $i = 0, 1, 2$. In particular, we have θ', $\theta'' \in \mathcal{N}_{\mathfrak{f}}(1) = \mathbb{F}_q$. Thus, $\mathfrak{f}^{(i)}$ is distinguished, for $0 \leq i \leq 2$. Therefore, $\mathfrak{a}^{(i)} = \left(\mathfrak{f}^{(i)}\right)^{-1} \in \mathbf{C}^{(i)}$ is distinguished, for $0 \leq i \leq 2$, and $D^{(i)} = \Psi^{-1}\left(\mathfrak{a}^{(i)}\right) \in \mathcal{R}_{\mathbf{C}^{(i)}}$, for $0 \leq i \leq 2$. If, in addition, $\mathfrak{f}$ is principal, then $\mathfrak{f} = \langle\alpha\rangle$, for some $\alpha \in K^*$, so $\mathfrak{f}^{(i)} = \left\langle\alpha^{(i)}\right\rangle$ is principal, so that $D^{(i)} = -\Psi^{-1}\left(\mathfrak{f}^{(i)}\right) \in \mathcal{R}$, for $0 \leq i \leq 2$.

Lastly, suppose that $\mathfrak{a} = \mathcal{O}$ is the product of totally ramified prime ideals. Since any prime ideal $\mathfrak{p} \mid \mathfrak{a}$ has ramification index 3, it has degree 1, so $N(\mathfrak{p}) = P$, for some monic irreducible polynomial $P \in \mathbb{F}_q[x]$. Thus, $\mathfrak{p}\mathfrak{p}'\mathfrak{p}'' = \langle N(\mathfrak{p})\rangle = \langle P\rangle = \mathfrak{p}^3$, and $\mathfrak{p}'\mathfrak{p}'' = \mathfrak{p}^2$. Now since $K/\mathbb{F}_q(x)$ is Galois, $\mathfrak{p}'$ and p'' are ideals of $\mathcal{O}$, and in particular, they are also prime ideals. By unique prime ideal factorization, we have $\mathfrak{p} = \mathfrak{p}' = \mathfrak{p}''$. It follows that $\mathfrak{a} = \mathfrak{a}' = \mathfrak{a}''$, so $D^{(i)} = \Psi^{-1}\left(\mathfrak{a}^{(i)}\right) = \Psi^{-1}(\mathfrak{a}) = D$, for $i = 1, 2$. □

As for ideals, we call D' and D'' (or $D^{(1)}$ and $D^{(2)}$) the *conjugates* of D. With the following discussion and the next two results, we classify the purely cubic function fields that are Galois.

By Theorem 3.2.1, if K is a purely cubic function field of unit rank 1, then $q \equiv 2 \pmod 3$, so a primitive cube root of unity, ι, does not exist in $\mathbb{F}_q$. Thus, cubic function fields of unit rank 1 are not Galois. In this case, if $\mathfrak{a}$ is an ideal of $\mathcal{O}$, then the conjugates of $\mathfrak{a}$ will generally not be ideals of $\mathcal{O}$. On the other hand, we have the following lemma of [LSY03].

Lemma 3.4.6 (Lemma 2.1 of [LSY03]) *Let $K/\mathbb{F}_q(x)$ be a cubic extension and $q \equiv 1 \pmod 3$. Then $K/\mathbb{F}_q(x)$ is purely cubic if and only if $K/\mathbb{F}_q(x)$ is cyclic (that is, Galois with Galois group $\mathbb{Z}/3\mathbb{Z}$).*

By Theorem 3.2.1, if K is a purely cubic function field of unit rank 2, then $q \equiv 1 \pmod 3$, so $\iota \in \mathbb{F}_q$, and $K/\mathbb{F}_q(x)$ is Galois. Thus, we have the following corollary.

Corollary 3.4.7 *If $K/\mathbb{F}_q(x)$ is a purely cubic function field of unit rank 2, then $K/\mathbb{F}_q(x)$ is Galois.*

Therefore, every purely cubic function field of unit rank 2 satisfies Lemma 3.4.5.

If $D \in \mathcal{R}$ is a distinguished principal divisor whose conjugates are also in $\mathcal{R}$, then we can easily determine their distance. Let $\mathfrak{a} = \langle \alpha \rangle$ be a distinguished ideal, for some $\alpha \in K^*$, whose conjugates are ideals of $\mathcal{O}$, and $D^{(i)} = \Psi^{-1}\left(\mathfrak{a}^{(i)}\right) \in \mathcal{R}$, for $0 \leq i \leq 2$. Since $\Psi\left(D^{(i)}\right) = \left\langle \alpha^{(i)} \right\rangle$ and $\delta(D) = \deg(\alpha)$ and $\delta(D) = (\deg(\alpha), \deg(\alpha'), \deg(\alpha''))$ in the unit rank 1 and 2 cases, respectively, we have

$$\delta\left(D^{(i)}\right) = \begin{cases} \deg\left(\alpha^{(i)}\right) \pmod{R_x} & \text{if } r = 1 \\ \left(\deg\left(\alpha^{(i)}\right), \deg\left(\alpha^{(i+1)}\right), \deg\left(\alpha^{(i+2)}\right)\right) \pmod{\Lambda} & \text{if } r = 2 \ , \end{cases} \tag{3.7}$$

where the superscripts are considered modulo 3 and Λ is the set of coordinate vectors of $\mathcal{E} = \mathcal{E}(\mathcal{O}) = \{\mathrm{div}(\eta_1), \mathrm{div}(\eta_2)\}$, with $\mathcal{U} = \{\eta_1, \eta_2\}$ determined as in Section 2.5.3. Considering the unit rank 2 portion of (3.7), this equation suggests that $\mathcal{R}$ is symmetric about 0; a divisor and its conjugates are equidistant from 0 and each other. In Chapter 5, we will show that $\mathcal{R}$ indeed possesses complete symmetry.

We use (3.7) to relate the sum of the distances of the conjugate of an infrastructure divisor to the regulator or system of fundamental units of a cubic function field of unit rank 1 or 2, respectively. This generalizes an equation in Section 4 of [JSS07b] from the analogous situation in quadratic function fields.

In unit rank 1 cubic function fields, we have $\delta(D)+\delta(D')+\delta(D'') \equiv \deg(\alpha)+\deg(\alpha')+\deg(\alpha'') = \deg(\alpha\alpha'\alpha'') = \deg(N(\alpha)) \pmod{R_x}$. Therefore, there is some $a \in \mathbb{Z}$ such that

$$\delta(D) + \delta(D') + \delta(D'') = aR_x + \deg(N(\mathfrak{a})) \ . \tag{3.8}$$

Since $0 \leq \delta\left(D^{(i)}\right) < R_x$, for $0 \leq i \leq 2$, we have $0 \leq \delta(D) + \delta(D') + \delta(D'') < 3R_x$, so $0 \leq a \leq 2$, and $a = 0$ if and only if $D = 0$. Similarly, in the unit rank 2 case, we have

$$\begin{aligned} \delta(D) + \delta(D') + \delta(D'') &\equiv (\deg(\alpha), \deg(\alpha'), \deg(\alpha'')) + (\deg(\alpha'), \deg(\alpha''), \deg(\alpha)) \\ &\quad + (\deg(\alpha''), \deg(\alpha), \deg(\alpha')) \\ &= (\deg(N(\alpha)), \deg(N(\alpha)), \deg(N(\alpha))) = \deg(N(\alpha))(1, 1, 1) \pmod{\Lambda} \ . \end{aligned} \tag{3.9}$$

Therefore, there is some unit $\eta \in \mathcal{O}^*$ such that

$$\begin{aligned} \delta(D) + \delta(D') + \delta(D'') &= (\deg(\eta), \deg(\eta'), \deg(\eta'')) + \deg(N(\alpha))(1,1,1) \\ &= (\deg(\eta), \deg(\eta'), \deg(\eta'')) + \deg(N(\mathfrak{a}))(1,1,1) \ . \end{aligned} \tag{3.10}$$

The unit eta is not canonical. For example, there is no canonical expression of η in terms of the system of fundamental units $\{\epsilon_1, \epsilon_2\}$, described later in Theorem 5.4.5. We give some examples to illustrate this.

Example 3.4.8 *Let $K = \mathbb{F}_7(C)$, where $C : Y^3 = (x^4 + x^3 + x^2 + 4)x^2$. In this example, $r = 2$, $R_x = 17$, and $\left\langle x^3 + 4x^2 + 6x \right\rangle = \mathfrak{a}\mathfrak{a}'\mathfrak{a}''$, so that $L\left(\mathfrak{a}^{(i)}\right) = x^3 + 4x^2 + 6x$ for all $0 \leq i \leq 2$. Let $D^{(i)} = \Psi^{-1}\left(\mathfrak{a}^{(i)}\right)$. Then $\delta(D) = (4, -1, 0)$, $\delta(D') = (-1, 0, 4)$, and $\delta(D'') = (3, -1, 1)$. For the system of fundamental units, $\{\epsilon_1, \epsilon_2\}$ given by Theorem 5.4.5, we have $\delta(\mathrm{div}(\epsilon_1)) = (5, -2, -3)$ and*

$\delta(\mathrm{div}(\epsilon_2)) = (-2, -3, 5)$. *Thus,*

$$\begin{aligned}\delta(D) + \delta(D') + \delta(D'') &= (4, -1, 0) + (-1, 0, 4) + (3, -1, 1) = (6, -2, 5) \\ &= (3, -5, 2) + (3, 3, 3) = \delta(\mathrm{div}(\epsilon_1\epsilon_2)) + \deg(N(\mathfrak{a}))(1, 1, 1) \ .\end{aligned}$$

Example 3.4.9 *Let* $K = \mathbb{F}_7(C)$*, where* $C : Y^3 = (x^4 + x^3 - x^2 - x - 1)x^2$*. In this example,* $r = 2$*,* $R = 37$*, and* $\langle x^2 + 6\rangle = \mathfrak{a}\mathfrak{a}'\mathfrak{a}''$*, so that* $L\left(\mathfrak{a}^{(i)}\right) = x^2+6$ *for all* $0 \leq i \leq 2$*. Let* $D^{(i)} = \Psi^{-1}\left(\mathfrak{a}^{(i)}\right)$*. Then* $\delta(D) = (3, -2, 1)$*,* $\delta(D') = (9, -9, 2)$*, and* $\delta(D'') = (5, -4, 1)$*. For the system of fundamental units,* $\{\epsilon_1, \epsilon_2\}$ *given by Theorem 5.4.5, we have* $\delta(\mathrm{div}(\epsilon_1)) = (11, -10, 1)$ *and* $\delta(\mathrm{div}(\epsilon_2)) = (-7, 3, 4)$*. Thus,*

$$\begin{aligned}\delta(D) + \delta(D') + \delta(D'') &= (3, -2, 1) + (9, -9, 2) + (5, -4, 1) = (17, -15, 4) \\ &= (15, -17, 2) + (2, 2, 2) = \delta\left(\mathrm{div}\left(\epsilon_1^2\epsilon_2\right)\right) + \deg(N(\mathfrak{a}))(1, 1, 1) \ .\end{aligned}$$

Let $\mathfrak{p}$ *be the ramified prime ideal above* $\langle x\rangle$*, and let* $D = \Psi^{-1}(\mathfrak{p})$*. Then* $\delta(D) = (4, -3, 0)$*. Now we have*

$$3\delta(D) = (12, -9, 0) = (11, -10, 1) + (1, 1, 1) = \delta(\mathrm{div}(\epsilon_1)) + \deg(x)(1, 1, 1) \ .$$

In this chapter, we restricted ourselves to purely cubic function fields, giving properties of each signature. We defined notions of reduced, i-distinguished, and distinguished, and proved that every divisor class of a cubic function field of signature not equal to $(1, 3)$ contains a unique 0-distinguished divisor. We then defined the infrastructures of such fields of unit rank 1 and 2 as a set of distinguished divisors whose associated ideals lie in the same ideal class, generalizing the discussion of [PR99] and [JSS07b]. This served to unify the traditional description of infrastructure as a set of ideals, as in [Sch00], [SS00], [Sch01], and [LSY03], with the divisor theoretic terminology and concepts of cubic function fields given in [Bau04]. This led to several new observations and results on the size of infrastructure divisors, the size of $\mathcal{R}_{\mathbf{C}}$ itself, and the structure of $\mathcal{R}_{\mathbf{C}}$. Furthermore, this description provides a foundation to extend these notions to infrastructures of higher degree function fields. Before we define the arithmetic operations of baby step and giant step on $\mathcal{R}_{\mathbf{C}}$, we need methods to compute the product of two ideals. This is the topic of the next chapter.

Chapter 4

Ideal Arithmetic with Canonical Bases

In this chapter, we build upon the work of Scheidler [Sch01] and Bauer [Bau04] to complete the description of ideal arithmetic in purely cubic function fields of characteristic not equal to 3. Throughout this chapter, we will assume that $K = K_x = \mathbb{F}_q(C)$, where $C : Y^3 = F(x) = G(x)H^2(x)$, $G(x)$, $H(x) \in \mathbb{F}_q[x]$ are monic, relatively prime, and square-free, $0 < \text{char}(K) \neq 3$, and $\mathcal{O}$ is the maximal order of K, with $\mathbb{F}_q[x]$-basis $\{1, \rho, \omega\}$, where $\rho \in K$ satisfies $\rho^3 = GH^2$ and $\omega = \rho^2/H$. In [Sch01], Scheidler described how to square an ideal and how to multiply two ideals whose norms are relatively prime, in such arbitrary fields K. Bauer later described how to invert an arbitrary ideal and how to multiply any two ideals for the case in which C is nonsingular, that is, $H(x) = 1$, in [Bau04]. He also showed how to find the unique distinguished ideal in a given class, where $\text{sig}(K) = (3, 1)$. In an unpublished manuscript, Bauer also stated results and outlined algorithms for computing inverses and products in full generality, but the results were largely unproven and given in terms of the $\mathbb{F}_q[x]$-basis, $\{1, Y, Y^2\}$, of $\mathcal{O}$ [Bau05]. Therefore, this chapter extends the results of these articles and provides a rigorous treatment to the statements in [Bau05] in terms of the $\mathbb{F}_q[x]$-basis, $\{1, \rho, \omega\}$, of $\mathcal{O}$. As such, most of the derivation and proof techniques are borrowed from [Sch01] and [Bau04].

While the results of this chapter are not particularly deep, they are vital for three main applications. First, the multiplication algorithms of this section will be used to perform giant steps in the infrastructure of a purely cubic function field in Section 5.3.3. Second, we use multiplication followed by reduction in order to compute in the ideal class group when K is totally ramified. This, in turn, allows us to implement algorithms to compute the class number of K, and we will describe this in Section 6.3. Finally, the results on ideal inversion will be used to speed up certain class number computation methods.

This chapter is organized as follows. In Section 4.1, we will define the particular ideal basis that lends itself well to efficient arithmetic and will state properties of this basis. In Section 4.2, we will describe the form of prime ideals of $\mathcal{O}$, and state several lemmas on how ideals of certain forms factor. These lemmas will be used to prove the results on general inversion, in Section 4.3, and general multiplication, in Section 4.4. In Section 4.4, we will also supply algorithms to multiply two ideals. Lastly, for the case in which K totally ramifies (so that $Cl(\mathcal{O}) \cong \mathcal{J}_K$), we will describe how to find the unique distinguished ideal equivalent to a given ideal in Section 4.5.

4.1 Canonical Bases

Recall from Section 2.5.1 that a primitive ideal, $\mathfrak{a}$, of $\mathcal{O}$ is an $\mathbb{F}_q[x]$-submodule of $\mathcal{O}$ of rank 3, so it may be written in the form $\mathfrak{a} = [\lambda, \mu, \nu]$, where $\lambda = l_0 + l_1\rho + l_2\omega$, $\mu = m_0 + m_1\rho + m_2\omega$, and $\nu = n_0 + n_1\rho + n_2\omega$, with $l_i, m_i, n_i \in \mathbb{F}_q[x]$, for $i = 0, 1, 2$. For the computational purposes of this chapter, however, it is more efficient to use a *canonical* basis to represent $\mathfrak{a}$. In this section, we define the canonical basis of an $\mathbb{F}_q[x]$-submodule of $\mathcal{O}$ and will show how to convert the basis of a primitive ideal into this basis. Next, we will give the necessary and sufficient conditions under which a submodule is a primitive ideal, and state several more properties that we will use throughout this chapter. We will then give results on ideal containment and equality, and will show how to compute a *minimal* canonical basis.

Definition 4.1.1 *A basis of an $\mathbb{F}_q[x]$-submodule of $\mathcal{O}$ is called* canonical *if it is of the form*

$$\{s,\, s'(u+\rho),\, s''(v+w\rho+\omega)\} \ , \quad \textit{where} \quad s,\, s',\, s'',\, u,\, v,\, w \in \mathbb{F}_q[x] \quad \textit{and} \quad ss's'' \neq 0 \ .$$

We note that in [Sch01], such a basis is called "triangular," reserving the term "canonical" for submodules that are primitive ideals. Given a canonical basis, we may choose s, s', and s'' to be monic. If an ideal, $\mathfrak{a}$, of $\mathcal{O}$ is given by such a basis, then $L(\mathfrak{a}) = s$ and $N(\mathfrak{a}) = ss's''$. By Proposition 2.5.2, every integral ideal, and in particular, every primitive ideal, has a triangular basis, which, we recall, is of the form $\{l_0,\, m_0 + m_1\rho,\, n_0 + n_1\rho + n_2\omega\}$.

The following lemma shows how to find the canonical basis of a primitive ideal in terms of a given basis. Its main application will be to convert the basis of a distinguished ideal associated with an infrastructure divisor to a canonical basis in order to perform a giant step. The proof of this lemma, and other results of [Sch01] that we will state, may be found in [Sch01].

Lemma 4.1.2 (Lemma 4.2 of [Sch01]) *Let $\mathfrak{a} = [L(\mathfrak{a}),\, \mu,\, \nu]$ be a primitive ideal, where $\mu = m_0 + m_1\rho + m_2\omega$ and $\nu = n_0 + n_1\rho + n_2\omega$, with $n_i,\, m_i \in \mathbb{F}_q[x]$, for $i = 0,\, 1,\, 2$. Then $\mathfrak{a} = [s,\, s'(u+\rho),\, s''(v + w\rho + \omega)]$, where*

$$s'' = \gcd(m_2,\, n_2) \ , \quad s' = \frac{m_1 n_2 - n_1 m_2}{s''} \ , \quad s = L(\mathfrak{a}) \ ,$$

$$u = \frac{m_0 n_2 - n_0 m_2}{s's''} \ , \quad v = \frac{am_0 + bn_0}{s''} \ , \quad w = \frac{am_1 + bn_1}{s''} \ ,$$

$$a = a' - \frac{a' - tn_2}{s''} \ , \quad b = b' + \frac{tm_2}{s''} \ ,$$

and $a',\, b',\, t \in \mathbb{F}_q[x]$ are chosen so that $a'm_2 + b'n_2 = s''$ and $s't \equiv a'm_1 + b'n_1 \pmod{s''}$.

The following theorem characterizes exactly when an $\mathbb{F}_q[x]$-submodule of $\mathcal{O}$ is a primitive ideal.

Theorem 4.1.3 (Theorem 4.7 of [Sch01]) *If $\mathfrak{a}$ is a non-zero $\mathbb{F}_q[x]$-submodule of $\mathcal{O}$ with a canonical basis $\{s,\, s'(u+\rho),\, s''(v+w\rho+\omega)\}$, then $\mathfrak{a}$ is a primitive ideal if and only if the canonical basis*

satisfies the following:

$$s's'' \mid s \ , \quad \gcd\left(\frac{s}{s_G s_H}, GH\right) = 1 \ , \quad \gcd(s', H) = 1 \ , \quad s'' \mid H \ , \tag{4.1}$$

$$H(uw - v) \equiv u^2 \pmod{s/s'} \ , \tag{4.2}$$

$$v \equiv Hw^2 \pmod{s's_H/s''} \ , \quad \textit{and} \tag{4.3}$$

$$H(G - vw) \equiv u(v - Hw^2) \pmod{s} \ , \tag{4.4}$$

where $s_G = \gcd(s, G)$ *and* $s_H = \gcd(s, H)$.

In particular, it follows from (4.1) that if $\mathfrak{a}$ is a primitive ideal, then $N(\mathfrak{a}) \mid L(\mathfrak{a})^2$.

Based on the conditions given in Theorem 4.1.3, we may derive the following properties, which will be useful at certain points in this chapter and Chapter 6.

Lemma 4.1.4 (Lemma 4.6 of [Sch01]) *If* $\mathfrak{a} = [s, s'(u + \rho), s''(v + w\rho + \omega)]$ *is a primitive ideal of* $\mathcal{O}$*, with the given canonical basis, then* $s_H \mid u$ *and* $u^3 \equiv -GH^2 \pmod{s/s'}$.

The following lemma gives necessary and sufficient conditions under which $\mathfrak{a}_1 \subseteq \mathfrak{a}_2$, where $\mathfrak{a}_1$ and $\mathfrak{a}_2$ are primitive ideals, in terms of their canonical bases. Following the adage "To contain is to divide," this lemma will be used to describe the product of two ideals, whose norms are relatively prime, in Theorem 4.2.2.

Lemma 4.1.5 (Lemma 4.1.1 of [Sch01]) *Let* $\mathfrak{a}_1 = [s_1, s_1'(u_1 + \rho), s_1''(v_1 + w_1\rho + \omega)]$ *and* $\mathfrak{a}_2 = [s_2, s_2'(u_2 + \rho), s_2''(v_2 + w_2\rho + \omega)]$ *be two primitive ideals given in terms of canonical bases. Then* $\mathfrak{a}_1 \subseteq \mathfrak{a}_2$ *if and only if*

$$\begin{aligned} s_2 \mid s_1 \ , \quad s_2' \mid s_1' \ , \quad s_2'' \mid s_1'' \ , \\ s_1'u_1 \equiv s_1'u_2 \pmod{s_2} \ , \\ s_1''w_1 \equiv s_1''w_2 \pmod{s_2'} \ , \\ s_1''v_1 \equiv s_1''(v_2 + u_2(w_1 - w_2)) \pmod{s_2} \ . \end{aligned}$$

We note that any primitive ideal has several possible canonical bases. As such, we desire a basis that identifies a particular ideal uniquely and also minimizes the space to represent the ideal. The following lemma characterizes ideal equality, and shows how to obtain a canonical basis of minimal size.

Lemma 4.1.6 (Lemma 4.1.2 of [Sch01])

1. *If* $\mathfrak{a}_i = [s_i, s_i'(u_i + \rho), s_i''(v_i + w_i\rho + \omega)]$ *is a primitive ideal, for* $i = 1, 2$*, such that each* s_i*,* s_i'*, and* s_i'' *is monic, then* $\mathfrak{a}_1 = \mathfrak{a}_2$ *if and only if* $s_1 = s_2$*,* $s_1' = s_2'$*,* $s_1'' = s_2''$*,* $u_1 \equiv u_2 \pmod{s_1/s_1'}$*,* $w_1 \equiv w_2 \pmod{s_1'}$*, and* $v_1 \equiv v_2 + u_2(w_1 - w_2) \pmod{s_1/s_1''}$.

2. *If* $\mathfrak{a} = [s, s'(u + \rho), s''(v + w\rho + \omega)]$ *is a primitive ideal, then all other canonical bases of* $\mathfrak{a}$ *are given by* $[s, s'(\tilde{u} + \rho), s''(\tilde{v} + \tilde{w}\rho + \omega)]$*, where*

$$\tilde{u} \equiv u \pmod{s/s'} \ , \quad \tilde{w} \equiv w \pmod{s'} \ , \quad \textit{and} \quad \tilde{v} \equiv v + u(\tilde{w} - w) \pmod{s/s''} \ .$$

This lemma allows us to find a canonical basis of a primitive ideal such that $\deg(u) < \deg(s/s')$, $\deg(v) < \deg(s/s'')$, and $\deg(w) < \deg(s')$. Moreover, this lemma implies that such a basis identifies a primitive ideal uniquely. We call such a canonical basis *minimal*.

In this section, we defined and established the properties of the canonical basis of a principal ideal and showed how to quickly produce the minimal canonical basis of an ideal. We will use this basis to describe the structure of prime ideals of $\mathcal{O}$ and basic ideal multiplication and inversion.

4.2 Partial Factorization and Multiplication of Ideals

In this section, we collect a number of lemmas that will be used to prove the major results on ideal inversion and multiplication in Sections 4.3 and 4.4, respectively. In Section 4.2.1, we describe the splitting behavior and the canonical basis of each type of prime ideal. In Section 4.2.2, we state the theorem of [Sch01] that describes how to multiply two ideals whose norms are relatively prime. We will also give results on divisibility properties of ideals having certain forms. Next, we describe the bases of products of prime ideals of a common type in Section 4.2.3 and show how they factor into ideals of special forms. These results are then combined in Section 4.2.4 to show how an arbitrary primitive ideal partially factors into ideals of special forms. Lastly, Section 4.2.5 describes how to multiply ideals having special forms.

4.2.1 Prime Ideals

The basis of the lemmas in this section is the following theorem on the structure of prime ideals of $\mathcal{O}$. This theorem was first proved by Voronoi for cubic number fields [Vor94], and was adapted to cubic function fields by Scheidler [Sch01].

Theorem 4.2.1 (Theorem 3.1 of [Sch01]) *Let $K = \mathbb{F}_q(C)$ be a purely cubic function field, where $C : Y^3 = F(x) = G(x)H^2(x)$ and $G, H \in \mathbb{F}_q[x]$ are monic, relatively prime, and square-free. If $P \in \mathbb{F}_q[x]$ is an irreducible polynomial, then the principal ideal $\langle P \rangle$ splits into prime ideals in $\mathcal{O}$ as follows:*

1. *If $P \mid G$, then $\langle P \rangle = \mathfrak{p}^3$, where $\mathfrak{p} = [P, \rho, \omega]$, and $\mathfrak{p}^2 = [P, P\rho, \omega]$. So $\mathfrak{p}$ ramifies and is called a* type 1 prime ideal.

2. *If $P \mid H$, then $\langle P \rangle = \mathfrak{p}^3$, where $\mathfrak{p} = [P, \rho, \omega]$, and $\mathfrak{p}^2 = [P, \rho, P\omega]$. So $\mathfrak{p}$ ramifies and is called a* type 2 prime ideal.

3. *If $P \nmid GH$, F is a cube modulo P, and $q^{\deg(P)} \equiv 2 \pmod 3$, then F has a unique cube root, X, modulo P in $\mathbb{F}_q[x]$. In this case $\langle P \rangle = \mathfrak{pq}$, where*

$$\mathfrak{p}^i = [P^i, -X_i + \rho, -X_i^2 r_i + \omega] \quad \text{and} \quad \mathfrak{q}^i = [P^i, P^i\rho, X_i^2 r_i + X_i r_i \rho + \omega] \ ,$$

for $i \in \mathbb{N}$, with

$$X_1 \equiv X r_1 H \pmod{PH} \quad \text{and} \quad X_{i+1} = X_i + A_i(D - X_i^3) \ ,$$

where

$$3X_i^2 A_i \equiv 1 \pmod{P^i} \quad \textit{and} \quad r_i H \equiv 1 \pmod{P^i} \ ,$$

for $i \in \mathbb{N}$. $\mathfrak{p}$ *and* $\mathfrak{q}$ *are called* type 3 prime ideals.

4. *If* $P \nmid GH$, F *is a cube modulo* P, *and* $q^{\deg(P)} \equiv 1 \pmod 3$, *then* F *has three distinct cube roots,* X, X', *and* X'' *modulo* P, *in* $\mathbb{F}_q[x]$. *In this case* $\langle P \rangle = \mathfrak{p}\mathfrak{p}'\mathfrak{p}''$, *where*

$$\begin{aligned} \mathfrak{p}^i &= [P^i,\ -X_i + \rho,\ -X_i^2 r_i + \omega] \ , \\ (\mathfrak{p}\mathfrak{p}')^i &= [P^i,\ P^i\rho,\ (X_i'')^2 r_i + X_i'' r_i \rho + \omega] \ , \quad \textit{and} \\ \mathfrak{p}^{i+j}(\mathfrak{p}')^i &= [P^{i+j},\ P^i(-X_j + \rho),\ (X_i'')^2 r_i + P^i Q_{ij} + X_i'' r_i \rho + \omega] \ , \end{aligned}$$

for $i, j \in \mathbb{N}$ *with* $X_1 \equiv X r_1 H \pmod{PH}$ *and* $X_{i+1} = X_i + A_i(D - X_i^3)$, *where* $3X_i^2 A_i \equiv 1 \pmod{P^i}$, $r_i H \equiv 1 \pmod{P^i}$, *and*

$$\frac{Q_{ij}(X_j - X_i'')}{H} \equiv \frac{(X_i'')^3 - F}{H^2 P^i} \pmod{P^j} \ ,$$

for $i \in \mathbb{N}$. *Similar congruences hold for* X' *and* X'' *and the corresponding* A' *and* A'' *for determining the basis for other such ideals for* $\mathfrak{p}'$, $\mathfrak{p}''$, *and so on.* $\mathfrak{p}$, $\mathfrak{p}'$, *and* $\mathfrak{p}''$ *are called* type 4 prime ideals.

5. *If* F *is not a cube modulo* P, *then* $\langle P \rangle = \mathfrak{p}$ *is inert, and* $\mathfrak{p} = [P, P\rho, P\omega]$ *is called a* type 5 prime ideal.

Notice that if $q \equiv 1 \pmod 3$, then there are no type 3 prime ideals and that type 5 prime ideals are not primitive. Also, if the curve, C, defining K is nonsingular, then $\mathcal{O}$ has no type 2 prime ideals.

4.2.2 Containment and Divisibility

In this section, we will give three results that are consequences of Lemma 4.1.5. The first result, due to Scheidler [Sch01], finds the product of two ideals whose norms are relatively prime. This theorem is very important for applications in which ideal multiplication is required, since most ideal multiplications will be between such ideals. We will also use this result to prove many lemmas in this section. The proof is given in [Sch01].

Theorem 4.2.2 (Theorem 4.4 of [Sch01]) *Let* $K = \mathbb{F}_q(C)$ *be a purely cubic function field, with* $C : Y^3 = G(x)H^2(x)$, *and, for* $i = 1, 2$, *let* $\mathfrak{a}_i = [s_i,\ s_i'(u_i + \rho),\ s_i''(v_i + w_i\rho + \omega)]$ *be two primitive ideals, given in terms of their canonical bases, with* $\gcd(s_1, s_2) = 1$. *Then* $\mathfrak{a}_1\mathfrak{a}_2 = [S,\ S'(U + \rho),\ S''(V + W\rho + \omega)]$, *where*

$$S = s_1 s_2 \ , \qquad S' = s_1' s_2' \ , \qquad S'' = s_1'' s_2'' \ ,$$

$$U \equiv \begin{cases} u_1 \pmod{s_1/s_1'} \\ u_2 \pmod{s_2/s_2'} \end{cases} , W \equiv \begin{cases} w_1 \pmod{s_1'} \\ w_2 \pmod{s_2'} \end{cases} , \textit{and} V \equiv \begin{cases} v_1 + u_1(W - w_1) \pmod{s_1/s_1''} \\ v_2 + u_2(W - w_2) \pmod{s_2/s_2''} \end{cases} .$$

The remaining two results of the section introduce canonical bases of special forms that facilitate inversion and multiplication of ideals having such bases. They will also prove helpful conditions for

divisibility in terms of these bases.

Lemma 4.2.3 *If* $\mathfrak{a}_1 = [s,\, u+\rho,\, v+w\rho+\omega]$ *and* $\mathfrak{a}_2$ *are primitive ideals, then* $\mathfrak{a}_2 \mid \mathfrak{a}_1$ *if and only if* $\mathfrak{a}_2 = [S,\, u+\rho,\, v+w\rho+\omega]$*, with* $S \mid s$*.*

Proof: Let $\mathfrak{a}_2 = [S,\, S'(U+\rho),\, S''(V+W\rho+\omega)]$. We have $\mathfrak{a}_2 \mid \mathfrak{a}_1$ if and only if $\mathfrak{a}_1 \subseteq \mathfrak{a}_2$. By Lemma 4.1.5 $\mathfrak{a}_1 \subseteq \mathfrak{a}_2$ if and only if $S \mid s$, $S' \mid 1$, $S'' \mid 1$, $u \equiv U \pmod{S}$, $w \equiv W \pmod{S'}$, and $v \equiv V + U(w-W) \pmod{S}$. Therefore, $S' = S'' = 1$, and by Part 2 of Lemma 4.1.6, we can then take $U = u$, $V = v$, and $W = w$. □

Lemma 4.2.4 *If* $\mathfrak{a}_1$ *is a primitive ideal such that* $s = s'$ *and* $s'' = 1$*, then* $\mathfrak{a}_1$ *can be written in the form* $[s,\, s\rho,\, v+w\rho+\omega]$*. Furthermore, if* $\mathfrak{a}_2 = [S,\, S(u_2+\rho),\, v_2+w_2\rho+\omega]$ *is another primitive ideal given by a basis of this form, then* $\mathfrak{a}_2 \mid \mathfrak{a}_1$ *if and only if* $\mathfrak{a}_2 = [S,\, S\rho,\, v+w\rho+\omega]$*, with* $S \mid s$*.*

Proof: First, we write $\mathfrak{a}_1 = [s,\, s(u+\rho),\, v+w\rho+\omega]$. Since $s/s' = 1$, we may replace u with 0 in the basis, by Part 2 of Lemma 4.1.6. From this, w and v are determined uniquely modulo s. Thus, $\mathfrak{a}_1 = [s,\, s\rho,\, v+w\rho+\omega]$.

Now let $\mathfrak{a}_2 = [S,\, S(U+\rho),\, V+W\rho+\omega]$. We have $\mathfrak{a}_2 \mid \mathfrak{a}_1$ if and only if $\mathfrak{a}_1 \subseteq \mathfrak{a}_2$. By Lemma 4.1.5, $\mathfrak{a}_1 \subseteq \mathfrak{a}_2$ if and only if $S \mid s$, $0 \equiv sU \pmod{S}$, $w \equiv W \pmod{S}$, and $v \equiv V \pmod{S}$. By Part 2 of Lemma 4.1.6, we can then take $U = 0$, $V = v$, and $W = w$. □

The basis forms we presented here will be used in the next section to describe ideals that are the product of prime ideals of a common type.

4.2.3 Products of Prime Ideals of a Common Type

In order to determine the inverse of an arbitrary ideal and the product of any two ideals, we will first consider ideals having the forms introduced in the previous section. The four lemmas in this section consider ideals whose factors are all prime ideals of a single type, first type 1, then type 2, and so on, and show how they factor into ideals of these special forms. We will not consider products of type 5 prime ideals, however, because such ideals are non-primitive. In the proofs of these lemmas, and throughout the remainder of the thesis, we will write $a^n \| b$, where a and b are members of any ring and $n \in \mathbb{N}$, to denote that a^n exactly divides b, that is, $a^n \mid b$, but $a^{n+1} \nmid b$.

Lemma 4.2.5 (Partial Factorization of Products of Type 1 Prime Ideals) *If* $\mathfrak{a}$ *is a primitive ideal of* $\mathcal{O}$ *that is the product of type 1 prime ideals of* $\mathcal{O}$*, then* $\mathfrak{a}$ *is of the form* $[s,\, s'\rho,\, \omega]$*, where* $s \mid G$ *and* $s' \mid s$ *and* $\mathfrak{a} = [s/s',\, \rho,\, \omega][s',\, s'\rho,\, \omega] = [s/s',\, \rho,\, \omega][s',\, \rho,\, \omega]^2$*.*

Proof: Suppose $\mathfrak{a} = [s,\, s'(u+\rho),\, s''(v+w\rho+\omega)]$ is the product of type 1 prime ideals. Then $\mathfrak{a}$ is the product of primes $\mathfrak{p} = [P,\, \rho,\, \omega]$ and squares of primes, $\mathfrak{p}^2 = [P,\, P\rho,\, \omega]$, where $P \mid G$ is an irreducible polynomial. Since $\mathfrak{a}$ is primitive, it is not divisible by $\mathfrak{p}^3$, for any prime ideal, $\mathfrak{p}$, so by Theorem 4.2.2, we have

$$\mathfrak{a} = \prod_i [P_i,\, \rho,\, \omega] \prod_j [P_j,\, \rho,\, \omega]^2 = \prod_i [P_i,\, \rho,\, \omega] \prod_j [P_j,\, P_j\rho,\, \omega] = \left[\prod_i P_i \prod_j P_j,\, \left(\prod_j P_j \right) \rho,\, \omega \right] ,$$

where the P_i and P_j are pairwise relatively prime irreducible polynomials in $\mathbb{F}_q[x]$. If $s = \prod_i P_i \prod_j P_j$ and $s' = \prod_j P_j$, then $s'' = 1$, $\mathfrak{a} = [s,\, s'\rho,\, \omega]$, and $s' \mid s \mid G$.

Now let $\mathfrak{b} = [s/s',\, \rho,\, \omega]$ and $\mathfrak{c} = [s',\, s'\rho,\, \omega] = [s',\, \rho,\, \omega]^2$. Since $s \mid G$, s is square-free, so we have $\gcd(s/s',\, s') = 1$. By Theorem 4.2.2, we have $\mathfrak{bc} = [s,\, s'\rho,\, \omega] = \mathfrak{a}$. □

Lemma 4.2.6 (Partial Factorization of Products of Type 2 Prime Ideals) *If $\mathfrak{a}$ is a primitive ideal of $\mathcal{O}$ that is the product of type 2 prime ideals of $\mathcal{O}$, then $\mathfrak{a}$ is of the form $[s,\, \rho,\, s''\omega]$, where $s \mid H$ and $s'' \mid s$, and $\mathfrak{a} = [s/s'',\, \rho,\, \omega][s'',\, \rho,\, s''\omega] = [s/s'',\, \rho,\, \omega][s'',\, \rho,\, \omega]^2$. Further, $\gcd(s/s'',\, s'') = 1$*

Proof: Suppose $\mathfrak{a} = [s,\, s'(u+\rho),\, s''(v + w\rho + \omega)]$ is the product of type 2 prime ideals. Then $\mathfrak{a}$ is the product of primes $\mathfrak{p} = [P,\, \rho,\, \omega]$ and squares of primes, $\mathfrak{p}^2 = [P,\, \rho,\, P\omega]$, where $P \mid H$ is an irreducible polynomial. Since $\mathfrak{a}$ is primitive, it is not divisible by $\mathfrak{p}^3$, for any prime ideal, $\mathfrak{p}$, so by Theorem 4.2.2, we have

$$\mathfrak{a} = \prod_i [P_i,\, \rho,\, \omega] \prod_j [P_j,\, \rho,\, \omega]^2 = \prod_i [P_i,\, \rho,\, \omega] \prod_j [P_j,\, \rho,\, P_j\omega] = \left[\prod_i P_i \prod_j P_j,\, \rho,\, \left(\prod_j P_j \right) \omega \right] \ ,$$

where the P_i and P_j are pairwise relatively prime irreducible polynomials in $\mathbb{F}_q[x]$. If $s = \prod_i P_i \prod_j P_j$ and $s'' = \prod_j P_j$, then $s' = 1$, $\mathfrak{a} = [s,\, \rho,\, s''\omega]$, and $s'' \mid s \mid H$.

Now let $\mathfrak{b} = [s/s'',\, \rho,\, \omega]$ and $\mathfrak{c} = [s'',\, \rho,\, s''\omega] = [s'',\, \rho,\, \omega]^2$. Since $s \mid H$, s is square-free, so $\gcd(s/s'',\, s'') = 1$. By Theorem 4.2.2, we have $\mathfrak{bc} = [s,\, \rho,\, s''\omega] = \mathfrak{a}$. □

Lemma 4.2.7 (Partial Factorization of Products of Type 3 Prime Ideals) *If $\mathfrak{a}$ is a primitive ideal of $\mathcal{O}$ that is the product of type 3 prime ideals, then $\mathfrak{a}$ is of the form $[s,\, s'(u+\rho),\, v+w\rho+\omega]$, with $s' \mid s$ and $\gcd(s,\, GH) = 1$, and $\mathfrak{a} = [s/s',\, u+\rho,\, v+w\rho+\omega][s',\, s'\rho,\, v+w\rho+\omega]$.*

Proof: Suppose $\mathfrak{a} = [s,\, s'(u+\rho),\, s''(v + w\rho + \omega)]$ is the product of type 3 prime ideals. If $P \mid s$, then $P \nmid GH$, since no ramified prime ideals divide $\mathfrak{a}$, and $\gcd(s,\, GH) = 1$. By (4.1), we have $s' \mid s$, $s'' \mid s$, and $s'' \mid H$, so we have $s'' = 1$. Now, if $\mathfrak{p} \mid \mathfrak{a}$, then, by Theorem 4.2.1, $\mathfrak{p}$ is either of the form $[P,\, u_P + \rho,\, v_P + \omega]$ or $[P,\, P\rho,\, v_P + w_P\rho + \omega]$. If $\langle P \rangle = \mathfrak{pq}$ and $\mathfrak{p}$ and $\mathfrak{q}$ are type 3 prime ideals such that $P^a \| s$, then since $\mathfrak{a}$ is primitive, either $\mathfrak{p}^a \| \mathfrak{a}$ or $\mathfrak{q}^a \| \mathfrak{a}$. Therefore, we have $\mathfrak{a} = \prod_i [P_i^{a_i},\, u_i + \rho,\, v_i + \omega] \prod_j \left[P_j^{b_j},\, P_j^{b_j}\rho,\, v_j + w_j\rho + \omega \right]$, where the P_i and P_j are pairwise relatively prime, $P_i^{a_i} \| s$, and $P_j^{b_j} \| s$, for all i and j. By Theorem 4.2.2, this product is of the form $[s_1,\, u_1 + \rho,\, v_1 + \omega][s_2,\, s_2'\rho,\, v_2 + w_2\rho + \omega]$, where $s_1 = \prod_i P_i^{a_i}$ and $s_2 = s_2' = \prod_j P_j^{b_j}$. Since $\gcd(s_1,\, s_2) = 1$, it also follows by Theorem 4.2.2 that this product is of the form $\mathfrak{a} = [s_1 s_2,\, s_2'(u+\rho),\, v + w\rho + \omega]$, so equating coefficients, we have $s_2 = s_2' = s'$ and $s_1 = s/s'$.

We claim that $[s/s',\, u_1 + \rho,\, v_1 + \omega] = [s/s',\, u+\rho,\, v + w\rho + \omega]$ and $[s',\, s'\rho,\, v_2 + w_2\rho + \omega] = [s',\, s'\rho,\, v + w\rho + \omega]$. By Part 1 of Lemma 4.1.6, it suffices to show that $u \equiv u_1 \pmod{s/s'}$, $w \equiv 0 \pmod{1}$, $w \equiv w_2 \pmod{s'}$, $v \equiv v_1 + u_1 w \pmod{s/s'}$, and $v \equiv v_2 \pmod{s'}$. However, these are precisely the same congruences given by Theorem 4.2.2 for the product $[s,\, s'(u+\rho),\, s''(v+w\rho+\omega)] = [s/s',\, u_1+\rho,\, v_1+\omega][s',\, s'\rho,\, v_2 + w_2\rho + \omega]$, and the desired result follows. □

Lemma 4.2.8 (Partial Factorization of Products of Type 4 Prime Ideals) *If $\mathfrak{a}$ is a primitive ideal of $\mathcal{O}$ that is the product of type 4 prime ideals, then $\mathfrak{a}$ is of the form $[s,\, s'(u+\rho),\, v+w\rho+\omega]$, with $s' \mid s$ and $\gcd(s,\, GH) = 1$, and factors as $\mathfrak{a} = [s/s',\, u+\rho,\, v+w\rho+\omega][s',\, s'\rho,\, v+w\rho+\omega]$.*

Proof: Suppose $\mathfrak{a} = [s,\, s'(u+\rho),\, s''(v+w\rho+\omega)]$ is the product of type 4 prime ideals. If $P \mid s$, then $P \nmid GH$, since no ramified prime ideals divide $\mathfrak{a}$, and $\gcd(s,\, GH) = 1$. By (4.1), we have $s' \mid s$, $s'' \mid s$, and $s'' \mid H$, so we have $s'' = 1$. Now, if $P|s$, then, by Theorem 4.2.1, we have $\langle P\rangle = \mathfrak{p}\mathfrak{p}'\mathfrak{p}''$. Since $\mathfrak{a}$ is primitive, at most two prime ideals over $\langle P\rangle$ divide $\mathfrak{a}$. If $P^a \| s$, then by Theorem 4.2.1, we have, without loss of generality, $\mathfrak{p}^a(\mathfrak{p}')^b \| \mathfrak{a}$, where $0 \le b \le a$. Since $\mathfrak{p}^a(\mathfrak{p}')^b$ is of the form $[P^a,\, P^b(u_P+\rho),\, v_P + w_P\rho + \omega]$, we have

$$\begin{aligned}\mathfrak{a} &= \prod_i \mathfrak{p}_i^{a_i}(\mathfrak{p}_i')^{b_i} = \prod_i \left[P_i^{a_i},\, P_i^{b_i}(u_i+\rho),\, v_i + w_i\rho + \omega\right] \\ &= \left[\prod_i P_i^{a_i},\, \left(\prod_i P_i^{b_i}\right)(u+\rho),\, v + w\rho + \omega\right] \quad ,\end{aligned}$$

where $P^{a_i} \| s$ and $0 \le b_i \le a_i$ for all i, so $s = \prod_i P_i^{a_i}$ and $s' = \prod_i P_i^{b_i}$. We may also factor

$$\mathfrak{a} = \left(\prod_i \mathfrak{p}_i^{a_i - b_i}\right)\left(\prod_i (\mathfrak{p}_i\mathfrak{p}_i')^{b_i}\right) \quad .$$

By Theorem 4.2.1, $\mathfrak{p}^a$ is of the form $[P^a,\, u_P+\rho,\, v_P+\omega]$ and $(\mathfrak{p}\mathfrak{p}')^b$ is of the form $[P^b,\, P^b\rho,\, v_P + w_P\rho + \omega]$, so

$$\mathfrak{a} = \prod_i \left[P_i^{a_i-b_i},\, u_i+\rho,\, v_i+\omega\right] \prod_i \left[P_i^{b_i},\, P_i^{b_i}\rho,\, v_i + w_i\rho + \omega\right] \quad .$$

Since $\gcd(P_i,\, P_j) = 1$ if and only if $i \ne j$, by Theorem 4.2.2, this product is of the form $\mathfrak{b}\mathfrak{c}$, where $\mathfrak{b} = [s_1,\, u_1+\rho,\, v_1+\omega]$, $\mathfrak{c} = [s_2,\, s_2'\rho,\, v_2 + w_2\rho + \omega]$, $s_1 = \prod_i P_i^{a_i-b_i}$, and $s_2 = s_2' = \prod_i P_i^{b_i}$. However, we showed earlier that $s' = \prod_i P_i^{b_i}$, so $s_2 = s_2' = s'$. Considering norms, we have $ss' = N(\mathfrak{a}) = N(\mathfrak{b}\mathfrak{c}) = N(\mathfrak{b})N(\mathfrak{c}) = s_1 s_2 s_2' = s_1(s')^2$, so $s_1 = s/s'$.

We claim that $\mathfrak{b} = [s/s',\, u+\rho,\, v+w\rho+\omega]$ and $\mathfrak{c} = [s',\, s'\rho,\, v+w\rho+\omega]$. By Part 1 of Lemma 4.1.6, it suffices to show that $u \equiv u_1 \pmod{s/s'}$, $w \equiv 0 \pmod 1$, $w \equiv w_2 \pmod{s'}$, $v \equiv v_1 + u_1 w \pmod{s/s'}$, and $v \equiv v_2 \pmod{s'}$. Since $\mathfrak{a} \subseteq \mathfrak{b}$ and $\mathfrak{a} \subseteq \mathfrak{c}$, however, these congruences are precisely those given by Lemma 4.1.5 for ideal containment, and the desired result follows. □

This completes the description of products of primitive prime ideals of a single type. In the next section, we will combine these results to show how ideals partially factor into ideals having special forms.

4.2.4 Ideal Factorization

The strategy we will use to prove general results on ideal inversion and multiplication will be to factor the operand(s) into a product of type 2 prime ideals, ideals such that $s' = s'' = 1$, and ideals in which $s = s'$. We will then show how ideals of these special forms invert and multiply, and recombine the factors. In this section, we give four results that describe the structure of ideals in various forms and how they partially factor into the desired forms. In particular, the last lemma

proves this for the most general case. The first result will be used to extract type 1 prime ideals from an ideal in which $s' = s'' = 1$ and proves more about the structure of such ideals.

Corollary 4.2.9 *If* $\mathfrak{a} = [s, u + \rho, v + w\rho + \omega]$ *is a primitive ideal and* $d = \gcd(s, GH)$*, then* $\mathfrak{a}$ *factors as*

$$\mathfrak{a} = \mathfrak{b}\mathfrak{c} \ , \quad \text{with} \quad \mathfrak{b} = [d, \rho, \omega] \quad \text{and} \quad \mathfrak{c} = \left[\frac{s}{d}, u + \rho, v + w\rho + \omega\right] \ ,$$

where $\mathfrak{b}$ *is the square-free product of type 1 and 2 prime ideals and* $\mathfrak{c}$ *is the product of type 3 prime ideals of degree* 1 *and 4 prime ideals, where for any type 4 prime ideal* $\mathfrak{p}$ *dividing* $\mathfrak{a}$*,* $\mathfrak{p}'$ *and* $\mathfrak{p}''$ *do not divide* $\mathfrak{a}$*.*

Proof: We write $\mathfrak{a} = \mathfrak{b}\mathfrak{c}$, where $\mathfrak{b}$ is the product of type 1 and 2 prime ideals and $\mathfrak{c}$ is the product of type 3 and 4 prime ideals. Since $s' = s'' = 1$, from the proofs of Lemmas 4.2.5 and 4.2.6, we see that $\mathfrak{b}$ must be square-free, and $\mathfrak{b}$ is of the desired form.

By Lemma 4.2.3, $\mathfrak{c} \mid \mathfrak{a}$ implies that $\mathfrak{c} = [c, u + \rho, v + w\rho + \omega]$, for some $c \in \mathbb{F}_q[x]$. Since $s = N(\mathfrak{a}) = N(\mathfrak{b})N(\mathfrak{c}) = dc$, we must have $c = s/d$. □

Corollary 4.2.10 *If* $\mathfrak{a} = [s, s\rho, v + w\rho + \omega]$ *is a primitive ideal, then* $\mathfrak{a}$ *is the product of squares of type 1 prime ideals, type 3 ideals,* $\mathfrak{q}$*, of degree* 2*, and ideals of the form* $\mathfrak{p}\mathfrak{p}'$*, where* $\mathfrak{p}$ *and* $\mathfrak{p}'$ *are type 4 prime ideals and* $\mathfrak{p}''$ *does not divide* $\mathfrak{a}$*.*

Proof: Since $s = s'$ and $s'' = 1$, $\mathfrak{a}$ cannot contain any type 2 prime ideals, by the proof of Lemma 4.2.6. Similarly, the proof of Lemma 4.2.5 yields that $\mathfrak{a}$ can only contain squares of type 1 prime ideals. Likewise, since $s = s'$, from the proofs of Lemmas 4.2.7 and 4.2.8 as well as Theorem 4.2.1, we see that the unramified factors of $\mathfrak{a}$ are of the form $\mathfrak{a}_P = [s_P, s_P\rho, v_P + w_P\rho + \omega]$, where $\mathfrak{a}_P$ is a power of a degree 2 type 3 prime ideal or a power of $\mathfrak{p}\mathfrak{p}'$, where $\mathfrak{p}$ is a type 4 prime ideal; in the latter case, $\mathfrak{p}''$ does not divide $\mathfrak{a}$, since $\mathfrak{a}$ is primitive. □

We combine Lemmas 4.2.5, 4.2.7, and 4.2.8 into the following result to describe how ideals that do not contain any type 2 prime ideals factor into ideals of certain forms. This lemma is a generalization of Equation 4.2 of [Bau04] from the case that K is defined by a nonsingular curve (i.e. H is taken to be 1) to the case that the defining curve is possibly singular.

Lemma 4.2.11 (Partial Factorization of Products of Non-Type 2 Prime Ideals)

1. *If* $\mathfrak{a}$ *is a primitive ideal of* $\mathcal{O}$ *that is the product of type 1, 3, and 4 prime ideals of* $\mathcal{O}$*, then* $\mathfrak{a}$ *is of the form* $\mathfrak{a} = [s, s'(u + \rho), v + w\rho + \omega]$*, where* $s' \mid s$ *and* $\gcd(s, H) = 1$*. If none of the prime ideals dividing* $\mathfrak{a}$ *ramify, then* $\gcd(s, GH) = 1$*.*

2. *If* $\mathfrak{a} = [s, s'(u + \rho), v + w\rho + \omega]$ *is a primitive ideal which is the product of type 1, 3, and 4 prime ideals of* $\mathcal{O}$*, then* $\mathfrak{a}$ *factors as*

$$\mathfrak{a} = \left[\frac{s}{s'}, u + \rho, v + w\rho + \omega\right][s', s'\rho, v + w\rho + \omega] \ .$$

Moreover, this factorization is unique in the following sense: if $\mathfrak{a} = \mathfrak{b}\mathfrak{c}$*, where* $\mathfrak{b} = [S/S', U_1 + \rho, V_1 + W_1\rho + \omega]$ *and* $\mathfrak{c} = [S', S'\rho, V_2 + W_2\rho + \omega]$ *for some* S*,* S'*,* U_1*,* V_1*,* V_2*,* W_1*,* $W_2 \in \mathbb{F}_q[x]$*, then* $\mathfrak{b} = [s/s', u + \rho, v + w\rho + \omega]$ *and* $\mathfrak{c} = [s', s'\rho, v + w\rho + \omega]$*.*

Proof: An arbitrary ideal is of the form $\mathfrak{a} = [s,\, s'(u+\rho),\, s''(v+w\rho+\omega)]$. By (4.1) and Lemmas 4.2.5, 4.2.7, and 4.2.8, if no type 2 prime ideal divides $\mathfrak{a}$, then $s'' = 1$ and $\gcd(s,\, H) = 1$. A type 1 prime, $\mathfrak{p}$, divides $\mathfrak{a}$, if and only if $L(\mathfrak{p}) \mid s$ and $L(\mathfrak{p}) \mid G$, by the previously stated lemmas. Therefore, $\mathfrak{a}$ is free of type 1 prime ideal factors if and only if $\gcd(s,\, G) = 1$.

For the second part, we may write $\mathfrak{a} = \mathfrak{b}\mathfrak{c}\mathfrak{d}$, where $\mathfrak{b} = [s_1,\, s_1'\rho,\, \omega]$, $\mathfrak{c} = [s_3,\, s_3'(u_3+\rho),\, v_3+w_3\rho+\omega]$, and $\mathfrak{d} = [s_4,\, s_4'(u_4+\rho),\, v_4+w_4\rho+\omega)]$ are the product of type 1, 3, and 4 prime ideals, respectively. By Lemmas 4.2.5, 4.2.7, and 4.2.8, $\mathfrak{b}$, $\mathfrak{c}$, and $\mathfrak{d}$ factor as $\mathfrak{b} = \mathfrak{b}_1\mathfrak{b}_2$, $\mathfrak{c} = \mathfrak{c}_1\mathfrak{c}_2$, and $\mathfrak{d} = \mathfrak{d}_1\mathfrak{d}_2$, where $\mathfrak{b}_1 = [s_1/s_1',\, \rho,\, \omega]$, $\mathfrak{b}_2 = [s_1',\, s_1'\rho,\, \omega]$, $\mathfrak{c}_1 = [s_3/s_3',\, u_3+\rho,\, v_3+w_3\rho+\omega]$, $\mathfrak{c}_2 = [s_3',\, s_3'\rho,\, v_3+w_3\rho+\omega]$, $\mathfrak{d}_1 = [s_4/s_4',\, u_4+\rho,\, v_4+w_4\rho+\omega]$, and $\mathfrak{d}_2 = [s_4',\, s_4'\rho,\, v_4+w_4\rho+\omega]$. Write $\mathfrak{a}_1 = \mathfrak{b}_1\mathfrak{c}_1\mathfrak{d}_1$ and $\mathfrak{a}_2 = \mathfrak{b}_2\mathfrak{c}_2\mathfrak{d}_2$ so that $\mathfrak{a} = \mathfrak{a}_1\mathfrak{a}_2$. Since the norms of $\mathfrak{b}_1$, $\mathfrak{c}_1$, and $\mathfrak{d}_1$ are pairwise relatively prime and the norms of $\mathfrak{b}_2$, $\mathfrak{c}_2$, and $\mathfrak{d}_2$ are pairwise relatively prime, by Theorem 4.2.2, we have

$$\mathfrak{a}_1 = \left[\frac{s_1s_3s_4}{s_1's_3's_4'},\, U_1+\rho,\, V_1+W_1\rho+\omega\right] \quad \text{and} \quad \mathfrak{a}_2 = [s_1's_3's_4',\, s_1's_3's_4'\rho,\, V_2+W_2\rho+\omega] \ ,$$

for some $U_1,\, V_1,\, V_2,\, W_1,\, W_2 \in \mathbb{F}_q[x]$. Since $\mathfrak{a} = \mathfrak{b}\mathfrak{c}\mathfrak{d} = [s_1,\, s_1'\rho,\, \omega][s_3,\, s_3'(u_3+\rho),\, v_3+w_3\rho+\omega][s_4,\, s_4'(u_4+\rho),\, v_4+w_4\rho+\omega)] = [s_1s_3s_4,\, s_1's_3's_4'(u+\rho),\, v+w\rho+\omega]$, we have $s_1's_3's_4' = s'$ and $(s_1s_3s_4)/(s_1's_3's_4') = s/s'$. Using an argument similar to the one in the proof of Lemma 4.2.8, $\mathfrak{a} \subseteq \mathfrak{a}_1$ and $\mathfrak{a} \subseteq \mathfrak{a}_2$ imply $\mathfrak{a}_1 = [s/s',\, u+\rho,\, v+w\rho+\omega]$ and $\mathfrak{a}_2 = [s',\, s'\rho,\, v+w\rho+\omega]$.

To show the uniqueness result, we invoke the proofs of Lemmas 4.2.5, 4.2.7, and 4.2.8. The factor $[s',\, s'\rho,\, v+w\rho+\omega]$ is the product of exactly the following ideals:

- $\mathfrak{p}^2$, where $\mathfrak{p}$ is a type 1 prime ideal, with $\mathfrak{p}^2 \| \mathfrak{a}$,
- $\mathfrak{p}^i$, where $\mathfrak{p}$ is a type 3 prime ideal of degree 2, with $\mathfrak{p}^i \| \mathfrak{a}$, and
- $(\mathfrak{p}\mathfrak{p}')^i$, where $\mathfrak{p}$ is a type 4 prime ideal, with $\mathfrak{p}^i \| \mathfrak{a}$ or $(\mathfrak{p}')^i \| \mathfrak{a}$.

Hence, the factor $[s/s',\, u+\rho,\, v+w\rho+\omega]$ is the product of exactly the following ideals:

- type 1 prime ideals, $\mathfrak{p}$, with $\mathfrak{p} \| \mathfrak{a}$,
- $\mathfrak{p}^i$, where $\mathfrak{p}$ is a type 3 prime ideal of degree 1, with $\mathfrak{p}^i \| \mathfrak{a}$, and
- $\mathfrak{p}^i$, where neither $\mathfrak{p}'$ nor $\mathfrak{p}''$ divide $\mathfrak{a}$.

The uniqueness now follows from the unique prime ideal factorization of $\mathfrak{a}$. □

The following lemma is the most general partial factorization result. As such, this generalizes Lemma 4.2.11 to include type 2 prime ideal factors, thus generalizing Equation 4.2 of [Bau04] further.

Lemma 4.2.12 *If $\mathfrak{a} = [s,\, s'(u+\rho),\, s''(v+w\rho+\omega)]$ is a primitive ideal given in terms of its minimal canonical basis, then $\mathfrak{a}$ factors as*

$$\begin{aligned}\mathfrak{a} &= [s_H,\, \rho,\, s''\omega]\left[\frac{s}{s_H},\, s'(u+\rho),\, v+w\rho+\omega\right] \\ &= [s_H,\, \rho,\, s''\omega]\left[\frac{s}{s's_H},\, u+\rho,\, v+w\rho+\omega\right][s',\, s'\rho,\, v+w\rho+\omega] \ ,\end{aligned}$$

where $s_H = \gcd(s, H)$, $[s_H, \rho, s''\omega]$ *is the product of type 2 prime ideals, and the other factors of* $\mathfrak{a}$ *are products of prime ideals of type 1, 3, and 4. Furthermore, this factorization is unique in the following sense: if* $\mathfrak{a} = \mathfrak{b}\mathfrak{c}\mathfrak{d}$, *where* $\mathfrak{b} = [S_H, \rho, S''\omega]$, $\mathfrak{c} = [S/S'S_H, U_1 + \rho, V_1 + W_1\rho + \omega]$, *and* $\mathfrak{c} = [S', S'\rho, V_2 + W_2\rho + \omega]$, *for some* $S, S', S'', U_1, V_1, V_2, W_1, W_2 \in \mathbb{F}_q[x]$, *with* $S_H = \gcd(S, H)$, *then* $\mathfrak{b} = [s_H, \rho, s''\omega]$, $\mathfrak{c} = [s/s's_H, u + \rho, v + w\rho + \omega]$, *and* $\mathfrak{d} = [s', s'\rho, v + w\rho + \omega]$. *In addition, we have* $\gcd(s/s'', s'') = 1$.

Proof: Write $\mathfrak{a} = \mathfrak{b}\mathfrak{c}$, where $\mathfrak{b} = [s_1, \rho, s''\omega]$ is the product of type 2 prime ideals and $\mathfrak{c} = [s_2, s'(u_2 + \rho), v_2 + w_2\rho + \omega]$ is the product of prime ideals of types 1, 3, and 4. Since $\gcd(s_1, s_2) = 1$, we have $s_1s_2 = s$, and since $s_1 \mid H$, we have $s_1 = s_H$ and $s_2 = s/s_H$, by Theorem 4.2.2. By Part 1 of Lemma 4.1.6, $\mathfrak{c} = [s/s_H, s'(u+\rho), v+w\rho+\omega]$ if and only if $u \equiv u_2 \pmod{s/(s's_H)}$, $w \equiv w_2 \pmod{s'}$, and $v \equiv v_2 + u_2(w - w_2) \pmod{s/s_H}$. Multiplying $\mathfrak{a} = \mathfrak{b}\mathfrak{c}$ using Theorem 4.2.2, we have these precise congruences to determine u, w, and v. Thus, $\mathfrak{a} = [s_H, \rho, s''\omega][s/s_H, s'(u+\rho), v+w\rho+\omega]$. Factoring $\mathfrak{c}$ via Lemma 4.2.11 yields the desired factorization.

To show the uniqueness result, note that the factor $[s_H, \rho, s''\omega]$ consists exactly of all the type 2 prime ideals dividing $\mathfrak{a}$ and their squares; the other two factors contain no type 2 prime ideals. So the uniqueness claim follows from Part 2 of Lemma 4.2.11.

Lastly, by Lemma 4.2.6, we have $\gcd(s_H/s'', s'') = 1$. Since $\gcd(s/s_H, H) = 1$ by construction, we have $\gcd(s/s'', s'') = 1$. □

4.2.5 Multiplication of Ideals of a Special Form

We conclude this section with two lemmas which contain a series of results concerning the product of ideals in special forms. The first deals exclusively with products of type 2 prime ideals and the second considers two forms of products of the prime ideals of types 1, 3, and 4. As noted earlier, these results will be used in conjunction with the factorizations we just derived to extend inversion and multiplication to general ideals.

Lemma 4.2.13 (Products of Type 2 Prime Ideals) *For* $i = 1, 2$, *let* $\mathfrak{a}_i = [s_i, \rho, s_i''\omega]$ *be primitive ideals which are the product of type 2 prime ideals. Then*

$$\mathfrak{a} = \mathfrak{a}_1\mathfrak{a}_2 = \langle d \rangle \left[\frac{s_1}{d}, \rho, \frac{s_1''}{d_2 d''}\omega\right] \left[\frac{s_2}{d}, \rho, \frac{s_2''}{d_1 d''}\omega\right] [d'', \rho, \omega] \ ,$$

where $d_1 = \gcd(s_1/s_1'', s_2'')$, $d_2 = \gcd(s_2/s_2'', s_1'')$, $d'' = \gcd(s_1'', s_2'')$, *and* $d = d_1d_2d''$. *If* $\mathfrak{a}$ *is primitive and* $d_H = \gcd(s_1, s_2)$, *then*

$$\mathfrak{a} = \left[\frac{s_1s_2}{d_H}, \rho, s_1''s_2''d_H\omega\right] = [s_1, \rho, \omega]\,[s_2, \rho, \omega]\,[s_1''s_2'', \rho, \omega] \ .$$

Proof: By Lemma 4.2.6, we factor $\mathfrak{a}_i = [s_i, \rho, s_i''\omega] = \mathfrak{b}_i\mathfrak{c}_i$, where $\mathfrak{b}_i = [s_i/s_i'', \rho, \omega]$ is square-free and $\mathfrak{c}_i = [s_i'', \rho, s_i''\omega] = [s_i'', \rho, \omega]^2$, for $i = 1, 2$. (Since each $\mathfrak{a}_i$ is primitive, no cube of any type 2 prime ideal can divide $\mathfrak{a}_i$.) Also by Lemma 4.2.6, each $s_i \mid H$, so s_i is square-free and $\gcd(s_i/s_i'', s_i'') = 1$ for $i = 1, 2$.

The non-primitive component of $\mathfrak{a}$ is the product of factors of $\mathfrak{a}_1\mathfrak{a}_2$ that are of the form $\mathfrak{p}^3$, for some type 2 prime ideal $\mathfrak{p}$. Since $\mathfrak{b}_1$ and $\mathfrak{b}_2$ are square-free and $\mathfrak{c}_1$ and $\mathfrak{c}_2$ are the products of squares,

$\mathfrak{b}_1\mathfrak{b}_2$ is primitive, but $\mathfrak{b}_1\mathfrak{c}_2$, $\mathfrak{b}_2\mathfrak{c}_1$, and $\mathfrak{c}_1\mathfrak{c}_2$ may have non-primitive factors. We consider the last three pairs to determine the non-primitive component of $\mathfrak{a}$ and the desired primitive factors that remain. First we consider $\mathfrak{b}_1\mathfrak{c}_2$ and let $d_1 = \gcd(s_1/s_1'', s_2'')$. If $P \in \mathbb{F}_q[x]$ is irreducible, $P \mid d_1$, and $\mathfrak{p} \mid \langle P \rangle$, then $\mathfrak{p}^3 \| \mathfrak{b}_1\mathfrak{c}_2$. Since $\gcd(s_i/s_i'', s_i'') = 1$, for $i = 1, 2$, we have $\mathfrak{p}^3 \| \mathfrak{a}$. It follows that the non-primitive component of $\mathfrak{b}_1\mathfrak{c}_2$ is $\langle d_1 \rangle$, $[d_1, \rho, \omega] \mid \mathfrak{b}_1$, and $[d_1, \rho, d_1\omega] \mid \mathfrak{c}_2$. By a similar argument, $\langle d_2 \rangle$ is the non-primitive component of $\mathfrak{b}_2\mathfrak{c}_1$, where $d_2 = \gcd(s_2/s_2'', s_1'')$, $[d_2, \rho, \omega] \mid \mathfrak{b}_2$, and $[d_2, \rho, d_2\omega] \mid \mathfrak{c}_1$. Since $\gcd(s_i/s_i'', s_i'') = 1$, for $i = 1, 2$, we have $\gcd(d_1, d_2) = 1$. Lastly, we consider $\mathfrak{c}_1\mathfrak{c}_2$ and let $d'' = \gcd(s_1'', s_2'')$. If $P \in \mathbb{F}_q[x]$ is irreducible, $P \mid d''$, and $\mathfrak{p} \mid \langle P \rangle$, then $\mathfrak{p}^4 \| \mathfrak{c}_1\mathfrak{c}_2$ and $\gcd(\mathfrak{c}_1, \mathfrak{c}_2) = [d'', \rho, \omega]^2$, by Lemma 4.2.6. Again, $\gcd(s_i/s_i'', s_i'') = 1$, for $i = 1, 2$, implies that $\mathfrak{p}^4 \| \mathfrak{a}$, $\gcd(d_1, d'') = 1$, and $\gcd(d_2, d'') = 1$. It follows that the non-primitive component of $\mathfrak{c}_1\mathfrak{c}_2$ is $\langle d'' \rangle$ and $[d'', \rho, \omega]$ is a factor of the primitive component of $\mathfrak{a}$. Multiplying the non-primitive components together, we have $\langle d \rangle = \langle d_1 d_2 d'' \rangle$.

Now factoring the non-primitive components out of $\mathfrak{b}_1$, $\mathfrak{c}_1$, $\mathfrak{b}_2$, and $\mathfrak{c}_2$, we have

$$\mathfrak{a} = \langle d \rangle \left[\frac{s_1}{s_1'' d_1}, \rho, \omega\right] \left[\frac{s_1''}{d_2 d''}, \rho, \frac{s_1''}{d_2 d''}\omega\right] \left[\frac{s_2}{s_2'' d_2}, \rho, \omega\right] \left[\frac{s_2''}{d_1 d''}, \rho, \frac{s_2''}{d_1 d''}\omega\right] [d'', \rho, \omega] \ .$$

Multiplying the first two primitive factors above and multiplying the third and fourth primitive factors above, both via Lemma 4.2.6, yield the desired result.

For the second part, we obtain the desired factorization by first writing $\mathfrak{a}_i = [s_i, \rho, \omega][s_i'', \rho, \omega]$, for $i = 1, 2$. Since $\mathfrak{a} = \mathfrak{a}_1\mathfrak{a}_2$ is primitive, we have $d'' = \gcd(s_1'', s_2'') = 1$, so $[s_1'', \rho, \omega][s_2'', \rho, \omega] = [s_1''s_2'', \rho, \omega]$.

To obtain the total product, we note that if $\mathfrak{a} = \mathfrak{a}_1\mathfrak{a}_2$ is primitive, then $d = d_1 = d_2 = d'' = 1$, so $\gcd(\mathfrak{b}_1, \mathfrak{c}_2) = \gcd(\mathfrak{b}_2, \mathfrak{c}_1) = \gcd(\mathfrak{c}_1, \mathfrak{c}_2) = \langle 1 \rangle$. Considering $\gcd(\mathfrak{b}_1, \mathfrak{b}_2)$, if $\mathfrak{p} \mid \gcd(\mathfrak{b}_1, \mathfrak{b}_2)$, then $\mathfrak{p}^2 \| \mathfrak{b}_1\mathfrak{b}_2$, and $\mathfrak{p}^2 \| \mathfrak{a}$, since $\mathfrak{a}$ is primitive. If $d_H'' = \gcd(N(\mathfrak{b}_1), N(\mathfrak{b}_2)) = \gcd(s_1/s_1'', s_2/s_2'')$, then by Lemma 4.2.6, we construct $\gcd(\mathfrak{b}_1, \mathfrak{b}_2) = [d_H'', \rho, \omega]$. Now by Lemma 4.2.6, s_1 and s_2 are square-free. If $d_H = \gcd(s_1, s_2)$, then obviously $d_H'' \mid d_H$. Let P be a prime divisor of d_H, so that $P \mid s_1, s_2$. Since $\gcd(s_1'', s_2'') = 1$, P does not divide at least one of s_1'', s_2''. Without loss of generality, assume $P \nmid s_1''$. Then $P \mid (s_1/s_1'')$. Since $\gcd(s_1/s_1'', s_2'') = 1$, $P \nmid s_2''$, so $P \mid (s_2/s_2'')$. It follows that $P \mid d_H''$, so that $d_H'' = d_H$ and $\gcd(\mathfrak{b}_1, \mathfrak{b}_2) = [d_H, \rho, \omega]$.

Therefore, $[d_H, \rho, \omega]^2 = [d_H, \rho, d_H\omega]$ is a factor of $\mathfrak{a}$. Factoring $[d_H, \rho, \omega]$ out of $\mathfrak{b}_1$ and $\mathfrak{b}_2$ yields

$$\mathfrak{a} = \left[\frac{s_1}{s_1'' d_H}, \rho, \omega\right] \left[\frac{s_2}{s_2'' d_H}, \rho, \omega\right] [s_1'', \rho, s_1''\omega][s_2'', \rho, s_2''\omega][d_H, \rho, d_H\omega] \ .$$

Since $d'' = \gcd(s_1'', s_2'') = \gcd(s_i'', d_H) = 1$, for each $i = 1, 2$, we have

$$[s_1'', \rho, s_1''\omega][s_2'', \rho, s_2''\omega][d_H, \rho, d_H\omega] = [s_1''s_2''d_H, \rho, s_1''s_2''d_H\omega] \ .$$

Further, since $\gcd(s_1/s_1''d_H, s_2/s_2''d_H) = 1$, we also have

$$\mathfrak{a} = \left[\frac{s_1 s_2}{s_1'' s_2'' d_H^2}, \rho, \omega\right] [s_1''s_2''d_H, \rho, s_1''s_2''d_H\omega] \ .$$

Multiplying these two ideals via Lemma 4.2.6 yields the total product. □

Lemma 4.2.14 (Products of Non-Type 2 Prime Ideals) *Suppose $\mathfrak{a}_1$ and $\mathfrak{a}_2$ are two primitive ideals that have no type 2 prime ideal factors.*

1. *If $\mathfrak{a}_i = [s_i,\, s_i\rho,\, v_i + w_i\rho + \omega]$, for $i = 1,\, 2$, and $\mathfrak{a}_1\mathfrak{a}_2$ is primitive, then $\mathfrak{a}_1\mathfrak{a}_2 = [s_1s_2,\, s_1s_2\rho,\, V + W\rho + \omega]$, for some polynomials $V,\, W \in \mathbb{F}_q[x]$.*

2. *If $\mathfrak{a}_i = [s_i,\, u_i + \rho, v_i + w_i\rho + \omega]$, for $i = 1,\, 2$, is given in terms of a minimal canonical basis, then $\mathfrak{a}_1\mathfrak{a}_2$ factors as $\mathfrak{a}_1\mathfrak{a}_2 = \mathfrak{a}_1'\mathfrak{a}_2'\mathfrak{a}_3'$, where*

$$\mathfrak{a}_i' = \left[\frac{s_i d_1}{d},\, u_i + \rho,\, v_i + w_i\rho + \omega\right] \quad , \quad \text{for} \quad i = 1,\, 2 \quad , \quad \text{and} \quad \mathfrak{a}_3' = \left[\frac{d}{d_1},\, \frac{d}{d_1}\rho,\, V + W\rho + \omega\right] \quad ,$$

   for some polynomials $V,\, W \in \mathbb{F}_q[x]$, where $d = \gcd(s_1,\, s_2)$ and $d_1 = \gcd(d,\, u_1 - u_2)/\gcd(d,\, G)$.

3. *With $\mathfrak{a}_1'$ and $\mathfrak{a}_2'$ as given in Part 2, there are polynomials $U,\, V,\, W \in \mathbb{F}_q[x]$ such that*

$$\mathfrak{a}_1'\mathfrak{a}_2' = \left[\frac{s_1 s_2 d_1^2}{d^2},\, U + \rho,\, V + W\rho + \omega\right] \quad .$$

Proof: For the first part of the lemma, let $\mathfrak{a}_i = [s_i,\, s_i\rho,\, v_i + w_i\rho + \omega]$, for $i = 1,\, 2$, such that $\mathfrak{a}_1\mathfrak{a}_2 = \mathfrak{a}$ is primitive. Since no type 2 prime ideals divide $\mathfrak{a}_1$ and $\mathfrak{a}_2$, there are polynomials $S,\, S',\, U,\, V,\, W \in \mathbb{F}_q[x]$ such that $\mathfrak{a} = [S,\, S'(U + \rho),\, V + W\rho + \omega]$, by Lemma 4.2.11. Since $\mathfrak{a} \subseteq \mathfrak{a}_i$, for $i = 1,\, 2$, we have $s_i \mid S' \mid S$, for $i = 1,\, 2$, so $s_1^2 s_2^2 \mid SS'$. However, we have $N(\mathfrak{a}_1\mathfrak{a}_2) = s_1^2 s_2^2 = SS' = N(\mathfrak{a})$, so $S = S' = s_1 s_2$. By Lemma 4.2.4, we have $U = 0$ and $\mathfrak{a} = [S,\, S\rho,\, V + W\rho + \omega] = [s_1 s_2,\, s_1 s_2\rho,\, V + W\rho + \omega]$.

For the second part of the lemma, let $\mathfrak{a}_i = [s_i,\, u_i + \rho, v_i + w_i\rho + \omega]$ and $D_i = \gcd(s_i, G)$, for $i = 1,\, 2$. By Corollary 4.2.9, we write $\mathfrak{a}_i = \mathfrak{b}_i\mathfrak{c}_i$, where $\mathfrak{b}_i = [D_i,\, \rho,\, \omega]$ is the product of type 1 prime ideals and $\mathfrak{c}_i = [s_i/D_i,\, u_i + \rho,\, v_i + w_i\rho + \omega]$ is the product of type 3 and 4 prime ideals, for $i = 1,\, 2$. For the proof, we will determine factors of $\mathfrak{b}_1\mathfrak{b}_2$ and $\mathfrak{c}_1\mathfrak{c}_2$ of the form $[S,\, S\rho,\, V + W\rho + \omega]$ and will divide the appropriate ideals that make up that factor out of $\mathfrak{b}_1$, $\mathfrak{b}_2$, $\mathfrak{c}_1$, and $\mathfrak{c}_2$. For $\mathfrak{b}_1$ and $\mathfrak{b}_2$, this will simply involve determining $\gcd(\mathfrak{b}_1,\, b_2)$ and factoring it out of $\mathfrak{b}_1\mathfrak{b}_2$. For $\mathfrak{c}_1$ and $\mathfrak{c}_2$, we will show that any factor of $\mathfrak{c}_1\mathfrak{c}_2$ of the form $[S,\, S\rho,\, V + W\rho + \omega]$ must come completely from products of the form $\mathfrak{p}\mathfrak{p}'$, where $\mathfrak{p}$ is a type 4 prime ideal. We will determine this by considering the ideal factors $\mathfrak{d}_i = [D,\, u_i\rho,\, v_i + w_i\rho + \omega]$ of $\mathfrak{c}_i$, for $i = 1,\, 2$, where $D = \gcd(s_1/D_1,\, s_2/D_2)$, and extracting the factor $[S,\, S\rho,\, V + W\rho + \omega]$ from $\mathfrak{d}_1\mathfrak{d}_2$. Multiplying the type 2 and non-type 2 factors back together will yield the final result.

First, let $\mathfrak{d}_G = \gcd(\mathfrak{b}_1,\, \mathfrak{b}_2)$. Using Theorem 4.2.1, it is easy to infer that $\mathfrak{d}_G = [D_G,\, \rho,\, \omega]$, where $D_G = \gcd(d,\, G) = \gcd(D_1,\, D_2)$ and $d = \gcd(s_1,\, s_2)$. Furthermore, we have $\mathfrak{d}_G^2 = [D_G,\, D_G\rho,\, \omega]$ by Lemma 4.2.5, so $\mathfrak{b}_1\mathfrak{b}_2 = [D_1/D_G,\, \rho,\, \omega][D_2/D_G,\, \rho,\, \omega][D_G,\, D_G\rho,\, \omega]$.

Now let $\mathfrak{d} = \gcd(\mathfrak{c}_1,\, \mathfrak{c}_2)$. If $\mathfrak{p}^n \| \mathfrak{d}$, for some $n \in \mathbb{N}$, then $\mathfrak{p}$ is a type 3 or 4 prime ideal, so by Theorem 4.2.1, $\mathfrak{p}^n$ is of the form $[P^n,\, u_P + \rho,\, v_P + w_P\rho + \omega]$ for some monic irreducible $P \in \mathbb{F}_q[x]$ and $u_P,\, v_P,\, w_P \in \mathbb{F}_q[x]$. From the containments $\mathfrak{a}_i \subseteq \mathfrak{p}^n$, for $i = 1,\, 2$, we have $u_P \equiv u_i \pmod{P^n}$, for $i = 1,\, 2$, by Lemma 4.1.5, so $P^n \mid (u_1 - u_2)$. Let $d_1 = \gcd(d, u_1 - u_2)/D_G$. By Lemma 4.2.3, we have $\mathfrak{d} = [d_1,\, u_i + \rho,\, v_i + w_i\rho + \omega]$, since $\mathfrak{d} \mid \mathfrak{c}_i$, for $i = 1,\, 2$. Since $\mathfrak{d}^2$ is the product of prime powers $\mathfrak{p}^{2n}$, we see that $\mathfrak{d}^2$ is of the form $[d_1^2,\, U' + \rho,\, V' + W'\rho + \omega]$, for some $U',\, V',\, W' \in \mathbb{F}_q[x]$, by Theorems 4.2.1 and 4.2.2.

Next, let $D = \gcd(s_1/D_1,\, s_2/D_2)$ and $\mathfrak{d}_i = [D,\, u_i + \rho,\, v_i + w_i\rho + \omega]$, for $i = 1,\, 2$. By Lemma 4.2.3, we have $\mathfrak{d} \mid \mathfrak{d}_i \mid \mathfrak{c}_i$, for $i = 1,\, 2$, and by construction, we have $\gcd(N(\mathfrak{c}_1/\mathfrak{d}_1), N(\mathfrak{c}_2/\mathfrak{d}_2)) = 1$. If $\mathfrak{p}$ is a

prime ideal such that $\mathfrak{p}^n \| (\mathfrak{d}_1/\mathfrak{d})$, then $\mathfrak{p} \nmid (\mathfrak{d}_2/\mathfrak{d})$ and $\mathfrak{p} | \langle P \rangle$, for some $P \mid (D/d_1)$. Thus, $(\mathfrak{p}')^n \| (\mathfrak{d}_2/\mathfrak{d})$. We must have $\mathfrak{p} \neq \mathfrak{p}'$ since $\mathfrak{p} \nmid \mathfrak{d}$. Further, since no type 3 prime ideals of degree 2 divide $\mathfrak{c}_1$ or $\mathfrak{c}_2$, by Corollary 4.2.9, $\mathfrak{p}$ and $\mathfrak{p}'$ are type 4 prime ideals over $\langle P \rangle$ and $(\mathfrak{p}\mathfrak{p}')^n \| (\mathfrak{d}_1\mathfrak{d}_2/\mathfrak{d}^2)$. In addition, by the construction of $\mathfrak{d}_1$ and $\mathfrak{d}_2$, we must have $(\mathfrak{p}\mathfrak{p}')^n \| \mathfrak{c}_1\mathfrak{c}_2$. By Theorem 4.2.1, $(\mathfrak{p}\mathfrak{p}')^n$ is of the form $[P^n, P^n\rho, V_P + W_P\rho + \omega]$, for some V_P, $W_P \in \mathbb{F}_q[x]$, so multiplying $(\mathfrak{d}_1/\mathfrak{d})(\mathfrak{d}_2/\mathfrak{d})$ by Theorem 4.2.2, we have $\mathfrak{d}_1\mathfrak{d}_2/\mathfrak{d}^2 = [D/d_1, (D/d_1)\rho, \tilde{V} + \tilde{W}\rho + \omega]$, for some polynomials $\tilde{V}$, $\tilde{W} \in \mathbb{F}_q[x]$.

Finally, for each $i = 1, 2$, we have $\mathfrak{d}_i/\mathfrak{d} = [D/d_1, u_i + \rho, v_i + w_i\rho + \omega]$, by Lemma 4.2.3, and we factor this out of $\mathfrak{c}_i$. With the given forms also given by Lemma 4.2.3, we multiply the appropriate factors together. Since $DD_G = d$, Corollary 4.2.9 states that

$$\mathfrak{a}_1' = \left(\frac{\mathfrak{b}_1}{\mathfrak{d}_G}\right)\left(\frac{\mathfrak{c}_1\mathfrak{d}}{\mathfrak{d}_1}\right) = \left[\frac{D_1}{D_G}, \rho, \omega\right]\left[\frac{s_1d_1}{DD_1}, u_1 + \rho, v_1 + w_1\rho + \omega\right] = \left[\frac{s_1d_1}{d}, u_1 + \rho, v_1 + w_1\rho + \omega\right]$$

and

$$\mathfrak{a}_2' = \left(\frac{\mathfrak{b}_2}{\mathfrak{d}_G}\right)\left(\frac{\mathfrak{c}_2\mathfrak{d}}{\mathfrak{d}_2}\right) = \left[\frac{D_2}{D_G}, \rho, \omega\right]\left[\frac{s_2d_1}{DD_2}, u_2 + \rho, v_2 + w_2\rho + \omega\right] = \left[\frac{s_2d_1}{d}, u_2 + \rho, v_2 + w_2\rho + \omega\right] \ .$$

By Part 1 of this lemma, there are polynomials V, $W \in \mathbb{F}_q[x]$ such that

$$\mathfrak{a}_3' = \mathfrak{d}_G^2\left(\frac{\mathfrak{d}_1\mathfrak{d}_2}{\mathfrak{d}^2}\right) = [D_G, D_G\rho, \omega]\left[\frac{D}{d_1}, \frac{D}{d_1}\rho, \tilde{V} + \tilde{W}\rho + \omega\right] = \left[\frac{d}{d_1}, \frac{d}{d_1}\rho, V + W\rho + \omega\right] \ .$$

Since $\mathfrak{a}_1\mathfrak{a}_2 = \mathfrak{b}_1\mathfrak{c}_1\mathfrak{b}_2\mathfrak{c}_2 = \mathfrak{a}_1'\mathfrak{a}_2'\mathfrak{a}_3'$, the desired result follows from multiplying the three products above.

For the third part of the lemma, the proof of Part 2 implies that if $\mathfrak{p}$ is a type 1 prime ideal such that $\mathfrak{p} \mid \mathfrak{a}_1'$, then $\mathfrak{p} \| \mathfrak{a}_1'\mathfrak{a}_2'$. Also, if $\mathfrak{p}$ is a type 3 prime ideal dividing $\mathfrak{a}_1'\mathfrak{a}_2'$, then it must be of degree 1. Lastly, if $\mathfrak{p}$ is a type 4 prime ideal dividing $\mathfrak{a}_1'$, then neither $\mathfrak{p}'$ nor $\mathfrak{p}''$ divide $\mathfrak{a}_2'$, and conversely $\mathfrak{p} \mid \mathfrak{a}_2'$ implies that $\mathfrak{p}'$, $\mathfrak{p}'' \nmid \mathfrak{a}_1'$, since factors of the form $\mathfrak{p}\mathfrak{p}'$ were collected into $\mathfrak{a}_3'$, the third factor of $\mathfrak{a}_1\mathfrak{a}_2$. Thus, $\mathfrak{a}_1'\mathfrak{a}_2'$ is of the form $[S, U + \rho, V + W\rho + \omega]$, for some polynomials S, U, V, $W \in \mathbb{F}_q[x]$. Since $N(\mathfrak{a}_1'\mathfrak{a}_2') = N(\mathfrak{a}_1')N(\mathfrak{a}_2') = S$, we have $S = s_1s_2d_1^2/d^2$. □

In this section, we identified the form of prime ideals and used that result to show how certain ideals partially factor into convenient factors. This was used, in turn, to describe products of type 2 prime ideals and products of special combinations of type 1, 3, and 4 prime ideals. We also showed how a general primitive ideal partially factors into ideals of a special form. Given these partial factorizations and descriptions of these ideals, we now describe how to find the inverse of an arbitrary primitive ideal.

4.3 Ideal Inversion

In this section, we will apply our results on the structure of prime ideals and partial factorization to describe the inverse of an arbitrary primitive ideal of $\mathcal{O}$. Specifically, for an ideal $\mathfrak{a}$, we will determine a minimal canonical basis of the primitive ideal, $\overline{\mathfrak{a}}$, such that $\mathfrak{a}\overline{\mathfrak{a}} = \langle L(\mathfrak{a}) \rangle$, as given in Lemma 2.5.7. In this way, the classes $[\overline{\mathfrak{a}}]$ and $[\mathfrak{a}]$ are inverses in $Cl(\mathcal{O})$. (In the canonical basis of $\mathfrak{a}$, $s = L(\mathfrak{a})$, so $\mathfrak{a}\overline{\mathfrak{a}} = \langle s \rangle$.) We will use the inverse of an ideal for three important applications. First,

they are used to derive the product of two arbitrary ideals in Section 4.4.1. In Section 4.5.3, we will use inverses to find the unique distinguished ideal in a given ideal class for the case in which K_x is totally ramified at infinity. Lastly, they will be used in Section 6.2 to speed up the computation of class numbers and regulators of purely cubic function fields of unit ranks 0 and 1. In the last case, we will first need to define the notion of the inverse of an infrastructure divisor; this will be done in Section 5.3.4.

We present three results in this section. The first finds the inverse of the product of type 2 prime ideals, the second finds the inverse of ideals not divisible by any type 2 prime ideal, and the third combines these two results to derive the inverse of any primitive ideal. This section generalizes Lemma 6.1 of [Bau04], and as such, many of the techniques are similar to those in [Bau04]. The main result of this section is found in Lemma 0.33 of [Bau05], though it is left unproved in that source.

Lemma 4.3.1 *If $\mathfrak{a} = [s,\, \rho,\, s''\omega]$ is a primitive ideal which is the product of type 2 prime ideals, then $\overline{\mathfrak{a}} = \langle s \rangle\, \mathfrak{a}^{-1} = [s,\, \rho,\, (s/s'')\omega]$.*

Proof: By Lemma 4.2.6, we have $\mathfrak{a} = [s/s'',\, \rho,\, \omega][s'',\, \rho,\, \omega]^2$, so $\mathfrak{a}[s/s'',\, \rho,\, \omega]^2[s'',\, \rho,\, \omega] = [s,\, \rho,\, \omega]^3$. For each irreducible polynomial $P \mid s$ and each type 2 prime ideal $\mathfrak{p} = [P,\, \rho,\, \omega]$ lying above $\langle P \rangle$, we have $\mathfrak{p}^3 = \langle P \rangle$, so $[s,\, \rho,\, \omega]^3 = \langle s \rangle$. Thus, $\overline{\mathfrak{a}} = \langle s \rangle\, \mathfrak{a}^{-1} = [s/s'',\, \rho,\, \omega]^2[s'',\, \rho,\, \omega] = [s,\, \rho,\, (s/s'')\omega]$, by Lemma 4.2.6. □

The following lemma generalizes Lemma 6.1 of [Bau04] from the case that the curve defining the function field K is nonsingular (i.e. H is taken to be 1) to the case that the curve is possibly singular. As such, similar techniques are used in the proof.

Lemma 4.3.2 *If $\mathfrak{a} = [s,\, s'(u+\rho),\, v + w\rho + \omega]$ is a primitive ideal, given in terms of its minimal canonical basis, and is the product of prime ideals of types 1, 3, and 4, then*

$$\overline{\mathfrak{a}} = \langle s \rangle\, \mathfrak{a}^{-1} = \left[s,\, \frac{s}{s'}(U+\rho),\, V + W\rho + \omega \right],$$

where

$$U \equiv -wH \pmod{s'} \;, \quad W \equiv -ur \pmod{s/s'} \;, \quad V \equiv -wWH - v \pmod{s} \;,$$

$rH \equiv 1 \pmod{s/s'}$, and the canonical basis is minimal.

Proof: By Lemma 2.5.7, $\overline{\mathfrak{a}} = \langle s \rangle\, \mathfrak{a}^{-1}$ is primitive. Since no type 2 prime ideal divides $\overline{\mathfrak{a}}$, it may be written in the form $\overline{\mathfrak{a}} = [S,\, S'(U+\rho),\, V + W\rho + \omega]$. We will first determine S and S', then U, W, and finally, V. Since $s \in \overline{\mathfrak{a}}$, we have $S \mid s$. Also, $S(v + w\rho + \omega) \in \langle s \rangle$, so $s \mid S$, and thus, $S = s$. Taking norms, we have $s^3 = N(\langle s \rangle) = N(\mathfrak{a}\overline{\mathfrak{a}}) = s^2 s' S'$, so $S' = s/s'$.

To obtain U, W, and V, we again consider $\mathfrak{a}\overline{\mathfrak{a}} = \langle s \rangle$. Since $S'(U+\rho)(v+w\rho+\omega) = S'(Uv+GH) + (v+Uw)\rho + (U+wH)\omega) \in \langle s \rangle$, in particular, we have $s \mid S'(U+wH)$, so $U \equiv -wH \pmod{s/S'}$. Since $s/S' = S/S' = s'$, this determines U uniquely.

To determine W, we consider the product

$$s'(u+\rho)(V + W\rho + \omega) = s'\left((uV + GH) + (uW + V)\rho + (u + WH)\omega\right) \in \langle s \rangle \quad .$$

In particular, we have $s \mid s'(u+WH)$, so $(s/s') \mid (u+WH)$. Since $S' = s/s'$ and $\gcd(S', H) = 1$, by (4.1), we have $W \equiv -ur \pmod{S'}$, where $rH \equiv 1 \pmod{S'}$.

Lastly, to determine V, we consider the product

$$(V+W\rho+\omega)(v+w\rho+\omega) = (vV+(w+W)GH)+(wV+vW+G)\rho+(V+v+wWH)\omega \in \langle s\rangle = \langle S\rangle \ .$$

In particular, we have $S \mid (V+v+wWH)$, so that $V \equiv -wWH - v \pmod{S}$. By construction, $\{S,\ S'(U+\rho),\ S''(V+W\rho+\omega)\}$ is a minimal canonical basis of $\overline{\mathfrak{a}}$. □

Combining Lemmas 4.3.1 and 4.3.2, we derive the inverse of a general primitive ideal. This result is stated without proof in the given source, but we provide the proof here.

Lemma 4.3.3 (Lemma 0.33 of [Bau05]) *If $\mathfrak{a} = [s,\ s'(u+\rho),\ s''(v+w\rho+\omega)]$ is a primitive ideal, given in terms of its minimal canonical basis, then $\overline{\mathfrak{a}} = \langle s\rangle\, \mathfrak{a}^{-1} = [S,\ S'(U+\rho),\ S''(V+W\rho+\omega)]$, where*

$$S = s \ , \quad S' = \frac{s}{s's_H} \ , \quad S'' = \frac{s_H}{s''} \ , \quad U \equiv -wH \pmod{\tfrac{S}{S'}} \ ,$$
$$W \equiv -ur_1 \pmod{S'} \ , \quad V \equiv -wWH - vs''r_2 \pmod{\tfrac{S}{S''}} \ ,$$

$r_1H \equiv 1 \pmod{S'}$, $r_2s'' \equiv 1 \pmod{s/s_H}$, and $s_H = \gcd(s, H)$. Furthermore, the given canonical basis for $\overline{\mathfrak{a}}$ is minimal.

Proof: For the proof, we will factor $\mathfrak{a}$ into two ideals: one which is a product of type 2 prime ideals, and the other which is free of type 2 prime ideal factors. We will then apply the previous two results to these factors and multiply the primitive parts together. First, we have $\mathfrak{a}\overline{\mathfrak{a}} = \langle L(\mathfrak{a})\rangle = \langle s\rangle$, by Lemma 2.5.7. Next, by Lemma 4.2.12, we may factor $\mathfrak{a}$ into $\mathfrak{a}_1\mathfrak{a}_2$, where $\mathfrak{a}_1 = [s_H,\ \rho,\ s''\omega]$ is the product of the type 2 prime ideal factors of $\mathfrak{a}$ and $\mathfrak{a}_2 = [s/s_H,\ s'(u+\rho),\ v+w\rho+\omega]$.

Likewise, we write $\overline{\mathfrak{a}} = \mathfrak{b}_1\mathfrak{b}_2$, where $\mathfrak{b}_1 = \overline{\mathfrak{a}}_1$ is the product of type 2 prime ideals and $\mathfrak{b}_2 = \overline{\mathfrak{a}}_2$ is free of type 2 prime ideal factors. Thus, we have $\gcd(L(\mathfrak{b}_1),\ L(\mathfrak{b}_2)) = 1$. By Lemmas 4.3.1 and 4.3.2, we have $\mathfrak{b}_1 = [s_H,\ \rho,\ (s_H/s'')\omega]$ and $\mathfrak{b}_2 = [s/s_H,\ (s/s's_H)(u_2+\rho),\ v_2+w_2\rho+\omega]$, where $u_2 \equiv -wH \pmod{s'}$, $w_2 \equiv -ur_1 \pmod{s/(s's_H)}$, $v_2 \equiv -ww_2H - v \pmod{s/s_H}$, and $r_1H \equiv 1 \pmod{s/(s's_H)}$. (We know $r_1 \in \mathbb{F}_q[x]$ exists because $\gcd(s/(s's_H),\ H) = 1$ by (4.1).)

Since $\gcd(s/s_H,\ s_H) = 1$, by (4.1), we apply Theorem 4.2.2 to recombine the factors:

$$\begin{aligned}\overline{\mathfrak{a}} = \mathfrak{b}_1\mathfrak{b}_2 &= \left[s_H,\ \rho,\ \frac{s_H}{s''}\omega\right]\left[\frac{s}{s_H},\ \frac{s}{s's_H}(u_2+\rho),\ v_2+w_2\rho+\omega\right] \\ &= \left[s,\ \frac{s}{s's_H}(U+\rho),\ \frac{s_H}{s''}(V+W\rho+\omega)\right] \ ,\end{aligned}$$

for some polynomials $U, V, W \in \mathbb{F}_q[x]$. From this, we have $S = s$, $S' = s/(s's_H)$, and $S'' = s_H/s''$. By Theorem 4.2.2, U satisfies $U \equiv 0 \pmod{s_H}$, and $U \equiv u_2 \equiv -wH \pmod{s'}$. Since $s_H \mid -wH$ and $S/S' = s's_H$, we have $U \equiv -wH \pmod{S/S'}$. Next, W satisfies $W \equiv 0 \pmod 1$ and $W \equiv w_2 \equiv -ur_1 \pmod{s/(s's_H)}$. Since $s/(s's_H) = S'$, we have $W \equiv -ur_1 \pmod{S'}$. Lastly, V satisfies $V \equiv 0 \pmod{s''}$ and $V \equiv v_2 + u_2(W-w_2) \equiv -ww_2H - v - wH(-ur_1+ur_1) \equiv -wWH - v \pmod{s/s_H}$. Since $\gcd(s/s_H,\ H) = 1$, by (4.1), we have $\gcd(s/s_H,\ s'') = 1$, so there is some polynomial $r_2 \in \mathbb{F}_q[x]$ such that $s''r_2 \equiv 1 \pmod{s/s_H}$. Then $-wWH - vs''r_2 \equiv -wWH - v \pmod{s/s_H}$ and

$-wWH - vs''r_2 \equiv 0 \pmod{s''}$. Since $S/S'' = ss''/s_H$, we have $V \equiv -wWH - vs''r_2 \pmod{S/S''}$. With the choices of U, V, and W, we see that the canonical basis is minimal. □

This lemma allows one to invert any ideal of a purely cubic function field of characteristic not equal to 3. In the next section, we will do a similar generalization of ideal multiplication.

4.4 Ideal Multiplication

In this section, we will use our results on partial factorization and multiplication of special ideals to compute the product of two ideals of $\mathcal{O}$ and will give algorithms for this ideal arithmetic. This section simultaneously generalizes Sections 4 and 5 of [Sch01], by showing how to multiply two different ideals whose norms are not coprime, and Section 7 of [Bau04], by extending the results to purely cubic function fields defined by a (possibly) singular curve. These results are also stated without proof in [Bau05]. We will also offer analogous results for multiplying fractional ideals, since the results of this section apply to the problem of computing giant steps in an infrastructure, and infrastructure divisors are frequently represented by corresponding distinguished fractional ideals. As such, the results of this section will be used to define the giant step operation in Section 5.3.3.

4.4.1 Ideal Multiplication Results

We will consider three main cases in this section. In addition, we have already stated Theorem 4.2.2, which described the product of two ideals whose norms are relatively prime. The first main result in this section finds the square of an arbitrary ideal and was first proved in [Sch01]. The last two results find the product of two ideals whose product is primitive and non-primitive, respectively. In the last case, we will determine both the primitive and non-primitive factors.

We use the following lemma to find the square of an ideal. The proof of it and the subsequent results of [Sch01] are found in the given source.

Lemma 4.4.1 (Lemma 5.1 of [Sch01]) *Let* $\{s,\ s'(u+\rho),\ v+w\rho+\omega\}$ *be a canonical basis of some primitive ideal* $\mathfrak{a}$ *such that* $\gcd(s,\ GH) = 1$. *Then there exists a polynomial* $f \in \mathbb{F}_q[x]$ *such that if* $w' = w + fs'$ *and* $v' = v + fus'$, *then* $\{s,\ s'(u+\rho),\ v' + w'\rho + \omega\}$ *is also a canonical basis of* $\mathfrak{a}$ *and* $\gcd(2v' + H(w')^2,\ s) = 1$.

The proof of Lemma 5.1 of [Sch01] is constructive, i.e. it gives an actual algorithm for finding a suitable polynomial $f \in \mathbb{F}_q[x]$. In practice, however, f is found by trial and error, and we usually have $f = 0$ or $f \in \mathbb{F}_q^*$.

Theorem 4.4.2 (Theorem 5.2 of [Sch01]) *Let* $K = \mathbb{F}_q(C)$ *be a purely cubic function field, with* $C : Y^3 = F(x) = G(x)H^2(x)$. *If* $\mathfrak{a} = [s,\ s'(u+\rho),\ s''(v+w\rho+\omega)]$ *is a primitive ideal, given in terms of its canonical basis, such that* $\gcd(2v+Hw^2,\ s/s_G s_H) = 1$, *where* $s_G = \gcd(s,\ G)$, $s_H = \gcd(s,\ H)$,

and $s'_G = \gcd(s', G)$*, then* $\mathfrak{a}^2 = \langle s'_G s'' \rangle\, [S,\, S'(U+\rho),\, S''(V + W\rho + \omega)]$*, where*

$$S = \frac{s^2}{s_G s_H}\ , \quad S' = \frac{(s')^2 s_G}{(s'_G)^3}\ , \quad S'' = \frac{s_H}{s''}\ , \quad U \equiv \begin{cases} 0 & (\mathrm{mod}\ \ s_H s'_G) \\ u - y(u^3 + F) & \left(\mathrm{mod}\ \ (ss'_G/s's_G s_H)^2\right) \end{cases},$$

$$W \equiv \begin{cases} 0 & (\mathrm{mod}\ \ s_G/s'_G) \\ w - z(Hw^3 - G) & \left(\mathrm{mod}\ \ (s'/s'_G)^2\right) \end{cases},\ \textit{and}$$

$$V \equiv \begin{cases} 0 & (\mathrm{mod}\ \ s_G s'') \\ v + U(W-w) + z\left(U(Hw^3 - G) + 2GHw - v(v + Hw^2)\right) & \left(\mathrm{mod}\ \ (s/s_G s_H)^2\right) \end{cases},$$

with $3u^2 y \equiv 1 \pmod{ss'_G/s's_G s_H}$ *and* $(2v + Hw^2)z \equiv 1 \pmod{s/s_G s_H}$.

Corollary 4.4.3 (Corollary 5.3 of [Sch01]) *Let* $\{1,\, s'(u+\rho)/s,\, s''(v+w\rho+\omega)/s\}$ *be a canonical basis of a fractional ideal* $\mathfrak{f}$*. Then* $\mathfrak{f}^2 = \left\langle d^{-1} \right\rangle \tilde{\mathfrak{f}}$*, where* $d = s_G s_H / s'_G s'' \in \mathbb{F}_q[x]$*,* $\tilde{\mathfrak{f}}$ *is a fractional ideal with canonical basis* $\{1,\, S'(U+\rho)/S,\, S''(V + W\rho + \omega)/S\}$*, and* S*,* S'*,* S''*,* U*,* V*, and* W *are as in Theorem 4.4.2.*

The second theorem in this section generalizes Theorem 4.4 of [Sch01] and Lemma 7.1 of [Bau04]. This result is also given in [Bau05], but without proof. Here, we modify the statement in the original source by using the basis, $\{1,\, \rho,\, \omega\}$, of $\mathcal{O}$, rather than $\{1,\, Y,\, Y^2\}$, and supply the proof.

Theorem 4.4.4 (Lemma 0.35 of [Bau05]) *Let* $K = \mathbb{F}_q(C)$ *be a purely cubic function field, with* $C : Y^3 = F(x) = G(x)H^2(x)$*, and, for* $i = 1,\, 2$*, let* $\mathfrak{a}_i = [s_i,\, s'_i(u_i + \rho),\, s''_i(v_i + w_i\rho + \omega)]$ *be two primitive ideals, given in terms of their minimal canonical bases. If* $\mathfrak{a} = \mathfrak{a}_1\mathfrak{a}_2$ *is primitive, then* $\mathfrak{a} = [S,\, S'(U+\rho),\, S''(V + W\rho + \omega)]$ *is given by a minimal canonical basis, where*

$$\begin{aligned}
S &= \frac{s_1 s_2 d_1}{D}\ , \quad S' = \frac{s'_1 s'_2 D}{d_1 d_H}\ , \quad S'' = s''_1 s''_2 d_H\ , \quad U = u - k\frac{S}{S' d_1}\ \left(\mathrm{mod}\ \ \tfrac{S}{S'}\right), \\
W &\equiv rW'\ (\mathrm{mod}\ \ S')\ , \quad V \equiv \frac{V'}{S''} + \frac{U}{S''}\left(S''W - W'\right)\ \left(\mathrm{mod}\ \ \tfrac{S}{S''}\right), \\
V' &= a_1 s_1 s''_2 v_2 + a_2 s'_1 s'_2 u_1 u_2 + a_3 s'_1 s''_2 (u_1 v_2 + GH) \\
&\quad + a_4 s_2 s''_1 v_1 + a_5 s'_2 s''_1 (u_2 v_1 + GH) + a_6 s''_1 s''_2 (v_1 v_2 + (w_1 + w_2)GH)\ , \ \textit{and} \\
W' &= a_1 s_1 s''_2 w_2 + a_2 s'_1 s'_2 (u_1 + u_2) + a_3 s''_2 s'_1 (u_1 w_2 + v_2) \\
&\quad + a_4 s_2 s''_1 w_1 + a_5 s'_2 s''_1 (u_2 w_1 + v_1) + a_6 s''_1 s''_2 (v_1 w_2 + v_2 w_1 + G)\ ,
\end{aligned}$$

where D*,* d_1*,* d_H*,* k*,* u*,* r*, and the* a_j*, for* $1 \le j \le 6$*, are defined as follows:*

$$\begin{aligned}
D &= \gcd\left(\frac{s_1}{s'_1},\, \frac{s_2}{s'_2}\right)\ , \quad d_1 = \frac{\gcd(D,\, u_1 - u_2)}{\gcd(D,\, GH)}\ , \qquad d_H = \gcd(D,\, H)\ , \\
r &\equiv (S'')^{-1}\ (\mathrm{mod}\ \ S')\ , \quad u \equiv \begin{cases} u_1 & \left(\mathrm{mod}\ \ \frac{s_1 d_1 d_H}{s'_1 D}\right) \\ u_2 & \left(\mathrm{mod}\ \ \frac{s_2 d_1 d_H}{s'_2 D}\right) \end{cases}, \\
S'' &= a_1 s_1 s''_2 + a_2 s'_1 s'_2 H + a_3 s'_1 s''_2 (u_1 + w_2 H) \\
&\quad + a_4 s_2 s''_1 + a_5 s'_2 s''_1 (u_2 + w_1 H) + a_6 s''_1 s''_2 (v_1 + v_2 + w_1 w_2 H)\ ,
\end{aligned}$$

and k is such that

$$d_1 \left| \left(\frac{u^3 + F}{S/S'd_1} - 3u^2 k \right) \right. .$$

Proof: Since $\mathfrak{a} = \mathfrak{a}_1\mathfrak{a}_2$ is primitive, $\mathfrak{a}$ is of the form $[S,\, S'(U+\rho),\, S''(V+W\rho+\omega)]$, with $\gcd(S',\, S'') = 1$. For the purposes of this proof, we define $s_{i,H} = \gcd(s_i,\, H)$, for $i = 1,\, 2$. By Lemma 4.2.12, we may factor each $\mathfrak{a}_i = \mathfrak{b}_i\mathfrak{c}_i\mathfrak{d}_i$, for $i = 1,\, 2$, where $\mathfrak{b}_i = [s_{i,H},\, \rho,\, s_i''\omega]$ is the product of prime ideals of type 2 and $\mathfrak{c}_i = [s_i/\left(s_i's_{i,H}\right),\, u_i + \rho,\, v_i + w_i\rho + \omega]$ and $\mathfrak{d}_i = [s_i',\, s_i'\rho,\, v_i + w_i\rho + \omega]$ are products of prime ideals of types 1, 3, and 4. For the proof, we will consider the subproducts $\mathfrak{b} = \mathfrak{b}_1\mathfrak{b}_2$, $\mathfrak{c} = \mathfrak{c}_1\mathfrak{c}_2$, and $\mathfrak{d} = \mathfrak{d}_1\mathfrak{d}_2$ to first determine S, S', and S''. We will then consider certain factors of $\mathfrak{b}$ and $\mathfrak{c}$ to determine U, then finally, we will multiply the bases of $\mathfrak{a}_1$ and $\mathfrak{a}_2$ to determine W and V.

By Lemma 4.2.13, we have $\mathfrak{b} = \mathfrak{b}_1\mathfrak{b}_2 = [s_{1,H}s_{2,H}/d_H,\, \rho,\, s_1''s_2''d_H\omega] = \mathfrak{b}_1'\mathfrak{b}_2'\mathfrak{b}_3'$, where $\mathfrak{b}_1' = [s_{1,H},\, \rho,\, \omega]$, $\mathfrak{b}_2' = [s_{2,H},\, \rho,\, \omega]$, $\mathfrak{b}_3' = [s_1''s_2'',\, \rho,\, \omega]$, and $d_H = \gcd(s_{1,H},\, s_{2,H})$. Now we are given that $D = \gcd(s_1/s_1',\, s_2/s_2')$. Since $\gcd(s_i',\, s_{i,H}) = 1$, for $i = 1,\, 2$, by (4.1), it follows that $d_H = \gcd(D,\, H)$, as given in the statement of the theorem. Considering $\mathfrak{b}$, we have $S'' = s_1''s_2''d_H$.

Now let $d = \gcd\left(s_1/s_1's_{1,H},\, s_2/s_2's_{2,H}\right)$ and $d_1' = \gcd(d,\, u_1 - u_2)/\gcd(d,\, G)$. By (4.1), we have $\gcd(d,\, H) = 1$, and hence, $\gcd(d,\, d_H) = 1$ as well. From this and the argument above, we notice that $D = dd_H$. By Lemma 4.1.4, we have $s_{i,H} \mid u_i$, for $i = 1,\, 2$, so $d_H \mid (u_1 - u_2)$, and $d_H = \gcd(d_H,\, u_1 - u_2) = \gcd(d_H,\, H)$. It follows that $d_1' = \gcd(d,\, u_1 - u_2)/\gcd(d,\, G) = \gcd(d,\, u_1 - u_2)\gcd(d_H,\, u_1 - u_2)/\left(\gcd(d,\, G)\gcd(d_H,\, H)\right) = \gcd(D,\, u_1 - u_2)/\gcd(D,\, GH)$, so that $d_1' = d_1$ as given in the statement of the theorem. Therefore, if $\mathfrak{c} = \mathfrak{c}_1\mathfrak{c}_2$ and $\mathfrak{d} = \mathfrak{d}_1\mathfrak{d}_2$, then by Parts 2 and 1 of Lemma 4.2.14, respectively, we have

$$\mathfrak{c} = \left[\frac{s_1d_1}{s_1's_{1,H}d},\, u_1 + \rho,\, v_1 + w_1\rho + \omega\right]\left[\frac{s_2d_1}{s_2's_{2,H}d},\, u_2 + \rho,\, v_2 + w_2\rho + \omega\right]\left[\frac{d}{d_1},\, \frac{d}{d_1}\rho,\, V_1 + W_1\rho + \omega\right] \tag{4.5}$$

and $\mathfrak{d} = [s_1's_2',\, s_1's_2'\rho,\, V_2 + W_2\rho + \omega]$, for some polynomials $V_1,\, W_1,\, V_2,\, W_2 \in \mathbb{F}_q[x]$. We denote the factors of $\mathfrak{c}$ in (4.5) by $\mathfrak{c}_1'$, $\mathfrak{c}_2'$, and $\mathfrak{c}_3'$, respectively. Since $\mathfrak{a}$ is primitive, we have $\mathfrak{c}_3'\mathfrak{d} = [s_1's_2'd/d_1,\, (s_1's_2'd/d_1)\,\rho,\, V_3 + W_3\rho + \omega]$ by Part 1 of Lemma 4.2.14, for some polynomials $V_3,\, W_3 \in \mathbb{F}_q[x]$. Also, by Part 3 of Lemma 4.2.14, $\mathfrak{c}_1'\mathfrak{c}_2' = [N(\mathfrak{c}_1'\mathfrak{c}_2'),\, U_4 + \rho,\, V_4 + W_4\rho + \omega]$, for some polynomials $U_4,\, V_4,\, W_4 \in \mathbb{F}_q[x]$. Since $\mathfrak{a}$ is primitive, $\mathfrak{b}(\mathfrak{c}_1'\mathfrak{c}_2')(\mathfrak{c}_3'\mathfrak{d})$ is the factorization of $\mathfrak{a}$ given in Lemma 4.2.12. It follows that $S' = s_1's_2'd/d_1$, and equating norms, we have $S = (s_1s_2d_1)/(dd_H)$.

Since $\gcd(d,\, H) = 1$, we also have $\gcd(d_1,\, H) = 1$, and hence $\gcd(d/d_1,\, H) = 1$. Let $S_H = \gcd(S,\, H)$, so that $S_H = \gcd(s_1s_2/d_H,\, H) = s_{1,H}s_{2,H}/d_H$. Now it also follows from Lemma 4.2.12 and the fact that $\mathfrak{a} = [S,\, S'(U+\rho),\, S''(V+W\rho+\omega)]$ is primitive that the factor $\mathfrak{c}_1'\mathfrak{c}_2'$ is of the form $[S/S'S_H,\, U+\rho,\, V+W\rho+\omega]$. Further, by the uniqueness result of Lemma 4.2.12, we have $S/S'S_H = N(\mathfrak{c}_1'\mathfrak{c}_2')$. In addition, since $\mathfrak{b}_1'\mathfrak{b}_2'$ is the product of type 2 prime ideals, it is relatively prime to $\mathfrak{c}_1'\mathfrak{c}_2'$, with $L(\mathfrak{b}_1'b_2') = s_{1,H}s_{2,H} = S_Hd_H$ and $L(\mathfrak{c}_1'\mathfrak{c}_2') = S/S'S_H$. By Theorem 4.2.2, we have $L((\mathfrak{b}_1'\mathfrak{b}_2')(\mathfrak{c}_1'\mathfrak{c}_2')) = Sd_H/S'$, and since $\mathfrak{c}_1'\mathfrak{c}_2' \mid \mathfrak{b}_1'\mathfrak{b}_2'\mathfrak{c}_1'\mathfrak{c}_2'$, Lemma 4.2.3 implies that $\mathfrak{b}_1'\mathfrak{b}_2'\mathfrak{c}_1'\mathfrak{c}_2' = [Sd_H/S',\, U+\rho,\, V+W\rho+\omega]$. We will use these products of $\mathfrak{b}_1'$, $\mathfrak{b}_2'$, $\mathfrak{c}_1'$, and $\mathfrak{c}_2'$ to determine U uniquely modulo S/S'.

Since $\mathfrak{b}_i'$ and $\mathfrak{c}_i'$ are relatively prime, for $i = 1,\, 2$, and $d = D/d_H$, Lemma 4.2.12 and Theorem 4.2.2 state that

$$\mathfrak{b}_i'\mathfrak{c}_i' = [s_{i,H},\, \rho,\, \omega]\left[\frac{s_id_1}{s_i's_{i,H}d},\, u_i + \rho,\, v_i + w_i\rho + \omega\right] = \left[\frac{s_id_1d_H}{s_i'D},\, u_i + \rho,\, v_i + w_i\rho + \omega\right] .$$

Now, since $\mathfrak{b}_1'\mathfrak{b}_2'\mathfrak{c}_1'\mathfrak{c}_2' \subseteq \mathfrak{b}_i'\mathfrak{c}_i'$, for $i = 1, 2$, Lemma 4.1.5 states that U satisfies

$$U \equiv u_1 \left(\mathrm{mod}\ \tfrac{s_1 d_1 d_H}{s_1' D}\right) \quad \text{and} \quad U \equiv u_2 \left(\mathrm{mod}\ \tfrac{s_2 d_1 d_H}{s_2' D}\right) . \tag{4.6}$$

This determines U uniquely modulo $l := \mathrm{lcm}\,(s_1 d_1 d_H/s_1' D,\ s_2 d_1 d_H/s_2' D)$. To determine l explicitly, notice that since $D = \gcd(s_1/s_1',\ s_2/s_2')$, we have $\gcd\,(s_1 d_1 d_H/s_1' D,\ s_2 d_1 d_H/s_2' D) = d_1 d_H$. Thus,

$$l = \left(\frac{s_1 d_1 d_H}{s_1' D}\right)\left(\frac{s_2 d_1 d_H}{s_2' D}\right) / \,(d_1 d_H) = \frac{s_1 s_2 d_1 d_H}{s_1' s_2' D^2} .$$

Now, $S/S' = \left(s_1 s_2 d_1^2 d_H\right) / \left(s_1' s_2' D^2\right)$, so substituting, we have $l = S/(S' d_1)$. Let $u \in \mathbb{F}_q[x]$ be any polynomial satisfying the system of congruences in (4.6). Then $U = u - kl = u - kS/(S' d_1) \pmod{S/S'}$, for some polynomial $k \in \mathbb{F}_q[x]$. By Lemma 4.1.4, we have $(S/S') \mid N(U+\rho)$, so S/S' divides

$$N(U+\rho) = N\left(u - k\frac{S}{S' d_1} + \rho\right) = u^3 + F - 3u^2 k\frac{S}{S' d_1} + 3uk^2\left(\frac{S}{S' d_1^2}\right)\left(\frac{S}{S'}\right) - k^3\left(\frac{S^2}{(S')^2 d_1^3}\right)\left(\frac{S}{S'}\right) . \tag{4.7}$$

Since $D = \gcd\,(s_1/s_1',\ s_2/s_2')$, we have $s_1/(s_1' D),\ s_2/(s_2' D) \in \mathbb{F}_q[x]$. From this it follows that $(s_1 s_2 d_H)/\left(s_1' s_2' D^2\right) = S/(S' d_1^2) \in \mathbb{F}_q[x]$. Therefore, $d_1^2 \mid (S/S')$, so S/S' divides the last two terms of (4.7), and S/S' must, in turn, divide the first three. Also, since $S/(S' d_1)$ divides the last three terms, it divides $u^3 + F$. Therefore, there is some polynomial $a \in \mathbb{F}_q[x]$ such that $aS/S' = u^3 + F - 3u^2 kS/(S' d_1)$. Dividing this through by $S/(S' d_1)$, we have $a d_1 = (u^3+F)/(S/S' d_1) - 3u^2 k$, so k is determined so that

$$d_1 \left| \left(\frac{u^3 + F}{S/(S' d_1)} - 3u^2 k\right) \right. .$$

With this polynomial k, $U = u - kS/(S' d_1) \pmod{S/S'}$ is determined uniquely.

To determine V and W, we multiply the given bases of $\mathfrak{a}_1 = [s_1,\ s_1'(u_1 + \rho),\ s_1''(v_1 + w_1\rho + \omega)]$ and $\mathfrak{a}_2 = [s_2,\ s_2'(u_2 + \rho),\ s_2''(v_2 + w_2\rho + \omega)]$. Since $\{S,\ S'(U + \rho),\ S''(V + W\rho + \omega)\}$ is a minimal canonical basis of $\mathfrak{a} = \mathfrak{a}_1\mathfrak{a}_2$, we have $\mathfrak{a} = [\alpha,\ \beta,\ \gamma,\ \delta,\ \epsilon,\ \zeta,\ \eta,\ \theta,\ \kappa]$, where

$$\begin{aligned}
&\alpha = s_1 s_2 \ , \quad \beta = s_1 s_2'(u_2 + \rho) \ , \quad \gamma = s_1 s_2''(v_2 + w_2\rho + \omega) \ , \quad \delta = s_2 s_1'(u_1 + \rho) \ , \\
&\epsilon = s_1' s_2'(u_1 u_2 + (u_1 + u_2)\rho + H\omega) \ , \quad \zeta = s_1' s_2''(u_1 v_2 + GH + (u_1 w_2 + v_2)\rho + (u_1 + w_2 H)\omega) \ , \\
&\eta = s_1'' s_2(v_1 + w_1\rho + \omega) \ , \quad \theta = s_1'' s_2'(u_2 v_1 + GH + (u_2 w_1 + v_1)\rho + (u_2 + w_1 H)\omega) \ , \quad \text{and} \\
&\kappa = s_1'' s_2''(v_1 v_2 + (w_1 + w_2)GH + (v_1 w_2 + v_2 w_1 + G)\rho + (v_1 + v_2 + w_1 w_2 H)\omega) .
\end{aligned}$$

We may write $S''(V + W\rho + \omega)$ as an $\mathbb{F}_q[x]$-linear combination of α, β, γ, δ, ϵ, ζ, η, θ, and κ:

$$S''(V + W\rho + \omega) = A\alpha + B\beta + a_1\gamma + C\delta + a_2\epsilon + a_3\zeta + a_4\eta + a_5\theta + a_6\kappa \ ,$$

with A, B, C, $a_i \in \mathbb{F}_q[x]$, for $1 \le i \le 6$.

We compare the coefficients of ω, ρ, and 1. First, for the ω terms, we see that

$$S'' = a_1 s_1 s_2'' + a_2 s_1' s_2' H + a_3 s_1' s_2''(u_1 + w_2 H) + a_4 s_2 s_1'' + a_5 s_2' s_1''(u_2 + w_1 H) + a_6 s_1'' s_2''(v_1 + v_2 + w_1 w_2 H) .$$

Considering the ρ terms, we have $S''W = Bs_1s_2' + Cs_1's_2 + W'$, with

$$\begin{aligned} W' = {} & a_1s_1s_2''w_2 + a_2s_1's_2'(u_1+u_2) + a_3s_2''s_1'(u_1w_2+v_2) \\ & + a_4s_2s_1''w_1 + a_5s_2's_1''(u_2w_1+v_1) + a_6s_1''s_2''(v_1w_2+v_2w_1+G) \ , \end{aligned}$$

Now $S' = (s_1's_2'd/d_1) \mid s_1's_2'd \mid s_1's_2'D \mid (s_1's_2's_1/s_1') = s_1s_2'$. Similarly, $S' \mid s_1's_2$, so $S''W \equiv W' \pmod{S'}$. By (4.1), we have $S'' \mid H$ and $\gcd(S', H) = 1$, so $\gcd(S', S'') = 1$ and there is some $r \in \mathbb{F}_q[x]$ such that $rS'' \equiv 1 \pmod{S'}$. Thus, $W \equiv rW' \pmod{S'}$, which determines W uniquely modulo S'.

Considering the constant terms, we have $S''V = As_1s_2 + Bs_1s_2'u_2 + Cs_2s_1'u_1 + V'$, where

$$\begin{aligned} V' = {} & a_1s_1s_2''v_2 + a_2s_1's_2'u_1u_2 + a_3s_1's_2''(u_1v_2+GH) \\ & + a_4s_2s_1''v_1 + a_5s_2's_1''(u_2v_1+GH) + a_6s_1''s_2''(v_1v_2+(w_1+w_2)GH) \ . \end{aligned}$$

To determine V uniquely, we first recall that $U \equiv u_i \pmod{s_id_1/s_i'd}$, for $i = 1, 2$, since $d = D/d_H$. Thus, there exist polynomials $a, b \in \mathbb{F}_q[x]$ such that $U = u_1 + a(s_1d_1/s_1'd)$ and $U = u_2 + b(s_2d_1/s_2'd)$. Therefore,

$$\begin{aligned} Bs_1s_2'u_2 + Cs_2s_1'u_1 &= Bs_1s_2'\left(U - b\frac{s_2d_1}{s_2'd}\right) + Cs_2s_1'\left(U - a\frac{s_1d_1}{s_1'd}\right) \\ &= (Bs_1s_2' + Cs_2s_1')U - \frac{s_1s_2d_1}{d}(Bb+Ca) \ . \end{aligned}$$

Note that $S = s_1s_2d_1/D = s_1s_2d_1/(dd_H)$, so $S \mid (s_1s_2d_1/d)$ and $(S/S'') \mid (s_1s_2d_1/d)$. Thus, $Bs_1s_2'u_2 + Cs_2s_1'u_1 \equiv (Bs_1s_2' + Cs_2s_1')U \equiv U(S''W - W') \pmod{S/S''}$. Also, $S \mid s_1s_2$, so $(S/S'') \mid s_1s_2$, which implies that $S''V = As_1s_2 + Bs_1s_2'u_2 + Cs_2s_1'u_1 + V' \equiv V' + U(S''W - W') \pmod{S/S''}$. By Lemma 4.1.4, we have $S'' \mid \gcd(S, H) \mid U$. In addition, $S'' \mid S \mid s_1s_2$, $S'' \mid S \mid (s_1s_2d_1/d)$, and $S''V = As_1s_2 + (Bs_1s_2' + Cs_2s_1')U + (s_1s_2d_1/d)(Bb + Ca) + V'$ imply that $S'' \mid V'$. Since $\gcd(S/S'', S'') = 1$, we have $V \equiv (V'/S'') + (U/S'')(S''W - W') \pmod{S/S''}$. Furthermore, with these choices of U, V, and W, we see that the canonical basis is minimal. □

With the exception of U, the computation of every other basis element of the product is clear. We will outline the steps to compute U in Algorithm 4.4.9.

For the last case of ideal multiplication, we consider two primitive ideals whose product is not primitive. This theorem extends Lemma 7.3 of [Bau04] to the case in which $H(x)$ is any square-free monic polynomial relatively prime to $G(x)$. Here we simplify and prove the corresponding statement in [Bau05], which was previously unproved.

Theorem 4.4.5 (Lemma 0.36 of [Bau05]) *Let $K = \mathbb{F}_q(C)$ be a purely cubic function field, with $C : Y^3 = G(x)H^2(x)$, and, for $i = 1, 2$, let $\mathfrak{a}_i = [s_i, s_i'(u_i + \rho), s_i''(v_i + w_i\rho + \omega)]$ be two primitive ideals, given in terms of their minimal canonical bases. Then $\mathfrak{a}_1\mathfrak{a}_2 = \langle d \rangle \mathfrak{a}$, where $d = d_1d_2d_3d_4d_5d''$,*

$\mathfrak{a} = \mathfrak{a}_1'\mathfrak{a}_2'\mathfrak{a}_3'$ *is primitive,* $\mathfrak{a}_3' = [d_3 d'', (w_1+w_2)H+\rho, -w_1w_2H+\omega]$,

$$\mathfrak{a}_1' = \left[\frac{s_1}{d}, \frac{s_1'}{d_2d_3}(u_1+\rho), \frac{s_1''}{d_5d''}(v_1+w_1\rho+\omega)\right] \ , \ \mathfrak{a}_2' = \left[\frac{s_2}{d}, \frac{s_2'}{d_1d_3}(u_2+\rho), \frac{s_2''}{d_4d''}(v_2+w_2\rho+\omega)\right] \ ,$$

$$d'' = \gcd(s_1'', s_2'') \ , \quad d_1 = \gcd\left(\frac{s_1}{s_1's_{1,H}}, s_2', u_1+w_2H\right) \ , \quad d_2 = \gcd\left(\frac{s_2}{s_2's_{2,H}}, s_1', u_2+w_1H\right) \ ,$$

$$d_3 = \gcd\left(\frac{s_1'}{d_2}, \frac{s_2'}{d_1}, v_1+v_2+w_1w_2H\right) \ , \ d_4 = \gcd\left(\frac{s_{1,H}}{s_1''}, s_2''\right) \ , \quad \textit{and} \quad d_5 = \gcd\left(\frac{s_{2,H}}{s_2''}, s_1''\right) \ ,$$

and $s_{i,H} = \gcd(s_i, H)$, *for* $i = 1, 2$.

Proof: By Lemma 4.2.12, we may factor each $\mathfrak{a}_i$ as $\mathfrak{a}_i = \mathfrak{b}_i\mathfrak{c}_i\mathfrak{d}_i$, for $i = 1, 2$, where

$$\mathfrak{b}_i = [s_{i,H}, \rho, s_i''\omega] \ , \quad \mathfrak{c}_i = \left[\frac{s_i}{s_i's_{i,H}}, u_i+\rho, v_i+w_i\rho+\omega\right] \ , \quad \text{and} \quad \mathfrak{d}_i = [s_i', s_i'\rho, v_i+w_i\rho+\omega] \ .$$

Here, $\mathfrak{b}_1$ and $\mathfrak{b}_2$ are the product of type 2 prime ideals and $\mathfrak{c}_1$, $\mathfrak{c}_2$, $\mathfrak{d}_1$, and $\mathfrak{d}_2$ are products of prime ideals of types 1, 3, and 4. The approach we will take for the proof is to determine the non-primitive factor of $\mathfrak{a}_1\mathfrak{a}_2$ by considering the products $\mathfrak{b} = \mathfrak{b}_1\mathfrak{b}_2$, and pairs of the other four factors. For each pair, we will determine and divide out the non-primitive factor to find a set of ideals whose product is the primitive part of $\mathfrak{a}_1\mathfrak{a}_2$.

First, by Lemma 4.2.13, we have $\mathfrak{b}_1\mathfrak{b}_2 = \langle d_4d_5d''\rangle\, \mathfrak{b}_1'\mathfrak{b}_2'\mathfrak{b}_3'$, where

$$\mathfrak{b}_1' = \left[\frac{s_{1,H}}{d_4d_5d''}, \rho, \frac{s_1''}{d_5d''}\omega\right] \ , \quad \mathfrak{b}_2' = \left[\frac{s_{2,H}}{d_4d_5d''}, \rho, \frac{s_2''}{d_4d''}\omega\right] \ , \quad \mathfrak{b}_3' = [d'', \rho, \omega] \ ,$$

$d_4 = \gcd(s_{1,H}/s_1'', s_2'')$, $d_5 = \gcd(s_{2,H}/s_2'', s_1'')$, and $d'' = \gcd(s_1'', s_2'')$.

Next, we consider $\mathfrak{c}_1$ and $\mathfrak{c}_2$, and claim that $\mathfrak{c}_1\mathfrak{c}_2$ is primitive. By Corollary 4.2.9, if $\mathfrak{p}$ is a type 1 prime ideal dividing $\mathfrak{c}_1$ and $\mathfrak{c}_2$, then we have $\mathfrak{p}^2 \| \mathfrak{c}_1\mathfrak{c}_2$. Similarly, if $\mathfrak{p}$ is a type 3 prime ideal dividing $\mathfrak{c}_1$ or $\mathfrak{c}_2$, then it is of degree 1. Lastly, if $\mathfrak{p}$ is a type 4 prime ideal dividing $\mathfrak{c}_1$ and $\mathfrak{p}^{(i)} \mid \mathfrak{c}_2$, for some $0 \le i \le 2$, then we cannot have $\mathfrak{p}\mathfrak{p}'\mathfrak{p}'' \mid \mathfrak{c}_1\mathfrak{c}_2$. Thus, $\mathfrak{c}_1\mathfrak{c}_2$ is primitive, so the remaining non-primitive factor of $\mathfrak{a}_1\mathfrak{a}_2$ is obtained from the other three pairs of ideals.

We now consider the products $\mathfrak{c}_1\mathfrak{d}_2$ and $\mathfrak{c}_2\mathfrak{d}_1$. We will show that the polynomials d_1, $d_2 \in \mathbb{F}_q[x]$, as defined in the theorem, satisfy $\mathfrak{c}_1\mathfrak{d}_2 = \langle d_1\rangle\, \mathfrak{c}_1'\mathfrak{d}_2'$ and $\mathfrak{c}_2\mathfrak{d}_1 = \langle d_2\rangle\, \mathfrak{c}_2'\mathfrak{d}_1'$, where $\mathfrak{c}_i' \mid \mathfrak{c}_i$ and $\mathfrak{d}_i' \mid \mathfrak{d}_i$, for $i = 1, 2$, and the products $\mathfrak{c}_1'\mathfrak{d}_2'$ and $\mathfrak{c}_2'\mathfrak{d}_1'$ are primitive. Multiplying the respective basis elements of $\mathfrak{c}_1$, $\mathfrak{c}_2$, $\mathfrak{d}_1$, and $\mathfrak{d}_2$, we see that d_1 and d_2 are the greatest common divisor of the coefficients of the ω terms of $\mathfrak{c}_1\mathfrak{d}_2$ and $\mathfrak{c}_2\mathfrak{d}_1$, respectively. Thus, we will show that

$$d_1 = \gcd\left(\frac{s_1}{s_1's_{1,H}}, s_2'H, u_1+w_2H, v_1+v_2+w_1w_2H, s_2', s_2'w_1H\right)$$

and

$$d_2 = \gcd\left(\frac{s_2}{s_2's_{2,H}}, s_1'H, u_2+w_1H, v_1+v_2+w_1w_2H, s_1', s_1'w_2H\right) \ .$$

We will simplify the right-hand side of the equations above, and will consider d_1; the simplifications for d_2 are analogous. First, $s_2' \mid s_2'H \mid s_2'w_1H$, so we may omit $s_2'H$ and $s_2'w_1H$. Next, since $\gcd(s_1/(s_1's_{1,H}), H) = 1$, there is a polynomial $r \in \mathbb{F}_q[x]$ such that $rH \equiv 1 \pmod{s_1/(s_1's_{1,H})}$. Then we have $(w_1 - ru_1 + r\rho)(u_1+\rho) = u_1w_1 - ru_1^2 + w_1\rho + rH\omega$, and by (4.2), we have $u_1w_1 - ru_1^2 \equiv$

$v_1 \pmod{s_1/(s_1' s_{1,H})}$. Thus, $v_1 + w_1\rho + \omega$ can be written as an $\mathcal{O}$-linear combination of $s_1/(s_1' s_{1,H})$ and $u_1 + \rho$. Since the term $v_1 + v_2 + w_1 w_2 H$ is obtained from the product $(v_1 + w_1\rho + \omega)(v_2 + w_2\rho + \omega)$, while the terms $s_1/(s_1' s_{1,H})$ and $u_1 + w_2 H$ are determined from the products $(s_1/(s_1' s_{1,H}))(v_2 + w_2\rho + \omega)$ and $(u_1 + \rho)(v_2 + w_2\rho + \omega)$, respectively, we must have $\gcd\left(s_1/(s_1' s_{1,H}),\, u_1 + w_2 H\right) \mid (v_1 + v_2 + w_1 w_2 H)$. Therefore, we indeed have

$$d_1 = \gcd\left(\frac{s_1}{s_1' s_{1,H}},\, s_2',\, u_1 + w_2 H\right) \quad \text{and} \quad d_2 = \gcd\left(\frac{s_2}{s_2' s_{2,H}},\, s_1',\, u_2 + w_1 H\right) \ .$$

Dividing the factors of $\langle d_1 \rangle$ and $\langle d_2 \rangle$ out of the respective ideals, we have

$$\mathfrak{c}_i' = \left[\frac{s_i}{s_{i,H} s_i' d_i},\, u_i + \rho,\, v_i + w_i\rho + \omega\right] \quad \text{and} \quad \mathfrak{d}_i' = \left[\frac{s_i'}{d_{3-i}},\, \frac{s_i'}{d_{3-i}}\rho,\, v_i + w_i\rho + \omega\right] \ ,$$

for $i = 1, 2$, by Lemmas 4.2.3 and 4.2.4. After factoring out these non-primitive components, we have $\mathfrak{a}_1\mathfrak{a}_2 = \langle d_1 d_2 d_4 d_5 d'' \rangle\, \mathfrak{b}_1'\mathfrak{b}_2'\mathfrak{b}_3'\mathfrak{c}_1'\mathfrak{c}_2'\mathfrak{d}_1'\mathfrak{d}_2'$ thus far.

Lastly, any remaining non-primitive factor of $\mathfrak{a}_1\mathfrak{a}_2$ must be a factor of $\mathfrak{d}_1'\mathfrak{d}_2'$. We will find $d_3 \in \mathbb{F}_q[x]$ such that $\mathfrak{d}_1'\mathfrak{d}_2' = \langle d_3 \rangle\, \mathfrak{d}''$, where $\mathfrak{d}''$ is primitive. Multiplying the basis elements of $\mathfrak{d}_1'$ and $\mathfrak{d}_2'$, we may obtain d_3 by taking the greatest common divisor of the ω terms. Thus, we will show that

$$d_3 = \gcd\left(\frac{s_1'}{d_2},\, \frac{s_1' s_2' H}{d_1 d_2},\, \frac{s_1' w_2 H}{d_2},\, \frac{s_2'}{d_1},\, \frac{s_2' w_1 H}{d_1},\, v_1 + v_2 + w_1 w_2 H\right) \ .$$

Since $(s_1'/d_2) \mid (s_1' s_2' H/d_1 d_2)$, $(s_1'/d_2) \mid (s_1' w_2 H/d_2)$, and $(s_2'/d_1) \mid (s_2' w_1 H/d_1)$, we indeed have

$$d_3 = \gcd\left(\frac{s_1'}{d_2},\, \frac{s_2'}{d_1},\, v_1 + v_2 + w_1 w_2 H\right) \ . \tag{4.8}$$

To determine the form of the ideal $\mathfrak{d}''$, we consider $\mathfrak{e}_i = [d_3,\, d_3\rho,\, v_i + w_i\rho + \omega]$, for $i = 1, 2$. By Lemma 4.2.4, we have $\mathfrak{e}_i \mid \mathfrak{d}_i'$, for $i = 1, 2$. We will determine ideals $\mathfrak{a}_3$, $\mathfrak{a}_4$, and $\mathfrak{a}_5$ such that $\mathfrak{d}_1'\mathfrak{d}_2' = \langle d_3 \rangle\, \mathfrak{d}'' = \langle d_3 \rangle\, \mathfrak{a}_3\mathfrak{a}_4\mathfrak{a}_5$. We write $\mathfrak{d}_1' = \mathfrak{a}_4\mathfrak{e}_1$ and $\mathfrak{d}_2' = \mathfrak{a}_5\mathfrak{e}_2$, so that $\mathfrak{d}_1'\mathfrak{d}_2' = (\mathfrak{e}_1\mathfrak{e}_2)\mathfrak{a}_4\mathfrak{a}_5$. To determine $\mathfrak{e}_1\mathfrak{e}_2$, and hence the ideal, $\mathfrak{a}_3$, we will find and multiply $\bar{\mathfrak{e}}_1$ and $\bar{\mathfrak{e}}_2$, showing that $\bar{\mathfrak{e}}_1\bar{\mathfrak{e}}_2$ is principal and divides $\langle d_3 \rangle$. We will then show that $\langle d_3 \rangle \,/\, (\bar{\mathfrak{e}}_1\bar{\mathfrak{e}}_2) = \mathfrak{a}_3$.

First, we find canonical bases for $\mathfrak{a}_4$ and $\mathfrak{a}_5$. Note that by Corollary 4.2.10, $\mathfrak{d}_1'$, $\mathfrak{d}_2'$, $\mathfrak{e}_1$, and $\mathfrak{e}_2$ are the products of squares of type 1 prime ideals, powers of type 3 prime ideals of degree 2, and powers of products $\mathfrak{p}\mathfrak{p}'$, where $\mathfrak{p}$ and $\mathfrak{p}'$ are type 4 prime ideals. Therefore, the same must be true of $\mathfrak{a}_4 = \mathfrak{d}_1'/\mathfrak{e}_1$ and $\mathfrak{a}_5 = \mathfrak{d}_2'/\mathfrak{e}_2$. By the same reasoning as in the uniqueness portion of the proof of Lemma 4.2.11, we have $\mathfrak{a}_4 = [s_4,\, s_4\rho,\, v_4 + w_4\rho + \omega]$ and $\mathfrak{a}_5 = [s_5,\, s_5\rho,\, v_5 + w_5\rho + \omega]$, for some polynomials s_4, s_5, v_4, v_5, w_4, $w_5 \in \mathbb{F}_q[x]$. By Lemma 4.2.4, however, we have $\mathfrak{a}_4 = [s_4,\, s_4\rho,\, v_1 + w_1\rho + \omega]$ and $\mathfrak{a}_5 = [s_5,\, s_5\rho,\, v_2 + w_2\rho + \omega]$. Finally, since $N(\mathfrak{a}_4)N(\mathfrak{e}_1) = N(\mathfrak{d}_1')$ and $N(\mathfrak{a}_5)N(\mathfrak{e}_2) = N(\mathfrak{d}_2')$, we have $s_4^2 d_3^2 = (s_1'/d_2)^2$ and $s_5^2 d_3^2 = (s_2'/d_1)^2$, so

$$\mathfrak{a}_4 = \left[\frac{s_1'}{d_2 d_3},\, \frac{s_1'}{d_2 d_3}\rho,\, v_1 + w_1\rho + \omega\right] \ , \quad \text{and} \quad \mathfrak{a}_5 = \left[\frac{s_2'}{d_1 d_3},\, \frac{s_2'}{d_1 d_3}\rho,\, v_2 + w_2\rho + \omega\right] \ .$$

Next, we will find a canonical basis for $\mathfrak{a}_3$. We will determine the structure of the ideals $\mathfrak{e}_1$, $\mathfrak{e}_2$, $\bar{\mathfrak{e}}_1$, and $\bar{\mathfrak{e}}_2$ in order to show that $\bar{\mathfrak{e}}_1\bar{\mathfrak{e}}_2$ has a particular form. By Lemma 4.3.2, we have $\bar{\mathfrak{e}}_i = [d_3,\, -w_i H +$

$\rho, -v_i + \omega]$, for $i = 1, 2$. We claim that $\overline{\mathfrak{e}}_1 \mid \mathfrak{e}_2$ and $\overline{\mathfrak{e}}_2 \mid \mathfrak{e}_1$. By Lemma 4.1.5, $\overline{\mathfrak{e}}_1 \mid \mathfrak{e}_2$ if and only if $d_3 \mid d_3$, $1 \mid d_3$, $1 \mid 1$, $0 \equiv d_3(-w_1 H) \pmod{d_3}$, $w_2 \equiv 0 \pmod{1}$, and $v_2 \equiv -v_1 - w_1 w_2 H \pmod{d_3}$. Each congruence is obvious except for the last. However, by the definition of d_3 in (4.8), we have $d_3 \mid (v_1 + v_2 + w_1 w_2 H)$, which establishes the congruence $v_2 \equiv -v_1 - w_1 w_2 H \pmod{d_3}$. Thus, $\overline{\mathfrak{e}}_1 \mid \mathfrak{e}_2$; by a similar argument, we have $\overline{\mathfrak{e}}_2 \mid \mathfrak{e}_1$.

We claim that no type 3 prime ideal divides $\mathfrak{e}_1$. To that end, suppose that $\mathfrak{q} \mid \mathfrak{e}_1$, where $\mathfrak{q}$ is a (degree 2) type 3 prime ideal. Then $\mathfrak{p} \mid \overline{\mathfrak{e}}_1 \mid \mathfrak{e}_2$, where $\mathfrak{p} = \overline{\mathfrak{q}}$ is a type 3 prime ideal of degree 1, which contradicts the fact that no degree 1 type 3 prime ideals divide $\mathfrak{e}_2$. Thus, $\mathfrak{e}_1$ and $\mathfrak{e}_2$ are the product of squares of type 1 prime ideals and powers of products $\mathfrak{p}\mathfrak{p}'$, where $\mathfrak{p}$ and $\mathfrak{p}'$ are type 4 prime ideals.

Next, we establish that the product $\overline{\mathfrak{e}}_1\overline{\mathfrak{e}}_2$ is also of this form. To that end, suppose that $\mathfrak{p} \mid \overline{\mathfrak{e}}_1$ and $\mathfrak{p} \mid \overline{\mathfrak{e}}_2$, where $\mathfrak{p}$ is a type 4 prime ideal. Then $\mathfrak{p}'\mathfrak{p}'' \mid \mathfrak{e}_1$ and $\mathfrak{p}'\mathfrak{p}'' \mid \mathfrak{e}_2$, where $\mathfrak{p}'$ and $\mathfrak{p}''$ are the conjugates of $\mathfrak{p}$. However, $\overline{\mathfrak{e}}_1 \mid \mathfrak{e}_2$ implies that $\mathfrak{p} \mid \mathfrak{e}_2$, contradicting the primitivity of $\mathfrak{e}_2$. It follows that $\overline{\mathfrak{e}}_1\overline{\mathfrak{e}}_2$ is the product of squares of type 1 prime ideals and powers of products $\mathfrak{p}\mathfrak{p}'$, where $\mathfrak{p}$ and $\mathfrak{p}'$ are type 4 prime ideals. By the same reasoning as in the uniqueness portion of the proof of Lemma 4.2.11, we have $\overline{\mathfrak{e}}_1\overline{\mathfrak{e}}_2 = [d_3, d_3\rho, v + w\rho + \omega]$, for some $v, w \in \mathbb{F}_q[x]$. In particular, this product is primitive and $\overline{\mathfrak{e}}_1\overline{\mathfrak{e}}_2 \mid \langle d_3 \rangle$.

We now apply Theorem 4.4.4 to $\overline{\mathfrak{e}}_1$ and $\overline{\mathfrak{e}}_2$ to determine the product $\overline{\mathfrak{e}}_1\overline{\mathfrak{e}}_2$. Since the coefficient of ω (the "s'' term") in the product is 1, Theorem 4.4.4 states that there exist polynomials $a_i \in \mathbb{F}_q[x]$, for $1 \leq i \leq 6$ such that

$$1 = a_1 d_3 + a_2 H + a_3(-w_1 H) + a_4 d_3 + a_5(-w_2 H) + a_6(-v_1 - v_2) \ .$$

However, since $\gcd(d_3, H) = 1$, we may choose a_1 and a_2 so that $a_1 d_3 + a_2 H = 1$, and hence, we have $a_3 = a_4 = a_5 = a_6 = 0$. Since the "$u$ term" of $\overline{\mathfrak{e}}_1\overline{\mathfrak{e}}_2$ is 0, applying Theorem 4.4.4, we obtain

$$v \equiv -a_1 v_2 d_3 + a_2 w_1 w_2 H^2 = -a_1 v_2 d_3 + w_1 w_2 H(1 - a_1 d_3) \equiv w_1 w_2 H \pmod{d_3}$$

and

$$w \equiv 0 - a_2(w_1 + w_2)H = -(w_1 + w_2)(1 - a_1 d_3) \equiv -(w_1 + w_2) \pmod{d_3} \ .$$

Therefore, $\overline{\mathfrak{e}}_1\overline{\mathfrak{e}}_2 = [d_3, d_3\rho, w_1 w_2 H - (w_1 + w_2)\rho + \omega]$. By Lemma 4.3.2, we have $\overline{\overline{\mathfrak{e}}_1\overline{\mathfrak{e}}_2} = [d_3, (w_1 + w_2)H + \rho, -w_1 w_2 H + \omega]$. Since $\mathfrak{e}_1\overline{\mathfrak{e}}_1 = \mathfrak{e}_2\overline{\mathfrak{e}}_2 = \langle d_3 \rangle$, we have

$$\begin{aligned} \mathfrak{e}_1\mathfrak{e}_2 &= \mathfrak{e}_1\mathfrak{e}_2\overline{\mathfrak{e}}_1\overline{\mathfrak{e}}_2 \left(\overline{\mathfrak{e}}_1\overline{\mathfrak{e}}_2\right)^{-1} = \langle d_3 \rangle^2 \left(\overline{\mathfrak{e}}_1\overline{\mathfrak{e}}_2\right)^{-1} = \langle d_3 \rangle \left(\langle d_3 \rangle \,/\, \left(\overline{\mathfrak{e}}_1\overline{\mathfrak{e}}_2\right)\right) \\ &= \langle d_3 \rangle \, \overline{\overline{\mathfrak{e}}_1\overline{\mathfrak{e}}_2} = \langle d_3 \rangle \, [d_3, (w_1 + w_2)H + \rho, -w_1 w_2 H + \omega] \ . \end{aligned}$$

Let $\mathfrak{a}_3 = [d_3, (w_1 + w_2)H + \rho, -w_1 w_2 H + \omega]$, so that $\mathfrak{e}_1\mathfrak{e}_2 = \langle d_3 \rangle \, \mathfrak{a}_3$, as desired.

Combining the factors we have derived, we have $\mathfrak{a}_1\mathfrak{a}_2 = \langle d_1 d_2 d_3 d_4 d_5 d'' \rangle \, \mathfrak{b}_1'\mathfrak{b}_2'\mathfrak{b}_3'\mathfrak{c}_1'\mathfrak{c}_2'\mathfrak{a}_3\mathfrak{a}_4\mathfrak{a}_5$. Let $\mathfrak{a}_1' = \mathfrak{b}_1'\mathfrak{c}_1'\mathfrak{a}_4$, $\mathfrak{a}_2' = \mathfrak{b}_2'\mathfrak{c}_2'\mathfrak{a}_5$, and $\mathfrak{a}_3' = \mathfrak{b}_3'\mathfrak{a}_3$ so that $\mathfrak{a}_1\mathfrak{a}_2 = \langle d \rangle \, \mathfrak{a} = \langle d \rangle \, \mathfrak{a}_1'\mathfrak{a}_2'\mathfrak{a}_3'$, where $d = d_1 d_2 d_3 d_4 d_5 d''$ and $\mathfrak{a}$ is primitive. By Lemma 4.2.12, the ideals

$$\left[\frac{s_1}{d}, \frac{s_1'}{d_2 d_3}(u_1 + \rho), \frac{s_1''}{d_5 d''}(v_1 + w_1\rho + \omega)\right] \ , \quad \left[\frac{s_2}{d}, \frac{s_2'}{d_1 d_3}(u_2 + \rho), \frac{s_2''}{d_4 d''}(v_2 + w_2\rho + \omega)\right] \ , \quad \text{and}$$
$$[d_3 d'', (w_1 + w_2)H + \rho, -w_1 w_2 H + \omega] \tag{4.9}$$

factor uniquely as $\mathfrak{b}_1'\mathfrak{c}_1'\mathfrak{a}_4$, $\mathfrak{b}_2'\mathfrak{c}_2'\mathfrak{a}_5$, and $\mathfrak{b}_3'\mathfrak{a}_3$, respectively, under the criteria in Lemma 4.2.12. Thus, $\mathfrak{a}_1'$, $\mathfrak{a}_2'$, and $\mathfrak{a}_3'$ are the respective ideals in (4.9), as in the statement of the theorem. □

As in the case of squaring, we present the equivalent product for fractional ideals. This corollary generalizes Theorem 4.8.2 of [Sch01].

Corollary 4.4.6 *Let* $\{1, s_i'(u_i+\rho)/s_i, s_i''(u_i+v_i\rho+\omega)/s_i\}$ *be the minimal canonical bases of two respective fractional ideals,* $\mathfrak{f}_i$*, for* $i = 1, 2$*. If* $(\langle s_1\rangle \mathfrak{f}_1)(\langle s_2\rangle \mathfrak{f}_2)$ *is primitive, then* $\mathfrak{f}_1\mathfrak{f}_2 = \langle S/(s_1s_2)\rangle \mathfrak{f}$ *and* $\mathfrak{f} = [1, S'(U+\rho)/S, S''(V+W\rho+\omega)/S]$*, where* S*,* S'*,* S''*,* U*,* V*, and* W *are as given in Theorem 4.4.4. If* $(\langle s_1\rangle \mathfrak{f}_1)(\langle s_2\rangle \mathfrak{f}_2)$ *is not primitive, then* $\mathfrak{f}_1\mathfrak{f}_2 = \langle d(\mathfrak{f})d/(s_1s_2)\rangle \mathfrak{f}$*, where* d *is as in Theorem 4.4.5.*

Proof: In Theorem 4.4.5, we have $(\langle s_1\rangle \mathfrak{f}_1)(\langle s_2\rangle \mathfrak{f}_2) = \mathfrak{a}_1\mathfrak{a}_2 = \langle d\rangle \mathfrak{a}$. Since $\mathfrak{a} = \langle d(\mathfrak{f})\rangle \mathfrak{f}$, we have $\mathfrak{f}_1\mathfrak{f}_2 = \langle (d(\mathfrak{f})d)/(s_1s_2)\rangle \mathfrak{f}$ and $1 \in \mathfrak{f}$, since $\mathfrak{a}$ is primitive. If $d = 1$, then $\mathfrak{a} = [S, S'(U+\rho), S''(V+W\rho+\omega)]$, where S, S', S'', U, V, and W are as given in Theorem 4.4.4. Therefore, $\mathfrak{f} = [1, S'(U+\rho)/S, S''(V+W\rho+\omega)/S]$ □

With Theorem 4.2.2, the results of this section conclude the description of ideal multiplication for all purely cubic function fields of characteristic not equal to 3. In the next section, we will describe algorithms corresponding to these results for use in implementations of this arithmetic.

4.4.2 Ideal Multiplication Algorithms

In this section, we present algorithms for squaring a primitive ideal and multiplying primitive ideals. Most of these algorithms will follow rather naturally from the statements of the corresponding theorems. We will not analyze the running times here, but we do note that for each algorithm, the number of operations in $\mathbb{F}_q[x]$ is polynomial in the degree(s) of the norm(s) of the input ideal(s). The first algorithm will apply to the vast majority of multiplication operations, since we expect two random distinguished ideals to have coprime norms. This algorithm follows from Theorem 4.2.2, Theorem 4.4 of [Sch01], and the Chinese Remainder Theorem.

Algorithm 4.4.7 Ideal Multiplication, $\gcd(s_1, s_2) = 1$
Input: *q; monic G, $H \in \mathbb{F}_q[x]$, relatively prime and square-free; and two primitive ideals, $\mathfrak{a}_1$ and $\mathfrak{a}_2$, of $\mathcal{O}_x = \mathcal{O}(\mathbb{F}_q(C))$, where $C : Y^3 = GH^2$, $\mathfrak{a}_i = [s_i, s_i'(u_i+\rho), s_i''(v_i+w_i\rho+\omega)]$, for $i = 1, 2$, and* $\gcd(s_1, s_2) = 1$.
Output: *The primitive ideal, $\mathfrak{a} = \mathfrak{a}_1\mathfrak{a}_2 = [S, S'(U+\rho), S''(V+W\rho+\omega)]$, with the minimal canonical basis.*

1. *Set $S := s_1s_2$, $S' := s_1's_2'$, and $S'' := s_1''s_2''$.*
2. *Compute a such that $a \equiv (s_1/s_1')^{-1} \pmod{s_2/s_2'}$.*
3. *Set $U :\equiv u_1 + a(s_1/s_1')(u_2-u_1) \pmod{S/S'}$.*
4. *Compute c such that $c \equiv (s_1')^{-1} \pmod{s_2'}$.*
5. *Set $W :\equiv w_1 + cs_1'(w_2-w_1) \pmod{s_1/s_1'}$.*
6. *Compute b such that $b \equiv (s_1/s_1'')^{-1} \pmod{s_2/s_2''}$.*

7. *Set* $V :\equiv (v_1 + u_1(W - w_1)) + b\frac{s_1}{s_1''}(v_2 - v_1 + u_1(W - w_1) - u_2(W - w_2)) \pmod{S/S''}$.

8. *Reduce the size of the* U, V, *and* W *via Lemma 4.1.6.*

9. *Output* $\mathfrak{a} := [S,\ S'(U + \rho),\ S''(V + W\rho + \omega)]$.

The squaring algorithm will be most useful for ideal exponentiation, and follows from Theorem 4.4.2 and Theorem 5.2 of [Sch01].

Algorithm 4.4.8 Ideal Squaring
Input: *q; monic* G, $H \in \mathbb{F}_q[x]$, *relatively prime and square-free; and a primitive ideal,* $\mathfrak{a} = [s,\ s'(u + \rho),\ s''(v + w\rho + \omega)]$, *of* $\mathcal{O}_x = \mathcal{O}(\mathbb{F}_q(C))$, *where* $C : Y^3 = GH^2$.
Output: *The primitive ideal* $\mathfrak{b} = [S,\ S'(U + \rho),\ S''(V + W\rho + \omega)]$, *with the minimal canonical basis, and* $d \in \mathbb{F}_q[x]$ *such that* $\mathfrak{b} = \langle d \rangle\, \mathfrak{a}^2$.

1. *Set* $s_G := \gcd(s,\ G)$, $s'_G = \gcd(s',\ G)$, $s_H := \gcd(s,\ H)$, *and* $d := s'_G s''$.

2. *While* $\gcd(2v + w^2 H,\ s/s_G s_H) \neq 1$

   a. *Choose* $f \in \mathbb{F}_q^*$.

   b. *Set* $w := w + fs'$ *and* $v := v + fus'$.

3. *Set* $z := (2v + Hw^2)^{-1} \pmod{s/s_G s_H}$ *and* $y := (3u^2)^{-1} \pmod{ss'_G/s_G s_H s'}$.

4. *Set* $S := s^2/s_G s_H$, $S' := (s')^2 s_G/(s'_G)^3$, *and* $S'' := s_H/s''$.

5. *Compute* $b :\equiv (s_H s'_G)^{-1} \left(\mathrm{mod}\ \ (ss'_G/s_G s_H s')^2\right)$.

6. *Set* $U :\equiv b\left(u - y\left(u^3 + GH^2\right)\right)(ss'_G/s_G s_H s')^2 \left(\mathrm{mod}\ \ s_H s'_G (ss'_G/s_G s_H s')^2\right)$.

7. *Compute* $a :\equiv (s_G/s'_G)^{-1} \left(\mathrm{mod}\ \ (s'/s'_G)^2\right)$.

8. *Set* $W :\equiv a(w - z(Hw^3 - G))(s'/s'_G)^2 \left(\mathrm{mod}\ \ s_G/s'_G (s'/s'_G)^2\right)$.

9. *Compute* $c :\equiv (s_G s'')^{-1} \left(\mathrm{mod}\ \ (s/s_G s_H)^2\right)$.

10. *Set* $V := c(v + U(W - w) + z(U(Hw^3 - G) + 2GHw - v(v + Hw^2)))(s/s_G s_H)^2$ *and* $V := V \left(\mathrm{mod}\ \ s_G s''(s/s_G s_H)^2\right)$.

11. *Reduce the size of* U, V, *and* W *via Lemma 4.1.6.*

12. *Output* $\mathfrak{b} := [S,\ S'(U + \rho),\ S''(V + W\rho + \omega)]$ *and* d.

The following algorithm applies to the case in which two given ideals have a primitive product, but whose norms are not necessarily coprime. This generalizes Algorithm 7.2 of [Bau04] and corrects the corresponding Algorithm 0.29 of [Bau05]. Its correctness follows mostly from Theorem 4.4.4; we will prove that the polynomial U, given in the algorithm, matches the polynomial given by Theorem 4.4.4.

Algorithm 4.4.9 (Algorithm 0.29 of [Bau05]) Ideal Multiplication - Primitive Product
Input: *q; monic G, $H \in \mathbb{F}_q[x]$, relatively prime and square-free; and two primitive ideals, $\mathfrak{a}_1$ and $\mathfrak{a}_2$, of $\mathcal{O}_x = \mathcal{O}(\mathbb{F}_q(C))$, where $C : Y^3 = GH^2$ and $\mathfrak{a}_i = [s_i, s_i'(u_i + \rho), s_i''(v_i + w_i\rho + \omega)]$, for $i = 1, 2$.*
Output: *The primitive ideal, $\mathfrak{a} = \mathfrak{a}_1\mathfrak{a}_2 = [S, S'(U + \rho), S''(V + W\rho + \omega)]$, with the minimal canonical basis.*

1. *Compute r_1 and $D := \gcd(s_1/s_1', s_2/s_2') = r_1 s_1/s_1 + r_2 s_2/s_2'$ via the half-extended Euclidean algorithm.*
2. *Compute: $d_1 := \gcd(D, u_1 - u_2)/\gcd(D, GH)$ and $d_H := \gcd(D, H)$ via the Euclidean algorithm.*
3. *Set $S := s_1 s_2 d_1/D$, $S' := s_1' s_2' D/(d_1 d_H)$, and $S'' := s_1'' s_2'' d_H$.*
4. *Set $u := u_1 + ((u_2 - u_1) r_1 s_1)/(s_1' D)$.*
5. *Compute r_3 and $d_2 := \gcd(d_1, 3u^2) = 3r_3u^2 + r_4 d_1$ via the half-extended Euclidean algorithm.*
6. *Set $U :\equiv u - r_3(u^3 + GH^2)/d_2 \pmod{S/S'}$.*
7. *Compute a_i, for $1 \le i \le 6$, using successive applications of the extended Euclidean algorithm,*

$$\begin{aligned} S'' := {} & a_1 s_1 s_2'' + a_2 s_1' s_2' H + a_3 s_1' s_2''(u_1 + w_2 H) + a_4 s_2 s_1'' \\ & + a_5 s_2' s_1''(u_2 + w_1 H) + a_6 s_1'' s_2''(v_1 + v_2 + w_1 w_2 H) \ . \end{aligned}$$

8. *Set r such that $rS'' \equiv 1 \pmod{S'}$.*
9. *Set*

$$\begin{aligned} V' := {} & a_1 s_1 s_2'' v_2 + a_2 s_1' s_2' u_1 u_2 + a_3 s_1' s_2''(u_1 v_2 + GH) \\ & + a_4 s_2 s_1'' v_1 + a_5 s_2' s_1''(u_2 v_1 + GH) + a_6 s_1'' s_2''(v_1 v_2 + (w_1 + w_2) GH) \ , \end{aligned}$$

   and

$$\begin{aligned} W' := {} & a_1 s_1 s_2'' w_2 + a_2 s_1' s_2'(u_1 + u_2) + a_3 s_2'' s_1'(u_1 w_2 + v_2) \\ & + a_4 s_2 s_1'' w_1 + a_5 s_2' s_1''(u_2 w_1 + v_1) + a_6 s_1'' s_2''(v_1 w_2 + v_2 w_1 + G) \ , \end{aligned}$$

10. *Set $W :\equiv rW' \pmod{S'}$ and $V :\equiv (V'/S'') + (U/S'')(S''W - W') \pmod{S/S''}$.*
11. *Output $\mathfrak{a} := [S, S'(U + \rho), S''(V + W\rho + \omega)]$.*

Every step of this algorithm follows from the statement of Theorem 4.4.4, except for the computation of U, so we provide a proof that U is the polynomial given in Theorem 4.4.4. The argument parallels the justification of Algorithm 6.5 of [Bau04].

Theorem 4.4.10 *Algorithm 4.4.9 computes the product of two primitive ideals, $\mathfrak{a}_1$, $\mathfrak{a}_2$, whose product is primitive.*

Proof: We use the notation of Theorem 4.4.4 and Algorithm 4.4.9. First, we show that u, found in Step 4, satisfies the congruences $u \equiv u_1 \pmod{s_1 d_1 d_H / s_1' D}$ and $u \equiv u_2 \pmod{s_2 d_1 d_H / s_2' D}$, as given in Theorem 4.4.4. We will then show that the polynomial U, computed in Step 6, matches the polynomial U given in Theorem 4.4.4. Since $d_1 = \gcd(D, u_1 - u_2)/\gcd(D, GH)$, we have $d_1 \mid (u_2 - u_1)$. Also, since $d_H = \gcd(D, H) = \gcd(s_1/s_1', s_2/s_2', H)$, we have $\gcd(d_1, d_H) = 1$. Further, Lemma 4.1.4 implies that $d_H \mid (u_2 - u_1)$. Therefore, $d_1 d_H \mid (u_1 - u_2)$, so $(s_1 d_1 d_H / s_1' D) \mid ((u_2 - u_1) r_1 s_1 / s_1' D)$ and $(s_2 d_1 d_H / s_2' D) \mid ((u_2 - u_1) r_2 s_2 / s_2' D)$. With u as defined in Step 4, we have

$$u = u_1 + (u_2 - u_1)\left(r_1 \frac{s_1}{s_1' D}\right) \equiv u_1 \pmod{\tfrac{s_1 d_1 d_H}{s_1' D}} \ .$$

Substituting

$$r_1 \frac{s_1}{s_1' D} = 1 - r_2 \frac{s_2}{s_2' D} \ ,$$

we have

$$u = u_2 + (u_1 - u_2)\left(r_2 \frac{s_2}{s_2' D}\right) \equiv u_2 \pmod{\tfrac{s_2 d_1 d_H}{s_2' D}} \ ,$$

so u satisfies the congruences as desired.

We now show that the polynomial U, computed in Step 6, is the same polynomial U given in Theorem 4.4.4. From Theorem 4.4.4, there is a polynomial $k \in \mathbb{F}_q[x]$ such that $d_1 \mid ((u^3 + F)/(S/S'd_1) - 3u^2 k)$, where $F = GH^2$; with this k, we have $U \equiv u - kS/(S'd_1) \pmod{S/S'}$. From the divisibility, we have $3u^2 k \equiv (u^3+F)/(S/S'd_1) \pmod{d_1}$. With $d_2 = \gcd(d_1, 3u^2) = 3r_3 u^2 + r_4 d_1$, as found in Step 5, we have $d_2 \mid d_1$ and $d_2 \mid 3u^2$, so $d_2 \mid \big((u^3 + F)/(S/S'd_1)\big)$. We claim that

$$k = r_3 \frac{u^3 + F}{d_2 (S/(S'd_1))}$$

satisfies the desired divisibility criteria. With this choice of k, we have

$$\begin{aligned} \frac{u^3 + F}{S/(S'd_1)} - 3u^2 k &= \frac{u^3 + F}{S/(S'd_1)} - 3u^2 r_3 \frac{u^3 + F}{d_2 (S/(S'd_1))} = \frac{u^3 + F}{S/(S'd_1)} \left(1 - \frac{3 r_3 u^2}{d_2}\right) \\ &= \frac{u^3 + F}{S/(S'd_1)} \left(\frac{r_4 d_1}{d_2}\right) = d_1 \frac{r_4 (u^3 + F)}{d_2 (S/(S'd_1))} \ , \end{aligned}$$

so $d_1 \mid ((u^3 + F)/(S/S'd_1) - 3u^2 k)$, as desired. Since $U \equiv u - kS/(S'd_1) \pmod{S/S'}$ in Theorem 4.4.4, we substitute for k to obtain

$$U \equiv u - k \frac{S}{S'd_1} = u - r_3 \frac{u^3 + F}{d_2 (S/S'd_1)} \left(\frac{S}{S'd_1}\right) = u - r_3 \frac{u^3 + F}{d_2} \pmod{\tfrac{S}{S'}} \ ,$$

which is the assignment given in Step 6. The rest of the algorithm is straightforward. □

Lastly, we give the most general ideal multiplication algorithm. We also expect that this algorithm, in its full generality, will be the least frequent algorithm used in the course of any computation involving several ideal multiplications. This algorithm follows from Theorem 4.4.5 and is given by Algorithm 0.31 of [Bau05].

Algorithm 4.4.11 (Algorithm 0.31 of [Bau05]) Ideal Multiplication - Non-primitive Product
Input: *q; monic G, $H \in \mathbb{F}_q[x]$, relatively prime and square-free; and two primitive ideals, $\mathfrak{a}_1$ and $\mathfrak{a}_2$, of $\mathcal{O}_x = \mathcal{O}(\mathbb{F}_q(C))$, where $C : Y^3 = GH^2$ and $\mathfrak{a}_i = [s_i,\, s_i'(u_i + \rho),\, s_i''(v_i + w_i\rho + \omega)]$, for $i = 1, 2$.*
Output: *The primitive ideal, $\mathfrak{a} = [S, S'(U + \rho), S''(V + W\rho + \omega)]$, with the minimal canonical basis, and $d \in \mathbb{F}_q[x]$ such that $\mathfrak{a}_1\mathfrak{a}_2 = \langle d \rangle\, \mathfrak{a}$.*

1. *Compute d_1, d_2, d_3, d_4, d_5, and d'', as given in the statement of Theorem 4.4.5, and set $d := d_1d_2d_3d_4d_5d''$.*
2. *Set*
$$\begin{aligned}
\mathfrak{a}_1' &:= \left[\frac{s_1}{d},\, \frac{s_1'}{d_2d_3}(u_1 + \rho),\, \frac{s_1''}{d_5d''}(v_1 + w_1\rho + \omega)\right] \ , \\
\mathfrak{a}_2' &:= \left[\frac{s_2}{d},\, \frac{s_2'}{d_1d_3}(u_2 + \rho),\, \frac{s_2''}{d_4d''}(v_2 + w_2\rho + \omega)\right] \ , \qquad \textit{and} \\
\mathfrak{a}_3' &:= [d_3d'', (w_1 + w_2)H + \rho, -w_1w_2H + \omega] \ .
\end{aligned}$$
3. *Reduce the size of the basis elements of $\mathfrak{a}_1'$, $\mathfrak{a}_2'$, and $\mathfrak{a}_3'$ via Lemma 4.1.6.*
4. *Set $\mathfrak{a}_4' := \mathfrak{a}_1'\mathfrak{a}_2'$ via Algorithm 4.4.9.*
5. *Set $\mathfrak{a} := [S, S'(U + \rho), S''(V + W\rho + \omega)] := \mathfrak{a}_4'\mathfrak{a}_3'$ via Algorithm 4.4.9.*
6. *Output $\mathfrak{a}$ and d.*

Lastly, we combine the four multiplication algorithms into one. Afterwards, we will give a brief justification for the step to test the primitivity of $\mathfrak{a}_1\mathfrak{a}_2$. For Algorithm 4.4.12, we do not include any possible non-primitive factors of the product $\mathfrak{a}_1\mathfrak{a}_2$ in the output since we will only need the primitive factor for the applications that use this algorithm.

Algorithm 4.4.12 General Ideal Multiplication
Input: *q; monic G, $H \in \mathbb{F}_q[x]$, relatively prime and square-free; and two primitive ideals, $\mathfrak{a}_1$ and $\mathfrak{a}_2$, of $\mathcal{O}_x = \mathcal{O}(\mathbb{F}_q(C))$, where $C : Y^3 = GH^2$ and $\mathfrak{a}_i = [s_i,\, s_i'(u_i + \rho),\, s_i''(v_i + w_i\rho + \omega)]$, for $i = 1, 2$.*
Output: *A primitive ideal, $\mathfrak{a} \sim \mathfrak{a}_1\mathfrak{a}_2 = [S,\, S'(U + \rho),\, S''(V + W\rho + \omega)]$, with the minimal canonical basis.*

1. *If $\gcd(s_1,\, s_2) = 1$, then apply Algorithm 4.4.7.*
2. *Else if $\mathfrak{a}_1 = \mathfrak{a}_2$, then apply Algorithm 4.4.8.*
3. *Else compute*
$$D := \gcd(s_1s_2'',\, s_1's_2'H,\, s_1's_2''(u_1 + w_2H),\, s_2s_1'',\, s_2's_1''(u_2 + w_1H),\, s_1''s_2''(v_1 + v_2 + w_1w_2H)) \ .$$
   a. *If $D = 1$ or $D = s_1''s_2'' \gcd(s_1,\, s_2,\, H)$, then apply Algorithm 4.4.9.*
   b. *Else, apply Algorithm 4.4.11.*

Each step of Algorithm 4.4.12 is straightforward with the exception of the criteria in Step 3 to determine whether $\mathfrak{a}_1\mathfrak{a}_2$ is primitive or non-primitive, so we explain this step. After multiplying the elements of the canonical bases of $\mathfrak{a}_1$ and $\mathfrak{a}_2$, the variable D is the greatest common divisor of the coefficients of ω. (See the proof of Theorem 4.4.4.) If $D = 1$, then clearly $D = d = S'' = 1$ and $\mathfrak{a}_1\mathfrak{a}_2$ is primitive. If $D = s_1''s_2'' \gcd(s_1, s_2, H)$, then this is precisely the coefficient S'' given for the product of two ideals whose product is $[S, S'(U+\rho), S''(V+W\rho+\omega)]$ in Theorem 4.4.4, and is also the coefficient of ω in the basis. Thus, $\mathfrak{a}_1\mathfrak{a}_2$ is primitive. In all other cases, $\mathfrak{a}_1\mathfrak{a}_2$ is not primitive.

In this section, we showed how to find the product of any two ideals, both in theory and in practice. For most applications in which we multiply ideals, we desire a distinguished ideal equivalent to the product, so we only use the primitive factor of any product. Recall that the product of two distinguished ideals is seldom distinguished. In Section 3.3 of the previous chapter, we discussed distinguished ideals of fields of positive unit rank, and in the next section, we will show how to find a distinguished ideal equivalent to a given ideal for totally ramified cubic function fields.

4.5 Ideal Reduction in Unit Rank 0

This section will only consider totally ramified purely cubic function fields, K. In this case, we have $Cl(\mathcal{O}) \cong \mathcal{J}_K$, so every ideal class of $\mathcal{O}$ contains a unique distinguished ideal by Corollary 3.3.17. In this section, we will show how to find this ideal, given any equivalent ideal. As in the previous sections, we will assume that $\mathrm{char}(K) \neq 3$.

In [GPS02], Galbraith, Paulus, and Smart provide a general reduction algorithm for reducing ideals in the ideal class group of a nonsingular superelliptic field of unit rank 0. In [Bau04], Bauer applied this result to purely cubic function fields, $K = \mathbb{F}_q(C)$, where C is nonsingular. These ideas are motivated by the ideal reduction techniques for number fields (see Section 6.5 of [Coh93]). In this section, we generalize the results of Bauer to the cases where C is possibly singular, though the techniques are similar. The reduction procedure we use is the same as that used in [GPS02] and [Bau04] to reduce a primitive ideal $\mathfrak{a}$ and has three main steps. First, we find the element, $\alpha \in \overline{\mathfrak{a}}$, whose norm has minimal degree in $\overline{\mathfrak{a}}$. Next, we compute a canonical basis for $\langle \alpha \rangle$, and finally, the distinguished ideal equivalent to $\mathfrak{a}$ will be the primitive factor of $\langle \alpha / L(\mathfrak{a}) \rangle \, \mathfrak{a}$. We will state this result more formally in Section 4.5.3, but first we will establish methods to perform each of these steps. The final contribution of the section, and hence the chapter, will be the algorithm to compute the distinguished representative of an ideal class.

4.5.1 Elements of Minimal Norm Degree

We will first prove the existence and uniqueness of an element whose norm has minimal degree in a primitive ideal and give an algorithm to find this element. The following lemma will help us find this element. This result generalizes Proposition 3.1 of [Bau04] to include singular curves, C, defining K. The proof uses the same techniques as the one in [Bau04].

Lemma 4.5.1 *Let $K = \mathbb{F}_q(C)$ be a totally ramified purely cubic function field, with $C : Y^3 = G(x)H^2(x)$ (that is, $r = 0$ and $\mathrm{sig}(K) = (3, 1)$), and $\alpha = a + b\rho + c\omega \in K^*$. Then*

$$\deg(N(\alpha)) = \max\left\{\deg\left(a^3\right), \deg\left(b^3GH^2\right), \deg\left(c^3G^2H\right)\right\} \quad .$$

Proof: From Corollary 3.2.2, if K is a purely cubic function field, then $\mathrm{sig}(K) = (3,1)$ if and only if $3 \nmid \deg\left(G(x)H^2(x)\right)$. This, in turn, holds if and only if $\deg(G) \not\equiv \deg(H) \pmod 3$.

From (3.1), we have $N(\alpha) = a^3 + b^3GH^2 + c^3G^2H - 3abcGH$. Since $\deg(G) \not\equiv \deg(H) \pmod 3$, it follows that $\deg\left(a^3\right)$, $\deg\left(b^3GH^2\right)$, and $\deg\left(c^3G^2H\right)$ each lie in distinct residue classes modulo 3. Therefore, $\deg(-3abcGH)$ must lie in one of these classes. We will show that

$$\deg\left(-3abcGH\right) < \max\left\{\deg\left(a^3\right), \deg\left(b^3GH^2\right), \deg\left(c^3G^2H\right)\right\} \quad ,$$

so that the degree of the norm must be what we claim.

First suppose that $\deg\left(abcGH\right) \geq \deg\left(c^3G^2H\right) = \max\left\{\deg\left(a^3\right), \deg\left(b^3GH^2\right), \deg\left(c^3G^2H\right)\right\}$. Then $2\deg(c) + \deg(G) \leq \deg(a) + \deg(b)$. By maximality, we have $\deg\left(a^3\right) \leq \deg\left(c^3G^2H\right)$, so

$$2\deg(c) + \deg(G) \leq \deg(a) + \deg(b) \leq \deg(c) + \frac{2}{3}\deg(G) + \frac{1}{3}\deg(H) + \deg(b)$$

and

$$\begin{aligned} \deg(c) + \frac{1}{3}\deg(G) - \frac{1}{3}\deg(H) &\leq \deg(b) \\ 3\deg(c) + \deg(G) - \deg(H) + \deg\left(GH^2\right) &\leq 3\deg(b) + \deg\left(GH^2\right) \\ \deg\left(c^3G^2H\right) &\leq \deg\left(b^3GH^2\right) \quad , \end{aligned}$$

which contradicts the maximality of $\deg\left(c^3G^2H\right)$.

Now suppose that $\deg(abcGH) \geq \deg\left(b^3GH^2\right) = \max\{\deg\left(a^3\right), \deg\left(b^3GH^2\right), \deg\left(c^3G^2H\right)\}$. Then $2\deg(b) + \deg(H) \leq \deg(a) + \deg(c)$. Again by maximality, we have $\deg\left(a^3\right) \leq \deg\left(b^3GH^2\right)$, so

$$2\deg(b) + \deg(H) \leq \deg(a) + \deg(c) \leq \deg(b) + \frac{1}{3}\deg(G) + \frac{2}{3}\deg(H) + \deg(c)$$

and

$$\begin{aligned} \deg(b) - \frac{1}{3}\deg(G) + \frac{1}{3}\deg(H) &\leq \deg(c) \\ 3\deg(b) - \deg(G) + \deg(H) + \deg\left(G^2H\right) &\leq 3\deg(c) + \deg\left(G^2H\right) \\ \deg\left(b^3GH^2\right) &\leq \deg\left(c^3G^2H\right) \quad , \end{aligned}$$

which contradicts the maximality of $\deg\left(b^3GH^2\right)$.

Finally, suppose that $\deg(abcGH) \geq \deg\left(a^3\right) = \max\left\{\deg\left(a^3\right), \deg\left(b^3GH^2\right), \deg\left(c^3G^2H\right)\right\}$. Then $2\deg(a) \leq \deg(b) + \deg(c) + \deg(G) + \deg(H)$. Again by maximality, we have $\deg\left(c^3G^2H\right) \leq \deg\left(a^3\right)$, so

$$\begin{aligned} &2\deg(a) \leq \deg(b) + \deg(c) + \deg(G) + \deg(H) \leq \deg(b) + \deg(a) + \frac{1}{3}\deg(G) + \frac{2}{3}\deg(H) \\ &3\deg(a) \leq 3\deg(b) + \deg(G) + 2\deg(H) \\ &\deg\left(a^3\right) \leq \deg\left(b^3GH^2\right) \quad , \end{aligned}$$

which contradicts the maximality of $\deg\left(a^3\right)$.

Therefore, $\deg(-3abcGH) < \max\left\{\deg\left(a^3\right), \deg\left(b^3GH^2\right), \deg\left(c^3G^2H\right)\right\}$, and the claim holds. □

The following theorem is a slight generalization of Theorem 5.1 in [Bau04], and proves the existence of a unique ideal element whose norm has minimal degree. The proof is a straightforward adaptation of the proof in [Bau04].

Theorem 4.5.2 *Let $K = \mathbb{F}_q(C)$ be a totally ramified purely cubic function field, with $C : Y^3 = F(x) = G(x)H^2(x)$ (that is, $r = 0$ and $\mathrm{sig}(K) = (3, 1)$), and $\mathcal{O}$ its maximal order. Every nonzero integral ideal, $\mathfrak{a}$, of $\mathcal{O}$, contains a nonzero element whose norm has minimal degree, which is unique up to multiplication by an element in $\mathbb{F}_q^*$.*

Proof: For $i = 1, 2$, suppose $\alpha_i = a_i + b_i\rho + c_i\omega$ are two nonzero elements of $\mathfrak{a}$ such that $\deg(N(\alpha_1)) = \deg(N(\alpha_2))$ and $\alpha_2 \notin \alpha_1\mathbb{F}_q^*$. By Lemma 4.5.1 and the fact that 1, $\deg\left(GH^2\right)$, and $\deg\left(G^2H\right)$ are in distinct residue classes modulo 3, there are three possibilities. We either have $\deg(a_1) = \deg(a_2)$ and $\deg(N(\alpha_i)) = \deg\left(a_i^3\right)$, $\deg(b_1) = \deg(b_2)$ and $\deg(N(\alpha_i)) = \deg\left(b_i^3GH^2\right)$, or $\deg(c_1) = \deg(c_2)$ and $\deg(N(\alpha_i)) = \deg\left(c_i^3G^2H\right)$, for $i = 1, 2$. In these respective cases, let $k = \mathrm{sgn}(a_1)/\mathrm{sgn}(a_2)$, $k = \mathrm{sgn}(b_1)/\mathrm{sgn}(b_2)$, or $k = \mathrm{sgn}(c_1)/\mathrm{sgn}(c_2)$; in each case, we have $k \in \mathbb{F}_q^*$. If $\alpha_3 = \alpha_1 - k\alpha_2$, then $\alpha_3 \in \mathfrak{a}$, but $N(\alpha_3) < N(\alpha_1)$, which is a contradiction. Thus, $\mathfrak{a}$ contains a unique nonzero element of minimal norm, up to a multiple in $\mathbb{F}_q^*$. □

The following algorithm finds an element of minimal norm in an ideal. The algorithm begins by putting each basis element of the ideal into a separate row of a 3×3 matrix, with the 1, ρ, and ω components of each basis element in columns 1, 2, and 3, respectively. To each matrix entry, the degree of its norm will be assigned a weight. By Lemma 4.5.1, the weights in every row will belong to different equivalence classes modulo 3, so each element in a given row has a different weight and the degree of the norm of the corresponding basis element is equal to the degree of the norm of the row entry having the highest weight in that row. We then use standard matrix row reduction to reduce the maximum weight of each row as follows. If the maximal weights of two rows are in a common column, so that the degrees of the norms of the corresponding basis elements are equal to those of the same component (1, ρ, or ω), then one can reduce the maximal weight of either row, and hence the degree of the norm of either element, as described in the proof of Theorem 4.5.2, creating a new element whose norm has smaller degree. Since Theorem 4.5.2 guarantees the existence of a unique (up to a multiple in $\mathbb{F}_q^*$) element whose norm has minimal degree, this algorithm must terminate in a finite number of steps with the correct output. This algorithm generalizes Algorithm 8.1 of [Bau04], which is a special case of the technique described in Section 6.4 of [GPS02], to include function fields defined by singular curves. If C is nonsingular, then $H(x) = 1$ and $s'' = 1$ in the input to the algorithm.

Algorithm 4.5.3 Minimal Element Algorithm

Input: *q; monic, relatively prime, and square-free polynomials G, $H \in \mathbb{F}_q[x]$ such that $3 \nmid \deg(GH^2)$; and a primitive ideal, $\mathfrak{a} = [s, s'(u + \rho), s''(v + w\rho + \omega)]$ of $\mathcal{O}_x = \mathcal{O}(\mathbb{F}_q(C))$, where $C : Y^3 = GH^2$.*

Output: *The element, $\alpha \in \mathfrak{a}$, of minimal norm.*

1. *Set* $B := \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} := \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix} := \begin{pmatrix} s & 0 & 0 \\ s'u & s' & 0 \\ s''v & s''w & s'' \end{pmatrix}$.

2. *For* $i = 1, 2, 3$, *set weights*

$$w_{i,1} := 3 \deg b_{i,1},$$

$$w_{i,2} := \begin{cases} 3 \deg b_{i,2} + \deg G + 2 \deg H & \text{if } b_{i,2} \neq 0 \\ 0 & \text{otherwise, and} \end{cases}$$

$$w_{i,3} := \begin{cases} 3 \deg b_{i,3} + 2 \deg G + \deg H & \text{if } b_{i,3} \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

3. *Set* $w_i := \max\{w_{i,1}, w_{i,2}, w_{i,3}\}$ *and* a_i *such that* $w_i = w_{i,a_i}$.

4. *Sort* $\{b_1, b_2, b_3\}$ *so that* $w_1 \leq w_2 \leq w_3$.

5. *While* $a_1 = a_2$, $a_1 = a_3$, *or* $a_2 = a_3$:

   a. *If* $a_1 = a_2$, *then*

      - *Find* $k, r \in \mathbb{F}_q[x]$ *such that* $b_{2,a_2} = b_{1,a_1} k + r$.
      - *Set* $b_2 := b_2 - k b_1$.
      - *Set the weights* $w_{2,1}$, $w_{2,2}$, *and* $w_{2,3}$ *as in Step 2.*
      - *Set* $w_2 := \max\{w_{2,1}, w_{2,2}, w_{2,3}\}$ *and* a_2 *such that* $w_2 = w_{2,a_2}$.

   b. *If* $a_1 = a_3$, *then*

      - *Find* $k, r \in \mathbb{F}_q[x]$ *such that* $b_{3,a_3} = b_{1,a_1} k + r$.
      - *Set* $b_3 := b_3 - k b_1$.
      - *Set the weights* $w_{3,1}$, $w_{3,2}$, *and* $w_{3,3}$ *as in Step 2.*
      - *Set* $w_3 := \max\{w_{3,1}, w_{3,2}, w_{3,3}\}$ *and* a_3 *such that* $w_3 = w_{3,a_3}$.

   c. *If* $a_2 = a_3$, *then*

      - *Find* $k, r \in \mathbb{F}_q[x]$ *such that* $b_{3,a_3} = b_{2,a_2} k + r$.
      - *Set* $b_3 := b_3 - k b_2$.
      - *Set the weights* $w_{3,1}$, $w_{3,2}$, *and* $w_{3,3}$ *as in Step 2.*
      - *Set* $w_3 := \max\{w_{3,1}, w_{3,2}, w_{3,3}\}$ *and* a_3 *such that* $w_3 = w_{3,a_3}$.

   d. *Sort* $\{b_1, b_2, b_3\}$ *and the associated weights so that* $w_1 \leq w_2 \leq w_3$.

6. *Output* $\alpha := b_{1,1} + b_{1,2}\rho + b_{1,3}\omega$.

4.5.2 Canonical Basis Construction

The second step in the procedure to compute a distinguished ideal requires the computation of the minimal canonical basis of the principal ideal generated by $\alpha = a + b\rho + c\omega$, produced by Algorithm 4.5.3. Here, we give the algorithm from [Bau05] and justify that it produces the intended output. Very briefly, we will begin with a matrix whose rows represent the coefficients of the elements α, $\alpha\rho$, and $\alpha\omega$ of $\langle\alpha\rangle$, and whose columns correspond with the $\mathbb{F}_q[x]$-basis elements of $\mathcal{O}$, namely 1, ρ, and ω. We will then reduce the matrix to one of the form given by B in Step 1 of Algorithm 4.5.3. From this, we can quickly obtain the non-primitive factor and the canonical basis elements. This algorithm also generalizes Algorithm 9.1 of [Bau04] to include the case in which the curve defining K is singular.

Algorithm 4.5.4 (Algorithm 0.27 of [Bau05]) Canonical Basis Algorithm
Input: *q; monic, relatively prime, and square-free polynomials G, $H \in \mathbb{F}_q[x]$ such that $3 \nmid \deg(GH^2)$; and an element $\alpha = a + b\rho + c\omega \in \mathbb{F}_q(C)$, where $C : Y^3 = GH^2$.*
Output: *$\langle \alpha \rangle = \langle d \rangle [s, s'(u+\rho), s''(v+w\rho+\omega)]$, in terms of a minimal canonical basis, where the ideal $[s, s'(u+\rho), s''(v+w\rho+\omega)]$ is primitive.*

1. *Row reduce* $\begin{pmatrix} a & b & c \\ cGH & a & bH \\ bGH & cG & a \end{pmatrix}$ *to* $\begin{pmatrix} a_3 & 0 & 0 \\ a_2 & b_2 & 0 \\ a_1 & b_1 & c_1 \end{pmatrix}$.
2. *Set $d := \gcd(b_2, c_1)$, $s := a_3/d$, $s' := b_2/d$, $s'' := c_1/d$, and $u :\equiv (a_2/b_2) \pmod{s/s'}$*
3. *Compute $r_1, r_2 \in \mathbb{F}_q[x]$ such that $r_1 s'' + r_2 s/s'' = 1$.*
4. *Compute $k, w \in \mathbb{F}_q[x]$ such that $\deg w < \deg s'$ and $r_1 b_1/d = ks' + w$.*
5. *Compute $v :\equiv r_1 a_1/d - s'ku \pmod{s/s''}$.*
6. *Output $\langle d \rangle [s, s'(u+\rho), s''(v+w\rho+\omega)]$.*

If $H(x) = 1$, then we must have $s'' = 1$ in the output, so we may take $d = c_1$ in Step 2 and $r_1 = 1$ and $r_2 = 0$ in Step 3. We give a brief justification that Algorithm 4.5.4 produces the intended output.

Proposition 4.5.5 *Algorithm 4.5.4 produces a minimal canonical basis of $\langle \alpha \rangle$, for any element, α, of a purely cubic function field, $K = K_x$.*

Proof: Let d, s, s', s'', u, v, and w be defined as in the output of Algorithm 4.5.4. We prove that the algorithm computes these quantities correctly. The elements α, $\alpha\rho$, $\alpha\omega \in \langle \alpha \rangle$ form an $\mathbb{F}_q[x]$-basis of the ideal. Since the columns of the matrix represent the elements 1, ρ, ω, the row vectors of the first matrix in Step 1 of Algorithm 4.5.4 represent the elements α, $\alpha\rho$, and $\alpha\omega$. After the elementary row operations, the row vectors of the second matrix in Step 1 still represent a basis of the ideal $\langle \alpha \rangle$, specifically $\{a_3, a_2 + b_2\rho, a_1 + b_1\rho + c_1\omega\}$.

Since $\gcd(s', s'') = 1$, by (4.1), in Step 2, we must have $\langle \alpha \rangle = \langle d \rangle \mathfrak{a}$, where $\mathfrak{a}$ is primitive. If $\mathfrak{a} = [S, S'(U+\rho), S''(V+W\rho+\omega)]$, then $\langle \alpha \rangle = \langle d \rangle \mathfrak{a} = [dS, dS'(U+\rho), dS''(V+W\rho+\omega)]$. It immediately follows that $dS'' = c_1$. Further, by Part 1 of Lemma 4.1.6, we have $s'' = S'' = c_1/d$. Considering the third basis elements, there exist polynomials m, $n \in \mathbb{F}_q[x]$ such that $a_1 + b_1\rho + c_1\omega = mdS + ndS'(U+\rho) + dS''(V+W\rho+\omega)$. Comparing coefficients, we have $a_1 = mdS + ndS'U + dS''V$ and $b_1 = ndS' + dS''W$, so $d \mid a_1$ and $d \mid b_1$. Now considering the second basis elements, there exist polynomials m, $n \in \mathbb{F}_q[x]$ such that $a_2 + b_2\rho = mdS + ndS'(U+\rho)$. By a similar argument, we see that $s' = S' = b_2/d$ and $d \mid a_2$. Finally, Part 1 of Lemma 4.1.6 and $dS = a_3$ imply that $s = S = a_3/d$.

Now, the second basis element of $\mathfrak{a}$, $S'(U+\rho) = a_2/d + (b_2/d)\rho = a_2/d + s'\rho$ must be expressible as a linear combination of s and $s'(u+\rho)$, i.e. there exist polynomials m, $n \in \mathbb{F}_q[x]$ such that $a_2/d + s'\rho = ms + ns'(u+\rho)$. Comparing coefficients at ρ immediately yields $n = 1$, so $a_2/d = ms + s'u$. Since $s' \mid s$, by (4.1), s' divides the right-hand side of this equality, so it also divides the left-hand side, a_2/d. Equivalently, $b_2 = ds'$ divides a_2, and $u = a_2/b_2 - ms/s' \equiv a_2/b_2 \pmod{s/s'}$. Thus, $u = a_2/b_2 \pmod{s/s'}$, as desired.

By Lemma 4.2.12, $\gcd(s/s'', s'') = 1$, so there are polynomials $r_1, r_2 \in \mathbb{F}_q[x]$ such that $r_1 s'' + r_2 s/s'' = 1$. Since v is unique modulo s/s'', w is determined uniquely modulo s', and $s' \mid (s/s'')$, we have $w \equiv r_1 b_1/d \pmod{s'}$ and by Lemma 4.1.6, we have $v \equiv r_1 a_1/d + u(w - r_1 b_1/d) \pmod{s/s''}$, which are the polynomials computed in Steps 2 through 5. □

4.5.3 Computing a Distinguished Ideal

The third and final step in the procedure to compute the distinguished representative of an ideal class will be covered in this section. Here, we will prove that given an ideal $\mathfrak{a}$, the three-step procedure of finding an element, $\alpha \in \overline{\mathfrak{a}}$, whose norm has minimal degree in $\overline{\mathfrak{a}}$, constructing a canonical basis of $\langle \alpha \rangle$, and computing $\langle \alpha \rangle / \overline{\mathfrak{a}}$, produces the distinguished ideal equivalent to $\mathfrak{a}$. In Algorithm 4.5.7, we will outline these steps formally. We will also define the *composition* operation in an ideal class group.

The following lemma shows how to use the element α, given by Theorem 4.5.2, and its basis, found by Algorithm 4.5.4, to find the unique distinguished ideal in a given class. The proof may be found in Corollary 5.2 of [Bau04] and Lemma 8 of [GPS02]. In both sources, the authors assume that $K = \mathbb{F}_q(C)$, with C nonsingular. However, in [GPS02], the proof of Lemma 8 holds for any totally ramified function field in which every ideal has an element whose norm has minimal degree. In particular, this lemma holds for all totally ramified purely cubic function fields, by Theorem 4.5.2.

Lemma 4.5.6 (Lemma 8 of [GPS02]) *Let K be a totally ramified cubic function field and $\mathcal{O}$ its maximal order. If $\mathfrak{a}$ is an ideal of $\mathcal{O}$ and α is the element of $\mathfrak{a}$ whose norm has minimal degree, then $\langle \alpha \rangle / \mathfrak{a}$ is the unique distinguished ideal in the ideal class $[\overline{\mathfrak{a}}]$.*

Equivalently, to find the unique distinguished ideal equivalent to $\mathfrak{a}$, we find the element $\alpha \in \overline{\mathfrak{a}}$ whose degree has minimal norm so that $\mathfrak{b} = \langle \alpha \rangle / \overline{\mathfrak{a}}$ is the unique distinguished ideal in $[\mathfrak{a}]$. In this context, we will write $\mathfrak{b} = \mathit{Reduce}(\mathfrak{a})$. However, since $\mathfrak{a}\overline{\mathfrak{a}} = \langle L(\mathfrak{a}) \rangle$, we have $\mathfrak{b} = \langle \alpha / L(\mathfrak{a}) \rangle \mathfrak{a}$. If $\langle \alpha \rangle = \langle d_1 \rangle \mathfrak{a}_2$, where $\mathfrak{a}_2$ is primitive, then we may multiply $\mathfrak{a}_2 \mathfrak{a} = \langle d_2 \rangle \mathfrak{b}$, where $d_2 \in \mathbb{F}_q[x]$ is discarded.

Lemma 4.5.6 and this discussion give rise to the following algorithm, which computes the distinguished ideal equivalent to a given ideal. This generalizes Algorithm 10.1 of [Bau04] to include purely cubic function fields defined by any totally ramified curve $C : Y^3 = G(x)H^2(x)$.

Algorithm 4.5.7 (Algorithm 10.1 of [Bau04]) Ideal Reduction, Unit Rank 0
Input: *q; monic, relatively prime and square-free polynomials G, $H \in \mathbb{F}_q[x]$, such that $3 \nmid \deg(GH^2)$; and a primitive ideal, $\mathfrak{a} = [s, s'(u + \rho), s''(v + w\rho + \omega)]$ of $\mathcal{O}_x = \mathcal{O}(\mathbb{F}_q(C))$, with given minimal canonical basis, where $C : Y^3 = GH^2$.*
Output: *The distinguished ideal, $\mathfrak{b} = [S, S'(U + \rho), S''(V + W\rho + \omega)] = \mathit{Reduce}(\mathfrak{a}) \sim \mathfrak{a}$, in terms of a minimal canonical basis.*

1. *Compute $\overline{\mathfrak{a}}$ via Lemma 4.3.3.*
2. *Compute the minimal element $\alpha \in \overline{\mathfrak{a}}$ via Algorithm 4.5.3.*
3. *Compute the minimal canonical basis, $\langle d_1 \rangle \mathfrak{a}_2$, of $\langle \alpha \rangle$ via Algorithm 4.5.4.*

4. *Compute* $\mathfrak{a}_2\mathfrak{a} = \langle d_2 \rangle \, \mathfrak{b} := \langle d_2 \rangle \, [S, \, S'(U+\rho), \, S''(V+W\rho+\omega)]$ *via Algorithm 4.4.12.*

5. *Output* $\mathfrak{b}$.

There are two potential variants of this algorithm noted in Section 10 of [Bau04]. In most applications, we will need to reduce the product of two ideals. If it is known, for example, that the norms of the two ideal factors have large degrees, while the norm degrees of their primitive inverses are smaller, then it would be faster to first invert the ideals, multiply the result, and use that product as the input to Algorithm 4.5.7. Bauer also notes that it is possible to combine Steps 3 and 4 in order to speed up the computation.

Lastly, $\mathfrak{a} = Reduce(\mathfrak{a}_1\mathfrak{a}_2)$ is called the *composition* of the ideals, $\mathfrak{a}_1 \in \mathbf{C}_1$ and $\mathfrak{a}_2 \in \mathbf{C}_2$, where $\mathbf{C}_1$, $\mathbf{C}_2 \in Cl(\mathcal{O})$, and we write $\mathfrak{a} = \mathfrak{a}_1 * \mathfrak{a}_2$. The ideal $\mathfrak{a}$ is the unique distinguished ideal in the ideal class $\mathbf{C}_1\mathbf{C}_2$.

In this chapter, we generalized the results of [Sch01] and [Bau04] to describe ideal inverses and multiplication in any purely cubic function field of characteristic not equal to 3 and showed how to find the unique distinguished ideal equivalent to a given ideal for such totally ramified function fields. In the next chapter, we will apply this arithmetic to the definition and computation of the giant step in the infrastructures of a purely cubic function field of unit ranks 1 and 2.

Chapter 5

Ideal Arithmetic with Reduced Bases

In Chapter 3, we defined the infrastructure of a cubic function field of positive unit rank and the distance measure on its divisors and discussed its structure. In this chapter, we will define arithmetic operations on the infrastructure and apply these operations to lay the theoretical foundation for the computation of the regulator and a system of fundamental units of a cubic function field, K, of positive unit rank. In Section 5.1, we will introduce the notion of a reduced basis of a fractional ideal. We will use such a basis to define and compute the first infrastructure operation, the baby step, on the divisor associated with the given fractional ideal. In Section 5.2, we will describe how to use reduced bases to reduce a fractional ideal to an equivalent distinguished fractional ideal by multiplying by a certain principal ideal, $\langle \psi^{-1} \rangle$, where $\psi \in K^*$ such that $\left|\deg\left(\psi^{(i)}\right)\right|$ is small for each $i = 0, 1, 2$. Furthermore, the description of ideal reduction in unit rank 2 infrastructures is new. We then apply these results to define and compute the infrastructure operations in Section 5.3, namely the baby step, giant step, and inverse operations. Using the divisor-theoretic description of the infrastructure of a cubic function field, we will present improved bounds on the length of a baby step and the number of reduction steps required to compute a giant step. We will also show how to compute a divisor close to or equal to a given distance in the unit rank 1 case. Moreover, the giant step and inverse operations will utilize the ideal arithmetic we developed in Chapter 4. Finally, we apply the baby step operation in particular to give results on computing the regulator and a system of fundamental units of a cubic function field of positive unit rank and also to describe new observations on the symmetry exhibited by the infrastructure of a purely cubic function field of unit rank 2 in Section 5.4.

As in previous chapters, $K = \mathbb{F}_q(C)$, with $C : Y^3 = GH^2$ and G, $H \in \mathbb{F}_q[x]$ monic, relatively prime, and square-free, will denote a purely cubic function field of unit rank $r > 0$ and genus g, and $\mathcal{O} = \mathcal{O}_x$ will denote its maximal order. We choose an $\mathbb{F}_q(x)$-basis, $\{1, \rho, \omega\}$, of K, where ρ is a fixed cube root of GH^2, and $\omega = \rho^2/H$. (Since we assume that GH^2 is monic, we may choose ρ so that $\operatorname{sgn}(\rho) = 1$, in which case $\operatorname{sgn}(\omega) = 1$ as well.) Since the definition of a reduced basis presented here is not valid in characteristic 2 or 3, we will assume that $\operatorname{char}(K) \geq 5$ in this chapter.

To highlight some relevant background material, much of the foundational work in cubic function fields was developed in the context of cubic number fields, based on Minkowski's geometry of numbers. In the function field case, this intuition is carried over into Mahler's geometry of Puiseux series [Mah41]. In many cases, we will develop a divisor theoretic intuition of certain concepts, but in some cases, it will be easier to use the language of ideals, particularly for defining baby steps, giant steps, and developing ideal and ideal basis reduction algorithms. The basis for many of the results in this chapter is found in the work of Voronoi [Vor94, Vor96], Delone and Fadeev [DF64],

and Buchmann, Williams, et al. [WZ72, WDS83, Buc85, WW87, BW88] for cubic number fields. This work was then applied to purely cubic function fields of positive unit rank by Scheidler, Stein, et al. [Sch00, SS00, Sch01, LSY03, Sch04].

5.1 Reduced Bases in Unit Ranks 1 and 2

This section defines the notion of a reduced basis of a fractional ideal, $\mathfrak{f}$, of $\mathcal{O}$ and states some important properties of these bases. We will then outline an algorithm to compute the reduced basis of any fractional ideal. These bases will provide a convenient means to perform arithmetic in the infrastructure of an ideal class. More specifically, one of the elements of the reduced basis of a distinguished fractional ideal will be used to apply the baby step operation to the corresponding infrastructure divisor. In addition, we will use these bases to find a distinguished fractional ideal from an equivalent fractional ideal for use in the giant step operation. Therefore, it will be to our advantage to supplement the divisor theory of infrastructure in this section with appropriate results on ideals.

To motivate the importance of reduced bases, we consider principal fractional ideals. By Lemma 3.3.8, an element, $\theta \in K^*$, is a minimum in $\mathcal{O}$ if and only if $\left\langle \theta^{-1} \right\rangle$ is a distinguished fractional ideal. However, θ can become as large as a unit of $\mathcal{O}$, so it will be computationally inefficient, and for larger function fields, infeasible, to represent a fractional ideal, $\mathfrak{f} = \left\langle \theta^{-1} \right\rangle$, by a single generating element. By comparison, elements of a reduced basis have very small degree when considered as elements in the completion, $K_{\infty_0} = \mathbb{F}_q \left\langle x^{-1} \right\rangle$, of K at ∞_0, hence the choice of the term.

The notions in this section were initially developed for cubic number fields by Voronoi [Vor94, Vor96] and were extended to purely cubic function fields of positive unit rank by Scheidler, Stein, et al. [Sch00, SS00, Sch01, LSY03, Sch04]. As such, we will forego any function field or divisor theoretic motivation in this section. However, the motivation from cubic number fields is found in work by Delone and Fadeev [DF64] and Williams et al. [WZ72, WCS80, WDS83].

5.1.1 Notation and Properties

Following the approach in [Sch00, SS00, Sch01, LSY03, Sch04], for any $\alpha = a + b\rho + c\omega \in K$, let:

$$\begin{array}{lll} \xi_\alpha & = b\rho + c\omega & = \frac{1}{3}(2\alpha - \alpha' - \alpha'') \ , \\ \eta_\alpha & = b\rho - c\omega & = \frac{1}{2\iota+1}(\alpha' - \alpha'') \ , \\ \zeta_\alpha & = 2a - b\rho - c\omega & = \alpha' + \alpha'' \ , \end{array}$$

where ι is a primitive cube root of 1 in $\overline{\mathbb{F}}_q$. If f, $g \in \mathbb{F}_q(x)$ and α, $\beta \in K^*$, then $\xi_{f\alpha+g\beta} = f\xi_\alpha + g\xi_\beta$, $\eta_{f\alpha+g\beta} = f\eta_\alpha + g\eta_\beta$, and $\zeta_{f\alpha+g\beta} = f\zeta_\alpha + g\zeta_\beta$. Also, for any $i = 0, 1, 2$, we have $\xi_{\alpha^{(i)}} = \xi_\alpha^{(i)}$, $\eta_{\alpha^{(i)}} = \eta_\alpha^{(i)}$, and $\zeta_{\alpha^{(i)}} = \zeta_\alpha^{(i)}$.

The following definition of a(n) (i-)reduced basis is the same as that for the unit rank 1 case in [Sch00] and [Sch01], but differs slightly from the definitions given in Algorithm 7.1 of [SS00] and Section 4 of [LSY03] for the unit rank 1 and 2 cases, respectively. However, we will show that if $\mathfrak{f}$ is distinguished, then a(n) (i-)reduced basis, as defined in [SS00] and [LSY03], is a(n) (i-)reduced basis as defined here and in [Sch00] and [Sch01].

Definition 5.1.1 *Let* $\mathfrak{f}$ *be a fractional ideal of* $\mathcal{O}$ *and* $\{1, \mu, \nu\}$ *an* $\mathbb{F}_q[x]$*-basis of* $\mathfrak{f}$*. A basis,* $\{1, \mu, \nu\}$*, is called* i-reduced *(or simply* reduced*, if* $r = 1$*) if*

$$\deg\left(\xi_\nu^{(i)}\right) < \deg\left(\xi_\mu^{(i)}\right) \ , \quad \deg\left(\eta_\mu^{(i)}\right) < 0 \le \deg\left(\eta_\nu^{(i)}\right) \ , \quad \deg\left(\zeta_\mu^{(i)}\right) < 0 \ , \quad \deg\left(\zeta_\nu^{(i)}\right) \le 0 \ ,$$

$$\textit{and if} \quad \deg\left(\eta_\nu^{(i)}\right) = 0 \ , \quad \textit{then} \quad \deg\left(\nu^{(i)}\right) \neq 0 \ .$$

If a fractional ideal is distinguished, then we may be more restrictive on the conditions needed for its basis to be (i-)reduced. The following lemma was given as the definition for a(n) (i-)reduced basis for distinguished fractional ideals in [SS00] for the unit rank 1 case, and in [LSY03] for the unit rank 2 case.

Lemma 5.1.2 *If* $\mathfrak{f}$ *is a distinguished fractional ideal, with a(n) (i-)reduced basis,* $\{1, \mu, \nu\}$*, and* $\tilde{\nu} = \nu - \lfloor \zeta_{\nu^{(i)}} \rfloor / 2$*, then* $\{1, \mu, \tilde{\nu}\}$ *is also a(n) (i-)reduced basis that satisfies* $\deg\left(\xi_{\tilde{\nu}}^{(i)}\right) < \deg\left(\xi_\mu^{(i)}\right)$*,* $\deg\left(\eta_\mu^{(i)}\right) < 0 \le \deg\left(\eta_{\tilde{\nu}}^{(i)}\right)$*,* $\deg\left(\zeta_\mu^{(i)}\right) < 0$*, and* $\deg\left(\zeta_{\tilde{\nu}}^{(i)}\right) < 0$*.*

Proof: The equality $\deg\left(\eta_\nu^{(i)}\right) = 0$, together with $\deg\left(\zeta_\nu^{(i)}\right) \le 0$, implies $\deg\left(\nu^{(i+1)}\right) \le 0$ and $\deg\left(\nu^{(i+2)}\right) \le 0$, where the superscripts are considered modulo 3. Since $\mathfrak{f}$ is distinguished, we must have $\deg\left(\nu^{(i)}\right) > 0$, so $\{1, \mu, \tilde{\nu}\}$ is a(n) (i-)reduced basis.

Since $\zeta_{\lfloor \zeta_{\nu^{(i)}} \rfloor} = 2\lfloor \zeta_{\nu^{(i)}} \rfloor$, we have $\deg\left(\zeta_{\tilde{\nu}}^{(i)}\right) = \deg\left(\zeta_{\nu^{(i)}} - \lfloor \zeta_{\nu^{(i)}} \rfloor\right) < 0$. Moreover, $\eta_\nu^{(i)} = \eta_{\tilde{\nu}}^{(i)}$ and $\xi_\nu^{(i)} = \xi_{\tilde{\nu}}^{(i)}$. Therefore, $\deg\left(\zeta_{\tilde{\nu}}^{(i)}\right) < 0$ and the basis $\{1, \mu, \tilde{\nu}\}$ satisfies the desired conditions. □

The following theorem establishes an important property of (i-)reduced bases for distinguished fractional ideals; such bases are unique up to constant factors.

Theorem 5.1.3 (Theorem 4.4 of [LSY03]) *Let* $\mathcal{O} = \mathcal{O}_x$ *be the maximal order of a purely cubic function field* $K = K_x$ *of positive unit rank* r *and* $\mathfrak{f}$ *and* $\mathfrak{g}$ *distinguished fractional ideals of* $\mathcal{O}$*. Then* $\mathfrak{f} = \mathfrak{g}$ *if and only if* $\mathfrak{f}$ *and* $\mathfrak{g}$ *have the same (i-)reduced bases (up to constant factors), as given by Lemma 5.1.2, for any* $i \in \{0, 1, 2\}$ *(or* $i = 0$ *if* $r = 1$*).*

If $\mathfrak{f} = \{1, \mu, \nu\}$ is a distinguished fractional ideal, with $\mu = m_0 + m_1\rho + m_2\omega$ and $\nu = n_0 + n_1\rho + n_2\omega$, then we will identify $\mathfrak{f}$ uniquely via its (i-)reduced basis, as given by Lemma 5.1.2, such that the leading coefficients, m_0 and n_0, of μ and ν are monic.

As noted, the terms reduced and i-reduced imply that the elements of a reduced basis are small. The following theorem applies to fractional ideals in fields of unit rank 1 or 2, and establishes an upper bound on the size of the basis elements.

Proposition 5.1.4 (Proposition 5.1 of [Sch00], and Proposition 4.5 of [LSY03]) *Let* $\mathcal{O} = \mathcal{O}_x$ *be the maximal order of a purely cubic function field* $K = K_x$ *of positive unit rank* r*,* $i \in \{0, 1, 2\}$ *(or* $i = 0$ *if* $r = 1$*), and* $\{1, \mu, \nu\}$ *a(n) (i-)reduced basis of a distinguished fractional ideal,* $\mathfrak{f}$*, of* $\mathcal{O}$*, as given by Lemma 5.1.2, where* $\mu = (m_0 + m_1\rho + m_2\omega)/d$ *and* $\nu = (n_0 + n_1\rho + n_2\omega)/d$ *with* $m_0, m_1, m_2, n_0, n_1, n_2, d \in \mathbb{F}_q[x]$ *and* $\gcd(m_0, m_1, m_2, n_0, n_1, n_2, d) = 1$*. Then*

1. $\lfloor m_0/d \rfloor = \lfloor m_1\rho/d \rfloor = \lfloor m_2\omega/d \rfloor = \lfloor \mu \rfloor / 3$,
2. $\deg\left(\nu^{(i)}\right) < \deg\left(\mu^{(i)}\right) \le \deg(\Delta(\mathfrak{f}))/2 = \deg(N(\mathfrak{f})) + g + 2 \le g + 2$,

3. $\deg\left(m_0^{(i)}\right) = \deg\left((m_1\rho)^{(i)}\right) = \deg\left((m_2\omega)^{(i)}\right) \leq \deg(\Delta)/2 = g+2$, *and*

4. $\deg\left(n_0^{(i)}\right) = \deg\left((n_1\rho)^{(i)}\right) = \deg\left((n_2\omega)^{(i)}\right) < \deg(\Delta)/2 = g+2$.

Having defined the notion of a reduced basis, the following section will show how to compute a(n) (i-)reduced basis of any fractional ideal.

5.1.2 Computing Reduced Bases

The following algorithm produces a(n) (i-)reduced basis from a non-reduced one. If $\mathcal{O}$ has unit rank 1, then we replace $\deg\left(\cdot^{(i)}\right)$ with $\deg\left(\cdot\right)$, since we assume $i = 0$ if $r = 1$. The unit rank 1 version of this algorithm is Algorithm 4.1 of [Sch00], Algorithm 7.1 of [SS00], Algorithm 6.3 of [Sch01], and Algorithm 6.2 of [Sch04]. Algorithm 4.6 of [LSY03] applies the same procedure on distinguished fractional ideals in the unit rank 2 case. The algorithm presented here simply combines the steps of the algorithms in the given sources. We will take as input any fractional ideal, $\mathfrak{f}$, of $\mathcal{O}$, of the form $\mathfrak{f} = [1, \mu, \nu]$, so this algorithm will differ from some of the sources because we do not assume that $\mathfrak{f}$ is distinguished. We will note those steps that may be omitted if $\mathfrak{f}$ is indeed distinguished. We will use this algorithm to compute the (i-)reduced basis of the product of two fractional ideals, since this product is not necessarily distinguished.

Algorithm 5.1.5 (Algorithm 4.1 of [Sch00]) Ideal Basis Reduction
Input: *A basis,* $\{1, \tilde{\mu}, \tilde{\nu}\}$*, of a fractional ideal,* $\mathfrak{f}$*, of* $\mathcal{O}$*, and* $i \in \{0, 1, 2\}$ *(*$i = 0$ *if* $r = 1$*).*
Output: *A(n) (i-)reduced basis,* $\{1, \mu, \nu\}$*, of* $\mathfrak{f}$.

1. *Set* $\mu := \tilde{\mu}$ *and* $\nu := \tilde{\nu}$.

2. *If* $\deg\left(\xi_\mu^{(i)}\right) < \deg\left(\xi_\nu^{(i)}\right)$ *or if* $\deg\left(\xi_\mu^{(i)}\right) = \deg\left(\xi_\nu^{(i)}\right)$ *and* $\deg\left(\eta_\mu^{(i)}\right) < \deg\left(\eta_\nu^{(i)}\right)$, *set*

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} := \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix} .$$

3. *If* $\deg\left(\eta_\mu^{(i)}\right) \geq \deg\left(\eta_\nu^{(i)}\right)$, *then*

   a. *While* $\deg\left((\xi_\nu\eta_\nu)^{(i)}\right) > \deg(\Delta(\mathfrak{f}))/2$, *set*

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} := \begin{pmatrix} 0 & 1 \\ -1 & \lfloor \xi_{\mu^{(i)}}/\xi_{\nu^{(i)}} \rfloor \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix} .$$

   b. *Set*

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} := \begin{pmatrix} 0 & 1 \\ -1 & \lfloor \xi_{\mu^{(i)}}/\xi_{\nu^{(i)}} \rfloor \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix} .$$

   c. *If* $\deg\left(\eta_\mu^{(i)}\right) = \deg\left(\eta_\nu^{(i)}\right)$, *set*

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} := \begin{pmatrix} 1 & -a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix} ,$$

where $a = \operatorname{sgn}(\eta_{\mu^{(i)}}) \operatorname{sgn}(\eta_{\mu^{(i)}}^{-1})$.

4. *a. If* $\mathfrak{f}$ *is not distinguished, then while* $\deg\left(\eta_{\nu}^{(i)}\right) < 0$, *set*

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} := \begin{pmatrix} 0 & 1 \\ -1 & \lfloor \xi_{\mu^{(i)}} / \xi_{\nu^{(i)}} \rfloor \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix} .$$

b. While $\deg\left(\eta_{\mu}^{(i)}\right) \geq 0$, *set*

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} := \begin{pmatrix} \lfloor \eta_{\nu^{(i)}} / \eta_{\mu^{(i)}} \rfloor & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix} .$$

5. *Set* $\mu := \mu - \lfloor \zeta_{\mu^{(i)}} \rfloor / 2$ *and* $\nu := \nu - \lfloor \zeta_{\nu^{(i)}} \rfloor / 2$.

6. *If* $\mathfrak{f}$ *is not distinguished and* $\deg\left(\nu^{(i)}\right) = \deg\left(\eta_{\nu}^{(i)}\right) = 0$, *then set* $\nu := \nu - \left\lfloor \nu^{(i)} \right\rfloor$.

7. *Output* $\{1, \mu, \nu\}$.

If $\mathfrak{f}$ is distinguished, then Steps 4.a and 6 may be omitted since they will not be called. The justification of Algorithm 5.1.5 follows quickly from the proofs of the corresponding algorithms in [Sch01] and [LSY03].

Theorem 5.1.6 (Thm. 6.4 of [Sch01] for $r = 1$ and Prop. 4.8 of [LSY03] for $r = 2$) *Algorithm 5.1.5 terminates and computes a reduced basis of the input ideal.*

Proof: The proof for the unit rank 1 case is given in Theorem 6.4 of [Sch01]. In the unit rank 2 case, if the input ideal, $\mathfrak{f}$, is distinguished, then the result holds by Proposition 4.8 of [LSY03]. If $\mathfrak{f}$ is not distinguished, then we must include Steps 4.a and 6. The inclusion of Step 4.a for non-distinguished input will be justified in Corollary 5.2.2, but is the generalization of the corresponding step in the unit rank 1 case. Likewise, the inclusion of Step 6 is justified precisely in the same way as in the proof of Theorem 6.4 of [Sch01] for the unit rank 1 case. □

5.2 Ideal Reduction in Unit Ranks 1 and 2

In this section, we will show how to reduce an arbitrary fractional ideal to an equivalent distinguished fractional ideal. The results of this section can be found in [Sch01] for the unit rank 1 case, but the analogous results for the unit rank 2 case are new. The techniques to derive the results, however, are identical to those in [Sch01]. We will first prove necessary and sufficient conditions to test whether or not a fractional ideal, given by a(n) (i-)reduced basis, is distinguished and then show how to compute this ideal in theory. Next, we will outline these steps in an algorithm, prove its correctness, and establish improved upper bounds on the number of required iterations given the type of fractional ideals that we input. Equivalently, this procedure will show how to find a distinguished divisor from any non-distinguished divisor.

5.2.1 Properties of Distinguished Fractional Ideals

The following theorem gives simple criteria for quickly determining whether or not a fractional ideal is distinguished. This theorem was proved in the unit rank 1 case in Theorem 6.6 of [Sch01] and we adapt the proof here for the unit rank 2 case.

Theorem 5.2.1 (Theorem 6.6 of [Sch01] for $r = 1$) *Let $K = K_x$ be a purely cubic function field of positive unit rank. If $\{1, \mu, \nu\}$ is a (0-)reduced basis of a fractional ideal $\mathfrak{f}$, of $\mathcal{O}_x$, then $\mathfrak{f} = [1, \mu, \nu]$ is distinguished if and only if $\max(\deg(\nu), \deg(\eta_\nu)) > 0$ and $\deg(\mu) > 0$.*

Proof: First suppose that $\mathfrak{f}$ is distinguished. Then $\mathcal{N}_\mathfrak{f}(1) = \mathbb{F}_q$, $\deg(\eta_\mu) < 0$, and $\deg(\zeta_\mu) < 0$. Since $\eta_\mu = (\mu' - \mu'')/(2\iota + 1)$ and $\zeta_\mu = \mu' + \mu''$, we have $\deg\left(((2\iota + 1)\eta_\mu + \zeta_\mu)/2\right) = \deg(\mu') < 0$. Likewise, $\deg(\mu'') < 0$. So we must have $\deg(\mu) > 0$, otherwise $\mu \in \mathcal{N}_\mathfrak{f}(1) = \mathbb{F}_q$. Similarly, if $\max\{\deg(\nu), \deg(\eta_\nu)\} \leq 0$, then since $\deg(\zeta_\nu) \leq 0$, we obtain $\deg(\nu), \deg(\nu'), \deg(\nu'') \leq 0$. This implies that $\nu \in \mathcal{N}_\mathfrak{f}(1)$, so $\nu \in \mathbb{F}_q^*$. However, this implies that $\deg(\nu) = \deg(\eta_\nu) = 0$, contradicting the fact that $\{1, \mu, \nu\}$ is a (0-)reduced basis. Therefore, we have $\max(\deg(\nu), \deg(\eta_\nu)) > 0$.

Conversely, suppose that $\deg(\mu) > 0$ and $\max(\deg(\nu), \deg(\eta_\nu)) > 0$. Let $\theta = l + m\mu + n\nu \in \mathcal{N}_\mathfrak{f}(1)$, where $l, m, n \in \mathbb{F}_q[x]$. We will show that $m = n = 0$ and $l \in \mathbb{F}_q$. Since $\theta \in \mathcal{N}_\mathfrak{f}(1)$, we have $\deg\left(\theta^{(j)}\right) \leq 0$ for $j = 0, 1, 2$. Therefore, $\deg(\zeta_\theta), \deg(\eta_\theta) \leq 0$, and since $\theta = (3\xi_\theta + \zeta_\theta)/2$, we have $\deg(\xi_\theta) \leq 0$.

Suppose first that $\deg(m) < \deg(n)$. Since $\{1, \mu, \nu\}$ is a (0-)reduced basis, we have $\deg(m\eta_\mu) < \deg(n\eta_\nu)$, so $0 \leq \deg(n) \leq \deg(n\eta_\nu) = \deg(m\eta_\mu + n\eta_\nu) = \deg(\eta_{m\mu+n\nu}) = \deg(\eta_{l+m\mu+n\nu}) = \deg(\eta_\theta) \leq 0$. So $\deg(n) = \deg(\eta_\nu) = 0$, which implies that $\deg(\nu) > 0$, and $\deg(m) < \deg(n) = 0$, and thus, $m = 0$. Now $\deg(l) = \deg(2l) = \deg(\zeta_{\theta - n\nu}) = \deg(\zeta_\theta - n\zeta_\nu) \leq 0$. Next, we have $0 = \deg(n) < \deg(n\nu) = \deg(\theta - l) \leq 0$, which is a contradiction. So $\deg(m) \geq \deg(n)$.

Suppose that $m \neq 0$. Then we have $\deg(n\xi_\nu) < \deg(m\xi_\mu)$, so $0 \leq \deg(m) < \deg(m\xi_\mu) = \deg(n\xi_\nu + m\xi_\mu) = \deg(\xi_{n\nu+m\mu}) = \deg(\xi_\theta) \leq 0$, another contradiction. Therefore, $m = n = 0$, and $\theta = l$. Since $\deg(\theta) \leq 0$, we have $\deg(l) \leq 0$, so $l = \theta \in \mathbb{F}_q$. Therefore, $\mathfrak{f}$ is distinguished. □

As an immediate consequence, we have the following corollary. The unit rank 1 result of the following is Corollary 6.7 in [Sch01].

Corollary 5.2.2 (Corollary 6.7 of [Sch01] for $r = 1$) *If $\{1, \mu, \nu\}$ is a (0-)reduced basis of a fractional ideal $\mathfrak{f}$, then $\mathfrak{f}$ is non-distinguished if and only if $\deg(\mu) \leq 0$ or $\deg(\nu) < \deg(\eta_\nu) = 0$.*

Given a non-distinguished fractional ideal, $\mathfrak{f}$, with given (0-)reduced basis, $\{1, \mu, \nu\}$, the following process finds a distinguished fractional ideal equivalent to $\mathfrak{f}$. This procedure is described in the unit rank 1 case in Section 4 of [Sch00] and in Section 6 of [Sch01], and the corresponding result for the unit rank 2 case is identical. To begin, let $\mathfrak{f} = \mathfrak{f}_0$ be a non-distinguished fractional ideal. We define a sequence $(\mathfrak{f}_n)_{n \in \mathbb{N}_0}$ of fractional ideals, and corresponding (0-)reduced bases, $\{1, \mu_n, \nu_n\}$, of $\mathfrak{f}_n$, as follows:

$$\mathfrak{f}_n = \left\langle \phi_n^{-1} \right\rangle \mathfrak{f}_{n-1} \ , \quad \text{where} \quad \phi_n = \begin{cases} \mu_{n-1} & \text{if } \deg(\mu_{n-1}) \leq 0 \\ \nu_{n-1} & \text{if } \deg(\mu_{n-1}) > 0 \end{cases} \quad \text{for } n \in \mathbb{N} \ . \tag{5.1}$$

We can condense this recurrence into

$$\mathfrak{f}_n = \left\langle \psi_n^{-1} \right\rangle \mathfrak{f} \ , \quad \text{where} \quad \psi_n = \prod_{i=1}^{n} \phi_i \text{ for } n \in \mathbb{N} \ . \tag{5.2}$$

In this way, each $\mathfrak{f}_n$ is divided by an element of of non-positive degree. The sequence, $(\mathfrak{f}_n)_{n\in\mathbb{N}_0}$, was shown to eventually terminate with a distinguished fractional ideal if $\mathcal{O}$ has unit rank 1 in [Sch00] and [Sch01], and we will prove this for the unit rank 2 case as well. As shown in [Sch00], the second case in (5.1) occurs at most once in the unit rank 1 case, and if so, then the ideal reduction procedure terminates. The following proposition shows that this condition holds in the unit rank 2 case as well.

Proposition 5.2.3 (Proposition 4.3 of [Sch00] for $r = 1$) *Let $\{1, \mu, \nu\}$ be a (0-)reduced basis of a non-distinguished fractional ideal $\mathfrak{f}$. If $\deg(\mu) > 0$, then $\mathfrak{g} = \left\langle \nu^{-1} \right\rangle \mathfrak{f}$ is distinguished with a (0-)reduced basis $\left\{1, \mu\nu^{-1}, \nu^{-1}\right\}$. By replacing ν^{-1} with $\nu^{-1} - \lfloor \zeta_{\nu^{-1}} \rfloor/2$, the basis of $\mathfrak{g}$ is in the form given by Lemma 5.1.2.*

Proof: Let $\tilde{\mu} = \mu\nu^{-1}$ and $\tilde{\nu} = \nu^{-1}$ so that $\mathfrak{g} = [1, \tilde{\mu}, \tilde{\nu}]$. We will first show that $\mathfrak{g}$ is distinguished. Since $\mathfrak{f}$ is not distinguished and $\deg(\mu) > 0$, we have $\deg(\nu) < \deg(\eta_\nu) = 0$ by Corollary 5.2.2, so $\max\{\deg(\tilde{\nu}), \deg(\eta_{\tilde{\nu}})\} \geq \deg(\tilde{\nu}) > 0$. Furthermore, $\deg(\tilde{\mu}) = \deg(\mu) - \deg(\nu) > \deg(\mu) > 0$. Therefore, $\mathfrak{g}$ is distinguished, by Theorem 5.2.1.

We now show that $\{1, \tilde{\mu}, \tilde{\nu}\}$ is a (0-)reduced basis of $\mathfrak{g}$. Since $\{1, \mu, \nu\}$ is a (0-)reduced basis of $\mathfrak{f}$, we also have $\deg(\zeta_\nu) \leq 0$, which combined with $\deg(\eta_\nu) = 0$, implies $\deg(\nu') = \deg(\nu'') = 0$. Likewise, $\deg(\zeta_\mu), \deg(\eta_\mu) < 0$ implies $\deg(\mu'), \deg(\mu'') < 0$. Therefore, $\deg(\tilde{\mu}') = \deg(\mu') - \deg(\nu') < 0$ and $\deg(\tilde{\mu}'') = \deg(\mu'') - \deg(\nu'') < 0$, so $\deg(\eta_{\tilde{\mu}}), \deg(\zeta_{\tilde{\mu}}) < 0$. Also, since $\deg(\nu') = \deg(\nu'') = 0$, we have $\deg(\tilde{\nu}') = \deg(\tilde{\nu}'') = 0$, so $\deg(\zeta_{\tilde{\nu}}) \leq 0$.

Next we will show that $\deg(\xi_{\tilde{\mu}}) > \deg(\xi_{\tilde{\nu}})$. Since $\deg(\tilde{\mu}) > 0$ and $\deg(\tilde{\mu}'), \deg(\tilde{\mu}'') < 0$, we have $\deg(\xi_{\tilde{\mu}}) = \deg(\tilde{\mu}) = \deg(\mu) - \deg(\nu) > \deg(\nu^{-1}) = \deg(\tilde{\nu})$. Now $\deg(\tilde{\nu}) > 0$ and $\deg(\tilde{\nu}') = \deg(\tilde{\nu}'') = 0$, so $\deg(\xi_{\tilde{\nu}}) = \deg(\tilde{\nu}) < \deg(\xi_{\tilde{\mu}})$.

Lastly, we prove that $\deg(\eta_{\tilde{\nu}}) \geq 0$. We have $\eta_{\tilde{\nu}} = (\tilde{\nu}' - \tilde{\nu}'')/(2\iota + 1) = \left(\left(\nu^{-1}\right)' - \left(\nu^{-1}\right)''\right)/(2\iota + 1) = (\nu'' - \nu')(\nu'\nu'')^{-1}/(2\iota + 1) = -\eta_\nu(\nu'\nu'')^{-1}$. Therefore, $\deg(\eta_{\tilde{\nu}}) = \deg(\eta_\nu) - \deg(\nu'\nu'') = \deg(\eta_\nu) \geq 0$. Therefore, $\{1, \tilde{\mu}, \tilde{\nu}\}$ is a (0-)reduced basis.

If $\deg(\zeta_{\tilde{\nu}}) = 0$, then by replacing $\tilde{\nu} := \tilde{\nu} - \lfloor \zeta_{\tilde{\nu}} \rfloor/2$, the resulting basis is in the form given by Lemma 5.1.2. □

5.2.2 Computing Distinguished Fractional Ideals

The procedure in (5.1) gives rise to the following algorithm to compute a distinguished fractional ideal. This algorithm was first given for the case that $r = 1$ in Algorithm 6.11 of [Sch01], but it generalizes to the unit rank 2 case. After stating the algorithm, we will prove its correctness and analyze its running time.

Algorithm 5.2.4 (Algorithm 6.11 of [Sch01] for $r = 1$) Ideal Reduction
Input: *A (0-)reduced basis, $\{1, \tilde{\mu}, \tilde{\nu}\}$, of a fractional ideal $\mathfrak{f}$.*
Output: *A (0-)reduced basis, $\{1, \mu, \nu\}$, of a distinguished fractional ideal equivalent to $\mathfrak{f}$.*

1. *Set* $\mu := \tilde{\mu}$ *and* $\nu := \tilde{\nu}$.

2. *While* $\deg(\mu) \leq 0$,

   a. *Set* $\tilde{\mu} := \mu^{-1}$ *and* $\tilde{\nu} := \nu\mu^{-1}$.

   b. *Compute a (0-)reduced basis,* $\{1, \mu, \nu\}$, *from the basis* $\{1, \tilde{\mu}, \tilde{\nu}\}$ *using Algorithm 5.1.5.*

3. *If* $\deg(\nu) < \deg(\eta_\nu) = 0$*:*

   a. *Set* $\mu := \mu\nu^{-1}$ *and* $\nu := \nu^{-1}$.

   b. *If* $\deg(\zeta_\nu) = 0$, *set* $\nu := \nu - \operatorname{sgn}(\zeta_\nu)$.

4. *Output* $\{1, \mu, \nu\}$.

If $\mathfrak{f} = [1, \tilde{\mu}, \tilde{\nu}]$ is the fractional ideal whose given (0-)reduced basis is the input to the ideal reduction algorithm above, and $\mathfrak{g} = [1, \mu, \nu]$ is the distinguished fractional ideal whose (0-)reduced basis is the corresponding output, then we will write $\mathfrak{g} = \mathit{Reduce}(\mathfrak{f})$. If $D = -\Psi^{-1}(\mathfrak{f})$ and $E = -\Psi^{-1}(\mathfrak{g})$, then we write $E = \mathit{Reduce}(D)$. For integral ideals, if $\mathfrak{a} = \Psi(D) = \mathfrak{f}^{-1}$, and $\mathfrak{b} = \Psi(E) = \mathfrak{g}^{-1}$ is a distinguished ideal, then we write $\mathfrak{b} = \mathit{Reduce}(\mathfrak{a})$. If $\mathfrak{g} = \langle \psi^{-1} \rangle \mathfrak{f}$, for some element $\psi \in K^*$, then $\mathfrak{b} = \langle \psi \rangle \mathfrak{a}$.

In order to prove that Algorithm 5.2.4 terminates and produces the intended output, we need the following lemma. This lemma was originally proved for the unit rank 1 case in [Sch01], but the proof is independent of unit rank, and is provided in the given source.

Lemma 5.2.5 (Lemma 6.1 of [Sch01]) *If* $\mathfrak{f}$ *is a fractional ideal containing* 1, *then* $|\mathcal{N}_\mathfrak{f}(1)| < \infty$.

The correctness of Algorithm 5.2.4 was first proved in Lemma 6.9 of [Sch01] for the unit rank 1 case, and the proof generalizes naturally to the unit rank 2 case.

Lemma 5.2.6 (Lemma 6.9 of [Sch01] for $r = 1$**)** *Algorithm 5.2.4 terminates after a finite number of steps, producing a distinguished fractional ideal,* $\mathfrak{g}$, *with a (0-)reduced basis. Specifically, if* $\mathfrak{f} = \mathfrak{f}_0$ *is a non-distinguished fractional ideal, then there exists* $m \in \mathbb{N}$ *such that* $\mathfrak{g} = \mathfrak{f}_m$ *is distinguished, where* $\mathfrak{f}_m$ *is given by the recurrence in* (5.1) *and* (5.2).

Proof: Let $\mathfrak{f}_i = [1, \mu_i, \nu_i]$ be a non-distinguished fractional ideal, with the given reduced basis. By Corollary 5.2.2, we have $\deg(\mu_i) \leq 0$ or $\deg(\nu_i) < \deg(\eta_{\nu_i}) = 0$. If $\deg(\mu_i) \leq 0$, then $\phi_i = \mu_i$, by (5.1). Since the basis of $\mathfrak{f}_i$ is reduced, we have $\deg(\eta_{\mu_i})$, $\deg(\zeta_{\mu_i}) < 0$, so $\deg(\mu_i')$, $\deg(\mu_i'') < 0$. Thus, $\mu_i = \phi_i \in \mathcal{N}_\mathfrak{f}(1)$.

Now suppose that $\deg(\mu_i) > 0$. Then $\deg(\nu_i) < \deg(\eta_{\nu_i}) = 0$, by Corollary 5.2.2, and $\phi_i = \nu_i$, by (5.1). Again, since the basis of $\mathfrak{f}_i$ is reduced, we have $\deg(\zeta_{\nu_i}) \leq 0$, so $\deg(\nu_i')$, $\deg(\nu'') \leq 0$. Thus, $\nu_i = \phi_i \in \mathcal{N}_\mathfrak{f}(1)$. Since $\psi_n = \prod_{i=1}^n \phi_i$, we have $\deg(\psi_i) \leq \deg(\psi_{i-1})$, $\deg(\psi_i') \leq \deg(\psi_{i-1}')$, and $\deg(\psi_i'') \leq \deg(\psi_{i-1}'')$, for all $1 \leq i \leq n$, and at least one of the inequalities is strict, so all the ψ_i are distinct. Also, $\deg(\psi_i)$, $\deg(\psi_i')$, $\deg(\psi_i'') \leq 0$, so $\psi_i \in \mathcal{N}_\mathfrak{f}(1)$ for all $1 \leq i \leq n$. If, for each $i \in \mathbb{N}_0$, $\mathfrak{f}_i = \langle \psi_i^{-1} \rangle \mathfrak{f}$ is not distinguished, then $(\psi_i)_{i \in \mathbb{N}_0}$ is an infinite sequence of distinct elements in $\mathcal{N}_\mathfrak{f}(i)$, contradicting Lemma 5.2.5. Thus, $\mathfrak{g} = \mathfrak{f}_m \in (\mathfrak{f}_i)_{i \in \mathbb{N}_0}$ is distinguished for some $m \in \mathbb{N}_0$.

If Step 3 of Algorithm 5.2.4 is not entered, then the basis of $\mathfrak{g}$ is (0-)reduced by construction. Otherwise, the basis is (0-)reduced by Proposition 5.2.3. □

The following theorem establishes an upper bound on the number of steps, m, required by Algorithm 5.2.4 to compute a distinguished fractional ideal, $\mathfrak{f}_m$, from an equivalent non-distinguished fractional ideal, $\mathfrak{f}_0$. We follow the method for the proof of Theorem 6.10 of [Sch01], which is the corresponding result for the unit rank 1 case, but we provide slightly better bounds.

Theorem 5.2.7 *Let $(\mathfrak{f}_i)_{i\in\mathbb{N}_0}$ be a sequence of fractional ideals and $m \in \mathbb{N}_0$, such that $\mathfrak{f}_m$ is distinguished and $\mathfrak{f}_n$ is not distinguished, for $n < m$, where $\mathfrak{f}_n$ is given by* (5.1)*. Then*

$$m \leq \frac{1}{2}\left(1 - \deg(N(\mathfrak{f}_0)) + \deg(N(\mathfrak{f}_m))\right) \leq \frac{1}{2}\left(1 - \deg(N(\mathfrak{f}_0))\right) \ .$$

Proof: If $\mathfrak{f}_0$ is distinguished, then $m = 0$ and we are done. Suppose that $\mathfrak{f}_0$ is not distinguished. Set $d_n = \deg(N(\mathfrak{f}_n))$. If $\mathfrak{f}_n = \langle \phi_n^{-1} \rangle \mathfrak{f}_{n-1}$ is given by a (0-)reduced basis, $\{1, \mu_n, \nu_n\}$, then by Proposition 5.2.3, we have $\phi_n = \mu_n$ for $1 \leq n \leq m-1$. Since $\deg(\mu_n) \leq 0$ and the basis of $\mathfrak{f}_i$ is (0-)reduced, we have $\deg(\eta_{\mu_i})$, $\deg(\zeta_{\mu_i}) < 0$, so $\deg(\mu_i')$, $\deg(\mu_i'') < 0$. Thus, $d_n \geq d_{n-1} + 2$ for $1 \leq n \leq m-1$.

If $\mathfrak{f}_m = \langle \phi_m^{-1} \rangle \mathfrak{f}_{m-1}$, then either $\phi_m = \mu_m$ or $\phi_m = \nu_m$, by (5.1). If $\phi_m = \mu_m$, then $\deg(N(\phi_m)) = \deg(N(\mu_m)) \leq -2$, as noted above. If $\phi_m = \nu_m$, then $\deg(\phi_m) = \deg(\nu_m) < \deg(\eta_{\nu_m}) = 0$, by Corollary 5.2.2, and since the basis of $\mathfrak{f}$ is (0-)reduced, we have $\deg(\zeta_{\nu_i}) \leq 0$, so $\deg\left(\phi_m^{(i)}\right) = \deg\left(\nu_m^{(i)}\right) \leq 0$, for $i = 1, 2$. In either case, we have $d_m \geq d_{m-1} + 1$. Inductively, we have $d_m \geq d_0 + 2(m-1) + 1 = d_0 + 2m - 1$. Therefore, $m \leq (1 + d_m - d_0)/2 = (1 + \deg(N(\mathfrak{f}_m)) - \deg(N(\mathfrak{f}_0)))/2 \leq (1 - \deg(\mathfrak{f}_0))/2$. □

In practice, the most common application of ideal reduction will be to reduce the product of two distinguished fractional ideals. In this case, we may be more specific about the number of reduction steps required to reduce a non-distinguished fractional ideal. This result improves on Proposition 4.4.3 of [Sch00].

Theorem 5.2.8 *Let $\mathfrak{f}_0$ be the product of two distinguished fractional ideals. If $\mathfrak{f}_m$ is distinguished, and $\mathfrak{f}_n$ is not distinguished, for $n < m$, where $\mathfrak{f}_n$ is given by* (5.1)*, then $m \leq g + (1 + \deg(N(\mathfrak{f}_m)))/2$. In particular, we have $m \leq g$.*

Proof: By Theorem 5.2.7, we have $m \leq (1 + \deg(N(\mathfrak{f}_m)) - \deg(N(\mathfrak{f}_0)))/2$. Since $\mathfrak{f}_0$ is the product of two distinguished fractional ideals, we have $-2g \leq \deg(N(\mathfrak{f}_0)) \leq 0$, by Proposition 3.4.3. Therefore, $m \leq (1 + \deg(N(\mathfrak{f}_m)) + 2g)/2 = g + (1 + \deg(N(\mathfrak{f}_m)))/2$. Since $\deg(N(\mathfrak{f}_m)) \leq 0$, we have $m \leq g + 1/2$, so we in fact have $m \leq g$. □

Under reasonable assumptions, we can be even more specific. If the norms of the distinguished ideals of $\mathcal{O}$ are uniformly distributed in $\{f(x) \in \mathbb{F}_q[x] \mid \deg(f(x)) \leq g\}$, then we first claim that $\deg(N(\mathfrak{a})) = g$, and hence $\deg(N(\mathfrak{f})) = -g$, with probability $1 - 1/q$, where $\mathfrak{a}$ is distinguished and $\mathfrak{f} = \mathfrak{a}^{-1}$. There are q^{g+1} polynomials in $\mathbb{F}_q[x]$ of degree at most g and $(q-1)q^g$ of degree g (that is, a nonzero x^g term), so a random polynomial of degree at most g has degree g with probability $(q-1)q^g/q^{g+1} = 1 - 1/q$. The claim then follows.[1] Therefore, for large q, we will have $\deg(N(\mathfrak{f}_0)) = -2g$ and $\deg(N(\mathfrak{f}_m)) = -g$ with high probability. In this case, we will have $m \leq (g+1)/2$.

The overall complexity of Algorithm 5.2.4 follows from Theorems 5.2.7 and 5.2.8.

[1] Based on experimental evidence, as stated in Section 6.6 of [Fon09], the actual probability appears to be $1 - 1/q + O\left(q^{-g/2}\right)$.

Corollary 5.2.9 *If the input to Algorithm 5.2.4 is an arbitrary fractional ideal,* $\mathfrak{f}$*, then Algorithm 5.2.4 requires at most* $1 - \deg(N(\mathfrak{f}))$ *inversions and* $(1 - \deg(N(\mathfrak{f})))/2$ *multiplications in* $\mathbb{F}_q\langle x^{-1}\rangle$ *and at most* $(1 - \deg(N(\mathfrak{f}_0)))/2$ *basis reductions. If the input is the product of two distinguished fractional ideals, then Algorithm 5.2.4 requires at most* $2g$ *inversions and* g *multiplications in* $\mathbb{F}_q\langle x^{-1}\rangle$ *and at most* g *basis reductions.*

In this section, we gave conditions on a (0-)reduced basis to determine whether or not a fractional ideal defined by that basis is distinguished. Moreover, these conditions are very easy to check. We also outlined an algorithm to produce a distinguished fractional ideal from an equivalent non-distinguished fractional ideal in a finite, usually very small, number of steps. In the next section, we will apply the techniques of basis and ideal reduction to define arithmetic on the infrastructures of a purely cubic function field.

5.3 Infrastructure Arithmetic in Unit Rank 1 and 2

This section discusses four operations on the infrastructures of a purely cubic function field: baby steps, giant steps, inverses, and what will be referred to as the divisor below a given $y \in \mathbb{N}_0$. Recall our discussion on the structure of infrastructures in Section 3.4.4. The baby step operation will correspond with finding a distinguished divisor adjacent to a given divisor when the associated minimal representatives (given by Theorem 3.3.18) are considered as discrete points on the circle or torus determined by $\mathcal{D}_0^S/\mathcal{P}^S$ in the unit rank 1 and 2 cases, respectively. The giant step, or composition, operation will be defined as the reduction of the sum of two divisors. Next, the inverse operation will only be defined for divisors, $D \in \mathcal{R}$, in the principal infrastructure, and will be given by *Reduce* $\left(\overline{D}\right)$, where $\overline{D} = \Psi^{-1}\left(\overline{\Psi(D)}\right)$. Finally, the divisor below y will only be defined for unit rank 1 infrastructures, and is the divisor, $D \in \mathcal{R}$ whose distance is maximal under the restriction $0 \leq \delta(D) \leq y < R_x$. The motivation for defining these operations will be to provide means to compute the regulator and a system of fundamental units of a purely cubic function field of positive unit rank. We will first define the notion of a neighbor, which will serve to define and also compute a baby step. Then we will define each of the aforementioned operations, giving algorithms for their computation in each case.

5.3.1 Neighbors

For any fractional ideal, $\mathfrak{f}$, of $\mathcal{O}$ and minimum $\theta \in \mathfrak{f}$, the notion of a(n) (i-)neighbor, $\phi \in \mathfrak{f}$, of θ will be defined and we will state results from [SS00] and [LSY03] that ϕ is a minimum in $\mathfrak{f}$ as well. Further, this notion will be extended to the corresponding distinguished divisors with the intent to define the baby step operation in terms of (i-)neighbors. Results from [SS00] and [LSY03] will then show how to compute this (i-)neighbor given an (i-)reduced basis of $\mathfrak{f}$.

To motivate the definition of a(n) (i-)neighbor, we consider the intent of the baby step operation, first in the unit rank 1 case, and then its generalization to the unit rank 2 setting. Given a divisor $D \in \mathcal{R}$ in the infrastructure of a cubic function field of unit rank 1, we wish a baby step to yield the unique divisor whose distance is minimal over all divisors in $\mathcal{R}$ having distance greater than $\delta(D)$. That is, we wish the baby step operation to find the adjacent distinguished divisor on the circle modulo R_x, with the divisors ordered by distance. By the definition of distinguished, there must

also be a strict decrease in the 1-component of distance. In the unit rank 2 setting, baby steps in three directions will be defined: the 0-, 1-, and 2-directions. The desired effect on distance for a baby step in the 0-direction, for example, will be a minimal increase in the 0-component of distance with no increase in the other two components. By the definition of distinguished, however, we will require a strict decrease in either the 1- or 2-component of distance. In this way, we wish a baby step from a divisor $D \in \mathcal{R}$ to yield a divisor that is close to it, it terms of distance, when considered as discrete points on the torus modulo Λ, the set of coordinate vectors of $\mathcal{E} = \mathcal{E}(\mathcal{O})$.

To realize these concepts, we must look to the minima corresponding with the principal distinguished divisors, since we obtain the distance measure on infrastructure divisors from appropriate functions in K^*. The following set and subsequent ordering arise from the discussion above and will be used to produce a(n) (i-)neighbor of a minimum of a fractional ideal. For reference, this set is the same set which was used in the proof of Theorem 5.1 of [SS00] for unit rank 1 infrastructures, and defined in Section 3 of [LSY03] for unit rank 2 infrastructures.

Definition 5.3.1 *Let $\mathfrak{f}$ be a fractional ideal of $\mathcal{O}$ and $\theta \in \mathfrak{f}$, and denote*

$$\mathcal{H}_{\mathfrak{f},i}(\theta) = \Big\{\alpha \in \mathfrak{f} \mid \deg\left(\alpha^{(i)}\right) > \deg\left(\theta^{(i)}\right), \deg\left(\alpha^{(j)}\right) \leq \deg\left(\theta^{(j)}\right) \text{ for all } j \neq i, \\ \text{and } \deg\left(\alpha^{(j)}\right) < \deg\left(\theta^{(j)}\right) \text{ for at least one } j \neq i\Big\} \quad . \tag{5.3}$$

If $r = 1$, then we will assume that $i = 0$ and write

$$\mathcal{H}_{\mathfrak{f}}(\theta) = \mathcal{H}_{\mathfrak{f},0}(\theta) = \{\alpha \in \mathfrak{f} \mid \deg(\alpha) > \deg(\theta), \deg(\alpha') < \deg(\theta')\} \quad .$$

The following ordering on functions in K^* will be used to pick out a minimal element in $\mathcal{H}_{\mathfrak{f},i}(\theta)$, for any $i = 0, 1, 2$. (See Section 3 of [LSY03].) Let $\alpha, \beta \in K^*$. If $\mathcal{O}$ has unit rank 1, write $\alpha \leq_0 \beta$ if the ordered pair $(\deg(\alpha), \deg(\alpha'))$ appears before $(\deg(\beta), \deg(\beta'))$ in lexicographical order. If $\mathcal{O}$ has unit rank 2, we write $\alpha \leq_i \beta$ if the ordered triple $\left(\deg\left(\alpha^{(i)}\right), \deg\left(\alpha^{(i+1)}\right), \deg\left(\alpha^{(i+2)}\right)\right)$ appears before $\left(\deg\left(\beta^{(i)}\right), \deg\left(\beta^{(i+1)}\right), \deg\left(\beta^{(i+2)}\right)\right)$ in lexicographical order, where the superscripts are considered modulo 3.

The following theorem guarantees the existence and uniqueness (up to a factor in $\mathbb{F}_q^*$) of the minimal element of $\mathcal{H}_{\mathfrak{f},i}(\theta)$ under the ordering $\leq_i$, and is proved in the given sources.

Theorem 5.3.2 (Theorem 5.1 of [SS00] and Theorem 3.4 of [LSY03]) *Let $\mathcal{O} = \mathcal{O}_x$ be the maximal order of a purely cubic function field, $K = K_x$, $\mathfrak{f}$ a distinguished fractional ideal of $\mathcal{O}$, and θ a minimum in $\mathfrak{f}$. For any $i \in \{0, 1, 2\}$, there exists an element $\phi \in \mathcal{H}_{\mathfrak{f},i}(\theta)$, unique up to a factor in $\mathbb{F}_q^*$, such that $\phi \leq_i \alpha$ for all $\alpha \in \mathcal{H}_{\mathfrak{f},i}(\theta)$. Furthermore, ϕ is a minimum in $\mathfrak{f}$.*

Ignoring constant factors, the minimal element, ϕ, of $\mathcal{H}_{\mathfrak{f},i}(\theta)$, identified in Theorem 5.3.2, is called the *(i-)neighbor* of θ, and is denoted by $\phi_{\mathfrak{f},i}(\theta)$. In the unit rank 1 case, ϕ is alternatively called the *minimum adjacent to* θ, keeping the cyclic behavior of unit rank 1 infrastructures in mind [SS00]. In terms of infrastructure divisors, if $\mathfrak{f} = \mathcal{O}$ and $D = \Psi^{-1}(\langle\theta\rangle) \in \mathcal{R}$, then we call $\Psi^{-1}(\langle\phi\rangle) \in \mathcal{R}$ the *(i-)neighbor* of (or, in the unit rank 1 case, the *divisor adjacent to*) D. If $\mathfrak{f} \in \mathbf{C}$, for some ideal class $\mathbf{C} \in Cl(\mathcal{O})$, and $\phi = \phi_{\mathfrak{f},i}(1)$, then we call $-\Psi^{-1}\left(\left\langle\phi^{-1}\right\rangle \mathfrak{f}\right) \in \mathcal{R}_{\mathbf{C}}$ the *(i-)neighbor* of (or, in the unit rank 1 case, the *divisor adjacent to*) $-\Psi^{-1}(\mathfrak{f}) \in \mathcal{R}_{\mathbf{C}}$. (Recall that $\left\langle\phi^{-1}\right\rangle \mathfrak{f}$ is a distinguished

fractional ideal by Lemma 3.3.8.) To show that this definition is well-defined, we have the following result from [SS00] and [LSY03].

Lemma 5.3.3 (Proposition 5.3 of [SS00] and Lemma 3.5 of [LSY03]) *If θ is a minimum in a distinguished fractional ideal, $\mathfrak{f}$, of $\mathcal{O}$, then $\theta\phi_{\langle\theta^{-1}\rangle\mathfrak{f},i}(1) = \phi_{\mathfrak{f},i}(\theta)$, for any $i \in \{0, 1, 2\}$ ($i = 0$ if $r = 1$).*

The i-neighbor of 1 in a distinguished fractional ideal, $\mathfrak{f}$, is easily determined from the (i-)reduced basis of $\mathfrak{f}$. The following two theorems give the results for the unit rank 1 and 2 cases, respectively, and will be used to compute a baby step. The proofs may be found in the given sources.

Theorem 5.3.4 (Theorem 7.5 of [SS00]) *If K has unit rank 1 and $\{1, \mu, \nu\}$ is a reduced basis of a distinguished fractional ideal, $\mathfrak{f}$, of $\mathcal{O}$, then μ is the neighbor of 1 in $\mathfrak{f}$.*

Theorem 5.3.5 (Theorem 4.2 of [LSY03]) *If K has unit rank 2, $i \in \{0, 1, 2\}$, and $\{1, \mu, \nu\}$ is an i-reduced basis of a distinguished fractional ideal, $\mathfrak{f}$, of $\mathcal{O}$, as given by Lemma 5.1.2, then the i-neighbor of 1 in $\mathfrak{f}$ is*

$$\phi_{\mathfrak{f},i}(1) = \begin{cases} \nu - \operatorname{sgn}\left(\nu^{(i+1)}\right) & \text{if } \deg\left(\nu^{(i+1)}\right) = 0 \\ \mu & \text{if } \deg\left(\nu^{(i+1)}\right) > 0 \end{cases}.$$

Combining this theorem and the definition of a neighbor with the following lemma, we have an important corollary.

Lemma 5.3.6 (Lemma 4.1 of [LSY03]) *Let $i \in \{0, 1, 2\}$ and $\beta \in K^*$ with $\deg\left(\beta^{(i)}\right) > 0$, $\deg\left(\beta^{(i+1)}\right) = 0$, and $\deg\left(\zeta_\beta^{(i)}\right) < 0$. If $\alpha = \beta - \operatorname{sgn}\left(\beta^{(i+1)}\right)$, then $\deg\left(\alpha^{(i)}\right) > 0$, $\deg\left(\alpha^{(i+1)}\right) < 0$, and $\deg\left(\alpha^{(i+2)}\right) = 0$.*

Corollary 5.3.7 (Corollary 4.3 of [LSY03]) *Let $\mathfrak{f}$ be a distinguished fractional ideal of $\mathcal{O}$. If $r = 1$ and $\phi = \phi_{\mathfrak{f}}(1)$, then $\deg(\phi) > 0$ and $\deg(\phi') = \deg(\phi'') < 0$. If $r = 2$, $0 \le i \le 2$, and $\phi = \phi_{\mathfrak{f},i}(1)$, then $\deg\left(\phi^{(i)}\right) > 0$, $\deg\left(\phi^{(i+1)}\right) < 0$, and $\deg\left(\phi^{(i+2)}\right) \le 0$, where the superscripts are considered modulo 3.*

It follows from this corollary that the distance of the (i-)neighbor of a divisor has the properties we intended for the baby step operation, which we will now formally define.

5.3.2 Baby Steps

In this section, we will define the first of the four infrastructure operations, the baby step operation. This operation arises naturally from Theorem 5.3.2 and the definition of a(n) (i-)neighbor. We will then give results on the length of a baby step, in terms of distance, including an important heuristic on this length in the case that the degree of the effective part of the input and output are both g. Lastly, we use Theorems 5.3.4 and 5.3.5 to compute a baby step.

Definition 5.3.8 *If $D \in \mathcal{R}_{\mathbf{C}}$ and $E \in \mathcal{R}_{\mathbf{C}}$ is the (i-)neighbor of D, for some $i \in \{0, 1, 2\}$ ($i = 0$ if $r = 1$), then the operation $D \mapsto E$ is called a* baby step (in the i-direction)*, and we write $bs_i(D) = E$ (or simply $bs(D) = E$ if $r = 1$).*

We have the following about the relative distance between a divisor and its (i-)neighbor.

Lemma 5.3.9 *If $D \in \mathcal{R}_{\mathbf{C}}$ and $bs_i(D) = E$, for some $i \in \{0, 1, 2\}$ ($i = 0$ if $r = 1$), then $\delta_D(E) = \deg(\phi)$ if $r = 1$ and $\delta_D(E) = (\deg(\phi), \deg(\phi'), \deg(\phi''))$ if $r = 2$, where $\phi = \phi_{\mathfrak{f},i}(1)$ and $\mathfrak{f} = \Psi(-D)$.*

Proof: Let $\mathfrak{g} = \Psi(-E)$. By definition, we have $\mathfrak{g} = \left\langle \phi^{-1} \right\rangle \mathfrak{f}$ and hence $\Psi(E) = \langle \phi \rangle \, \Psi(D)$. The result then follows from Lemma 3.4.2. □

Since the element ϕ in Lemma 5.3.9 is the minimal element of $\mathcal{H}_{\mathfrak{f},i}(1)$ under the ordering $\leq_i$, the i-component of the relative distance $\delta_D(E)$ will be small. This explains the choice of the term "baby step."

In the following discussion, we will determine the relative distance between i-neighbors and the difference between the associated minimal representatives in $\mathcal{D}_0^S$. If $E = bs_i(D)$, for some $D \in \mathcal{R}_{\mathbf{C}}$ and $i = 0, 1, 2$, then there is some $\phi \in K^*$ such that $\Psi(E) = \langle \phi \rangle \, \Psi(D)$. Let D_∞, $E_\infty \in \mathcal{D}_0^S$ be the minimal representatives of $[D_\infty]$, $[E_\infty] \in \mathcal{D}_0^S / \mathcal{P}^S$ corresponding with D and E, respectively. By Theorem 3.3.18, we have $\operatorname{div}(\phi) = \operatorname{div}(\phi)_S - \deg(N(\phi))\infty_i + A_\infty$, where $A_\infty \equiv E_\infty - D_\infty \pmod{\mathcal{E}(\mathcal{O})}$. Note that $\deg(N(\phi)) = \deg(\operatorname{div}(\phi)_S)$. Combining this with Corollary 5.3.7 and Lemma 5.3.9 yields the following identities.

$$\operatorname{div}(\phi)^S = \deg\left(\operatorname{div}(\phi)_S\right) \infty_0 + a\left(\infty_1 - 2\infty_0\right) = -\delta_D(E)\infty_0 + a\infty_1 \ , \tag{5.4}$$

where $a = \deg(\operatorname{div}(\phi)_S) + \delta_D(E) > 0$ in the unit rank 1 case and

$$\begin{aligned} \operatorname{div}(\phi)^S &= \deg\left(\operatorname{div}(\phi)_S\right) \infty_i + a(\infty_{i+1} - \infty_i) + b(\infty_{i+2} - \infty_i) \\ &= -\left(\delta_{D,i}(E)\infty_i + \delta_{D,i+1}(E)\infty_{i+1} + \delta_{D,i+2}(E)\infty_{i+2}\right) \ , \end{aligned} \tag{5.5}$$

where $a = -\delta_{D,i+1}(E) > 0$, $b = -\delta_{D,i+2}(E) \geq 0$, $\delta_{D,i}(E)$ is the i-component of the relative distance $\delta_D(E)$, and the subscripts are considered modulo 3.

We use these equations to establish upper and lower bounds on the length of baby steps in $\mathcal{R}_{\mathbf{C}}$. This result improves on previous baby step bounds given in Proposition 5.1 of [Sch00].

Theorem 5.3.10 *Let $K = K_x$ be a purely cubic function field of unit rank $r > 0$ and genus g, $\mathcal{O}$ its maximal order, $D \in \mathcal{R}_{\mathbf{C}}$, for some $\mathbf{C} \in Cl(\mathcal{O})$, and $E = bs_i(D)$, for some $i \in \{0, 1, 2\}$ ($i = 0$ if $r = 1$). If $r = 1$, then*

1. *$1 \leq \delta_D(E) \leq g + 2 - \deg(D_S) \leq g + 2$; and*
2. *if $\deg(D_S) \leq \deg(E_S) + 1$, then $\deg(E_S) - \deg(D_S) + 2 \leq \delta_D(E)$.*

If $r = 2$, then

3. *$1 \leq \delta_{D,i}(E) \leq g + 1 - \deg(D_S) \leq g + 1$; and*
4. *if $\deg(D_S) \leq \deg(E_S)$, then $\deg(E_S) - \deg(D_S) + 1 \leq \delta_{D,i}(E)$,*

where $\delta_{D,i}(E)$ is the i-component of the relative distance $\delta_D(E)$.

Proof: In either unit rank 1 or 2, we must have $0 < \delta_{D,i}(E)$, which establishes the lower bound of Parts 1 and 3. For the remaining cases in both unit ranks, we will first establish some notation.

If $r = 1$, then let $d = 2$ and $i = 0$, and if $r = 2$, then let $d = 1$. If $\mathfrak{f} = \Psi_i(-D)$, then there is an element $\phi = \phi_{\mathfrak{f},i}(1) \in K^*$, unique up to a factor in $\mathbb{F}_q^*$, such that $\Psi_i(E) = \langle \phi \rangle \Psi_i(D)$. Thus, $\mathrm{div}(\phi)_S = E_S - D_S$. We will choose an i-distinguished divisor, B, near D and define $\theta \in K^*$ such that $\Psi(B) = \langle \theta \rangle \Psi(D)$. We will show that $\theta \in \mathcal{H}_{\mathfrak{f},i}(1)$, invoking the fact that D is distinguished, so that $\phi \leq_i \theta$. We will then use information about $\deg\left(\phi^{(i)}\right)$ and $\deg\left(\theta^{(i)}\right)$ in terms of B, D, and E to establish the desired upper bounds on $\delta_{D,i}(E)$ in Parts 1 and 3. Parts 2 and 4 will be a straightforward consequence of statements made near the beginning of the argument. We consider both the unit rank 1 and 2 cases together.

By Lemma 5.3.9, we have $\delta_D(E) = \deg(\phi)$ if $r = 1$, and $\delta_D(E) = (\deg(\phi), \deg(\phi'), \deg(\phi''))$ if $r = 2$. If $r = 1$, then by (5.4), we have $\mathrm{div}(\phi) = E_S - D_S - \deg(E_S - D_S)\infty_0 + a(\infty_1 - 2\infty_0)$, for some integer $a > 0$, and if $r = 2$, then by (5.5), we have $\mathrm{div}(\phi) = E_S - D_S - \deg(E_S - D_S)\infty_i + a(\infty_{i+1} - \infty_i) + b(\infty_{i+2} - \infty_i)$, for integers $a > 0$ and $b \geq 0$. (For the remainder of the proof, all superscripts and subscripts will be considered modulo 3.) If $r = 1$, then let $b = 0$. It follows that $\delta_{D,i}(E) = \deg\left(\phi^{(i)}\right) = \deg(E_S) - \deg(D_S) + da + b$. To establish Parts 2 and 4, if $\deg(D_S) \leq \deg(E_S) + d - 1$, then $1 \leq \deg(E_S) - \deg(D_S) + d \leq \deg(E_S) - \deg(D_S) + da + b = \deg\left(\phi^{(i)}\right) = \delta_{D,i}(E)$, as desired.

To prove the upper bounds on $\delta_{D,i}(E)$ in both Parts 1 and 3, we will show that $\deg(E_S) + da + b \leq g + d$. Let B be the i-distinguished divisor whose corresponding infinite divisor in $\mathcal{D}_0^S$, under Theorem 3.3.18, is $B_\infty \equiv D_\infty + (\infty_{i+1} - d\infty_i) \pmod{\mathcal{E}}$, where D_∞ is the minimal representative of $[D_\infty]$ corresponding to D, under Theorem 3.3.18, and $\mathcal{E} = \{\mathrm{div}(\eta_1), \mathrm{div}(\eta_r)\}$ is as defined in Section 2.5.3. Also let $\theta \in K^*$ such that $\Psi(B) = \langle \theta \rangle \Psi(D)$. We first show that $\theta \in \mathcal{H}_{\mathfrak{f},i}(1)$. Notice that $\mathrm{div}(\theta)_S = B_S - D_S \geq -D_S = \Phi^{-1}(\mathfrak{f})$, so $\theta \in \mathfrak{f}$. In addition, $\deg(N(\theta)) = \deg(\mathrm{div}(\theta)_S) = \deg(B_S - D_S)$ and $B_\infty - D_\infty \equiv \infty_{i+1} - d\infty_i \pmod{\mathcal{E}}$, so by Theorem 3.3.18 we have $\mathrm{div}(\theta)^S = -\deg(B_S - D_S)\infty_i + (\infty_{i+1} - d\infty_i)$. Thus, $\deg\left(\theta^{(i)}\right) = \deg(B_S) - \deg(D_S) + d$ and $\deg\left(\theta^{(i+1)}\right) = -1$, and if $r = 2$, then $\deg\left(\theta^{(i+2)}\right) = 0$ as well. Therefore, it suffices to show that $\deg\left(\theta^{(i)}\right) > 0$. To this end, suppose that $\deg\left(\theta^{(i)}\right) \leq 0$. Then $\mathrm{div}(\theta)^S > 0$ and $\deg(B_S) + d \leq \deg(D_S)$. If $A = D + \mathrm{div}(\theta)$, then $A \sim D$, $A_S = B_S \geq 0$, $\deg(A_S) = \deg(B_S) < \deg(B_S) + d \leq \deg(D_S)$, and $A^S = D^S + \mathrm{div}(\theta)^S > D^S$, contradicting the fact that D is distinguished. Thus, $\deg\left(\theta^{(i)}\right) > 0$ and $\theta \in \mathcal{H}_{\mathfrak{f},i}(1)$.

By the minimality of ϕ under $\leq_i$, we have $\phi \leq_i \theta$. Therefore, $\deg\left(\phi^{(i)}\right) \leq \deg\left(\theta^{(i)}\right)$, so $\deg(E_S) - \deg(D_S) + da + b \leq \deg(B_S) - \deg(D_S) + d$ and hence, $\deg(E_S) + da + b \leq \deg(B_S) + d$. Since B is i-distinguished, it is reduced, by Theorem 3.3.15, so $\deg(B_S) \leq g$, whence it follows that $\deg(E_S) + da + b \leq g + d$. Therefore, $\delta_{D,i}(E) = \deg\left(\phi^{(i)}\right) = \deg(E_S) - \deg(D_S) + da + b \leq g + d - \deg(D_S) \leq g + d$, since $-\deg(D_S) \leq 0$. Subsituting $d = 2$ if $r = 1$ and $d = 1$ if $r = 2$ establishes the upper bounds on $\delta_{D,i}(E)$ in Parts 1 and 3, respectively. □

We briefly remark that this result explains the existence of holes, or "distance gaps" around some infrastructure divisors. If $D \in \mathcal{R}_{\mathbf{C}}$, with $\deg(D_S) < g$, then certain distances are impossible for infrastructure divisors around D. More specifically, if D_∞ is the minimal representative of $[D_\infty]$ corresponding with D, then the 0-distinguished divisors (such as B in the proof above), corresponding to the minimal representatives whose classes are near $[D_\infty]$ on $\mathcal{D}_0^S/\mathcal{P}^S$, are not always distinguished. In particular, since $1 \leq \deg(bs(0)_S)$, Theorem 5.3.10 implies that there is no $D \in \mathcal{R}$ such that $\delta(D) = 1$ or $\delta(D) = 2$ in unit rank 1 infrastructures, and in unit rank 2 infrastructures, there is a similar gap around 0.

If $bs_i(0) = D$ and $\deg(D_S) = g$, then Theorem 5.3.10 states that $\delta(D) = g + 2$ in the unit rank 1 case and $\delta_i(D) = g + 1$ in the unit rank 2 case. This is analogous to the corresponding situation in the infrastructure of a (unit rank 1) hyperelliptic function field,[2] in which the first baby step results in a divisor whose distance is $g + 1$. Moreover, if for any $D \in \mathcal{R}_{\mathbf{C}}$ such that $\deg(D_S) = g$ and $\deg(bs_i(D)_S) = g$, then we can be more specific about the relative distance $\delta_D(bs_i(D))$.

Corollary 5.3.11 *If $D \in \mathcal{R}_{\mathbf{C}}$, for some $\mathbf{C} \in Cl(\mathcal{O})$, $E = bs_i(D)$, for some $i \in \{0, 1, 2\}$ ($i = 0$ if $r = 1$), and $\deg(D_S) = \deg(E_S) = g$, then $\delta_D(E) = 2$ if $r = 1$ and $\delta_{D,i}(E) = 1$, $\delta_{D,i+1}(E) = -1$, and $\delta_{D,i+2}(E) = 0$ if $r = 2$, where $\delta_{D,j}(E)$ is the j-component of the relative distance, $\delta_D(E)$, for $j = 0, 1, 2$, and the subscripts are considered modulo 3.*

Proof: If $\deg(D_S) = \deg(E_S) = g$ and $r = 1$, then by Theorem 5.3.10, we have $\deg(E_S) - \deg(D_S) + 1 = 1 < \delta_D(E) \leq g + 2 - \deg(D_S) = 2$, so $\delta_D(E) = 2$.

If $r = 2$, then by Theorem 5.3.10, we have $\deg(E_S) - \deg(D_S) = 0 < \delta_{D,i}(E) \leq g + 1 - \deg(D_S) = 1$, so $\delta_{D,i}(E) = 1$. Let $\phi \in K^*$ such that $\Psi(E) = \langle \phi \rangle \Psi(D)$. Since $\deg(D_S) = \deg(E_S) = g$, we have $\deg(N(\phi)) = 0$. Thus, $\deg\left(\phi^{(i)}\right) = 1$ and $\deg\left(\phi^{(i+1)}\right) + \deg\left(\phi^{(i+2)}\right) = -1$. By definition, $\phi = \phi_{\mathfrak{f},i}(1)$, where $\mathfrak{f} = \Psi(-D)$, so by Corollary 5.3.7, we have $\deg\left(\phi^{(i+1)}\right) < 0$ and $\deg\left(\phi^{(i+2)}\right) \leq 0$. Thus, $\deg\left(\phi^{(i+1)}\right) = -1$ and $\deg\left(\phi^{(i+2)}\right) = 0$, and the result follows by the definition of distance. □

Under reasonable assumptions, which are verified experimentally, the following result will show that the hypothesis of Corollary 5.3.11 is satisfied very frequently for large q. Furthermore, this heuristic will be used to analyze the running times of certain algorithms in Chapter 6.

Corollary 5.3.12 *If K has unit rank 1, then a baby step in $\mathcal{R}_{\mathbf{C}}$ has length 2 with probability $1 - O(1/q)$, assuming that the norms of the distinguished ideals of $\mathcal{O}$ are uniformly distributed in $\{f(x) \in \mathbb{F}_q[x] \mid \deg(f(x)) \leq g\}$.*

Proof: Given the assumption that the norms of the distinguished ideals of $\mathcal{O}$ are uniformly distributed in $\{f(x) \in \mathbb{F}_q[x] \mid \deg(f(x)) \leq g\}$, along with the experimental evidence noted in Footnote 1 in Section 5.2.2, the number of divisors $D \in \mathcal{R}_{\mathbf{C}}$, with $\deg(D_S) = g$, is $|\mathcal{R}_{\mathbf{C}}|\left(1 - 1/q + O\left(q^{-g/2}\right)\right)$. Similarly, there are $|\mathcal{R}_{\mathbf{C}}|\left(1/q - O\left(q^{-g/2}\right)\right)$ divisors $D \in \mathcal{R}_{\mathbf{C}}$ with $\deg(D_S) < g$. Therefore, at most $|\mathcal{R}_{\mathbf{C}}|\left(1/q - O\left(q^{g/2}\right)\right)$ divisors whose finite part has degree g map, via the baby step operation, to divisors whose finite part has smaller degree, leaving at least $|\mathcal{R}_{\mathbf{C}}|\left(1 - 2/q + O\left(q^{-g/2}\right)\right)$ divisors, D, such that $\deg(D_S) = \deg(bs(D)_S) = g$. Therefore, if $D \in \mathcal{R}_{\mathbf{C}}$ is chosen randomly, then $\deg(D_S) = \deg(bs(D)_S) = g$ with probability at least $1 - 2/q + O(q^{-g/2})$. By Corollary 5.3.11, we have $\delta_D(bs(D)) = 2$ in this case, and the desired result follows. □

As another consequence, Theorem 5.3.10 improves some of the bounds on the size of the elements of a(n) (i-)reduced basis of a distinguished fractional ideal given in Part 2 of Proposition 5.1.4.

Corollary 5.3.13 *Let $\mathfrak{f} = [1, \mu, \nu]$ be a nontrivial distinguished fractional ideal of $\mathcal{O}$ with the given (i-) reduced basis. If $\mathcal{O}$ has unit rank 1, then $\deg(\nu) < \deg(\mu) \leq g + 1$. If $\mathcal{O}$ has unit rank 2 and $\deg\left(\nu^{(i+1)}\right) > 0$, then $\deg\left(\nu^{(i)}\right) < \deg\left(\mu^{(i)}\right) \leq g$. If the norms of the distinguished ideals of $\mathcal{O}$ are uniformly distributed in $\{f(x) \in \mathbb{F}_q[x] \mid \deg(f(x)) \leq g\}$, then a random distinguished fractional ideal, $\mathfrak{f} = [1, \mu, \nu]$, of $\mathcal{O}$, will satisfy $\deg\left(\nu^{(i)}\right) < \deg\left(\mu^{(i)}\right) \leq 2$ with probability $1 - 1/q$.*

[2] See the discussion at the end of Section 5 of [JSS07b].

Proof: If $\mathcal{O}$ has unit rank 1, and $\mathfrak{f} = [1, \mu, \nu]$ is nontrivial, then $-g \leq \deg(N(\mathfrak{f})) \leq -1$, so Part 2 of Proposition 5.1.4 gives $\deg(\nu) < \deg(\mu) \leq g + 1$ immediately. If $\mathcal{O}$ has unit rank 2 and $\deg\left(\nu^{(i+1)}\right) > 0$, then $\phi_{\mathfrak{f},i}(1) = \mu$, by Theorem 5.3.5. If $D = -\Psi^{-1}(\mathfrak{f})$, then $E = bs_i(D) = -\Psi^{-1}\left(\left\langle \mu^{-1} \right\rangle \mathfrak{f}\right)$, so $\deg(\mu) = \delta_{D,i}(E) = \delta_i(E) - \delta_i(D)$. Since $D \neq 0$, Theorem 5.3.10 states that $1 \leq \delta_{D,i}(E) \leq g + 1 - \deg(D_S) \leq g$.

Finally, if the norms of the distinguished ideals of $\mathcal{O}$ are uniformly distributed in $\{f(x) \in \mathbb{F}_q[x] \mid \deg(f(x)) \leq g\}$, then $\deg(N(\mathfrak{f})) = -g$ with probability $1 - 1/q$. In this case, we have $\deg\left(\nu^{(i)}\right) < \deg\left(\mu^{(i)}\right) \leq -g + g + 2 = 2$, by Part 2 of Proposition 5.1.4. □

We combine the definitions of a(n) (i-)neighbor and a baby step with the results on computing (i-) neighbors in Theorems 5.3.4 and 5.3.5 for the unit rank 1 and 2 cases, respectively, to compute a baby step. The relative distance of the baby step is determined by Lemma 3.4.2. Thus, the correctness of the following algorithm follows from these results and definitions. This algorithm is the key component of Algorithms 6.7 of [SS00] and 6.1 of [LSY03], which compute the regulator of a purely cubic function field of unit rank 1 and 2, respectively.

Algorithm 5.3.14 Baby Steps
Input: *A distinguished divisor,* $D \in \mathcal{R}_{\mathbf{C}}$, *and* $i \in \{0, 1, 2\}$ *(*$i = 0$ *if* $r = 1$*).*
Output: $E = bs_i(D) \in \mathcal{R}_{\mathbf{C}}$ *and the relative distance,* $\delta = \delta_D(E)$.

1. *Set* $\mathfrak{f} := \Psi(-D)$*, and compute the* i*-reduced basis,* $\{1, \mu, \nu\}$*, of* $\mathfrak{f}$ *via Algorithm 5.1.5.*
2. *If* $r = 1$ *or* $r = 2$ *and* $\deg\left(\nu^{(i+1)}\right) > 0$*, then:*
   a. *Set* $\{1, \tilde{\mu}, \tilde{\nu}\} := \left\{1, \mu^{-1}, \nu\mu^{-1}\right\}$.
   b. *If* $r = 1$*, set* $\delta := \deg(\mu)$.
   c. *If* $r = 2$*, set* $\delta := (\deg(\mu), \deg(\mu'), \deg(\mu''))$.
3. *Else (*$r = 2$ *and* $\deg\left(\nu^{(i+1)}\right) = 0$*), then:*
   a. *Set* $\alpha := \nu - \operatorname{sgn}\left(\nu^{(i+1)}\right)$ *and* $\{1, \tilde{\mu}, \tilde{\nu}\} := \left\{1, \mu\alpha^{-1}, \alpha^{-1}\right\}$.
   b. *Set* $\delta := (\deg(\alpha), \deg(\alpha'), \deg(\alpha''))$.
4. *Compute the* 0*-reduced basis,* $\{1, \mu, \nu\}$*, of* $\mathfrak{g} := [1, \tilde{\mu}, \tilde{\nu}]$ *via Algorithm 5.1.5.*
5. *Output* $E := -\Psi^{-1}(\mathfrak{g})$ *and* $\delta = \delta_D(E)$.

In this section, we defined the notion of a baby step in terms of neighbors and showed how to compute a baby step. We also showed that this operation yields a divisor very close to the operand, in terms of distance, and provided upper bounds on this relative distance. The next operation we will consider is the infrastructure equivalent of ideal composition.

5.3.3 Giant Steps

This section considers the second infrastructure operation, the giant step. While the baby step operation on $\mathcal{R}_{\mathbf{C}}$ is unary and finds an infrastructure divisor close to the operand, in terms of distance, the giant step operation is binary and corresponds with the composition of two ideals,

that is, ideal multiplication followed by reduction. If at least one of the operands is in the principal infrastructure, then the giant step operation produces a divisor whose distance is roughly the sum of the distances of the operands, hence the name. Giant steps are discussed in Section 7 of [Sch01] for purely cubic function fields of unit rank 1, and we extend that discussion to the unit rank 2 case. As in the previous section, we will prove results on the distance of a giant step and will outline a procedure to compute giant steps at the end of the section.

Definition 5.3.15 *Let* $\mathbf{C}_1, \mathbf{C}_2 \in Cl(\mathcal{O})$*,* $D_1 \in \mathcal{R}_{\mathbf{C}_1}$*, and* $D_2 \in \mathcal{R}_{\mathbf{C}_2}$*. The distinguished divisor* $D = Reduce(D_1 + D_2) \in \mathcal{R}_{\mathbf{C}_1\mathbf{C}_2}$ *is called the* composition *of* D_1 *and* D_2*, and is written* $D_1 \oplus D_2 = Reduce(D_1 + D_2)$*. The operation* $D_1 \oplus D_2 \rightarrow D$ *is called a* giant step*.*

Clearly, this operation is commutative, but because of the reduction step, composition is non-associative in general. However, if D_1, D_2, $D_3 \in \mathcal{R}$, then $D_1 \oplus (D_2 \oplus D_3)$ will be close to $(D_1 \oplus D_2) \oplus D_3$ in terms of distance.

There are two main obstructions to using the giant step operation in full generality. First, composition is only closed in the principal infrastructure, $\mathcal{R}$. However, if we choose to fix, say $D_2 \in \mathcal{R}$, and if $D_1 \in \mathcal{R}_{\mathbf{C}}$, then $D_1 \oplus D_2 \in \mathcal{R}_{\mathbf{C}}$. Second, if $D_1 \in \mathcal{R}_{\mathbf{C}_1}$, $D_2 \in \mathcal{R}_{\mathbf{C}_2}$, and $\mathbf{C}_1$, $\mathbf{C}_2 \neq [\mathcal{O}]$, then the distance of $D_1 \oplus D_2 \in \mathcal{R}_{\mathbf{C}_1\mathbf{C}_2}$ relative to a fixed distinguished divisor in $\mathcal{R}_{\mathbf{C}_1\mathbf{C}_2}$ is not trivial to determine. As such, we will henceforth assume that $D_1 \in \mathcal{R}_{\mathbf{C}}$, for some ideal class $\mathbf{C}$, and $D_2 \in \mathcal{R}$. We note that for our main applications, computing the S-regulator and a system of fundamental units of $\mathcal{O}$, both input divisors will in fact always belong to $\mathcal{R}$, so this is a reasonable assumption.

The following result describes the distance of the composition of two infrastructure divisors relative to the distances of the operands.

Lemma 5.3.16 *If* $\mathbf{C} \in Cl(\mathcal{O})$*,* $D_1 \in \mathcal{R}_{\mathbf{C}}$*,* $E \in \mathcal{R}_{\mathbf{C}}$ *a fixed distinguished divisor,* $D_2 \in \mathcal{R}$*, and* $D = D_1 \oplus D_2$*, then*

$$\delta_E(D) - \delta_E(D_1) + \delta(D_2) = \begin{cases} \deg(\psi) \pmod{R_x} & \text{if } r = 1 \\ (\deg(\psi), \deg(\psi'), \deg(\psi'')) \pmod{\Lambda} & \text{if } r = 2 \ , \end{cases}$$

where $\psi \in K^*$ *such that* $\Psi(D) = \langle \psi \rangle \, \Psi(D_1)\Psi(D_2)$ *and* Λ *is the set of coordinate vectors of* $\mathcal{E} = \mathcal{E}(\mathcal{O})$*.*

Proof: By the definition of a giant step, we have $D = Reduce(D_1 + D_2)$ and by Lemma 5.2.6, we have $\Psi(D) = \langle \psi \rangle \, \Psi(D_1 + D_2) = \langle \psi \rangle \, \Psi(D_1)\Psi(D_2)$. The desired result for both unit rank cases then follows from Lemma 3.4.2. □

Theorem 5.2.8 gives bounds on the number of reduction steps required to find $D_1 \oplus D_2$ from $D_1 + D_2$. This next result derives a bound on each $\deg\left(\psi^{(i)}\right)$, for $i = 0, 1, 2$, directly, to give bounds on the difference between the relative distance of the composition of two distinguished divisors and the sum of their relative distances. In particular, this result shows that the reduction procedure produces a distinguished divisor "close to" $D_1 + D_2$, in the sense that each $\left|\deg\left(\psi^{(i)}\right)\right|$ is small. The following generalizes and sharpens the bounds given in Theorems 7.1 and 7.2 of [Sch01].

Theorem 5.3.17 *Let $D_1 \in \mathcal{R}_{\mathbf{C}}$ and $D_2 \in \mathcal{R}$, where $\mathbf{C} \in Cl(\mathcal{O})$. If $\Psi(D_1 \oplus D_2) = \langle \psi \rangle \, \Psi(D_1)\Psi(D_2)$, then for each $i \in \{0, 1, 2\}$, we have*

$$-2g \leq -\left(\deg((D_1)_S) + \deg((D_2)_S)\right) \leq \deg(N(\psi)) \leq \deg\left(\psi^{(i)}\right) \leq 0 \ .$$

Proof: If $D = D_1 \oplus D_2$, then $\deg(D_S) = \deg(N(\psi)) + \deg((D_1)_S) + \deg((D_2)_S)$. From the proof of Lemma 5.2.6, we have $\deg\left(\psi^{(i)}\right) \leq 0$ for each $i = 0, 1, 2$. Since $N(\psi) = \psi\psi'\psi''$, we have $\deg(N(\psi)) \leq \deg\left(\psi^{(i)}\right) \leq 0$, which establishes the right-most two inequalities. Since $0 \leq \deg((D_j)_S) \leq g$, for $j = 1, 2$, we have $0 \leq \deg((D_1)_S) + \deg((D_2)_S) \leq 2g$. It follows that $-2g \leq -\deg((D_1)_S) + \deg((D_2)_S) = \deg(N(\psi)) - \deg(D) \leq \deg(N(\psi))$, which establishes the left-most two inequalities. □

Combining Lemma 5.3.16 and Theorem 5.3.17, we have the following result on the length of the relative distance between $D_1 + D_2$ and their composition.

Corollary 5.3.18 *Let $D_1 \in \mathcal{R}_{\mathbf{C}}$ and $D_2 \in \mathcal{R}$, where $\mathbf{C} \in Cl(\mathcal{O})$. If $\Psi(D_1 \oplus D_2) = \langle \psi \rangle \, \Psi(D_1)\Psi(D_2)$, then for each $i \in \{0, 1, 2\}$, we have*

$$-2g \leq \deg(N(\psi)) \leq \delta_{E,i}(D) - (\delta_{E,i}(D_1) + \delta_i(D_2) \pmod{R_x}) \leq 0 \ ,$$

where $D = D_1 \oplus D_2$ and $\delta_{E,i}(D)$ is the i-component of $\delta_E(D)$.

Remark 5.3.19 *Assuming, as in Corollary 5.3.12, that the norms of the distinguished ideals of $\mathcal{O}$ are uniformly distributed in $\{f(x) \in \mathbb{F}_q[x] \mid \deg(f(x)) \leq g\}$, we have $\deg((D_1)_S) + \deg((D_2)_S) = 2g$ and $\deg(D_S) = g$, where $D = D_1 \oplus D_2$, and therefore, $\deg(N(\psi)) = -g$, with probability $1 - O(1/q)$. Furthermore, this assumption is supported by experimental evidence in the principal infrastructure of function fields of unit ranks 1 and 2. Concerning $\deg(\psi)$, and hence the relative distance $\delta(D) - (\delta(D_1) + \delta(D_2))$ in principal infrastructures of unit rank 1 cubic function fields, we have $\deg(\psi) = -\lfloor g/3 \rfloor$ if $g \not\equiv 1 \pmod 3$ and $\deg(\psi) = -(g+2)/3$ if $g \equiv 1 \pmod 3$, all with probability $1 - O(1/q)$, based on experimental results for each genus $3 \leq g \leq 10$.*

Given two infrastructure divisors, $D_1 \in \mathcal{R}_{\mathbf{C}}$ and $D_2 \in \mathcal{R}$, with a fixed $E \in \mathcal{R}_{\mathbf{C}}$, the following algorithm computes their composition, $D = D_1 \oplus D_2$, along with the relative distance $\delta_E(D) - \delta_E(D_1) - \delta(D_2)$. In Section 4.4, we gave results and algorithms for ideal multiplication, and they will be used here to facilitate the computation of a giant step. In this way, we generalize Algorithm 7.4 of [Sch01], which computes giant steps in infrastructures of purely cubic function fields of unit rank 1, to include unit rank 2 function fields as well.

Algorithm 5.3.20 (Algorithm 7.4 of [Sch01] for $r = 1$) Giant Steps
Input: *$D_1 \in \mathcal{R}_{\mathbf{C}}$ and $D_2 \in \mathcal{R}$.*
Output: *$D = D_1 \oplus D_2 \in \mathcal{R}_{\mathbf{C}}$, with the relative distance, $\delta = \delta_E(D) - (\delta_E(D_1) + \delta(D_2))$, where $E \in \mathcal{R}_{\mathbf{C}}$ is a fixed distinguished divisor.*

1. *Set $\mathfrak{f}_i := \Psi(-D_i)$, with the $\mathfrak{f}_i$ in terms of a 0-reduced basis, and $\mathfrak{a}_i := \langle d(\mathfrak{f}_i) \rangle \, \mathfrak{f}_i$, for $i = 1, 2$.*
2. *Compute canonical bases for $\mathfrak{a}_1$ and $\mathfrak{a}_2$ via Lemma 4.1.2.*
3. *If $\gcd(L(\mathfrak{a}_1), L(\mathfrak{a}_2)) = 1$, compute $\mathfrak{a} := \mathfrak{a}_1\mathfrak{a}_2$ via Algorithm 4.4.7, and set $\delta := 0$.*

4. *Else, if* $\mathfrak{a}_1 = \mathfrak{a}_2$*:*

   a. *Compute* $\mathfrak{a} := \langle d^{-1} \rangle \, \mathfrak{a}_1^2$ *via Algorithm 4.4.8, where* $d \in \mathbb{F}_q[x]$.

   b. *If* $r = 1$, *set* $\delta := \deg(d) - \deg(\gcd(L(\mathfrak{a}_1), GH))$.

   c. *If* $r = 2$, *set* $\delta := (\deg(d) - \deg(\gcd(L(\mathfrak{a}_1), GH)))(1, 1, 1)$.

5. *Else, if* $\mathfrak{a}_1\mathfrak{a}_2$ *is primitive (checked via Step 3 of Algorithm 4.4.12):*

   a. *Compute* $\mathfrak{a} := \mathfrak{a}_1\mathfrak{a}_2$ *via Algorithm 4.4.9.*

   b. *If* $r = 1$, *set* $\delta := \deg(L(\mathfrak{a})) - \deg(L(\mathfrak{a}_1)) - \deg(L(\mathfrak{a}_2))$.

   c. *If* $r = 2$, *set* $\delta := (\deg(L(\mathfrak{a})) - \deg(L(\mathfrak{a}_1)) - \deg(L(\mathfrak{a}_2)))(1, 1, 1)$.

6. *Else (*$\mathfrak{a}_1 \neq \mathfrak{a}_2$ *and* $\mathfrak{a}_1\mathfrak{a}_2$ *is not primitive)*

   a. *Compute* $\mathfrak{a} := \langle d^{-1} \rangle \, \mathfrak{a}_1\mathfrak{a}_2$ *via Algorithm 4.4.11, where* $d \in \mathbb{F}_q[x]$.

   b. *If* $r = 1$, *set* $\delta := \deg(L(\mathfrak{a})) + \deg(d) - \deg(L(\mathfrak{a}_1)) - \deg(L(\mathfrak{a}_2))$.

   c. *If* $r = 2$, *set* $\delta := (\deg(L(\mathfrak{a})) + \deg(d) - \deg(L(\mathfrak{a}_1)) - \deg(L(\mathfrak{a}_2)))(1, 1, 1)$.

7. *Compute the (*0*-)reduced basis,* $\{1, \mu, \nu\}$, *of* $\mathfrak{f} := \mathfrak{a} / \langle L(\mathfrak{a}) \rangle$ *via Algorithm 5.1.5.*

8. *While* $\deg(\mu) \leq 0$*:*

   a. *If* $r = 1$, *set* $\delta := \delta + \deg(\mu)$. *If* $r = 2$, *set* $\delta := \delta + (\deg(\mu), \deg(\mu'), \deg(\mu''))$.

   b. *Set* $\mathfrak{f} := \langle \mu^{-1} \rangle \, \mathfrak{f}$. *(See Algorithm 5.2.4, Step 2.a.)*

   c. *Compute the (*0*-)reduced basis,* $\{1, \mu, \nu\}$, *of* $\mathfrak{f}$ *via Algorithm 5.1.5.*

9. *If* $\deg(\nu) < \deg(\eta_\nu) = 0$*:*

   a. *If* $r = 1$, *set* $\delta := \delta + \deg(\nu)$. *If* $r = 2$, *set* $\delta := \delta + (\deg(\nu), \deg(\nu'), \deg(\nu''))$.

   b. *Set* $\mathfrak{f} := \langle \nu^{-1} \rangle \, \mathfrak{f}$. *(See Algorithm 5.2.4, Step 3.a.)*

   c. *If* $\deg(\zeta_\nu) = 0$, *set* $\nu := \nu - \operatorname{sgn}(\zeta_\nu)$.

10. *Output* $D := -\Psi^{-1}(\mathfrak{f}) \in \mathcal{R}_{\mathbf{C}}$ *and* δ.

Proposition 5.3.21 *Algorithm 5.3.20 computes the composition,* $D \in \mathcal{R}_{\mathbf{C}}$, *of two distinguished divisors,* $D_1 \in \mathcal{R}_{\mathbf{C}}$ *and* $D_2 \in \mathcal{R}$, *and the relative distance* $\delta_E(D) - (\delta_E(D_1) + \delta(D_2))$, *for some fixed* $E \in \mathcal{R}_{\mathbf{C}}$.

Proof: The correctness of the composition of D_1 and D_2 follows from the correctness of the respective algorithms that are called in Steps 3, 4, 5, and 6 and the definition of a giant step. The relative distances computed in Steps 4, 5, and 6 follow from Lemma 5.3.16 and Corollaries 4.4.3 and 4.4.6. Finally, the reduction portion of the algorithm, Steps 8 and 9, and its contribution to the relative distance is valid by Lemmas 5.2.6 and 5.3.16. □

Since the ideals $\mathfrak{a}_1$ and $\mathfrak{a}_2$, computed in Step 1 of Algorithm 5.3.20 will tend to have norms of large degree, $2g$ in most cases, another approach to composing two divisors is to compute and multiply $\overline{\mathfrak{a}_1}$ and $\overline{\mathfrak{a}_2}$, then find the primitive inverse of the remaining product. Since inversion is not

trivial in purely cubic function fields, it is not clear if this approach would be faster. By comparison, the assignments for $\mathfrak{f}_i$ and $\mathfrak{a}_i$ in Step 1 and for $\mathfrak{f}$ in Step 7 are trivial, so Algorithm 5.3.20 is the more straightforward description of the giant step operation.

In this section, the composition operation was defined on divisors and we showed how to compute a giant step. Theorem 5.3.17 also provided bounds on the length of the reduction portion of Algorithm 5.3.20. One of the main applications of giant steps will be to speed up the computation of the regulator of a purely cubic function field of unit rank 1. We will describe this process in more depth in Section 6.2.2, though it is important to note for our current discussion that we may speed up such computations further by defining and applying the notion of an inverse of a infrastructure divisor. This is the motivation for the next section.

5.3.4 Inverses in the Principal Infrastructure

In Section 4.3, we showed how to compute the primitive component of the inverse of an ideal of any purely cubic function field. In this section, we will extend that notion to the principal infrastructure of a purely cubic function field. We will discuss potential obstructions in computing the distance of this inverse in practice and will prove a result that determines its distance. Lastly, we will outline an algorithm to compute this inverse.

Definition 5.3.22 *If $D \in \mathcal{R}$, then we denote $\overline{D} = \Psi^{-1}\left(\overline{\Psi(D)}\right)$. We call Reduce$\left(\overline{D}\right) \in \mathcal{R}$ the* inverse *of D, and write $Inverse(D) =$ Reduce$\left(\overline{D}\right)$.*

Unlike the hyperelliptic function field case, if $D \in \mathcal{R}$, then $\overline{D}$ is not distinguished in general, and hence not in $\mathcal{R}$. Thus, inversion is not a trivial calculation in the infrastructure of a cubic function field and generally requires the reduction step as well.

As noted, the main application of inverses will be to compute the regulator of a purely cubic function field of unit rank 1. For this, given a divisor $D \in \mathcal{R}$, we will need to determine the divisor $D \oplus Inverse(D) \in \mathcal{R}$, its distance, and from this, $\delta(Inverse(D))$. The following discussion will hold for infrastructures of both unit rank 1 and 2. Let $\mathfrak{a} = \Psi(D)$ and $\mathfrak{b} = \Psi(Inverse(D))$. From the definition of the inverse of D and reduction, there is some element $\psi_1 \in K^*$ such that $\mathfrak{b} = \langle \psi_1 \rangle \, \overline{\mathfrak{a}}$. Next, there is some element $\psi_2 \in K^*$ such that $\Psi(D \oplus Inverse(D)) = \langle \psi_2 \rangle \, \mathfrak{a}\mathfrak{b}$. Thus, $\Psi(D \oplus Inverse(D)) = \mathfrak{a} * \mathfrak{b} = Reduce(\mathfrak{a}\mathfrak{b}) = \langle \psi_2 \rangle \, \mathfrak{a}\mathfrak{b} = \langle \psi_1\psi_2 \rangle \, \mathfrak{a}\overline{\mathfrak{a}}$. By Lemma 2.5.7, $\overline{\mathfrak{a}}$ is a primitive ideal such that $\mathfrak{a}\overline{\mathfrak{a}} = \langle L(\mathfrak{a}) \rangle$, so $\Psi(D \oplus Inverse(D)) = \langle \psi_1\psi_2 L(\mathfrak{a}) \rangle$.

Restricting now to the unit rank 1 case, we have $0 \leq \deg(L(\mathfrak{a})) \leq \deg(\mathfrak{a}) \leq g$, since $\mathfrak{a}$ is distinguished and $-2g \leq \deg(\psi_2) \leq 0$, by Theorem 5.3.17. To find bounds on $\deg(\psi_1)$, we first note that $0 \geq \deg(\psi_1) \geq \deg(N(\psi_1)) = \deg(\mathfrak{b}) - \deg(\overline{\mathfrak{a}})$. Next, since $\mathfrak{a}\overline{\mathfrak{a}} = \langle L(\mathfrak{a}) \rangle$, we have $-\deg(\overline{\mathfrak{a}}) = \deg(\mathfrak{a}) - 3\deg(L(\mathfrak{a})) \geq -2\deg(\mathfrak{a}) \geq -2g$. Therefore, $0 \geq \deg(\psi_1) \geq \deg(N(\psi_1)) = \deg(\mathfrak{b}) - \deg(\overline{\mathfrak{a}}) \geq 0 - 2g = -2g$. Combining the degree bounds for ψ_1, ψ_2, and $L(\mathfrak{a})$, it follows that

$$R_x - 4g \leq \delta(D \oplus Inverse(D)) \leq R_x - 1 \quad \text{or} \quad 0 \leq \delta(D \oplus Inverse(D)) \leq g \ . \tag{5.6}$$

Thus, $D \oplus Inverse(D)$ will be close, if not equal, to $0 \in \mathcal{R}$, in the distance sense. If $D \oplus Inverse(D) = 0$, then by Theorem 5.3.10, we either arrive at 0 by taking at most $4g$ baby steps from $D \oplus Inverse(D)$, or we find $D \oplus Inverse(D)$ by applying at most g baby steps from 0. Thus, $D \oplus Inverse(D)$, and hence its distance, is determined my computing at most $5g$ baby steps.

The following result expresses the distance of $Inverse(D)$ in terms of $\delta(D)$ and other quantities that are obtained as we described in the discussion above.

Lemma 5.3.23 *Let $D \in \mathcal{R}$ and $E = Inverse(D) \in \mathcal{R}$ the inverse of D. If $A = D \oplus E$, then*

$$\delta(E) \equiv \begin{cases} \delta(A) - \delta(D) - \delta \pmod{R_x} & \text{if } r = 1 \\ \delta(A) - \delta(D) - \delta \pmod{\Lambda} & \text{if } r = 2 \ , \end{cases}$$

where δ is the degree output of Algorithm 5.3.20 on input D and E. If $A = 0$, then

$$\delta(E) \equiv \begin{cases} -\delta(D) + \deg(\psi) + \deg(L(\mathfrak{a})) \pmod{R_x} & \text{if } r = 1 \\ -\delta(D) + (\deg(\psi), \deg(\psi'), \deg(\psi'')) + \deg(L(\mathfrak{a}))(1, 1, 1) \pmod{\Lambda} & \text{if } r = 2 \ , \end{cases}$$

where $\mathfrak{a} = \Psi(D)$ and $\Psi(E) = \langle \psi \rangle \, \Psi\left(\overline{D}\right)$.

Proof: For the first part, Lemma 5.3.16 states that $\delta(A) \equiv \delta(D) + \delta(E) + \deg(\psi_1) \pmod{R_x}$ and $\delta(A) \equiv \delta(D) + \delta(E) + (\deg(\psi_1), \deg(\psi_1'), \deg(\psi_1'')) \pmod{\Lambda}$ in the unit rank 1 and 2 cases, respectively, where $\Psi(A) = \langle \psi_1 \rangle \, \Psi(D)\Psi(E)$. The result follows from the fact that the relative distance output of Algorithm 5.3.20 is $\delta = \deg(\psi_1)$ or $\delta = (\deg(\psi_1), \deg(\psi_1'), \deg(\psi_1''))$, in the unit rank 1 and 2 cases, respectively.

In the second part, we have $\delta(A) = 0$ or $\delta(A) = (0, 0, 0)$, in the unit rank 1 and 2 cases, respectively. From our earlier discussion, since $\langle 1 \rangle = \Psi(D \oplus E) = \langle \psi\psi_1 L(\mathfrak{a}) \rangle$, we have $\deg\left((\psi\psi_1 L(\mathfrak{a}))^{(i)}\right) = 0$, for all $0 \leq i \leq 2$. Therefore, in the unit rank 1 case, we have $\delta = -\deg(\psi L(\mathfrak{a}))$ and in the unit rank 2 case, we have $\delta = -(\deg(\psi L(\mathfrak{a})), \deg(\psi' L(\mathfrak{a})), \deg(\psi'' L(\mathfrak{a})))$. Substituting these into the first part of the lemma yields the desired result. □

The following algorithm will compute the inverse of a distinguished divisor. Its correctness follows naturally from the definition.

Algorithm 5.3.24 Inverses in the Principal Infrastructure
Input: $D \in \mathcal{R}$.
Output: $E = Inverse(D)$.

1. *Set $\mathfrak{f} := \Psi(-D)$, given in terms of a canonical basis, and $\mathfrak{a} := \langle d(\mathfrak{f}) \rangle \, \mathfrak{f}$.*
2. *Compute the canonical basis of $\mathfrak{a}$ via Lemma 4.1.2.*
3. *Compute $\mathfrak{b} := \overline{\mathfrak{a}}$ via Lemma 4.3.3.*
4. *Compute the reduced basis of $\mathfrak{g} := \mathfrak{b} / \langle L(\mathfrak{b}) \rangle$ via Algorithm 5.1.5.*
5. *Compute $\mathfrak{g} := Reduce(\mathfrak{g})$ via Algorithm 5.2.4.*
6. *Output $E := -\Psi^{-1}(\mathfrak{g})$.*

In this section, we defined and described the inverse operation in the principal infrastructure of a purely cubic function field and showed how to compute the inverse of an infrastructure divisor and find its distance. In the next section, we describe a final arithmetic operation that we will also apply to the problem of computing the regulator of a purely cubic function field of unit rank 1.

5.3.5 Computing Divisors Close to a Given Distance

As in the previous section, we will only consider the principal infrastructure here. To compute the regulator of a purely cubic function field of unit rank 1, we will need an infrastructure operation analogous to exponentiation in $Cl(\mathcal{O})$. Specifically, given $y \in \mathbb{N}_0$, this operation finds the infrastructure divisor, $D \in \mathcal{R}$, whose distance is maximal under the restriction $0 \leq \delta(D) \leq y < R_x$. We will show how to compute D and its distance, and will analyze the running time of the algorithm, both in the worst case and in the most frequent case.

Since the divisors in $\mathcal{R}$ are ordered by distance, for any integer $0 \leq y < R_x$, there is a unique divisor $D \in \mathcal{R}$ such that $\delta(D) \leq y < \delta(bs(D))$. This divisor D is called the infrastructure divisor *below* y and is denoted $D = D(y)$. For the applications described in Chapter 6, R_x is unknown, so we extend the definition of $D(y)$ to all $y \in \mathbb{N}_0$ via $D(y) = D(y \pmod{R_x})$.

While $D(y)$ can be computed quickly by generalizing standard binary exponentiation procedures to the infrastructure of a function field, via the method described in Section 4B of [SW99], for example, faster techniques are possible by expressing y in terms of its *non-adjacent form*, rather than in binary. By Reitswiesner [Rei60], every $n \in \mathbb{N}_0$ has a unique representation, $n = \sum_{i=0}^{l} b_i 2^{l-i}$, where $b_0 = 1$, $b_i \in \{-1, 0, 1\}$, for $1 \leq i \leq l$, and no two consecutive digits, b_i, are nonzero; this expression of n is called the non-adjacent form of n. The non-adjacent form of n has at most one more digit than the binary representation of n. More importantly, we expect 2/3 of the digits to be 0 in the non-adjacent form, as opposed to 1/2 for the binary representation. Therefore, the number of extra giant steps (in Step 5.d of Algorithm 5.3.26) that we expect to compute in determining $D(y)$ is reduced significantly. The following algorithm shows how to determine the non-adjacent form of an integer $n \in \mathbb{N}_0$ from its binary representation.

Algorithm 5.3.25 (Reitswiesner [Rei60]) Computation of the Non-adjacent Form of n
Input: $n = \sum_{i=0}^{k} n_i 2^i \in \mathbb{N}_0$ *in binary.*
Output: $n = \sum_{i=0}^{l} b_i 2^{l-i}$ *in non-adjacent form.*

1. *Set* $i := 0$.
2. *While* $n > 0$ *do:*
   - *If* n *is odd, then set* $z_i := 2 - n \pmod{4}$.
   - *Else, set* $z_i := 0$.
   - *Set* $n := (n - z_i)/2$.
   - *Set* $i := i + 1$.
3. *Set* $l := i - 1$.
4. *For* i *from* 0 *to* l, *set* $b_i := z_{l-i}$.
5. *Output* $\sum_{j=0}^{l} b_j 2^{l-j}$.

In addition, we have either $l = k$ or $l = k + 1$ in Algorithm 5.3.25.

Algorithm 5.3.26 will outline the steps to compute $D(y)$, but we first describe the procedure. The idea is to compute $D(s\delta(D_0))$, where $D_0 = bs(0)$ and $s = \lfloor y/\delta(D_0) \rfloor$, then take baby steps

from $D(s\delta(D_0))$ to determine $D(y)$. We will express s in terms of its non-adjacent form, $s = 2^l s_0 + 2^{l-1}s_1 + \cdots + 2s_{l-1} + s_l$, and loop through the digits, s_i. In each step, i, we will determine $E_i = D(t_i\delta(D_0))$, where $t_i = 2^i s_0 + 2^{i-1}s_1 + \cdots + 2s_{i-1} + s_i$. There are four main differences between binary exponentiation and the analogous procedure for infrastructures via the non-adjacent form. First, for each $1 \leq i \leq l$, we apply baby steps after each "doubling" $(E_{i-1} \oplus E_{i-1})$ to determine $D(2t_{i-1}\delta(D_0))$. Second, we compute another giant step with D_0 or $Inverse(D_0)$, if $s_i = 1$ or $s_i = -1$, respectively. Next, we apply baby steps to determine $D(t_i\delta(D_0)) = D((2t_{i-1} + s_i)\delta(D_0))$. Finally, we apply another series of baby steps to compute $D(y)$ from $D(t_l\delta(D_0)) = D(s\delta(D_0))$.

The following procedure was described in Algorithms VAR-DIST1 and FIXED-DIST1 of [JSS07a] for the analogous computation in hyperelliptic function fields. Since computing the inverse of a divisor in $\mathcal{R}$ is not trivial, as it is in the hyperelliptic case, and since the distance of this inverse is not trivial to determine, we make a significant change, particularly Step 5.d.iv, to generalize the procedure to the infrastructure of a purely cubic function field of unit rank 1.

Algorithm 5.3.26 (Jacobson-Scheidler-Stein [JSS07a]) Computation of $D(y)$: The Divisor Below y
Input: $y \in \mathbb{N}$.
Output: *The divisor* $D = D(y) \in \mathcal{R}$ *and* $\delta = \delta(D)$.

1. *Compute* $E := D_0 := bs(0)$ *via Algorithm 5.3.14 and set* $s := \lfloor y/\delta(D_0) \rfloor$.
2. *If* $s = 0$, *output* $D := 0$ *and* $\delta := 0$.
3. *Compute* $D_1 := Inverse(D_0)$ *via Algorithm 5.3.24.*
4. *Put* $s = \sum_{i=0}^{l} s_i 2^{l-i}$ *into non-adjacent form via Algorithm 5.3.25 and set* $t := s_0$.
5. *For* i *from* 1 *to* l*:*
   a. *Compute* $E_0 := E \oplus E$ *via Algorithm 5.3.20 and set* $j := 0$.
   b. *While* $\delta(E_j) \leq 2t\delta(D_0)$*:*
      i. *Compute* $E_{j+1} := bs(E_j)$ *via Algorithm 5.3.14.*
      ii. *Set* $j := j + 1$.
   c. *Set* $E := E_{j-1}$ *and* $t := 2t + s_i$.
   d. *If* $s_i \neq 0$*:*
      i. *If* $s_i = 1$, *set* $D' := D_0$.
      ii. *If* $s_i = -1$, *set* $D' = D_1$.
      iii. *Compute* $E' := E \oplus D'$ *via Algorithm 5.3.20.*
      iv. *If* $s_i = -1$ *and* $\delta(E') > t\delta(D_0)$, *then set* $D'_0 := D_0$ *and do:*
          - *Compute and reassign* $D'_0 := bs(D'_0)$ *via Algorithm 5.3.14.*

          While $\delta(D'_0) - 3g < \delta(D_0)$.
          - *Compute* $E' := E \oplus Inverse(D'_0)$ *via Algorithms 5.3.24 and 5.3.20.*

      v. *Set* $E'_0 := E'$ *and* $j := 0$.
      vi. *While* $\delta(E'_j) \leq t\delta(D_0)$.

- *Compute* $E'_{j+1} := bs(E'_j)$ *via Algorithm 5.3.14.*
- *Set* $j := j + 1$.

vii. Set $E := E'_{j-1}$.

6. *Set* $E_0 := E$ *and* $j := 0$.

7. *While* $\delta(E_j) \leq y$*:*

   a. *Compute* $E_{j+1} := bs(E_j)$ *via Algorithm 5.3.14.*

   b. *Set* $j := j + 1$.

8. *Output* $D := E_{j-1}$ *and* $\delta := \delta(D)$.

We note that even faster methods to compute $D(y)$ via the use of non-adjacent forms are possible by taking extra baby steps to define D_0 and by taking advantage of the heuristics in Corollary 5.3.12 and Remark 5.3.19. For the application of these ideas to hyperelliptic infrastructures, see the discussion in Section 3.3 of [JSS07a].

We give a justification for and an analysis of the running time of Algorithm 5.3.26.

Proposition 5.3.27 *For any integer* $y \in \mathbb{N}_0$*, Algorithm 5.3.26 computes* $D(y)$.

Proof: First, $0 \leq y - \lfloor y/\delta(D_0) \rfloor \delta(D_0) = y - s\delta(D_0)$, so if $\delta(E) \leq s\delta(D_0)$ in Step 6, then Step 7 indeed computes $D(y)$. Thus, it suffices to show that Steps 1 through 5 computes $D(s\delta(D_0))$. Let $t_i = 2^i s_0 + 2^{i-1} s_1 + \cdots + 2s_{i-1} + s_i$, for $0 \leq i \leq l$, so that $t_l = s$. We will show that $E = D(t_i \delta(D_0))$ at the end of each iteration of Step 5. We proceed via induction. In Step 1, and hence, at the beginning of Step 5, with $i = 1$, we have $E = D_0 = D(t_0 \delta(D_0))$. Assume that at the beginning of Step 5, we have $E = D(t_{i-1}\delta(D_0))$ for some $1 \leq i \leq l$. Step 5.a computes $E_0 = E \oplus E$, so by Lemma 5.3.16 and Theorem 5.3.17, we have $\delta(E_0) \leq 2\delta(E) \leq 2t_{i-1}\delta(D_0)$. Thus, Step 5.b will take baby steps, if necessary, to determine $D(2t_{i-1}\delta(D_0))$, and Step 5.c assigns E to this divisor.

At this point, we consider the three possibilities of $s_i \in \{-1, 0, 1\}$. If $s_i = 0$, then $E = D(2t_{i-1}\delta(D_0)) = D(t_i\delta(D_0))$. If $s_i = 1$, then Step 5.d.iii computes $E' = E \oplus D_0$, so that $\delta(E') \leq \delta(E) + \delta(D_0) \leq 2t_{i-1}\delta(D_0) + \delta(D_0) = t_i\delta(D_0)$. Taking baby steps in Step 5.d.vi, we therefore determine the divisor $D(t_i\delta(D_0))$, assigning E to this in Step 5.d.vii. Finally, if $s_i = -1$, and $\delta(E') \leq t_i\delta(D_0)$ then the loop in Step 5.d.iv is not entered. Otherwise, the loop is entered; we claim that it terminates after a finite number of steps and that the divisor, E', computed in Step 5.4.iv satisfies $\delta(E') \leq t_i\delta(D_0)$. The loop is initialized with $D'_0 = D_0$ and computes and reassigns the divisors, $D'_0 := bs(D'_0)$, so that $\delta(D'_0)$ is strictly increasing with each iteration. Therefore, after a finite number of steps, we obtain a divisor, D'_0, such that $3g - \delta(D'_0) \leq -\delta(D_0)$. If $D'_1 = Inverse(D'_0)$, then the next step computes $E' = E \oplus D'_1$. Corollary 5.3.18 states that $\delta(E') = \delta(E \oplus D'_1) \leq (\delta(E) + \delta(D'_1) \pmod{R_x})$. Lemma 5.3.23 then implies that $\delta(E) + \delta(D'_1) \equiv \delta(E) + \delta(D'_0 \oplus D'_1) - \delta(D'_0) - \deg(\psi) \pmod{R_x}$, where $\Psi(D'_1) = \langle\psi\rangle\, \Psi(D'_0)$. By (5.6), $\delta(D'_0 \oplus D'_1)$ is either close to R_x or close to 0; we consider both cases. First, if $R_x - 4g \leq \delta(D'_0 \oplus D'_1) \leq R_x - 1$, then by Theorem 5.3.17, we have $\delta(E) - 4g - \delta(D'_0) \leq (\delta(E) + \delta(D'_0 \oplus D'_1) - \delta(D'_0) - \deg(\psi) \pmod{R_x})$ and

$$\begin{aligned}(\delta(E) + \delta(D'_0 \oplus D'_1) - \delta(D'_0) - \deg(\psi) \pmod{R_x}) &\leq \delta(E) - 1 - \delta(D'_0) + 2g \\ &< \delta(E) + 3g - \delta(D'_0) \\ &\leq 2t_{i-1}\delta(D_0) - \delta(D_0) = t_i\delta(D_0)\ .\end{aligned}$$

Next, if $0 \leq \delta(D_0' \oplus D_1') \leq g$, then Theorem 5.3.17 gives

$$\delta(E) + \delta(D_0' \oplus D_1') - \delta(D_0') - \deg(\psi) \leq \delta(E) + g - \delta(D_0') + 2g \leq 2t_{i-1}\delta(D_0) - \delta(D_0) = t_i\delta(D_0) \ .$$

Therefore, after a finite number of steps, Step 5.d.iv determines E' such that $\delta(E') \leq t_i\delta(D_0)$, as claimed. Taking baby steps in Step 5.d.vi, we determine the divisor $D(t_i\delta(D_0))$. For a fixed $1 \leq i \leq l$ and each possible s_i, Step 5 finishes with the divisor, $E = D(t_i\delta(D_0))$. Thus, in Step 6, we have $E = D(t_l\delta(D_0)) = D(s\delta(D_0))$, so $\delta(E) \leq s\delta(D_0)$, as desired. □

Proposition 5.3.28 *Algorithm 5.3.26 requires at most $O(g\log(y))$ baby steps and at most $O(\log(y))$ giant steps and inverses.*

Proof: Let $t_i = 2^i s_0 + 2^{i-1}s_1 + \cdots + 2s_{i-1} + s_i$, for $0 \leq i \leq l$, so that $t_l = s$. Step 1 performs one baby step and Step 3 makes one inversion. Step 5 then makes $l \leq \lceil \lg(y/\delta(D_0)) \rceil \leq \lceil \lg(y) \rceil$ iterations. Now if a positive integer is given in non-adjacent form, then no two consecutive digits can be nonzero. It follows that of the l steps taken in Step 5, at least half of the s_i are 0, so at most $\lfloor l/2 \rfloor$ are -1. In Step 5, Step 5.a makes one giant step. Now by Theorem 5.3.10, the length of a baby step is between 1 and $g+2$, so $0 \leq t_{i-1}\delta(D_0) - \delta(E) \leq g+1$ and hence $0 \leq 2t_{i-1}\delta(D_0) - 2\delta(E) \leq 2g+2$. In addition, by Theorem 5.3.17, we have $0 \leq 2\delta(E) - \delta(E_0) \leq 2g$, so $0 \leq 2t_{i-1}\delta(D_0) - \delta(E_0) \leq 4g+2$. Therefore, $D(2t_{i-1}\delta(D_0))$ is reached after at most $4g+2$ baby steps in Step 5.b, and the step requires an additional baby step to verify that we have indeed computed $D(2t_{i-1}\delta(D_0))$. In Step 5.d.iii, we make another giant step. Now suppose that $s_i = -1$ and $\delta(E') > t_i\delta(D_0)$. In this case, Step 5.d.iv computes D_0' from D_0 via baby steps until $\delta(D_0') - 3g \geq \delta(D_0)$. Thus, $3g \leq \delta(D_0') - \delta(D_0) \leq (3g-1) + (g+2) = 4g+1$, so computing D_0' requires at most $3g$ baby steps. In addition, Step 5.d.iv computes one infrastructure inverse and one giant step. Next, by reasoning similar to that for Step 5.b, Step 5.d.vi will perform at most $3g+2$ baby steps. Thus, each iteration of Step 5 requires at most $10g+5$ baby steps, 3 giant steps, and 1 inverse if $s_i = -1$, at most $7g+5$ baby steps and 2 giant steps if $s_i = 1$, and at most $4g+3$ baby steps and 1 giant step if $s_i = 0$. Lastly, in Step 7, we will have $E = D(s\delta(D_0))$ and $0 \leq y - s\delta(D_0) \leq g+1$, so we will determine $D(y)$ from E via at most $g+2$ baby steps. In total, Algorithm 5.3.26 requires at most $g+3+(10g+5)\lfloor l/2 \rfloor + (4g+3)\lfloor l/2 \rfloor = (14g+8)\lfloor l/2 \rfloor + g + 3 = O(g\log(y))$ baby steps, at most $3\lfloor l/2 \rfloor = O(\log(y))$ giant steps, and at most $\lfloor l/2 \rfloor + 1 = O(\log(y))$ inverses. □

In practice, the number of baby steps, giant steps, and inverses computed by Algorithm 5.3.26 will be much lower than the worst case. We assume that q is large and that the norms of the distinguished principal ideals are uniformly distributed in $\{f(x) \in \mathbb{F}_q[x] \mid \deg(f(x)) \leq g\}$. By Corollary 5.3.12, the length of a baby step is 2 with probability $1 - O(1/q)$. From this it follows that the average value of $s_{i-1} - \delta(D_{i-1})$, over all $1 \leq i \leq k$ and all purely cubic function fields of genus g and unit rank 1, is close to 1/2. Now let $\rho(g) = \lfloor g/3 \rfloor$ if $g \not\equiv 1 \pmod 3$, and $\rho(g) = (g+2)/3$ if $g \equiv 1 \pmod 3$. By Remark 5.3.19, we have $2\delta(D_{i-1}) - \delta(D_{i,0}) = \rho(g) \approx g/3$ with probability $1 - O(1/q)$, based on experimental evidence given in Remark 5.3.19. Therefore, in Step 5.b, we will have $2t_{i-1}\delta(D_0) - 2\delta(E) = 2$ and $2\delta(E) - \delta(E') = \rho(g)$ with probability $1 - O(1/q)$, which yields $(\rho(g)+3)/2 \approx g/6 + 3/2$ baby steps. Next, if $s = -1$, then with high probability, the loop in Step 5.d.iv will not be entered. By similar reasonings as for Step 5.b, we will perform

an average of $\rho(g)/2 + 1/4 \approx g/6 + 1/4$ baby steps in Step 5.d.vi, with probability $1 - O(1/q)$. Similarly, we will perform an average of $(g+5)/4$ baby steps in Step 7, with probability $1 - O(1/q)$. Now, in Step 1, we have $s \approx y/(g+2)$, so that $l \leq \lceil \lg(y/(g+2)) \rceil$. With s in non-adjacent form, we will have $s_i = 0$ with probability 2/3. Therefore, with high probability, we will compute about $1 + ((g/6 + 3/2) + (1/3)(g/6 + 5/4))l + (g+5)/4 = (2g/9 + 23/12)l + (g+9)/4$ baby steps, $(1 + (1/3)(1))l = (4/3)l$ giant steps and 1 inverse.

In this section we defined and showed how to compute the baby step and giant step operations in the infrastructures of a purely cubic function field of positive unit rank, the inverse operation in the principal infrastructure, and the divisor below $y \in \mathbb{N}_0$ in the principal infrastructure of a unit rank 1 cubic function field. In the next section and subsequent chapter, we will apply these operations to compute the regulator and a system of fundamental units of a purely cubic function field of positive unit rank.

5.4 Applications of Infrastructure Arithmetic

In Chapter 6, we will formally outline several procedures to compute the regulator of a purely cubic function field of positive unit rank. In this section, however, we will apply the baby step operation in particular to lay the theoretical foundation for these algorithms. We will also prove results on the symmetric structure of infrastructures in the unit rank 2 case.

The notion of an i-chain, which we will discuss in Section 5.4.1, and methods to compute fundamental units and regulators using these i-chains were first described for use in purely cubic number fields by Voronoi [Vor96], using the language of binary forms. His methods were rewritten in the language of fractional ideals by Delone and Fadeev [DF64], implemented for purely cubic number fields by Williams et al. in [WCS80], and then improved using infrastructure techniques in [WDS83, Wil85]. In addition, Buchmann generalized Voronoi's method to number fields of unit rank 1 and 2 in [Buc85], and to number fields of any unit rank in [Buc87a] and [Buc87b], applying infrastructure methods in [Buc87c] and [BW88]. These methods were then adapted to purely cubic function fields of positive unit rank by Scheidler, Stein, et al. in [SS00, LSY03] for the unit rank 1 and 2 cases, respectively, using the language of fractional ideals and minima. In this section, we will complement the description of these techniques using the language of divisors.

5.4.1 i-Chains

In this section, two notions of an i-chain will be defined: one as a sequence of divisors computed via baby steps, and a second via associated minima of $\mathcal{O}$. These i-chains will be applied to the computation of regulators and systems of fundamental units, so it will be necessary to utilize the correspondence between principal distinguished divisors and minima in $\mathcal{O}$.

Let $D = D_0 \in \mathcal{R}_{\mathbf{C}}$, for some ideal class $\mathbf{C} \in Cl(\mathcal{O})$. For a fixed $i \in \{0, 1, 2\}$, let $D_n = bs_i(D_{n-1})$, for all $n \in \mathbb{N}$. The sequence $(D_n)_{n \in \mathbb{N}_0} \subseteq \mathcal{R}_{\mathbf{C}}$ is called the *i-chain* of D in $\mathcal{R}_{\mathbf{C}}$ and is denoted $\mathcal{R}_{\mathbf{C},i}(D)$. If $\mathbf{C} = [\mathcal{O}]$, then we simply write $\mathcal{R}_i(D) = (D_n)_{n \in \mathbb{N}_0} \subseteq \mathcal{R}$. In addition, if $D = 0$, then we write $(D_n)_{n \in \mathbb{N}_0} = \mathcal{R}_i$. In unit rank 1 infrastructures, $\mathcal{R}_0$ is called the *Voronoi chain*. Since $\mathcal{R}_{\mathbf{C}}$ is finite for any ideal class $\mathbf{C} \in Cl(\mathcal{O})$, by Proposition 3.4.4, any i-chain of $\mathcal{R}_{\mathbf{C}}$ must be periodic. $\mathcal{R}_{\mathbf{C},i}(D)$ is said to be *periodic* if there are minimal integers $p \geq 0$ and $l > 0$ such that $bs_i(D_{p+l-1}) = D_p$, and

purely periodic if $p = 0$. The subsequences $(D_n)_{0 \leq n < p}$ and $(D_n)_{p \leq n < p+l}$ of $\mathcal{R}_{\mathbf{C},i}(D)$ are called the *preperiod* and the *(primitive) period* of $\mathcal{R}_{\mathbf{C},i}(D)$, respectively. Identifying periods of these i-chains will be the key component for computing a system of fundamental units and the regulator of $\mathcal{O}$.

In the principal infrastructure, we will define an analogous i-chain composed of minima of $\mathcal{O}$. Let $(D_n)_{n \in \mathbb{N}_0} = \mathcal{R}_i(D_0)$ and θ_0 a minimum of $\mathcal{O}$ such that $\Psi(D_0) = \langle \theta_0 \rangle$. Since $D_{n+1} = bs_i(D_n)$, for all $n \in \mathbb{N}_0$, there is an element $\phi_n \in \mathcal{O}$ such that $\Psi(D_{n+1}) = \langle \phi_n \rangle \Psi(D_n)$. For all $n \in \mathbb{N}_0$, we recursively define $\theta_{n+1} = \phi_n \theta_n$, and call the sequence $(\theta_n)_{n \in \mathbb{N}_0}$ the *i-chain* of θ_0. If $r = 1$ and $\theta_0 = 1$, then $(\theta_n)_{n \in \mathbb{N}_0}$ is called the *Voronoi chain* of $\mathcal{O}$. By construction, we have $\langle \theta_n \rangle = \Psi(D_n)$, for all $n \in \mathbb{N}_0$, but Theorem 5.3.10 states that $\deg\left(\phi_n^{(i)}\right) > 0$, so $\deg\left(\theta_{n+1}^{(i)}\right) > \deg\left(\theta_n^{(i)}\right)$, for all $n \in \mathbb{N}_0$, and the i-chain of θ_0 is not periodic.

Considering now the unit rank 1 case, from our discussion in Section 3.4.4, $\mathcal{R}_{\mathbf{C}}$ can be viewed as being embedded into the circle $\mathbb{Z}/R_x\mathbb{Z}$ via the distance function (relative to some $E \in \mathcal{R}_{\mathbf{C}}$). Therefore, we would expect that every divisor in $\mathcal{R}_{\mathbf{C}}$ is on the 0-chain of any divisor of $\mathcal{R}_{\mathbf{C}}$; in other words, that any unit rank 1 infrastructure is purely periodic. The following result, whose proof is based on that of Proposition 6.1 of [SS00], shows that this is indeed true.

Proposition 5.4.1 *If $K = K_x$ is a purely cubic function field of unit rank 1, $\mathbf{C} \in Cl(\mathcal{O})$, and $D \in \mathcal{R}_{\mathbf{C}}$, then $\mathcal{R}_{\mathbf{C},0}(D) = \mathcal{R}_{\mathbf{C}}$ is purely periodic. In particular, if $\mathbf{C} = [\mathcal{O}]$, then we have $\mathcal{R}_0 = \mathcal{R}$.*

Proof: Let $(D_n)_{0 \leq n < p+l} = \mathcal{R}_{\mathbf{C},0}(D)$, for any divisor $D = D_0 \in \mathcal{R}_{\mathbf{C}}$, where p and l are the preperiod and period length of $\mathcal{R}_{\mathbf{C},0}(D)$, respectively, and let D be the distinguished divisor in $\mathcal{R}_{\mathbf{C}}$ to which all distances are relative. We first show that $\mathcal{R}_{\mathbf{C},0}(D)$ is purely periodic. By construction, D_{p+l-1} is the infrastructure divisor of largest distance relative to D and $bs(D_{p+l-1}) = D_p$. By definition, there is some $\phi_{p+l-1} \in K^*$ such that $\Psi(D_p) = \langle \phi_{p+l-1} \rangle \Psi(D_{p+l-1})$ and $\delta_D(D_p) \equiv \delta_D(D_{p+l-1}) + \deg(\phi_{p+l-1}) \pmod{R_x}$. Let $\alpha \in K^*$ such that $\Psi(D) = \langle \alpha \rangle \Psi(D_{p+l-1})$. Since $\delta_D(D) = 0 \leq \delta_D(D_p)$, we must have $0 < \deg(\alpha) \leq \deg(\phi_{p+l-1})$.

If $\deg(\alpha') \geq 0$, then $\mathrm{div}(\alpha)^S < 0$ and $\deg(\mathrm{div}(\alpha)_S) > 0$. Then $D - \mathrm{div}(\alpha) \sim D$, $D_S - \mathrm{div}(\alpha)_S = (D_{p+l-1})_S \geq 0$, $\deg(D_S) \geq \deg(D_S - \mathrm{div}(\alpha)_S)$, and $D^S \leq D^S - \mathrm{div}(\alpha)^S$, which contradicts the assumption that D is distinguished. Now notice that $\mathrm{div}(\alpha)_S = D_S - (D_{p+l-1})_S \geq -(D_{p+l-1})_S$, so that $\alpha \in \mathfrak{f} = \Psi(-D_{p+l-1})$. Since $\deg(\alpha') < 0$, we have $\alpha \in \mathcal{H}_{\mathfrak{f}}(1)$, so $\alpha \leq_0 \phi_{p+l-1}$. By the minimality of ϕ_{p+l-1} under $\leq_0$, we must have $\alpha = \phi_{p+l-1}$. Thus, $D = D_p$, so $p = 0$ and $\mathcal{R}_{\mathbf{C},0}(D)$ is purely periodic.

Now let $E \in \mathcal{R}_{\mathbf{C}}$. To show that $\mathcal{R}_{\mathbf{C},0}(D) = \mathcal{R}_{\mathbf{C}}$, it suffices to show that $E = D_k$ for some $0 \leq k < l$. Since $\mathcal{R}_{\mathbf{C},0}(D)$ is purely periodic, $(\delta(D_n))_{0 \leq n < l}$ is a strictly increasing sequence of non-negative integers, and $0 \leq \delta(E) < R_x$, there is some integer $0 \leq k < l$ such that either $\delta(D_k) \leq \delta(E) < \delta(D_{k+1})$ or $\delta(D_k) \leq \delta(E) < R_x$. In either case, there are elements α, $\phi_k \in K^*$ such that $\Psi(E) = \langle \alpha \rangle \Psi(D_k)$ and $\Psi(bs(D_k)) = \langle \phi_k \rangle \Psi(D_k)$. Since $\mathrm{div}(\alpha)_S = E_S - (D_k)_S \geq -(D_k)_S$, we have $\alpha \in \mathfrak{f}_k = \Psi(-D_k)$. From the inequalities of the distances, we have $0 \leq \deg(\alpha) < \deg(\phi_k)$.

If $\deg(\alpha') < 0$, then $\alpha \in \mathcal{H}_{\mathfrak{f}_k}(1)$, which would contradict the minimality of ϕ_k under $<_0$. Thus, $\deg(\alpha') \geq 0$, so that $\mathrm{div}(\alpha)^S \leq 0$ and $\deg(\mathrm{div}(\alpha)_S) \geq 0$. We have $E - \mathrm{div}(\alpha) \sim E$, $E_S - \mathrm{div}(\alpha)_S = (D_k)_S \geq 0$, $\deg(E_S - \mathrm{div}(\alpha)_S) \leq \deg(E_S)$, and $E^S - \mathrm{div}(\alpha)^S \geq E^S$. Since E is distinguished, we must have $E = E - \mathrm{div}(\alpha)$, that is $\alpha \in \mathbb{F}_q^*$, so that $E = D_k$, as desired. □

The following result, from [SS00], proves the analogous property of the Voronoi chain of $\mathcal{O}$. Its proof may be found in the given source.

Proposition 5.4.2 (Proposition 6.1 of [SS00]) *Let $K = K_x$ be a purely cubic function field of unit rank 1 and $(\theta_n)_{n\in\mathbb{N}_0}$ the Voronoi chain of $\mathcal{O} = \mathcal{O}_x$. If θ is a minimum of $\mathcal{O}$ of positive degree, then $\theta = a\theta_n$ for some $a \in \mathbb{F}_q^*$ and $n \in \mathbb{N}_0$.*

Unlike the unit rank 1 case, if $\mathcal{O}$ has unit rank 2, then $\mathcal{R}_{\mathbf{C}}$ cannot be expressed as a single i-chain in general. Furthermore, using Riemann-Roch spaces, Fontein has found examples in which some infrastructure divisors cannot be reached via baby steps from any other infrastructure divisor. For example, if $C : Y^3 = x^6 + 4x^5 + 4x^4 + 5x^3 + 2x^2$, then the principal infrastructure of $\mathbb{F}_7(C)$ contains three divisors which are not on any baby step chain of any other divisor of $\mathcal{R}$. Fontein [Fon08a] notes that each i-chain of 0 has a nontrivial preperiod in this example. For the infrastructure of any field of unit rank 2, he expects that if the i-chains of 0 in $\mathcal{R}$ are not purely periodic, then there exist divisors that are not on the i-chain of any other divisor in $\mathcal{R}$. Based on this and a number of observations, we have the following conjecture.

Conjecture 5.4.3 *If $\mathcal{O} = \mathcal{O}_x$ has unit rank 2 and there exists an i-chain of 0 in $\mathcal{R}$ which is not purely periodic, then there exists a divisor $D \in \mathcal{R}$ which cannot be found by taking a baby step from another divisor of $\mathcal{R}$.*

Moreover, he expects that almost every infrastructure of a function field of unit rank 2 will contain divisors that cannot be reached via baby steps from another divisor, assuming that the infrastructure is sufficiently large.

We have defined the notion of an i-chain of divisors and of minima. We note that while i-chains of divisors are periodic, i-chains of minima are not. This is the main idea behind using these i-chains to compute regulators and a system of fundamental units of $\mathcal{O}$, and we will discuss this in more depth in the next section.

5.4.2 Computing the Regulator and a System of Fundamental Units of $\mathcal{O}$

Though i-chains of minima are not periodic, we will show that the period of the associated i-chain of divisors will allow us to identify nontrivially associate minima. In this section, we will describe specifically how this process computes the regulator and a system of fundamental units of a cubic function field of positive unit rank. In addition, we will use the divisor-theoretic description of i-chains, along with our results on the structure of infrastructures, to give an intuitive explanation for the effectiveness of these methods. Since the correspondence between minima and divisors only exists in the principal class, we will restrict ourselves to the principal infrastructure of a cubic function field in this section.

For some $i \in \{0, 1, 2\}$ and $D = D_0 \in \mathcal{R}$, let $(D_n)_{n\in N_0} = \mathcal{R}_i(D)$ be the i-chain of D, with preperiod length p and period length l, and $(\theta_n)_{n\in\mathbb{N}_0}$ the Voronoi chain of θ_0. For $0 \le n < p + l$, we have $\deg(\theta_n) = \delta(D_n)$ in unit rank 1 infrastructures and $(\deg(\theta_n), \deg(\theta'_n), \deg(\theta''_n)) = \delta(D_n)$ in unit rank 2 infrastructures, by Lemma 5.3.9. However, since $D_p = bs_i(D_{p+l-1})$, we have $\langle\theta_p\rangle = \Psi(D_p) = \langle\phi_{p+l-1}\rangle\,\Psi(D_{p+l-1}) = \langle\phi_{p+l-1}\theta_{p+l-1}\rangle = \langle\theta_{p+l}\rangle$. This fact, combined with $\deg\left(\theta_{p+l}^{(i)}\right) > \deg\left(\theta_p^{(i)}\right)$, implies that $\theta_{p+l}\theta_p^{-1} = \eta$, for some nontrivial unit $\eta \in \mathcal{O}^*$ such that $\deg\left(\eta^{(i)}\right) > 0$. We call η the *primitive unit* of the i-chain, though it is only unique up to a multiple in $\mathbb{F}_q^*$.

In the unit rank 1 case, the following result shows that η, the primitive unit of $\mathcal{R} = \mathcal{R}_0$, is in fact the fundamental unit of $\mathcal{O}$ of positive degree, unique up to a multiple in $\mathbb{F}_q^*$. The proof may be found in the given source.

Theorem 5.4.4 (Theorem 6.1.1 of [Sch04]) *If $K = K_x$ is a purely cubic function field of unit rank 1 and $(\theta_n)_{n\in\mathbb{N}_0}$ is the Voronoi chain of $\mathcal{O} = \mathcal{O}_x$, then $\epsilon = \theta_l$ is the fundamental unit of $\mathcal{O}$ of positive degree, up to a multiple in $\mathbb{F}_q^*$, where $l = |\mathcal{R}|$ is the period of the Voronoi chain.*

To compute the S-regulator, we have $R_x = \deg(\epsilon)$ and $R^S = R_x/2$, by definition. However, since $\mathcal{R} = \left\{D_n = \Psi^{-1}\left(\langle\theta_n\rangle\right) \mid 0 \le n < l\right\}$ and $\Psi(D_{n+1}) = \langle\phi_n\rangle\,\Psi(D_n)$, where ϕ_n is the neighbor of 1 in the principal fractional ideal $\left\langle\theta_n^{-1}\right\rangle$, then we may determine R_x without computing ϵ by noting that

$$R_x = \deg(\epsilon) = \deg(\theta_l) = \deg\left(\prod_{n=0}^{l-1}\phi_n\right) = \sum_{n=0}^{l-1}\deg(\phi_n) = \sum_{n=0}^{l-1}\delta_{D_n}(D_{n+1}) = \delta(D_{l-1}) + \delta_{D_{l-1}}(0)\ . \tag{5.7}$$

In this way, we replace the multiplication of elements in $\mathcal{O}$ with addition in $\mathbb{Z}$.

In unit rank 2 infrastructures, the toroidal structure of $\mathcal{R}$ implies that there are two periods to navigate, so it is not as clear how to obtain a system of fundamental units, $\{\epsilon_1, \epsilon_2\}$, by performing baby steps in $\mathcal{R}$. Theorem 5.4.5 will describe one way to determine a system of fundamental units, but before stating it, we will discuss the idea behind it. Informally, traversing the two periods of $\mathcal{R}$ via two different chains will yield two pairs of associate minima in $\mathcal{O}$ and therefore two nontrivial units of $\mathcal{O}^*$. In Figure 5.1, we demonstrate this for the infrastructure of $\mathbb{F}_7(C)$, where $C : Y^3 = x^6 - x^5 + 3x^4 + 2x^3 - x^2$. We consider the parallelogram, $\mathcal{D}_0^S/\mathcal{P}^S$, drawn in the coordinate system with an $(\infty_1 - \infty_0)$-axis and an $(\infty_0 - \infty_2)$-axis. In Figure 5.1, $\mathcal{D}_0^S/\mathcal{P}^S$ is determined by $\operatorname{div}(\epsilon_1) = 28(\infty_1 - \infty_0) - 4(\infty_0 - \infty_2)$ and $\operatorname{div}(\epsilon_2) = -8(\infty_1 - \infty_0) + 12(\infty_0 - \infty_2)$.

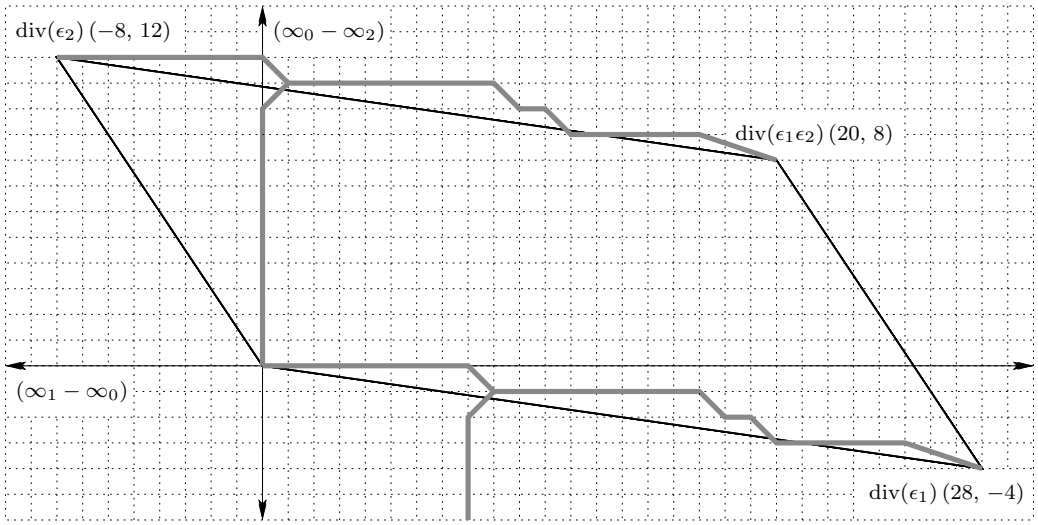

Figure 5.1: The Infrastructure of $\mathbb{F}_7(C)$, $C : Y^3 = x^6 - x^5 + 3x^4 + 2x^3 - x^2$, with Respect to $\{\epsilon_1, \epsilon_2\}$

In the figure, the corners of the parallelogram represent divisors of units of $\mathcal{O}$, which in each case is the divisor $0 \in \mathcal{R}$. The lattice points represent 0-distinguished divisors; most, but not all of these are in fact distinguished divisors and hence, elements of $\mathcal{R}$. However, we identify divisors that

differ by the divisor of a unit, so that $\mathcal{R}$ may be considered as a subset of the lattice point in the interior of the parallelogram, including lattice points in the interior of the left and lower edges and the identity, 0.

The 0-chain, $\mathcal{R}_0$, finds a period, beginning with $D_p \in \mathcal{R}_0$, corresponding with the "length" of the parallelogram, roughly parallel to the $(\infty_1 - \infty_0)$-axis. In Figure 5.1, two copies of $\mathcal{R}_0$ are drawn as gray lines, beginning at (0, 0) and $(-8, 12)$, respectively, and proceed to the right. Note that $\mathcal{R}_0$ is purely periodic in the example, so $D_p = 0$. The 2-chain of D_p, $\mathcal{R}_2(D_p)$, finds a period corresponding with the "height" of the parallelogram, roughly parallel to the $(\infty_0 - \infty_2)$-axis. In Figure 5.1, two segments of $\mathcal{R}_2$ are drawn as gray lines, beginning at (0, 0) and $(8, -6)$, respectively, and proceed upwards. $\mathcal{R}_2$ intersects $\mathcal{R}_0$ at the identified points $(9, -1)$ and (1, 11), which yields $\operatorname{div}(\epsilon_2)$. In general, we are not guaranteed that $\mathcal{R}_2(D_p)$ is purely periodic, but it will intersect with $\mathcal{R}_0$. The period is therefore completed by "backtracking" along $\mathcal{R}_0$ to reach D_p again. Both periods yield nontrivially associate minima in $\mathcal{O}$, and hence a pair of units. Given the structure of $\mathcal{R}$, these units generate a subgroup of $\mathcal{O}^*$ of finite index, though our intuition suggests that these units form a system of fundamental units.

The following theorem shows specifically how this method computes a system of fundamental units of $\mathcal{O}$. The theorem was proved in the number field case in Theorems 2.1 and 2.2 in [Buc85] and restated in the function field setting in [LSY03]. As stated in [LSY03], the proofs in [Buc85] adapt perfectly to the function field setting.

Theorem 5.4.5 (Theorems 3.7 of [LSY03] and 6.1.2 of [Sch04]) *Suppose $\mathcal{O} = \mathcal{O}_x$ has unit rank 2. Let $\{i, j, k\} = \{0, 1, 2\}$, θ a minimum in $\mathcal{O}$, $(\theta_n)_{n\in\mathbb{N}_0}$ the i-chain of $\theta = \theta_0$, and $(\phi_n)_{n\in\mathbb{N}_0}$ the j-chain of $\theta_p = \phi_0$, where p is the preperiod of $(\theta_n)_{n\in\mathbb{N}_0}$. Suppose that for some n, there are elements θ_n and θ_{n+1} in the i-chain of θ such that $\deg\left(\theta_{n+1}^{(k)}\right) \neq \deg\left(\theta_n^{(k)}\right)$.*

1. *There exists an element $\phi_m \in (\phi_n)_{n\in\mathbb{N}_0}$ such that $\phi_m \neq \theta_t$ and $\phi_m = \epsilon\theta_t$, for some element $\theta_t \in (\theta_n)_{n\in\mathbb{N}_0}$ and unit $\epsilon \in \mathcal{O}^* \setminus \mathbb{F}_q$.*

2. *Let $m, t \in \mathbb{N}$ be minimal such that θ_t and ϕ_m are nontrivially associate. If ϵ_1 is the primitive unit of the i-chain of θ and $\epsilon_2 = \phi_m\theta_t^{-1}$, then $\{\epsilon_1, \epsilon_2\}$ is a system of fundamental units of $\mathcal{O}$.*

In the context of computing in unit rank 2 infrastructure, we will take $i = 0$ and $j \in \{1, 2\}$. We will call p the *preperiod length* of $\mathcal{R}$, l the *(primary) period length* of $\mathcal{R}$, $m = m_j$ the *j-secondary* period length of $\mathcal{R}$, and $t = t_j$ the *j-twist* of $\mathcal{R}$.

As in the unit rank 1 case, we may use relative distances to compute the regulator, $R_x = R^S$, of a cubic function field of unit rank 2. Taking $i = 0$ and $j = 2$, Theorem 5.4.5 uses the 0- and 2-chains

$$\mathcal{R}_0 = \left\{D_n = \Psi^{-1}\left(\langle\theta_n\rangle\right) \in \mathcal{R} \mid 0 \leq n \leq p + l\right\}$$

and

$$\mathcal{R}_2\left(D_p\right) = \left\{E_n = \Psi^{-1}\left(\langle\phi_n\rangle\right) \in \mathcal{R} \mid 0 \leq n \leq m\right\} \quad ,$$

where $E_0 = D_p$. Recall from Section 3.2.5 that $R_x = |\deg(\epsilon_1)\deg(\epsilon_2') - \deg(\epsilon_2)\deg(\epsilon_1')|$. Let $\alpha_n \in K^*$, for $p \leq n < p + l$, such that $\Psi(D_{n+1}) = \langle\alpha_n\rangle\,\Psi(D_n)$, and $\beta_n \in K^*$, for $0 \leq n < m$, such

that $\Psi(E_{n+1}) = \langle \beta_n \rangle \, \Psi(E_n)$. Then for $i = 0, 1, 2$, we have

$$\begin{aligned}
\deg\left(\epsilon_1^{(i)}\right) &= \deg\left(\left(\theta_{p+l}\theta_p^{-1}\right)^{(i)}\right) = \deg\left(\left(\prod_{n=p}^{p+l-1}\alpha_n\right)^{(i)}\right) = \sum_{n=p}^{p+l-1}\deg\left(\alpha_n^{(i)}\right) \\
&= \sum_{n=p}^{p+l-1}\delta_{D_n,i}(D_{n+1}) \quad \text{and} \qquad (5.8) \\
\deg\left(\epsilon_2^{(i)}\right) &= \deg\left(\left(\phi_m\theta_t^{-1}\right)^{(i)}\right) = \deg\left(\left(\prod_{n=0}^{m-1}\beta_n\prod_{j=p}^{t-1}\alpha_j^{-1}\right)^{(i)}\right) = \sum_{n=0}^{m-1}\deg\left(\beta_n^{(i)}\right) - \sum_{j=p}^{t-1}\deg\left(\alpha_j^{(i)}\right) \\
&= \sum_{n=0}^{m-1}\delta_{E_n,i}(E_{n+1}) - \sum_{j=p}^{t-1}\delta_{D_j,i}(D_{j+1}) \ ,
\end{aligned}$$

where $\delta_{D_n,i}(D_{n+1})$ and $\delta_{E_n,i}(E_{n+1})$ are the i-components of the relative distances $\delta_{D_n}(D_{n+1})$ and $\delta_{E_n}(E_{n+1})$, respectively.

In this section, we showed how baby steps and i-chains are used to compute the regulator and a system of fundamental units of a cubic function field of positive unit rank. Further, we considered the structure of infrastructures to help explain why Theorems 5.4.4 and 5.4.5 are true. In Section 6.1, we will apply the results of Theorems 5.4.4 and 5.4.5 to outline an algorithm for computing a system of fundamental units and will modify these algorithms via (5.7) and (5.8) to compute the S-regulator of a purely cubic function field of positive unit rank.

5.4.3 Symmetries in Unit Rank 2 Infrastructures

We conclude this chapter by making further observations on the structure of the infrastructure of a purely cubic function field of unit rank 2 based on our results on baby steps and i-chains. The first result will show that taking each of the three possible i-baby steps in succession is the identity operation with high probability for large q. The subsequent discussion will then describe the three-fold symmetry of the baby step operation, followed by its extension to a similar symmetry exhibited by i-chains. While the aim of this discussion is to ultimately apply these observations to improve fundamental unit and regulator computations, we will not consider how this could be accomplished in this thesis.

Proposition 5.4.6 *Let $\{i, j, k\} = \{0, 1, 2\}$. If $\mathcal{O}$ has unit rank 2 and $D_1, D_2, D_3 \in \mathcal{R}_{\mathbf{C}}$ such that $\deg((D_n)_S) = g$, for all $1 \le n \le 3$, $D_2 = bs_i(D_1)$, and $D_3 = bs_j(D_2)$, then $D_1 = bs_k(D_3)$. In other words, $D_1 = bs_k(bs_j(bs_i(D_1)))$.*

Proof: Without loss of generality, suppose $i = 0$, $j = 1$, and $k = 2$. Let $D_4 = bs_2(D_3)$. We will show that $D_4 = D_1$. For $1 \le n \le 3$, let $\phi_n \in K^*$ such that $\Psi(D_{n+1}) = \langle \phi_n \rangle \, \Psi(D_n)$. Since $\deg((D_n)_S) = g$, for all $1 \le n \le 3$, we have $\delta_{D_1}(D_2) = (1, -1, 0)$ and $\delta_{D_2}(D_3) = (0, 1, -1)$ by Corollary 5.3.11.

Suppose that $\deg((D_4)_S) < g$. By (5.5), we have $\operatorname{div}(\phi_3)^S = (\deg((D_4)_S) - g)\infty_2 + a(\infty_0 - \infty_2) + b(\infty_1 - \infty_2)$, where $a > 0$ and $b \ge 0$, so that $\delta_{D_3}(D_4) = (-a, -b, g - \deg((D_4)_S) + a + b)$. Thus, $\delta_{D_1}(D_4) = (1 - a, -b, g - \deg((D_4)_S) + a + b - 1)$. Now if $\alpha \in K^*$ such that $\Psi(D_1) = \langle \alpha \rangle \, \Psi(D_4)$, then $\deg(\alpha) = 1 - a \le 0$ and $\deg(\alpha') = -b \le 0$. If $\deg(\alpha'') \le 0$, then $D := D_1 + \operatorname{div}(\alpha)$ would

be such that $D \sim D_1$, $D_S = (D_4)_S$ is effective, $\deg(D_S) = \deg((D_4)_S) < g = \deg((D_1)_S)$, and $D^S = D_1^S + \mathrm{div}(\alpha)^S \geq D_1^S$, contradicting the fact that D_4 is distinguished. Thus, $\deg(\alpha'') = g - \deg((D_4)_S) + a + b - 1 > 0$, so $\deg(\phi_3'') = g - \deg((D_4)_S) + a + b \geq 2$. Let $\mathfrak{f}_n = \Psi(-D_n)$, for $1 \leq n \leq 4$. Then $\mathfrak{f}_3 = \left\langle (\phi_1\phi_2)^{-1} \right\rangle \mathfrak{f}_1$ and $\mathfrak{f}_4 = \left\langle \phi_3^{-1} \right\rangle \mathfrak{f}_3$. By assumption, $\phi_3 \in \mathcal{H}_{\mathfrak{f}_3,2}(1)$ is minimal under $\leq_2$. Also, since $1 \in \mathfrak{f}_1$, we have $(\phi_1\phi_2)^{-1} \in \mathfrak{f}_3$, and by construction, we have $\deg\left((\phi_1''\phi_2'')^{-1}\right) = 1$, $\deg\left((\phi_1\phi_2)^{-1}\right) = -1$, and $\deg\left((\phi_1'\phi_2')^{-1}\right) = 0$. Therefore, $(\phi_1\phi_2)^{-1} \in \mathcal{H}_{\mathfrak{f}_3,2}(1)$ and $(\phi_1\phi_2)^{-1} \leq_2 \phi_3$, which contradicts the minimality of ϕ_3 under $\leq_2$.

Thus, $\deg((D_4)_S) = g$, so $\delta_{D_3}(D_4) = (-1, 0, 1)$ and $\delta_{D_1}(D_4) = (0, 0, 0)$. By Theorem 3.3.18, no two infrastructure divisors have the same distance, so $D_1 = D_4$. □

Based on the assumption that the norms of the distinguished ideals of $\mathcal{O}$ are uniformly distributed in $\{f(x) \in \mathbb{F}_q[x] \mid \deg(f(x)) \leq g\}$, we expect that if $D \in \mathcal{R}_{\mathbf{C}}$ is chosen at random, then $D = bs_k(bs_j(bs_i(D)))$ with probability $1 - O(1/q)$. To see this, we note that Theorem 5.3.10 suggests that it is possible for more than one divisor to map to a divisor, $D \in \mathcal{R}_{\mathbf{C}}$, in which $\deg(D_S) < g$, via the baby step operation. Based on experimental evidence, we have $|\{bs_i(D) \mid D \in \mathcal{R}_{\mathbf{C}}\}| = |\mathcal{R}_{\mathbf{C}}|\left(1 - 1/q + O\left(q^{-g/2}\right)\right)$, for any fixed $i = 0, 1, 2$. In other words, bs_i is almost bijective. From our assumption, we also have $|\mathcal{R}_{\mathbf{C}}|\left(1 - 1/q + O\left(q^{-g/2}\right)\right)$ divisors $D \in \mathcal{R}_{\mathbf{C}}$ such that $\deg(D_S) = g$, at most $|\mathcal{R}_{\mathbf{C}}|\left(2/q - O\left(q^{-g/2}\right)\right)$ of which map to a divisor whose finite part has degree less than g. Thus, the probability that $\deg(D_S) = \deg(bs_i(D)_S) = g$ is $1 - O(1/q)$. By a similar argument, the probability that $\deg(bs_j(bs_i(D))_S) = g$ as well is $1 - O(1/q)$. So, by Proposition 5.4.6, we have $D = bs_k(bs_j(bs_i(D)))$ with probability $1 - O(1/q)$.

The following lemma will be used to establish the results on symmetry.

Lemma 5.4.7 *If K is a purely cubic function field of unit rank 2, $\mathfrak{f}$ a fractional ideal of $\mathcal{O}$, and $\phi = \phi_{\mathfrak{f},i}(1)$, for some $0 \leq i \leq 2$, then $\phi' = \phi_{\mathfrak{f}',i+2}(1)$ and $\phi'' = \phi_{\mathfrak{f}'',i+1}(1)$, where the subscripts are taken modulo 3.*

Proof: If $\phi = \phi_{\mathfrak{f},i}(1)$, then the ordered triple $\left(\deg\left(\phi^{(i)}\right), \deg\left(\phi^{(i+1)}\right), \deg\left(\phi^{(i+2)}\right)\right)$ is minimal in $\mathcal{H}_{\mathfrak{f},i}(1)$ under the lexicographical ordering. (Throughout the proof, all subscripts and superscripts will be considered modulo 3.) Since $\deg\left(\phi^{(i)}\right) > 0$ and $\deg\left(\phi^{(i+1)}\right), \deg\left(\phi^{(i+2)}\right) \leq 0$, with the inequality being strict in at least one case, we have $\deg\left((\phi')^{(i+2)}\right) > 0$ and $\deg\left((\phi')^{(j)}\right) \leq 0$ for both $j \not\equiv i + 2 \pmod 3$, and $\deg\left((\phi')^{(j)}\right) < 0$ for at least one $j \not\equiv i + 2 \pmod 3$. Since $\phi' \in \mathfrak{f}'$, it follows that $\phi' \in \mathcal{H}_{\mathfrak{f}',i+2}(1)$. Now suppose $\theta = \phi_{\mathfrak{f}',i+2}(1)$, so that $\theta \leq_{i+2} \phi'$. Then $\left(\deg\left(\theta^{(i+2)}\right), \deg\left(\theta^{(i)}\right), \deg\left(\theta^{(i+1)}\right)\right)$ appears before $\left(\deg\left(\phi^{(i)}\right), \deg\left(\phi^{(i+1)}\right), \deg\left(\phi^{(i+2)}\right)\right)$ under the lexicographical ordering. However, this fact and the minimality of ϕ in $\mathcal{H}_{\mathfrak{f},i}(1)$ under $\leq_i$, imply that $\theta'' \leq_i \phi \leq_i \theta''$, so $\theta'' = a\phi$, for some $a \in \mathbb{F}_q^*$. Thus, $\theta = a\phi'$, and $\phi' = \phi_{\mathfrak{f}',i+2}(1)$. Using a similar argument, we have $\phi'' = \phi_{\mathfrak{f}'',i+1}(1)$. □

We now show that the i-neighbors of a divisor, $D \in \mathcal{R}_{\mathbf{C}}$, exhibit an interesting symmetry with the i-neighbors of the conjugates of D in their respective infrastructures.

Theorem 5.4.8 *If K is a purely cubic function field of unit rank 2, $\mathfrak{a} \in \mathbf{C} \in Cl(\mathcal{O})$ a distinguished ideal, $D = \Psi^{-1}(\mathfrak{a}) \in \mathcal{R}_{\mathbf{C}}$, and $E = bs_i(D)$, for some $0 \leq i \leq 2$, then $E' = bs_{i+2}(D') \in \mathcal{R}_{\mathbf{C}'}$ and $E'' = bs_{i+1}(D'') \in \mathcal{R}_{\mathbf{C}''}$, where the subscripts are considered modulo 3.*

Proof: By Corollary 3.4.7, if $\mathcal{O}$ has unit rank 2, then $K/\mathbb{F}_q(x)$ is Galois, so if $\mathfrak{a} \in \mathbf{C}$ is a distinguished ideal of $\mathcal{O}$, then $\mathfrak{a}^{(j)} \in \mathbf{C}^{(j)}$ is a distinguished ideal of $\mathcal{O}$ and $D^{(j)} = \Psi^{-1}\left(\mathfrak{a}^{(j)}\right) \in \mathcal{R}_{\mathbf{C}^{(j)}}$, for $j = 1, 2$, by Lemma 3.4.5. Let $0 \leq i \leq 2$. If $E = bs_i(D)$ and $\mathfrak{b} = \Psi(E)$, then there is some $\phi \in K^*$ such that $\mathfrak{b} = \langle \phi \rangle \mathfrak{a}$. If $\mathfrak{f} = \mathfrak{a}^{-1}$, then $\phi = \phi_{\mathfrak{f},i}(1)$. By Lemma 5.4.7, $\phi' = \phi_{\mathfrak{f}',i+2}(1)$ and $\phi'' = \phi_{\mathfrak{f}'',i+1}(1)$. Therefore, $\mathfrak{b}' = (\langle \phi \rangle \mathfrak{a})' = \langle \phi' \rangle \mathfrak{a}'$ and $E' = \Psi^{-1}(\mathfrak{b}') = \Psi^{-1}\left(\langle \phi' \rangle \mathfrak{a}'\right)$, so $E' = bs_{i+2}(D')$. By a similar argument, we have $E'' = bs_{i+1}(D'') \in \mathcal{R}_{\mathbf{C}''}$. □

If $D \in \mathcal{R}_{\mathbf{C}}$ such that $\Psi(D)$ is the product of ramified primes, then we can say more.

Corollary 5.4.9 *Suppose K is a purely cubic function field of unit rank 2. Let $\mathfrak{a} \in \mathbf{C} \in Cl(\mathcal{O})$ be a distinguished ideal such that either $\mathfrak{a} = \mathcal{O}$ or $\mathfrak{a}$ is the product of ramified prime ideals. If $D = \Psi^{-1}(\mathfrak{a}) \in \mathcal{R}_{\mathbf{C}}$ and $E = bs_i(D)$, for some $0 \leq i \leq 2$, then $E' = bs_{i+2}(D) \in \mathcal{R}_{\mathbf{C}}$ and $E'' = bs_{i+1}(D) \in \mathcal{R}_{\mathbf{C}}$, where the subscripts are considered modulo 3.*

Proof: By Lemma 3.4.5, if $\mathfrak{a} = \mathcal{O}$ or is the product of ramified prime ideals and $D = \Psi^{-1}(\mathfrak{a})$, then $D = D' = D''$. Since $\mathfrak{a}$ is distinguished, $D^{(j)} \in \mathcal{R}_{\mathbf{C}}$, for $0 \leq j \leq 2$. If $E = bs_i(D)$, for some $0 \leq i \leq 2$, then by Theorem 5.4.8, we have $E' = bs_{i+2}(D) \in \mathcal{R}_{\mathbf{C}}$ and $E'' = bs_{i+1}(D) \in \mathcal{R}_{\mathbf{C}}$, where the subscripts are considered modulo 3. □

In the next two results, we extend the results of Theorem 5.4.8 and Corollary 5.4.9 to i-chains to show how this symmetry extends more broadly.

Theorem 5.4.10 *If K is a purely cubic function field of unit rank 2, $\mathfrak{a} \in \mathbf{C} \in Cl(\mathcal{O})$ a distinguished ideal, $D = D_0 = \Psi^{-1}(\mathfrak{a}) \in \mathcal{R}_{\mathbf{C}}$, and $\mathcal{R}_{\mathbf{C},i}(D) = (D_j)_{j \in \mathbb{N}_0}$, where $D_{j+1} = bs_i(D_j)$ is the i-chain of D, for some $0 \leq i \leq 2$, then $\mathcal{R}_{\mathbf{C}',i+2}(D') = (D'_j)_{j \in \mathbb{N}_0}$ and $\mathcal{R}_{\mathbf{C}'',i+1}(D'') = (D''_j)_{j \in \mathbb{N}_0}$, where the subscripts are considered modulo 3.*

Proof: If $D_1 = bs_i(D)$, then from Theorem 5.4.8, we have $bs_{i+2}(D') = D'_1 \in \mathcal{R}_{\mathbf{C}'}$ and $bs_{i+1}(D'') = D''_1 \in \mathcal{R}_{\mathbf{C}''}$. The desired result then follows inductively. □

Corollary 5.4.9 and Theorem 5.4.10 give rise to the following result.

Corollary 5.4.11 *Suppose K is a purely cubic function field of unit rank 2. Let $\mathfrak{a} \in \mathbf{C} \in Cl(\mathcal{O})$ be a distinguished ideal such that either $\mathfrak{a} = \mathcal{O}$ or $\mathfrak{a}$ is the product of ramified prime ideals. If $D = D_0 = \Psi^{-1}(\mathfrak{a}) \in \mathcal{R}_{\mathbf{C}}$, and $\mathcal{R}_{\mathbf{C},i}(D) = (D_j)_{j \in \mathbb{N}_0}$, where $D_{j+1} = bs_i(D_j)$ is the i-chain of D, for some $0 \leq i \leq 2$, then $\mathcal{R}_{\mathbf{C},i+2}(D) = (D'_j)_{j \in \mathbb{N}_0}$ and $\mathcal{R}_{\mathbf{C},i+1}(D) = (D''_j)_{j \in \mathbb{N}_0}$, where the subscripts are considered modulo 3.*

Proof: By Lemma 3.4.5, if $\mathfrak{a} = \mathcal{O}$ or is the product of ramified prime ideals and $D = \Psi^{-1}(\mathfrak{a})$, then $D = D' = D''$. Since $\mathfrak{a}$ is distinguished, $D^{(j)} \in \mathcal{R}_{\mathbf{C}}$, for $0 \leq j \leq 2$. If $\mathcal{R}_{\mathbf{C},i}(D) = (D_j)_{j \in \mathbb{N}_0}$, then by Theorem 5.4.10, we immediately have $\mathcal{R}_{\mathbf{C},i+2}(D) = (D'_j)_{\mathbb{N}_0}$ and $\mathcal{R}_{\mathbf{C},i+1}(D) = (D''_j)_{j \in \mathbb{N}_0}$, where the subscripts are considered modulo 3. □

These results imply that there is a three-fold symmetry in principal unit rank 2 infrastructures. Specifically, this symmetry is via the action of $\mathrm{Gal}(K/\mathbb{F}_q(x))$. In light of (3.7), we see that a divisor, $D \in \mathcal{R}$ and its conjugates are symmetric about 0 in terms of distance. Corollary 5.4.11 also implies that we may compute any i-chain of 0 in $\mathcal{R}$ from another chain of 0 by simply conjugating the divisors. In practice, this requires conjugating the basis elements of the corresponding fractional

ideals and reducing the bases. We also immediately know the distances of the elements of the i-chain, by (3.7).

In this chapter, we defined the notion of a reduced basis of a fractional ideal and showed how to compute such a basis. We then used this basis to describe how to reduce a fractional ideal to an equivalent reduced fractional ideal "close to" it and how to compute a baby step. We then combined the ideal reduction techniques with the ideal arithmetic of Chapter 4 to define the giant step operation, inverses, and the divisor below $y \in \mathbb{N}_0$. Finally, we applied the baby step operation and our results on the structure of infrastructures to show, in theory, how to compute the S-regulator and a system of fundamental units of a purely cubic function field of positive unit rank. In the next chapter, we will outline the steps required to compute the S-regulator and system of fundamental units in practice and will show how to use the infrastructure arithmetic described in this chapter to speed the regulator calculations up significantly.

Chapter 6

Computing Class Numbers and Regulators

In this chapter, we will present a number of algorithms for computing systems of fundamental units, S-regulators, and divisor class numbers of purely cubic function fields. While none of the algorithms are new, we will present results on the first implementations of the algorithm of Scheidler and Stein [SS07] to compute the divisor class number of a purely cubic function field. We will also apply their method to compute the S-regulator of a purely cubic function field of unit rank 1, extending the corresponding algorithm for hyperelliptic function fields in [SW99] to the cubic setting. (In the next chapter, we will offer some thoughts on how this algorithm may be adapted to purely cubic function fields of unit rank 2.)

In Section 6.1, we will describe the fundamental unit and regulator algorithms due to Scheidler, Stein, et al. in [SS00] and [LSY03], for purely cubic function fields of unit rank 1 and 2, respectively. In Section 6.2, we describe and compare the running times of two very important algorithms that are the core of several methods used to compute class numbers and regulators: Shanks' Baby Step-Giant Step algorithm [Sha71] and Pollard's Kangaroo (or Lambda) algorithm [Pol78]. In Section 6.3, we will describe Scheidler and Stein's method to compute the divisor class number of a purely cubic function field, and will present algorithms to compute these class numbers for function fields of unit ranks 0 and 1. In the unit rank 1 case, we will show how to compute the S-regulator as well. In particular, the regulator algorithm will run much faster than the corresponding algorithm in Section 6.1. We will present results on implementations of these methods, including data to optimize the expected running times, in Section 6.4. In particular, we computed the 28 decimal digit divisor class numbers of two purely cubic function fields of genus 3: one of unit rank 0 and one of unit rank 1. We also computed the 25 decimal digit divisor class numbers of two purely cubic function fields of genus 4: one of unit rank 0 and one of unit rank 1. For the unit rank 1 examples, we factored the divisor class numbers into the ideal class numbers and the respective 26 and 24 decimal digit S-regulators. Bauer, Teske, and Weng [BTW05, Wen06] have computed much larger divisor class numbers of genus 3 purely cubic function fields, but the 28 and 25 decimal digit examples are the largest divisor class numbers computed for any cubic function field not generated by a Picard curve. Likewise, the S-regulators are the largest ever computed for any cubic function field. We conclude by estimating the time required to compute divisor class numbers and S-regulators of cubic function fields of unit rank 0 and 1 for a range of larger base fields, $\mathbb{F}_q$.

6.1 Fundamental Units and Regulators

In this section, we will describe procedures to compute the system of fundamental units and regulators of cubic function fields of unit ranks 1 and 2. The algorithms follow from applying baby steps to our discussion in Section 5.4.2 on i-chains in the principal infrastructure, $\mathcal{R}$, and their relationship to units. The first algorithm to compute the system of fundamental units of a purely cubic function field was described by Mang [Man87], but is only feasible for function fields of small characteristic and genus. The procedures described here are based on the method due to Voronoi [Vor96] for computing the fundamental unit of a cubic number field. Voronoi's method was extended to compute fundamental units in number fields of unit rank 1 and 2 by Buchmann [Buc85]. Buchmann's method, in turn, was modified to compute fundamental units and regulators of purely cubic function fields of unit rank 1 by Scheidler and Stein in [SS00] and of unit rank 2 by Lee, Scheidler, and Yarrish in [LSY03]. We describe the fundamental unit algorithms first and will then follow with the modifications to yield the regulator directly.

6.1.1 Computing Fundamental Units

In this section, we present two algorithms: the first will compute the fundamental unit of positive degree, ϵ, of a purely cubic function field of unit rank 1, and the second finds a system of fundamental units, $\{\epsilon_1, \epsilon_2\}$ of a purely cubic function field of unit rank 2. In each case, we will briefly describe the procedure, formalize the steps in an algorithm, give a justification for its correctness, and finally analyze the running time.

To compute ϵ in the unit rank 1 case, we will traverse the entire infrastructure via baby steps, beginning at 0. The algorithm will compute the minima of the Voronoi chain corresponding to each infrastructure divisor until we determine the element θ_l of the Voronoi chain, where $l = |\mathcal{R}|$. This procedure is also outlined in Algorithm 6.4 of [SS00].

Algorithm 6.1.1 (Algorithm 6.4 of [SS00]) Fundamental Unit Computation - Unit Rank 1
Input: $q \equiv 2 \pmod 3$*; monic, relatively prime, and square-free polynomials* $G, H \in \mathbb{F}_q[x]$ *such that* $3 \mid \deg(GH^2)$*; and* $K = \mathbb{F}_q(C)$, $C : Y^3 = GH^2$, *so that* $\operatorname{sig}(K) = (1, 1; 1, 2)$.
Output: *The fundamental unit of positive degree,* ϵ, *of* K.

1. *Set* $D := 0$ *and* $\epsilon := 1$.
2. *Do:*
   a. *Compute* $E := bs_0(D)$ *and* α, *such that* $\Psi(E) = \langle \alpha \rangle \, \Psi(D)$, *via Algorithm 5.3.14.*
   b. *Set* $D := E$ *and* $\epsilon := \alpha\epsilon$.

   While $D \neq 0$.
3. *Output* ϵ.

The correctness of this algorithm follows from Theorem 5.4.4. If $\mathcal{R} = \{0 = D_0, D_1, \ldots, D_{l-1}\}$, then at each iteration, i, of Step 2, the variable ϵ stores the minimum θ_i corresponding to $D_i \in \mathcal{R}$. By Theorem 5.4.4, the fundamental unit is $\epsilon = \theta_l$, which is found when the baby steps arrive back at 0.

The running time of Algorithm 6.1.1 is given by Theorem 6.5 of [SS00], but we give a slightly different argument to improve the bound on l by a factor of 2.

Proposition 6.1.2 *Algorithm 6.1.1 requires $l \leq R^S = O(q^g)$ baby steps in the principal infrastructure, $\mathcal{R}$, and multiplications in K.*

Proof: First, Steps 1 and 3 have negligible time and storage requirement, so we will only consider Step 2. As such, Algorithm 6.1.1 requires $l = |\mathcal{R}|$ baby steps in $\mathcal{R}$ and multiplications in K. From Proposition 3.4.4, we have $l \leq R^S$. Since $R^S \mid h$, the Hasse-Weil bounds on h give the upper bound $\left(\sqrt{q}+1\right)^{2g}$ on h, and hence, on R^S. Thus, Algorithm 6.1.1 requires $l = O(q^g)$ baby steps in $\mathcal{R}$ and multiplications in $\mathcal{O}$. □

The corresponding algorithm to determine the system of fundamental units, $\{\epsilon_1, \epsilon_2\}$, of a unit rank 2 cubic function field combines Algorithms 5.1 and 5.2 of [LSY03] and is motivated by Theorem 5.4.5. To find ϵ_1, we find the period of $\mathcal{R}_0$, the 0-chain of $\mathcal{R}$. The first unit is the primitive unit of $\mathcal{R}_0$. To find the second unit, if the period of $\mathcal{R}_0$ begins at D, then we compute $\mathcal{R}_2(D)$, the 2-chain of D, until we intersect $\mathcal{R}_0$ again. The second unit is then determined from minima associated with the infrastructure divisors on the two chains. The algorithm we present is slightly different from the algorithms presented in [LSY03]. Here, we reduce the storage requirement by storing the elements, α_i, found in each baby step, rather than the minima, θ_i, corresponding to the divisors on $\mathcal{R}_0$, the 0-chain of $\mathcal{R}$.

Algorithm 6.1.3 (Algorithms 5.1 and 5.2 of [LSY03]) System of Fundamental Units Computation - Unit Rank 2
Input: $q \equiv 1 \pmod 3$*; monic, relatively prime, and square-free polynomials* $G, H \in \mathbb{F}_q[x]$ *such that* $3 \mid \deg(GH^2)$*; and* $K = \mathbb{F}_q(C)$*,* $C : Y^3 = GH^2$*, so that* $\operatorname{sig}(K) = (1, 1; 1, 1; 1, 1)$.
Output: *A system of fundamental units,* $\{\epsilon_1, \epsilon_2\}$*, of* K.

1. *Set* $I := N := \varnothing$*,* $D := 0$*, and* $i := 0$.
2. *Do:*
   a. *Set* $I[i] := D$.
   b. *Compute* $E := bs_0(D)$ *and* α*, such that* $\Psi(E) = \langle \alpha \rangle \Psi(D)$*, via Algorithm 5.3.14.*
   c. *Set* $N[i] := \alpha$*,* $D := E$*, and* $i := i + 1$.

   While $D \notin I$.
3. *If* $D = I[p]$*, remove the first* p *elements of* I *and* N.
4. *Set* $l := |I|$ *and* $\epsilon_1 := \alpha \prod_{i=0}^{l-1} N[i]$.
5. *Set* $\epsilon_2 := 1$ *and* $D := I[0]$.
6. *Do:*
   a. *Compute* $E := bs_2(D)$ *and* α*, such that* $\Psi(E) = \langle \alpha \rangle \Psi(D)$*, via Algorithm 5.3.14.*
   b. *Set* $D := E$ *and* $\epsilon_2 := \epsilon_2 \alpha$.

While $D \notin I$.

7. *If* $D = I[t]$, *set* $\epsilon_2 := \epsilon_2 \prod_{i=0}^{t-1} N[i]^{-1}$.

8. *Output* $\{\epsilon_1, \epsilon_2\}$.

The correctness of Algorithm 6.1.3 follows from Theorem 5.4.5 and an argument similar to that of Algorithm 6.1.1. Here, we give a formal analysis of the running time of Algorithm 6.1.3.

Proposition 6.1.4 *Algorithm 6.1.3 requires at most* $2R^S = O(q^g)$ *baby steps in the principal infrastructure,* $\mathcal{R}$, *at most* $3R^S = O(q^g)$ *multiplications in* K, *and at most* $R^S = O(q^g)$ *inversions in* K. *The storage requirement is at most* $R^S = O(q^g)$ *divisors and elements of* K.

Proof: The steps that dominate the running time of Algorithm 6.1.3 are Steps 2, 4, 6, and 7. All other steps require a negligible amount of time. We will first analyze the computation of ϵ_1; that is, Steps 2 and 4. Step 2 terminates when the array I has stored each divisor of $\mathcal{R}_0$ and a match is found, determining the period of $\mathcal{R}_0$. Since $\mathcal{R}_0 \subseteq \mathcal{R}$ and $|\mathcal{R}| \leq R^S$, by Proposition 3.4.4, the number of infrastructure divisors stored by I and the number of functions, $\alpha_i \in K$, stored by N are each bounded above by R^S. Since $R^S \mid h$, the Hasse-Weil bounds on h give the upper bound $\left(\sqrt{q}+1\right)^{2g}$ on h, and hence, on R^S and the size of I and N. It follows that the storage requirement of Algorithm 6.1.3 is $|I| = |N| \leq R_x = O(q^g)$ divisors and elements of K. Likewise, Step 2 requires $|\mathcal{R}_0| \leq |\mathcal{R}| \leq R^S = O(q^g)$ baby steps and Step 4 requires $|\mathcal{R}_0| \leq |\mathcal{R}| \leq R^S = O(q^g)$ multiplications in K.

Next, we consider the computation of ϵ_2 in Steps 6 and 7. Each iteration of Step 6 yields distinct infrastructure divisors until a divisor on the period of $\mathcal{R}_0$ is found. We note that if these divisors are not all distinct, then Step 6 would find a period in $\mathcal{R}_2(D)$, the 2-chain of $D = I[p]$, before finding a match in $\mathcal{R}_0$. If this occurs, then the loop in Step 6 would never terminate, contradicting Theorem 5.4.5, which guarantees that this loop is finite. It follows that the elements of $\mathcal{R}_2(D)$ found in Step 6 are distinct from each other and distinct from the divisors in the period of $\mathcal{R}_0$, until the last iteration. Thus, Step 6 requires at most $|\mathcal{R}| - l$ baby steps, where l is the length of the period of $\mathcal{R}_0$. Again, by Proposition 3.4.4, Step 6 requires at most $|\mathcal{R}| - l < |\mathcal{R}| \leq R^S = O(q^g)$ baby steps and multiplications in $\mathcal{O}$. Finally, Step 7 requires $t \leq l = |\mathcal{R}_0| \leq R^S = O(q^g)$ multiplications and inversions in K. In total, the computational requirement of Algorithm 6.1.3 is at most $2|\mathcal{R}| \leq R^S = O(q^g)$ baby steps, at most $3|\mathcal{R}| \leq R^S = O(q^g)$ multiplications in K, and at most $|\mathcal{R}| \leq R^S = O(q^g)$ inversions in K. □

Although we may find the S-regulator of a cubic function field from its system of fundamental units, a far more efficient means of computing R^S only stores the relative distances of each step rather than the minima. This is the main idea behind the following modifications to compute the S-regulator of a cubic function field.

6.1.2 Computing Regulators

In this section, we will modify Algorithms 6.1.1 and 6.1.3, applying (5.7) and (5.8), to compute the S-regulator, R^S, of purely cubic function fields of unit ranks 1 and 2 directly. As such, the regulator algorithms follow from Theorems 5.4.4 and 5.4.5, respectively.

In the unit rank 1 case, we replace the multiplication of the elements, α, in Step 2.b of Algorithm 6.1.1 with addition by $\deg(\alpha) = \delta_D(bs_0(D))$ in Step 2.a here. In this way, we will compute the distance of each divisor in $\mathcal{R}_0$ at each step. If R_x is the desired output, then multiply the result by 2.

Algorithm 6.1.5 (Algorithm 6.7 of [SS00]) S-Regulator Computation - Unit Rank 1
Input: $q \equiv 2 \pmod 3$*; monic, relatively prime, and square-free polynomials* G, $H \in \mathbb{F}_q[x]$ *such that* $3 \mid \deg(GH^2)$*; and* $K = \mathbb{F}_q(C)$, $C : Y^3 = GH^2$*, so that* $\operatorname{sig}(K) = (1, 1; 1, 2)$.
Output: *The* S*-regulator,* R^S*, of* K.

1. *Set* $D := 0$ *and* $R := 0$.
2. *Do:*
   a. *Compute* $E := bs_0(D)$ *and set* $R := R + \delta_D(E)$ *via Algorithm 5.3.14.*
   b. *Set* $D := E$.

   While $D \neq 0$.
3. *Output* $R^S := R/2$.

The variable R stores the distance of the divisor E in Step 2.a. Thus, the correctness of Algorithm 6.1.5 follows from Theorem 5.4.4, (5.7), and the fact that distances in $\mathcal{R}$ are unique modulo $\deg(\epsilon) = R_x = 2R^S$, by definition.

The proof of the running time is completely analogous to the proof of Proposition 6.1.2, so we omit it.

Proposition 6.1.6 *Algorithm 6.1.5 requires* $l \leq R^S = O(q^g)$ *baby steps in the principal infrastructure.*

Though the asymptotic running times of the fundamental unit and the regulator algorithms are the same, the latter is significantly faster because we replace costly function field multiplications with fast integer additions in Step 2. The factor of this speed up is therefore absorbed by the O-constant.

The corresponding algorithm for unit rank 2 cubic function fields follows from Theorem 5.4.5 and (5.8) and is Algorithm 6.1 of [LSY03]. We will replace storage of the elements $\alpha \in N$ in Step 2 of Algorithm 6.1.3 with the storage of the 0- and 1-components of $\delta_D(bs_0(D)) = (\delta_0, \delta_1, \delta_2)$. As with Algorithm 6.1.5, we replace the multiplication of the elements α in Step 4 of Algorithm 6.1.3 with the addition of the 0- and 1-components of these relative distances: $\deg(\alpha) = \delta_0$ and $\deg(\alpha') = \delta_1$. This will determine $\deg(\epsilon_1)$ and $\deg(\epsilon_1')$. Likewise, we replace the multiplication of α in Step 6 with the addition of the 0- and 1-components of $\delta_D(bs_2(D))$. Finally, we replace the inversions and multiplications in Step 7 with the subtraction of the components of the corresponding relative distances along the period of $\mathcal{R}_0$ to compute $\deg(\epsilon_2)$ and $\deg(\epsilon_2')$. Lastly, we use the definition of the regulator of a cubic function field of unit rank 2, found in Section 3.2.5, to find $R_x = R^S$.

Algorithm 6.1.7 (Algorithm 6.1 of [LSY03]) Regulator Computation - Unit Rank 2
Input: $q \equiv 1 \pmod 3$*; monic, relatively prime, and square-free polynomials* G, $H \in \mathbb{F}_q[x]$ *such that* $3 \mid \deg(GH^2)$*; and* $K = \mathbb{F}_q(C)$, $C : Y^3 = GH^2$*, so that* $\operatorname{sig}(K) = (1, 1; 1, 1; 1, 1)$.
Output: *The regulator,* R^S*, of* K.

1. *Set* $I := N := \varnothing$, *indexed from* 0, $D := 0$, *and* $i := 0$.

2. *Do:*

   a. *Set* $I[i] := D$.

   b. *Compute* $E := bs_0(D)$ *and set* $N[i] := (\delta_0, \delta_1)$, *where* $\delta_D(E) = (\delta_0, \delta_1, \delta_2)$, *via Algorithm 5.3.14.*

   c. *Set* $D := E$ *and* $i := i + 1$.

   While $D \notin I$.

3. *If* $D = I[p]$, *remove the first* p *elements of* I *and* N.

4. *Set* $l := |I|$, $e_{1,1} := \sum_{i=0}^{l-1} N[i]_0$, *and* $e_{1,2} := \sum_{i=0}^{l-1} N[i]_1$, *where* $N[i]_j$ *is the* j*-component of distance in* $N[i]$.

5. *Set* $e_{2,1} := e_{2,2} := 0$ *and* $D := I[0]$.

6. *Do:*

   a. *Compute* $E := bs_2(D)$ *and* $(\delta_0, \delta_1, \delta_2) := \delta_D(E)$ *via Algorithm 5.3.14.*

   b. *Set* $D := E$, $e_{2,1} := e_{2,1} + \delta_0$, *and* $e_{2,2} := e_{2,2} + \delta_1$.

   While $D \notin I$.

7. *If* $D = I[t]$, *set* $e_{2,1} := e_{2,1} - \sum_{i=0}^{t-1} N[i]_0$, *and* $e_{2,2} := e_{2,2} - \sum_{i=0}^{t-1} N[i]_1$.

8. *Output* $R^S := |e_{1,1}e_{2,2} - e_{1,2}e_{2,1}|$.

The array N stores the relative distances of the divisors on $\mathcal{R}_0$ so that $e_{1,1} = \deg(\epsilon_1)$ and $e_{1,2} = \deg(\epsilon_1')$ in Step 4. Similarly, we have $e_{2,1} = \deg(\epsilon_2)$ and $e_{2,2} = \deg(\epsilon_2')$, by Theorem 5.4.5. Thus, R^S is the quantity computed in Step 8, by definition.

As with the unit rank 1 example, we speed up regulator computation by replacing the function field multiplications in Steps 4, 6b, and 7 of Algorithm 6.1.3 with integer additions in the same steps of Algorithm 6.1.7. Again, we attain a constant-time speed up that is absorbed by the O-constant. The proof of the following theorem is analogous to the proof of Proposition 6.1.4 so we also omit it.

Proposition 6.1.8 *Algorithm 6.1.7 requires at most* $2R^S = O(q^g)$ *baby steps in the principal infrastructure. The storage requirement is at most* $R^S = O(q^g)$ *divisors and at most* $2R^S = O(q^g)$ *integers.*

In this section, we described algorithms to compute the fundamental units and regulator of a purely cubic function field of positive unit rank. In each case, the algorithms require $O(q^g)$ baby steps. In addition, the fundamental unit algorithms, require $O(q^g)$ function field multiplications, and the unit rank 2 algorithms store $O(q^g)$ divisors and integers in the fundamental unit and regulator algorithms, respectively. In the next section, we will discuss some specific details regarding the implementation of these algorithms.

6.1.3 Implementation Notes

Here, we discuss issues surrounding the representation of function field elements and appropriate storage methods for the unit rank 2 algorithms, Algorithms 6.1.3 and 6.1.7.

By Section 2.4, we may represent any function field element $\alpha \in K$, as an element in one of its completions. From Sections 3.2.4 and 3.2.5, we have $K_{\infty_0} = \mathbb{F}_q \left\langle x^{-1} \right\rangle$ in both unit rank cases. Thus, for the algorithms in this section, multiplication and inversion in K will be performed in the field of Puiseux series, $\mathbb{F}_q \left\langle x^{-1} \right\rangle$. In addition, the baby step operation requires a(n) (i-)reduced basis, which in turn requires arithmetic to be performed in $\mathbb{F}_q \left\langle x^{-1} \right\rangle$. Since it is impossible to store an element $\alpha \in \mathbb{F}_q \left\langle x^{-1} \right\rangle$ with infinite precision, we instead use an *approximation* $\hat{\alpha}_n$, of α, of *precision* n. If $\alpha = \sum_{i=m}^{\infty} a_i x^{-i}$, then $\hat{\alpha}_n = \sum_{i=m}^{n} a_i x^{-i}$, so that $|\alpha - \hat{\alpha}_n| < q^{-n}$. In Section 8 of [SS00], Scheidler and Stein have found that $n = \deg(\Delta)/2$ is a sufficiently high level of precision for Algorithm 5.1.5 to give the correct results, based on experimental results. A deeper analysis, given by Scheidler in Section 7 of [Sch00], supports the choice of $n = \deg(\Delta)/2$ as a sufficient level of precision for the algorithms in this section. In the case that giant steps are being performed however, it will be necessary to reduce a fractional ideal, $\mathfrak{f}$, which is the product of two distinguished fractional ideals. In this case, the analysis in [Sch00] suggests that a precision of $n = 2g \geq \deg(d(\mathfrak{f}))$ is sufficient.

To store the function field elements in Step 2 of Algorithm 6.1.3 and the infrastructure divisors in Step 2 of Algorithm 6.1.7, we also suggest using a binary search tree or a hash table, rather than a linear list. Searching for matches would dominate the computation time of larger computations if the 0-chain divisors of $\mathcal{R}$, $\mathcal{R}_0$, were stored in a linear list. If nonlinear storage is used, however, a separate table will be needed to record the position of the function field or infrastructure elements. To use a hash table, we need a fixed size. Thus, the upper bound $|\mathcal{R}_0| \leq R^S \leq (\sqrt{q}+1)^{2g} = O(q^g)$ in Propositions 6.1.4 and 6.1.8 suffices.

Results on the implementation of Algorithm 6.1.7 are given in Table 3 of [LSY03]. In contrast to the proven upper bounds on storage and the number of baby steps, if p, l, and m are the lengths of the preperiod of $\mathcal{R}_0$, the primitive period of $\mathcal{R}_0$, and the 2-secondary period, respectively, then the computations in [LSY03] suggest that the upper bound, R^S, on $p+l$ and m tends to be far from sharp. In fact, most of the time, it was the case that $p \leq l$ and $lm \leq R_x$. Based on these examples, we conclude that the bounds given in Propositions 6.1.4 and 6.1.8 are not very sharp in general.[1] However, this will likely not make any difference in the asymptotic running time. In Section 7.2, we will discuss possibilities of improving the running time to compute the regulator of a unit rank 2 purely cubic function field using the methods of Shanks and Pollard.

There are methods to reduce the time to compute the S-regulator of a cubic function field of unit rank 1. The methods due to Shanks and Pollard were adapted to the problem of computing the regulator of a real hyperelliptic function field by Stein, Teske, and Williams [SW98, SW99, ST02b, ST05] to obtain significantly faster running times. In the next section, we will describe the Baby Step-Giant Step and Kangaroo algorithms and will apply them to the problem of computing S-regulators of purely cubic function fields of unit rank 1.

[1] In Table 3 of [LSY03], the number of infrastructure divisors stored in I is given by $p + l$, where p and l are the preperiod and period length, respectively.

6.2 Generic Group Order Computation Algorithms

There are three algorithms designed to compute the order of a generic finite abelian group, Γ: Shanks' Baby Step-Giant Step algorithm [Sha71], Pollard's Rho algorithm [Pol75], and Pollard's Kangaroo (or Lambda) algorithm [Pol78]. The heuristic running time for the Rho Method is $O\left(\sqrt{\gamma}\right)$ group operations, where $\gamma = |\Gamma|$ is the order of Γ, and does not require any information about either Γ or γ. The Baby Step-Giant Step and Kangaroo methods, however, require at least an upper bound, U, on γ, and have a deterministic and probabilistic running time of $O\left(\sqrt{U}\right)$ group operations, respectively.[2]

In this section, we will assume that we are given integers E, $U \in \mathbb{N}$ such that $\gamma \in (E-U,\, E+U)$. In this case, the Baby Step-Giant Step and Kangaroo methods may be optimized to compute γ with a deterministic and probabilistic running time of $O\left(\sqrt{U}\right)$ group operations, respectively. Assuming that the interval is sufficiently small, these methods will be faster than Pollard's Rho method. In Table 6.1 at the end of this section, we will show that, based on theoretical results and the timing of arithmetic operations, the expected time to compute the divisor class number of a purely cubic function field using the Baby Step-Giant Step method is faster than the expected time using the Kangaroo method, when the Jacobian is sufficiently small. The Baby Step-Giant Step method, however, requires the storage of $O\left(\sqrt{U}\right)$ group elements and cannot be parallelized efficiently. For larger computations, the Kangaroo method is preferable, since variants of this algorithm require very little storage and can be parallelized. In this section, we will describe both the Baby Step-Giant Step and Kangaroo methods and explain important improvements that apply in particular to the problem of computing the divisor class number, h of a purely cubic function field. We will then formalize each method in an algorithm and analyze the running time and storage requirement.

Therefore, for the remainder of this section, we will take $\Gamma = \mathcal{J}_K$ and $\gamma = h$, where $K/\mathbb{F}_q(x)$ is a cubic function field of unit rank r and genus g. In this case, we have known bounds for h, namely the Hasse-Weil interval, that is, $h \in \left(\left(\sqrt{q}-1\right)^{2g},\, \left(\sqrt{q}+1\right)^{2g}\right)$, so that one could use $E = q^g$ and $U = \left(\sqrt{q}+1\right)^{2g} - q^g + 1$. However, the approximation, E, of h is generally too rough, and the upper bound, U, on the error, $|h-E|$, is generally too large to make either the Baby Step-Giant Step or the Kangaroo methods efficient. This motivates the algorithm discussed in Section 6.3. More specifically, we will describe a method from [SS07] to compute a better estimate $E \in \mathbb{N}$, of h and a sharper upper bound, $U \in \mathbb{N}$, on the error, $|h-E|$, (so that $h \in (E-U,\, E+U)$) than those given by the Hasse-Weil bounds when $g \geq 3$. For a fixed q and g, these new parameters, E and U, are not fixed as they are with the Hasse-Weil bounds, but are generally different for each function field. In this section, we will assume that $E = O(q^g)$, being an approximation of h, and that $U < E$. We will also assume that $2(E-U) > E+U$. In this way, if we determine $h_0 \in (E-U,\, E+U)$ to be a multiple of h, then $h_0/2 < (E+U)/2 < E-U$. Thus, $h_0 = h$. It is important to note then, that this is not an unreasonable restriction for the integers E and U produced by Scheidler and Stein's method in [SS07]; only function fields of very small characteristic fail this criterion.

For both the Baby Step-Giant Step and Kangaroo methods, we will first consider totally ramified cubic function fields, and then describe adjustments to purely cubic function fields of unit rank 1.

[2] Terr's triangle method [Ter00] is a variant of the Baby Step-Giant Step method that does not require information about γ, and has a deterministic running time of $O\left(\sqrt{\gamma}\right)$ group operations. This method is slower for the situation which we will consider in this chapter, so we will not consider this variant further. See Sections 2.2 and 2.3 of [ST05] for a comparison of the running time of this variant with others.

In the unit rank 0 case, we will operate in the ideal class group instead, since $\mathcal{J}_K \cong Cl(\mathcal{O})$, and will represent the ideal classes by their unique distinguished representatives and will write $\mathfrak{g}^d$ to mean $Reduce(\mathfrak{g}^d)$, for any ideal $\mathfrak{g}$ and $d \in \mathbb{Z}$, for ease of notation. In the unit rank 1 case, we will focus on the infrastructure to determine a multiple, h_0, of R^S. (In most cases, we will in fact have $h = h_0$.)

6.2.1 Shanks' Baby Step-Giant Step Algorithm for Groups

In this section, we will describe and analyze the Baby Step-Giant Step algorithm of Shanks [Sha71] as it applies to groups, and will make specific application to the divisor class group of a purely cubic function field of signature $(3, 1)$. Recall that in this case, we have $\mathcal{J}_K \cong Cl(\mathcal{O})$, so we will consequently describe the algorithm in the language of ideals. We will give three general versions, beginning with the original version, which searches for h from the left side of the interval, $(E - U, E + U)$, to the right. The second method searches for h from the center of the interval to the ends; we will note several possible improvements that apply specifically to our case. Finally, we will show how to perform the Baby Step-Giant Step algorithm when there exist known integers d, $p \in \mathbb{N}$ such that $h \equiv d \pmod{p}$. In each case, we will briefly explain the procedure, outline the main steps, note why the algorithm works, and will analyze its running time and storage requirement. For the third version, however, we will give more extensive details of the algorithm, incorporating each improvement.

Though we will use the terms "baby step" and "giant step" here, these terms have a different meaning than the infrastructure operations of the same names, and will be defined in terms of group operations in this section. Here, baby steps are multiplications by some fixed group element and giant steps are multiplications by some fixed power, M, of that group element. The terminology stems from the following observation. If the fixed group element, $\mathfrak{g}$, is a generator of $Cl(\mathcal{O})$, then one could think of the "distance" of any group element, $\mathfrak{a} = \mathfrak{g}^d$, as the exponent d. In this way, baby steps result in an increase of 1 in distance, while giant steps, computed by multiplying by $\mathfrak{g}^M$, yield an increase of M in distance.

Originally, Shanks applied his method to compute class numbers of real quadratic number fields [Sha71], but his method generalizes to compute group orders, element orders, and discrete logarithms in any finite abelian group. Improvements to Shanks' method are described in [SW88] and generalizations to compute the regulator of a real quadratic function field are found in [SZ91, Ste92].

We first describe the original version. If $M = \lceil \sqrt{2U - 1} \rceil$, then there are unique integers, i and j, with $0 \leq i \leq M$ and $0 \leq j < M$, such that $h = E - U + iM + j$. Let $\mathfrak{g}$ be a distinguished ideal. The Baby Step-Giant Step method is motivated by the fact that $\mathfrak{g}^{E-U} * \left(\mathfrak{g}^M\right)^i = \mathfrak{g}^{-j}$. Thus, the original version has two main steps.

Algorithm 6.2.1 (Shanks [Sha71]) Baby Step-Giant Step Algorithm, Original Version

1. *Compute* baby steps*:* $\mathcal{B} = \left\{\mathfrak{g}^{-j} \mid 0 \leq j < M\right\}$ *and set* $\mathfrak{a} := \mathfrak{g}^{E-U}$.
2. *While* $\mathfrak{a} \notin \mathcal{B}$*, compute* giant steps $\mathfrak{a} := \mathfrak{a} * \mathfrak{g}^M$.

In Step 1, if $\mathfrak{g}^{-j} = \langle 1 \rangle$, then $j \mid h$, and we may restart the algorithm with this new information about h and a new ideal $\mathfrak{g}$. (We will describe later how to take advantage of such knowledge about h.) Otherwise, in Step 2, we have $\mathfrak{a} = \mathfrak{g}^{E-U+iM} = \mathfrak{g}^{-j} \in \mathcal{B}$, for some $0 \leq i \leq M$ and some $0 \leq j < M$, so from the assumption that $2(E - U) > E + U$, we are guaranteed that $h = E - U + iM + j$.

We give a brief analysis of the running time and storage. In Step 1, we require one inversion, $M-2$ ideal compositions ($\mathfrak{g}^{-2} = \mathfrak{g}^{-1} * \mathfrak{g}^{-1}, \ldots, \mathfrak{g}^{-(M-1)} = \mathfrak{g}^{-(M-2)} * \mathfrak{g}^{-1}$), and an exponentiation. In Step 2, we need one ideal composition and inversion to compute $\mathfrak{g}^M = \left(\mathfrak{g}^{-(M-1)} * \mathfrak{g}^{-1}\right)^{-1}$, and at most M ideal compositions. Therefore, this algorithm requires at most $2M - 1 + O(\log(E-U)) = 2\sqrt{2U} + O(g\log(q)) = O\left(\sqrt{U}\right)$ ideal compositions, two inversions, and stores $M = 2\sqrt{2U} + O(1) = O\left(\sqrt{U}\right)$ distinguished ideals. Although the ideal $\mathfrak{g}$ is chosen randomly, the only multiple of $\mathrm{ord}(\mathfrak{g})$ in $(E-U,\, E+U)$ is h, so the giant step phase terminates at the same i regardless of the choice of $\mathfrak{g}$. Thus, the algorithm is deterministic.

Though this choice of M optimizes the running time of the worst-case scenario (see Section 2.1 of [ST05]), we may improve the average running time, over all totally ramified cubic function fields over $\mathbb{F}_q(x)$ having genus g, with a different choice of M. In the following discussion, we analyze a slight modification to the original method.

If h is symmetrically distributed about E in the interval $(E-U,\, E+U)$, then the average value of h over all totally ramified cubic function fields over $\mathbb{F}_q(x)$ and of genus g is very close, if not equal, to E. Given this fact, if we begin making giant steps from $E-U+1$, then the expected difference between this starting point and h is $E-(E-U+1) = U-1$. If M is the length of a giant step, then we expect to make $(U-1)/M$ giant steps. To minimize the overall expected running time, we choose M so that the expected number of giant steps is also M. Thus, choosing $M = \left\lceil \sqrt{U-1} \right\rceil$, we expect to compute $(U-1)/M = \sqrt{U-1} + O(1)$ giant steps and compose $2\sqrt{U-1} + O(1) = 2\sqrt{U} + O(1)$ ideals in total. We will also require two inversions. This minimizes the average run time over all cubic function fields over a fixed constant field and genus. Since we must store each baby step, the storage requirement is $M = \sqrt{U} + O(1)$ ideals.

With more information about h or the timing of various arithmetic operations, however, we will obtain a constant-time improvement over the original approach. In the following discussion, we will consider the second version of the Baby Step-Giant Step algorithm, which searches for h beginning at the center of the interval.

Based on experimental evidence, which will be given in Section 6.4, h is indeed distributed symmetrically about E and further, h tends to lie near the middle of the interval, $(E-U,\, E+U)$, when considered over all cubic function fields over $\mathbb{F}_q(x)$ having genus g. Such a distribution is called an *Increasing Hazard Rate* (IHR) distribution in [BT00] and [ST05].[3] In this case, we can improve the expected running time by concentrating our effort on the center of the interval. In other words, rather than searching from the left edge of the interval, we will search from the middle of the interval out in both directions. This situation is described in Sections 2.1 and 2.3.2 of [ST05]. For convenience, we simply let $M = \left\lceil \sqrt{U} \right\rceil$. Then there exist unique integers $-M \le i \le M$ and $0 \le j < M$ such that $h = E + iM + j$. Since $|i|$ will tend to be closer to 0 than to M, we will take giant steps so that we will test increasing values of $|i|$ for matches. The resulting version, originally due to Stephens and Williams [SW88], has the following two steps.

Algorithm 6.2.2 (Stephens–Williams [SW88]) Baby Step-Giant Step Algorithm with Inverses

[3]This name originates from statistical analyses of the probability of a mechanism failing within a short interval of time. In our case, a "failure" is the event of finding a solution. Thus, in a system with an increasing hazard rate, the probability of a failure occurring within a short interval increases over time. This is the case for the situation we are concerned with.

1. *Compute baby steps:* $\mathcal{B} = \left\{\mathfrak{g}^{E+j} \mid 0 \le j < M\right\}$ *and set* $\mathfrak{a} := \langle 1 \rangle$.

2. *While* $\mathfrak{a}$, $\mathfrak{a}^{-1} \notin \mathcal{B}$, *compute giant steps* $\mathfrak{a} := \mathfrak{a} * \mathfrak{g}^M$ *and* $\mathfrak{a}^{-1}$.

If $\mathfrak{g}^{iM} = \mathfrak{g}^{E+j} \in \mathcal{B}$, then $h = E - iM + j$. In this way, the giant steps, $\mathfrak{g}^{iM}$, with $i > 0$, search the left half of the interval while the inverses, $\mathfrak{g}^{iM}$, with $i \le 0$, search the right half. Unless we know more about the distribution of $h \in (E - U,\, E + U)$, however, we cannot improve upon the asymptotic running time of $2\sqrt{U} + O(1)$ ideal compositions that we derived previously.

Now let $\alpha = \alpha(q,\, g)$ be the expected mean value of $|h - E|/U$ over all cubic function fields of characteristic q and genus g. If h were distributed uniformly in $(E - U,\, E + U)$, we would have $\alpha = 1/2$, but because h tends to lie closer to E than to either endpoint, we have $\alpha < 1/2$. If the time to compute an inverse in $Cl(\mathcal{O})$ is the same as the time to compose two ideals, Proposition 2.4 of [ST05], states that $M = \left\lceil \sqrt{2\alpha U} \right\rceil$ minimizes the expected running time over all cubic function fields of a fixed characteristic, q, and genus, g, resulting in a total expected running time equivalent to $2\sqrt{2\alpha U} + O(1)$ ideal compositions and a storage of $M = \sqrt{2\alpha U} + O(1)$ distinguished ideals. This results in an improvement over the previous version by a factor of $\sqrt{(2\alpha)^{-1}}$. This version is called the IHR version in [BT00] and [ST05].

In general, however, α is difficult to determine precisely, and is best approximated via experimentation. In Table 6.5 of Section 6.4, we will give experimental results that approximate α for $3 \le g \le 7$ and a few values of q for each genus. Assuming that a good approximation, $\hat{\alpha} = \hat{\alpha}(q,\, g)$, of α is known, we will use the approximation in practice. For the previous case, for example, we will choose $M = \left\lceil \sqrt{2\hat{\alpha} U} \right\rceil$.

Later, in Table 6.5, we will give experimental results that show that the computation of the inverse of an ideal in a totally ramified cubic function field is faster than a giant step, but is not trivial. Thus, we can improve the average running time to compute h over all cubic function fields of characteristic q and genus g by adjusting the choice of M to take advantage of cheap inverses. This version is called the *Cheap Inverses* (CI) version. (See the related discussion in Section 2.5 of [ST05].) For this, let $T_{I,0}$ be the time required to compute *Reduce* $(\overline{\mathfrak{a}})$, for a given distinguished ideal, $\mathfrak{a}$, (henceforth called the "ideal inverse" of $\mathfrak{a}$), $T_{G,0}$ the time required to compose two ideals in $Cl(\mathcal{O})$, and $\tau_1 = 1 + T_{I,0}/T_{G,0}$. We will show that the expected running time is minimized by choosing $M = \left\lceil \sqrt{\alpha\tau_1 U} \right\rceil$, though in practice we will choose $M = \left\lceil \sqrt{\hat{\alpha}\tau_1 U} \right\rceil$. The algorithm again proceeds exactly as before with this choice. Proposition 2.4 of [ST05] analyzes this algorithm for the cases in which $T_{I,0} = 0$ and $T_{I,0} = T_{G,0}$, but in the following proposition, we generalize the result in [ST05] slightly to include all possible timings, $T_{I,0}$, and apply it to totally ramified cubic function fields.

Proposition 6.2.3 *Let K be a cubic function field of signature $(3, 1)$ and suppose that h has an IHR distribution in the interval $(E - U,\, E + U)$. Let $T_{I,0}$ and $T_{G,0}$ be the respective times required to compute an ideal inverse and to compose two ideals in $Cl(\mathcal{O})$. Then the expected running time, over all cubic function fields over $\mathbb{F}_q(x)$ of genus g, of Algorithm 6.2.2 is minimized by computing $M = \left\lceil \sqrt{\alpha\tau_1 U} \right\rceil$ baby steps, where $\alpha = \alpha(q,\, g) \le 1/2$ is the mean value of $|h - E|/U$ over all cubic function fields over $\mathbb{F}_q(x)$ and of genus g, and $\tau_1 = 1 + T_{I,0}/T_{G,0}$. With this choice of M, the total expected running time is $\left(2\sqrt{\alpha\tau_1 U} + O(g \log(q))\right) T_{G,0}$. The storage requirement is $\sqrt{\alpha\tau_1 U} + O(1)$ distinguished ideals.*

Proof: If the mean value of $|h-E|/U$, over all cubic function fields over $\mathbb{F}_q(x)$ and genus g, is α and we compute $M = \lceil \sqrt{\alpha\tau_1 U} \rceil$ baby steps, then we expect to compute $\alpha U/M = \sqrt{\alpha U/\tau_1}+O(1)$ giant steps and inverses. The baby step phase takes time $MT_{G,0} = \left(\sqrt{\alpha\tau_1 U} + O(1)\right) T_{G,0}$ and we expect the giant step phase to finish in time $\left(\sqrt{\alpha U/\tau_1} + O(1)\right)(T_{G,0} + T_{I,0}) = \left(\sqrt{\alpha U/\tau_1} + O(1)\right)(\tau_1 T_{G,0}) = \left(\sqrt{\alpha\tau_1 U} + O(1)\right) T_{G,0}$. Therefore, the two phases of the algorithm have the same expected running time, which minimizes the total expected running time of the algorithm. We also require two exponentiations, namely $\mathfrak{g}^E$ and $\mathfrak{g}^M$, for some ideal $\mathfrak{g}$. These operations take total time $O(\log(E) + \log(M))T_{G,0} = O(g\log(q))T_{G,0}$. Thus, adding the expected running time of the two phases, the total expected running time for Algorithm 6.2.2 is $\left(2\sqrt{\alpha\tau_1 U} + O(g\log(q))\right) T_{G,0}$. Lastly, we will store $M = \sqrt{\alpha\tau_1 U} + O(1)$ ideals from the baby step phase. □

We will give estimates of τ_1, for certain q and g in Table 6.5 of Section 6.4.

The third version also searches for h beginning from the center of the interval, but only in a given congruence class. This adaptation is described in Section 2.4 of [Tes01]. Suppose it is known that there exist integers a, $b \in \mathbb{N}$ such that $h \equiv a \pmod{b}$, with $0 \leq a < b$. (In Section 6.3.7, we will give four results that will state circumstances under which such a and b can be determined.) We first change the estimate $E := E - (E \pmod{b}) + a$, so that we now have $E \equiv a \pmod{b}$ for the revised value of E. Then there exist integers $|i| \leq U/(bM)$ and $0 \leq j < M$ such that $h = E + ibM + jb$, where M is the number of baby steps that we compute. For this, we will perform baby steps and giant steps in the subinterval $\{x \in (E-U, E+U) \mid x \equiv a \pmod{b}\}$. Thus, the baby steps will have length b: $\mathcal{B} = \left\{\mathfrak{g}^E, \mathfrak{g}^{E+b}, \ldots, \mathfrak{g}^{E+(M-1)b}\right\}$ and the giant steps will be $\mathfrak{g}^{\pm bM}$, $\mathfrak{g}^{\pm 2bM}$, ... In Algorithm 6.2.4, we outline this procedure formally, incorporating the improvements of the IHR and CI versions as well.

Algorithm 6.2.4 Baby Step-Giant Step Algorithm for Class Number Computation - Unit Rank 0
Input: *q; monic, relatively prime, and square-free G, $H \in \mathbb{F}_q[x]$ such that $3 \nmid \deg(GH^2)$; a, $b \in \mathbb{N}_0$ such that $h \equiv a \pmod{b}$;*[4] *$K = \mathbb{F}_q(C)$, where $C : Y^3 = GH^2$, so that $\mathrm{sig}(K) = (3,1)$; and integers E, $U \in \mathbb{N}$ such that $|h-E| < U$ and $2(E-U) > E+U$.*
Output: *The divisor class number, h, of K.*

1. *Set $g := \deg(GH) - 1$.*
2. *Find an estimate, $\hat{\alpha} := \hat{\alpha}(q, g)$, of the mean value of $|h-E|/U$ via Table 6.5.*
3. *Set $M := \left\lceil \sqrt{\tau_1 \hat{\alpha} U/b} \right\rceil$, where τ_1 is the appropriate value in Table 6.5, and $\mathcal{B} := \varnothing$, indexed from 0.*
4. *Generate a random ideal, $\mathfrak{g}$, via Algorithm 6.3.18.*
5. *Set $E := E - (E \pmod{b}) + a$, $\mathcal{B}[0] := \mathfrak{g}^E$, and $\mathfrak{a} := \mathfrak{g}^b$.*
6. *(Baby Steps) For $1 \leq j < M$:*
   - *Set $\mathcal{B}[j] := \mathcal{B}[j-1] * \mathfrak{a}$.*
7. *Set $\mathfrak{a} := \mathfrak{g}^{bM}$, $\mathfrak{b}_0 := \mathfrak{b}_1 := \langle 1 \rangle$, and $i := 0$.*

[4] We use $b = 1$ and $a = 0$ if no non-trivial b is known.

8. *(Giant Steps) While* $\mathfrak{b}_0, \mathfrak{b}_1 \notin \mathcal{B}$*:*

   - *Set* $\mathfrak{b}_0 := \mathfrak{b}_0 * \mathfrak{a}$ *and* $i := i + 1$*.*
   - *Compute* $\overline{\mathfrak{b}_0}$ *via Lemma 4.3.3 and* $\mathfrak{b}_1 := Reduce\left(\overline{\mathfrak{b}_0}\right)$ *via Algorithm 4.5.7.*

9. *If* $\mathfrak{b}_0 = \mathfrak{g}^j$*, output* $h := E - ibM + j$*. If* $\mathfrak{b}_1 = \mathfrak{g}^j$*, output* $h := E + ibM + j$*.*

To explain why this algorithm works, we noted earlier that if $h \equiv E \equiv a \pmod{b}$, then we can write $h = E - ibM + jb$, where $|i| \leq U/(bM)$ and $0 \leq j < M$, so that $\mathfrak{g}^{E-ibM+jb} = \langle 1 \rangle$. Therefore, $\mathfrak{g}^{E+jb} = \mathfrak{g}^{ibM}$; we have $\mathfrak{g}^{E+jb} \in \mathcal{B}$ and $\mathfrak{g}^{ibM}$ is computed during the giant step phase. The proof of the running time of Algorithm 6.2.4 is analogous to the proof of Proposition 6.2.3. Thus, the result follows from the fact that the subinterval of $(E - U, E + U)$ under our consideration includes only integers in the arithmetic progression $a \pmod{b}$ and thus is shorter by a factor of b. This result therefore combines Proposition 6.2.3 with well-known results on performing the Baby Step-Giant Step procedure in a congruence class, for example, the description in Section 2.4 of [Tes01].

Proposition 6.2.5 *Let K be a cubic function field of signature $(3, 1)$. Suppose that there exist integers $a, b \in \mathbb{N}$ such that $h \equiv a \pmod{b}$ and h has an IHR distribution in the interval $(E - U, E + U)$. Let $T_{I,0}$ and $T_{G,0}$ be the respective times to compute an inverse and to compose two ideals in $Cl(\mathcal{O})$. Then the expected running time, over all cubic function fields over $\mathbb{F}_q(x)$ and of genus g, to compute h via Algorithm 6.2.4 is minimized by computing $M = \left\lceil \sqrt{\alpha\tau_1 U/b} \right\rceil$ baby steps, where $\alpha = \alpha(q, g) \leq 1/2$ is the mean value of $|h - E|/U$ over all cubic function fields over $\mathbb{F}_q(x)$ and of genus g, and $\tau_1 = 1 + T_{I,0}/T_{G,0}$. With this choice of M, the total expected running time of Algorithm 6.2.4 is $\left(2\sqrt{\alpha\tau_1 U/b} + O(g \log(q))\right) T_{G,0}$, and is deterministic. The storage requirement is $\sqrt{\alpha\tau_1 U/b} + O(1)$ distinguished ideals.*

This yields a speed-up by a factor of $\sqrt{b}$ over the situation described in Proposition 6.2.3.

We have described Shanks' Baby Step-Giant Step algorithm for groups and outlined improvements based on the distribution of h in $(E-U, E+U)$, the time to compute inverses and compositions, and finally, if it is known that h lies in a certain congruence class. In the next section, we will adapt the methods described here to operate in the infrastructure of a cubic function field of unit rank 1.

6.2.2 Shanks' Baby Step-Giant Step Algorithm for Infrastructures

If K is a purely cubic function field of unit rank 1, then the adjustments necessary to operate in the principal infrastructure, $\mathcal{R}$, of K, rather than in the Jacobian or ideal class group, as was done in the unit rank 0 case, are very natural. In the unit rank 1 setting, however, we will search for a multiple, $h_0 = h^* R^S$, of R^S by first finding a multiple, $2h_0 = h^* R_x$, of $R_x = 2R^S$. Details on determining R^S from h_0 will be discussed in Section 6.3.9. We will first describe the adaptations to the infrastructure of K and detail them in Algorithm 6.2.6. The corresponding algorithm for real quadratic function fields was given in [SW99], and we adapt their methods to the cubic setting, using the infrastructure arithmetic in [Sch01] and Section 5.3. Finally, we will prove the correctness of the algorithm, its running time, and storage requirement. As in the unit rank 0 case, we can find an estimate, E, of h and an upper bound, U, on the error $|h - E|$, and we assume that we have a good approximation, $\hat{\alpha}$, of $\alpha = \alpha(q, g)$, the expected mean value of $|h - E|/U$ over all cubic function

fields of a fixed characteristic, q, and genus, g. However, since distances are unique modulo R_x, we will search for $2h_0 \in (2(E-U),\, 2(E+U))$. Since the ideal class number, h_x, tends to be small, we expect that $h_0 = h$ and $h^* = h_x$ in most cases.

We note more differences between the unit rank 0 and unit rank 1 cases. First, in the unit rank 1 scenario, we will need to compute the infrastructure divisor below y, $D(y)$, for certain values of y, to replace the notion of computing $[\mathfrak{g}^y]$ in $Cl(\mathcal{O})$, for some ideal class $[\mathfrak{g}]$ in the unit rank 0 setting.

In addition, we will take advantage of faster baby steps in the infrastructure setting. In the infrastructure of a real hyperelliptic function field, the time to compute a baby step is significantly less than the time to compute a giant step, specifically $4g + O(1)$ versus $17g^2 + O(g)$ finite field operations [Ste01]. In Table 6.7, we will give experimental evidence to show that similar giant step to baby step timing ratios exist in the infrastructure of a cubic function field of unit rank 1. Analogously to the unit rank 0 case, we define $T_{B,1}$, $T_{G,1}$, and $T_{I,1}$ as the times required to compute a baby step, a giant step, and the inverse of a divisor in $\mathcal{R}$, respectively, and $\tau_2 = (T_{G,1} + T_{I,1})/T_{B,1}$.

When performing the Baby Step-Giant Step method in the infrastructure, we do not set a fixed number of baby steps, but rather, we fix the minimal distance, M, that is spanned by $\mathcal{B}$. In this way, we will have $\mathcal{B} = \{D_0, \ldots, D_{\lambda-1}\}$, where $D_0 = D(2E)$, $D_j = bs(D_{j-1})$, for $1 \leq j < \lambda$, and λ is minimal such that the relative distance $\delta_{D_0}(D_{\lambda-1}) \geq M$. We also store the distances $\delta(D_j) = \delta_0(D_j)$, for $0 \leq j < \lambda$. (We assume that the distance measure δ is relative to 0.)

For the giant steps, we set $A := D(\delta(D_{\lambda-1}) - \delta(D_0))$, so that $\delta(A)$ is not greater than the distance spanned by $\mathcal{B}$. As in the situation in Algorithm 6.2.4, we set $A_0 := 0$ and define A_i, for $i \in \mathbb{N}$, recursively by $A_i := A_{i-1} \oplus A$, determining $\delta(A_i)$ at each step. When $A_i = D_j \in \mathcal{B}$ or $Inverse(A_i) = D_j \in \mathcal{B}$, we can determine h_0. The following algorithm adapts Steps 2-5 of Algorithm 3.8 (or Step 3 of Algorithm 4.4) of [SW99] to purely cubic function fields of unit rank 1.

Algorithm 6.2.6 Baby Step-Giant Step Algorithm for Partial S-Regulator Computation - Unit Rank 1

Input: $q \equiv 2 \pmod 3$*; monic, relatively prime, and square-free polynomials* $G, H \in \mathbb{F}_q[x]$ *such that* $3 \mid \deg(GH^2)$*;* $K = \mathbb{F}_q(C)$*,* $C : Y^3 = GH^2$*, so that* $\mathrm{sig}(K) = (1, 1; 1, 2)$*; and integers* $E, U \in \mathbb{N}$ *such that* $|h - E| < U$*.*

Output: *A multiple,* h_0*, of the* S*-regulator,* R^S*, of* K*.*

1. *Set* $g := \deg(GH) - 2$*.*
2. *Find an estimate,* $\hat{\alpha} := \hat{\alpha}(q, g)$*, of the expected value of* $|h - E|/U$ *via Table 6.5.*
3. *Set* $M := \sqrt{2\hat{\alpha}\tau_2 U}$*, where* τ_2 *is determined from Table 6.7.*
4. *Set* $\mathcal{B} := N := \varnothing$*, indexed from* 0*, and* $j := 0$*.*
5. *Compute* $\mathcal{B}[0] := D(2E)$*, and* $N[0] := \delta(D(2E))$ *via Algorithm 5.3.26.*
6. *(Baby Steps) While* $N[j] - N[0] < M$*:*
   a. *Set* $j := j + 1$*.*
   b. *Set* $\mathcal{B}[j] := bs(\mathcal{B}[j-1])$ *and* $\delta := \delta(\mathcal{B}[j]) - \delta(\mathcal{B}[j-1])$ *via Algorithm 5.3.14.*
   c. *Set* $N[j] := N[j-1] + \delta$*.*

d. *If* $\mathcal{B}[j] = \mathcal{B}[0]$, *output* $R^S := (N[j] - N[0])/2$.

e. *If* $\mathcal{B}[j] = 0$, *output* $h_0 := N[j]/2$.

7. *Compute* $A := D(N[j] - N[0])$ *and* $a := \delta(A)$ *via Algorithm 5.3.26.*

8. *Set* $D_0 := D_1 := 0$ *and* $d := 0$.

9. *(Giant Steps) While* D_0, $D_1 \notin \mathcal{B}$:

   a. *Compute* $D_0 := D_0 \oplus A$ *and* $\delta := \delta(D_0) - d - a$ *via Algorithm 5.3.20.*

   b. *Set* $d := d + a + \delta$.

   c. *Set* $\mathfrak{b}_0 := \Psi(D_0)$, *compute* $\overline{\mathfrak{b}_0}$ *via Lemma 4.3.3, and set* $\mathfrak{f}_0 := \left(\overline{\mathfrak{b}_0}\right)^{-1}$.

   d. *Compute* $\mathfrak{f}_1 := Reduce\,(\mathfrak{f}_0)$ *via Algorithm 5.2.4 and set* $D_1 := -\Psi^{-1}(\mathfrak{f}_1)$.

10. *If* $D_0 = \mathcal{B}[j]$, *output* $h_0 := (N[j] - d)/2$.

11. *If* $D_1 = \mathcal{B}[j]$:

    a. *Compute* $B := D_0 \oplus D_1$ *and* $\delta := \delta(B) - \delta(D_1) - d$ *via Algorithm 5.2.4.*

    b. *If* $B = 0$, *output* $h_0 := (N[j] + d + \delta)/2$.

    c. *If* $B \neq 0$ *and* 0 *is found via baby steps from* B, *via Algorithm 5.3.14, then output* $h_0 := (N[j] + d + \delta + \delta_B(0))/2$, *where* $\delta_B(0)$ *is determined from adding the outputs of Algorithm 5.3.14.*

    d. *If* $B \neq 0$ *and* B *is found via baby steps from* 0, *via Algorithm 5.3.14, then output* $h_0 := (N[j] + d + \delta - \delta(B))/2$, *where* $\delta(B)$ *is determined from adding the outputs of Algorithm 5.3.14.*

We give a justification that Algorithm 6.2.6 indeed produces a multiple of $\mathcal{R}^S$.

Proposition 6.2.7 *Algorithm 6.2.6 computes a multiple,* h_0, *of the* S*-regulator,* R^S, *of a purely cubic function field of signature* $(1, 1; 1, 2)$.

Proof: We will consider the six possible outputs. The first two concern the case of finding h_0 in the baby step phase in Step 6 and the last four concern the giant step phase in Steps 10 and 11. First, if some baby step $D_j = 0$, for some $0 \leq j < \lambda$, then we know immediately that $h_0 = N[j]/2 = \delta(D_j)/2$ is a multiple of R^S. Similarly, if j is minimal such that $D_j = D_0$, then the baby step set $\{D_0, \ldots, D_{j-1}\} = \mathcal{R}$, so $R_x = N[j] - N[0] = \delta(D_j) - \delta(D_0)$ and $R^S = R_x/2$.

If neither of these situations hold, then we perform giant steps. If $A_i = D_j \in \mathcal{B}$, then $\delta(A_i) \equiv \delta(D_j) \pmod{R_x}$, so $h_0 = (\delta(D_j) - \delta(A_i))/2 = (N[j] - d)/2$ is a multiple of R^S. If $B_i = Inverse(A_i)$ and $B_i = D_j \in \mathcal{B}$, then $h_0 = (\delta(D_j) - \delta(B_i))/2$ is a multiple of R^S. The issue now is to determine $\delta(B_i)$ in this situation. From Lemma 5.3.23, we have $\delta(B_i) \equiv \delta(B) - \delta(A_i) - \delta \pmod{R_x}$, where $B = B_i \oplus A_i$ and δ are the outputs of Algorithm 5.3.20 with inputs B_i and A_i. If $B = 0$, then clearly, $\delta(B) = 0$ and $\delta(B_i) \equiv -\delta(A_i) - \delta \pmod{R_x}$ so that $h_0 := (N[j] + d + \delta)/2$. If $B \neq 0$, then there are two final cases to consider. First, if 0 is found via baby steps from B, then $\delta(B) = R_x - b$, where $b = \delta_B(0)$ is the relative distance from B to 0 and is small. Thus, $\delta(B_i) \equiv -b - \delta(A_i) - \delta \pmod{R_x}$ so that $h_0 = (\delta(D_j) + \delta(A_i) + \delta + b)/2 = (N[j] + d + \delta + \delta_B(0))/2$. Lastly, if B is found via baby

steps from 0, then $\delta(B) = \delta_0(B) \in \mathbb{N}$ is small as well. Thus, $h_0 = (\delta(D_i) + \delta(A_j) + \delta - \delta(B))/2 = (N[j] + d + \delta - \delta(B))/2$ is a multiple of R^S. □

The following proposition analyzes the running time of Algorithm 6.2.6, generalizing Propositions 2.4 and 2.5 of [ST05] and applying them specifically to the infrastructure of a cubic function field of unit rank 1.

Proposition 6.2.8 *Let K be a cubic function field of signature $(1, 1; 1, 2)$, assume that the norms of the distinguished principal ideals are uniformly distributed in $\{f(x) \in \mathbb{F}_q[x] \mid \deg(f(x)) \leq g\}$, and suppose that h has an IHR distribution in the interval $(E - U, E + U)$. Let $T_{I,1}$, $T_{B,1}$, and $T_{G,1}$ be the respective times required to compute an inverse, a baby step, and a giant step in $\mathcal{R}$. Then the expected running time of Algorithm 6.2.6, over all cubic function fields over $\mathbb{F}_q(x)$ and of genus g, is minimized by computing the baby step set, $\mathcal{B}$, such that the distance spanned by the divisors of $\mathcal{B}$ is $M = \lceil 2\sqrt{\alpha\tau_2 U} \rceil$, where $\alpha = \alpha(q, g) \leq 1/2$ is the mean value of $|h - E|/U$ over all cubic function fields over $\mathbb{F}_q(x)$ and of genus g, and $\tau_2 = (T_{G,1} + T_{I,1})/T_{B,1}$. With this choice of M, the total expected running time of Algorithm 6.2.6 is $\left(2\sqrt{\alpha\tau_2 U} + O(g^2 \log(q))\right) T_{B,1}$. Under our initial assumption, the storage requirement is $\sqrt{\alpha\tau_2 U} + O(1)$ distinguished divisors.*

Proof: If $M = \lceil 2\sqrt{\alpha\tau_2 U} \rceil$ is the lower bound on the distance spanned by $\mathcal{B}$, then by Theorem 5.3.10, the actual distance spanned by $\mathcal{B}$ is at most $M + g + 1 = 2\sqrt{\alpha\tau_2 U} + O(1)$. With the assumption on the norms of the distinguished divisors, then by Corollary 5.3.12, we will compute $\lambda - 1 = M/2 + O(1) = \sqrt{\alpha\tau_2 U} + O(1)$ baby steps in time $\lambda T_{B,1} = \left(\sqrt{\alpha\tau_2 U} + O(1)\right) T_{B,1}$. It also follows that $\lambda = \sqrt{\alpha\tau_2 U} + O(1)$ distinguished divisors is the storage requirement.

Next, the expected distance between $2E$ and $2h$ is $2\alpha U$. The giant step, $A = D(\delta(\mathcal{B}[\lambda - 1]) - \delta(\mathcal{B}[0]))$ is within one baby step of $\delta(\mathcal{B}[\lambda - 1]) - \delta(\mathcal{B}[0])$, so $\delta(A) = M + O(1)$. Therefore, we expect to compute $2\alpha U/(M + O(1)) = \sqrt{\alpha U/\tau_2} + O(1)$ giant steps and inverses, taking time $\left(\sqrt{\alpha U/\tau_2} + O(1)\right)(T_{G,1} + T_{I,1}) = \left(\sqrt{\alpha U/\tau_2} + O(1)\right)(\tau_2 T_{B,1}) = \left(\sqrt{\alpha\tau_2 U} + O(1)\right) T_{B,1}$. The two phases of the algorithm have the same expected running time, which minimizes the total expected running time.

In addition, by Proposition 5.3.28, the computation of A and $\mathcal{B}[0] = D(2E)$ requires $O(g \log(U))$ and $O(g \log(E))$ baby steps and $O(\log(U))$ and $O(\log(E))$ giant steps, respectively, and takes total time $O(g \log(E) + g \log(U))T_{B,1} = O(g^2 \log(q))T_{B,1}$. Finally, additional baby steps may be required if Step 11 is entered. In this case, determining $\delta_B(0)$ or $\delta(B)$ requires at most $5g$ baby steps, by Equation (5.6); this time is absorbed by the component $O(g^2 \log(q))T_{B,1}$.

Thus, adding the expected running time of the baby step and giant step phases, the total expected running time to compute a multiple, h_0, of R^S is $\left(2\sqrt{\alpha\tau_2 U} + O(g^2 \log(q))\right) T_{B,1}$. □

For larger computations in either unit rank 0 or 1 cubic function fields, we will not be able to store the optimal number of baby steps, due to storage constraints. To some extent, we may compute the maximum number of baby steps that memory will allow and use a giant step with a smaller than optimal exponent, or a fixed divisor, A, of a smaller than optimal distance, depending on the unit rank. Further analysis of this situation is in Section 2.4 of [ST05]. In order to reduce the storage requirement to compute h or R^S, we turn instead to another algorithm, Pollard's Kangaroo method.

6.2.3 Pollard's Kangaroo Algorithm for Groups

Pollard's Kangaroo method was originally developed to compute discrete logarithms in $\mathbb{F}_q$, but has been extended to compute discrete logarithms (and group orders in particular) and regulators in quadratic function fields in [ST99, ST02b] and in more general situations [vOW99, Pol00]. The original description of the algorithm was as a serial algorithm [Pol78], but van Oorschot and Wiener described a parallelized version that achieved a linear speed-up in the number of processors [vOW99]. Pollard subsequently gave another parallelized version of his algorithm in [Pol00]. In this section, we describe and analyze the running time of the parallelized version on m processors to compute the divisor class number, h, of a purely cubic function field, K, of unit rank 0. In the next section, we will describe the necessary adaptations to compute a multiple, h_0, of R^S of a purely cubic function field of unit rank 1, operating in the principal infrastructure, $\mathcal{R}$.

As in the description of the Baby Step-Giant Step algorithm, we will assume that $h \in (E-U,\ E+U)$, for some $E, U \in \mathbb{N}$, and that $2(E-U) > E+U$. The Kangaroo algorithm uses two *herds* of *kangaroos*, a herd of *tame* kangaroos, $\{T_1, \ldots, T_t\}$, and a herd of *wild* kangaroos, $\{W_1, \ldots, W_w\}$.[5] A kangaroo is a sequence of distinguished ideals. As with the Baby Step-Giant Step algorithm, which requires a *collision* of a baby step and a giant step to yield the group order, the Kangaroo method requires a collision between a tame and a wild kangaroo to obtain h. The tame kangaroos will begin their jumps at distinct, known points near the middle of the interval $(E-U,\ E+U)$, and the wild kangaroos will begin their jumps at points near h, whose location in the interval is unknown, hence their respective names. The running time of the Kangaroo method on a given input varies in practice, since the kangaroos in each herd will be chosen randomly. Therefore, the Kangaroo method is a heuristic algorithm, in contrast to the deterministic Baby Step-Giant Step method.

The idea of the algorithm is as follows. Let $\mathfrak{g}$ be a distinguished ideal. We define a set of random positive integers, $\{s_1, \ldots, s_k\}$, the *jump set* $J = \{\mathfrak{g}^{s_1}, \ldots, \mathfrak{g}^{s_k}\}$, and a hash function $v : \mathcal{I}(\mathcal{O}) \to \{1, \ldots, k\}$. Typically, k is chosen to be 64, and the hash function is typically chosen by taking the constant term of any basis element, for example $L(\mathfrak{a})(0)$, of the ideal, $\mathfrak{a}$, modulo k. (We will describe restrictions on the s_i in order to obtain an optimal running time later.) We initialize each tame kangaroo, T_i, at a distinguished ideal $\mathfrak{t}_{0,i} = \mathfrak{g}^{E+(i-1)\nu}$, for some small $\nu \in \mathbb{Z}$ and $1 \le i \le t$, and each wild kangaroo, W_j, at a distinguished ideal $\mathfrak{w}_{0,j} = \mathfrak{g}^{(j-1)\nu}$, for $1 \le j \le w$. The kangaroos jump through $Cl(\mathcal{O})$ via:

$$\mathfrak{t}_{l+1,i} = \mathfrak{t}_{l,i} * \mathfrak{g}^{s_{v(\mathfrak{t}_{l,i})}} \text{ and } \mathfrak{w}_{l+1,j} = \mathfrak{w}_{l,j} * \mathfrak{g}^{s_{v(\mathfrak{w}_{l,j})}} \ , \text{ for } l \in \mathbb{N}_0 \ , \ 1 \le i \le t \ , \text{ and } 1 \le j \le w \ .$$

The computation of $\mathfrak{t}_{l+1,i}$ from $\mathfrak{t}_{l,i}$, and $\mathfrak{w}_{l+1,j}$ from $\mathfrak{w}_{l,j}$ is called a *(kangaroo) jump*. The *distance* of the i-th tame kangaroo, T_i (or j-th wild kangaroo, W_j) at step l is the discrete logarithm of the ideal $\mathfrak{t}_{l,i}$ (or $\mathfrak{w}_{l,i}$) with respect to the base ideal $\mathfrak{g}$ and is denoted by $d_l(T_i)$ and $d_l(W_j)$, for tame and wild kangaroos, respectively. (We note that this definition of distance is distinct from the infrastructure notion of distance.) Specifically, we initialize $d_0(T_i) = E + (i-1)\nu$ and $d_0(W_j) = (j-1)\nu$, for each $1 \le i \le t$ and $1 \le j \le w$, so that

$$d_{l+1}(T_i) = d_l(T_i) + s_{v(\mathfrak{t}_{l,i})} \quad \text{and} \quad d_{l+1}(W_j) = d_l(W_j) + s_{v(\mathfrak{w}_{l,j})} \ , \quad \text{for} \quad l \in \mathbb{N}_0 \ .$$

[5] We may use different sized herds, but the performance of the method is optimized if m is even and $t = w = m/2$.

Therefore, $\mathfrak{t}_{l,i} = \mathfrak{g}^{d_l(T_i)}$ and $\mathfrak{w}_{l,j} = \mathfrak{g}^{d_l(W_j)}$ for each $l \in \mathbb{N}_0$, $1 \leq i \leq t$, and $1 \leq j \leq w$. For each jump, we store each kangaroo, its type (tame or wild), and its corresponding distance. If $\mathfrak{t}_{A,m} = \mathfrak{w}_{B,n}$, for some A, $B \in \mathbb{N}_0$, $1 \leq m \leq t$, and $1 \leq n \leq w$, then we have a collision and $\mathfrak{g}^{d_A(T_m)} = \mathfrak{g}^{d_B(W_n)}$. If $d_A(T_m) - d_B(W_n) \in (E - U, E + U)$, then we are guaranteed that $h = d_A(T_m) - d_B(W_n)$.

If there is a collision between any two kangaroos, then they will continue on the same path. Therefore, if there is a collision between two kangaroos of the same herd, then we cannot obtain any information about h, so we must re-initialize one of the two kangaroos. Without loss of generality, suppose that the two tame kangaroos T_1 and T_2 collide at the distance $d_A(T_1) = d_B(T_2)$. For a small $c \in \mathbb{N}$, set $\mathfrak{t}_{A+1,1} = \mathfrak{t}_{A,1} * \mathfrak{g}^c$ and $d_{A+1}(T_1) = d_A(T_1) + c$, then let T_1 continue jumping on its new path as usual. T_2 may continue along the same path as before without being interrupted. In practice, there are few collisions, but a method due to Pollard [Pol00] guarantees that no two kangaroos of the same herd collide. We briefly discuss this method.

We first choose t and w such that $t + w \leq m$, the number of processors, and $\gcd(t, w) = 1$. Let $J = \{\mathfrak{g}^{s_1 tw}, \ldots, \mathfrak{g}^{s_k tw}\}$ be the jump set, and initialize $\mathfrak{t}_{0,i} = \mathfrak{g}^{E+(i-1)w}$ and $\mathfrak{w}_{0,j} = \mathfrak{g}^{(j-1)t}$, for $1 \leq i \leq t$ and $1 \leq j \leq w$. The jumping proceeds exactly as before. However, there is exactly one pair, (i, j), that satisfies $E + (i-1)w \equiv h + (j-1)t \pmod{tw}$, namely $i \equiv 1 + (h - E)w^{-1} \pmod{t}$ and $j \equiv 1 - (h - E)t^{-1} \pmod{w}$. Therefore, only one pair of kangaroos, T_i and W_j, will collide to yield h. One drawback to this variant is that if we desire a collision-free environment within each herd, then additional kangaroos on other processors cannot be added once a computation has begun. Theoretical analysis and implementation results of this method may be found in [Pol00] and [ST05].

A key feature of the Kangaroo algorithm is that there is no need to store every jump. Using the idea of van Oorschot and Wiener [vOW99], we will only store *distinguished points*. In order to distinguish this concept from distinguished divisors and ideals, such points will be called *(kangaroo) traps* instead. To this end, we define another hash function, $z : \mathcal{I}(\mathcal{O}) \to \{0, \ldots, \theta - 1\}$, where θ is typically a sufficiently large power of 2. In practice, z finds the lowest $\lg(\theta)$ bits of some integer quickly derived from a kangaroo, $\mathfrak{k}$, where lg denotes the base 2 logarithm. We install a trap, that is, store a kangaroo, $\mathfrak{k}$, if $z(\mathfrak{k}) = 0$. In this way, we expect to set a trap every θ jumps. If θ is sufficiently large, then the storage requirement will be very small; we will provide a discussion of how best to choose θ below. Note that we only detect collisions between traps, but since colliding kangaroos travel along the same path following their first collision, a collision in a trap will eventually be found.

As in our discussion of the Baby Step-Giant Step algorithm, if it is known that there exist integers a, $b \in \mathbb{N}_0$ such that $0 \leq a < b$ and $h \equiv a \pmod{b}$, then we can make adjustments to the jump set and initializations to only operate within the congruence class $a \pmod{b}$. We change the estimate E to $E - (E \pmod{b}) + a$, so that $E \equiv a \pmod{b}$ for the revised value of E, and choose ν and the jump distances such that $b \mid \nu$ and $b \mid s_i$, for each $1 \leq i \leq k$. The remaining initializations are the same. In this way, we have $d_l(T_i) \equiv a \pmod{b}$ and $d_l(W_j) \equiv 0 \pmod{b}$, for any tame kangaroo T_i, with $1 \leq i \leq t$, wild kangaroo W_j, with $1 \leq j \leq w$, and $l \in \mathbb{N}_0$. Thus, if there is a collision, $\mathfrak{g}^{d_A(T_m)} = \mathfrak{g}^{d_B(W_n)}$, then $d_A(T_m) - d_B(W_n) \equiv a \pmod{b}$.

In Algorithm 6.2.9, we formalize the procedures that we have described above. Its correctness follows from our discussion. Afterwards, we will optimize the expected running time of this algorithm, over all cubic function fields over $\mathbb{F}_q(x)$ and genus g, to justify the choices of particular variables.

Algorithm 6.2.9 Kangaroo Algorithm for Class Number Computation - Unit Rank 0

Input: *q; monic, relatively prime, and square-free $G, H \in \mathbb{F}_q[x]$ such that $3 \nmid \deg(GH^2)$; $a, b \in \mathbb{N}_0$ such that $h \equiv a \pmod{b}$;*[6] *$K = \mathbb{F}_q(C)$, where $C : Y^3 = GH^2$, so that $\mathrm{sig}(K) = (3, 1)$; integers $E, U \in \mathbb{N}$ such that $|h - E| < U$ and $2(E - U) > E + U$; and an even m, the number of processors.*
Output: *The divisor class number, h, of K.*

1. *Set $g := \deg(GH) - 1$.*
2. *Find an estimate, $\hat{\alpha} := \hat{\alpha}(q, g)$, of the expected value of $|h - E|/U$ via Table 6.5.*
3. *If $h \equiv a \pmod{b}$, set $\beta := \left\lceil (m/2)\sqrt{\hat{\alpha} b U} \right\rceil$, $\nu := [2\beta/m] - ([2\beta/m] \pmod{b})$, $\theta := 2^{[\lg(\beta)/2]}$, and $E := E - (E \pmod{b}) + a$.*
4. *For $i = 1, \ldots, 64$, choose random integers $0 < s_i \leq 2\beta$, such that $Mean(\{s_i\}) = \beta$ and $b \mid s_i$.*
5. *Generate a random ideal, $\mathfrak{g}$, via Algorithm 6.3.18.*
6. *Define hash functions $v : \mathcal{I} \to \{1, \ldots, 64\}$ and $z : \mathcal{I} \to \{0, \ldots, \theta - 1\}$.*
7. *For $i = 1, \ldots, m/2$, initialize the tame kangaroos, T_i: $\mathfrak{t}_{0,i} := \mathfrak{g}^{E+(i-1)\nu}$, wild kangaroos, W_i: $\mathfrak{w}_{0,i} := \mathfrak{g}^{(i-1)\nu}$, and their distances: $d_0(T_i) := E + (i-1)\nu$ and $d_0(W_i) := (i-1)\nu$, and set $j := 0$.*
8. *If $z(\mathfrak{t}_{0,i}) = 0$ or $z(\mathfrak{w}_{0,i}) = 0$, for some $i = 1, \ldots, m/2$, then store the respective ideal and its distance.*
9. *While a collision between a tame and a wild kangaroo has not been found:*
   a. *For $i = 1, \ldots, m/2$, compute $\mathfrak{t}_{j+1,i} := \mathfrak{t}_{j,i} * \mathfrak{g}^{s_{v(\mathfrak{t}_{j,i})}}$ and $\mathfrak{w}_{j+1,i} := \mathfrak{w}_{j,i} * \mathfrak{g}^{s_{v(\mathfrak{w}_{j,i})}}$.*
   b. *For $i = 1, \ldots, m/2$, set $d_{j+1}(T_i) := d_{j+1}(T_i) + s_{v(\mathfrak{t}_{j,i})}$ and $d_{j+1}(W_i) := d_{j+1}(W_i) + s_{v(\mathfrak{w}_{j,i})}$.*
   c. *Set $j := j + 1$.*
   d. *If $z(\mathfrak{t}_{j,i}) = 0$ or $z(\mathfrak{w}_{j,i}) = 0$, for some $i = 1, \ldots, m/2$, then store the respective ideal and its distance.*
10. *If $\mathfrak{t}_{A,\,i} = \mathfrak{w}_{B,\,j}$, output $h := d_A(T_i) - d_B(W_j)$.*

Here, we will minimize the expected running time of the Kangaroo method, in terms of the number of kangaroo jumps. For our analysis, we assume that the initial spacing between kangaroos is $\nu \leq \beta/(m/2)$, so that at each step, the expected distance between the leading and trailing kangaroos of either herd is at most β (see Section 5.2 of [Tes03]). In this way, the spacing between the leading and trailing kangaroos of either herd is also very small relative to the group order, so that each herd may be viewed as a single entity within the group. The *distance* between the two herds is defined as the absolute value of the difference between the distances of the leading kangaroos of the two herds. Finally, we will assume that the number of processors, m, is even and that the size of each herd is $u := t = w = m/2$. The following analysis combines the results found in the discussions of Section 5.1 (specifically Equation 6) of [vOW99], Section 2.1 (specifically Equation 2.6) of [ST02b], and Sections 6 and 9.1 of [Tes03].

[6] We use $b = 1$ and $a = 0$ if no non-trivial b is known.

Proposition 6.2.10 (van Oorschot, Wiener, Stein, and Teske) *Let K be a purely cubic function field of signature $(3, 1)$. Suppose that there exist integers a, $b \in \mathbb{N}$ such that $h \equiv a \pmod{b}$ and h has an IHR distribution in the interval $(E-U, E+U)$. Then the expected heuristic running time, over all cubic function fields over $\mathbb{F}_q(x)$ of genus g, to compute h via Algorithm 6.2.9 is minimized by choosing an average jump distance of $\beta = \left\lfloor (m/2)\sqrt{\alpha b U} \right\rfloor$, where m is the (even) number of processors and $\alpha = \alpha(q, g) < 1/2$ is the mean value of $|h-E|/U$ over all cubic function fields, K, over $\mathbb{F}_q(x)$ of genus g. For this choice of β, the total expected heuristic running time of Algorithm 6.2.9 for each kangaroo is $(4/m)\sqrt{\alpha U/b} + \theta + O(1)$ ideal compositions, as $q \to \infty$, where traps are set on average every θ jumps.*

Proof: To minimize the total expected heuristic running time of Algorithm 6.2.9, we must determine the optimal value of the average jump distance, β. There are three main stages in any implementation of Algorithm 6.2.9. The end of the first stage occurs when the distance of the leading kangaroo of one herd is greater than each initial distance of the other herd; in other words, when the trailing herd has caught up to the starting point of the leading herd. The end of the second stage is marked by a kangaroo of one herd jumping onto the path of a kangaroo of the other herd. Finally, the end of the third stage, and hence the computation, occurs when one kangaroo has fallen into a trap set by a kangaroo of a different herd. The expected number of jumps required to find a collision among a trapped tame and wild kangaroo is θ, and we will assume that θ is chosen sufficiently small so that the expected number of jumps required to find a collision among a trapped tame and wild kangaroo, θ, is negligible compared to the expected number of jumps required by the other two stages. Therefore, we will optimize Algorithm 6.2.9 by ignoring this third stage and minimizing the total expected time for the first and second stages of the algorithm.

From the definition of the initializations of each herd, the initial distance between the two herds is $\kappa := |h - E|$, and we denote the expected value of κ by $\overline{\kappa}$. In the first stage, we expect each kangaroo to make $\overline{\kappa}/\beta$ jumps. From this point on, we expect a fixed kangaroo of one herd to land on the path of a fixed kangaroo of the other herd with probability $(\beta/b)^{-1}$, since the distance of each kangaroo lies in a single congruence class modulo b. Since there are $u^2 = m^2/4$ such possible pairings of colliding tame and wild kangaroos, we expect the paths of any two kangaroos of different herds to intersect with probability $bm^2/(4\beta)$. Thus, we expect each kangaroo to make $4\beta/(bm^2)$ jumps in the second stage. Finally, once a collision between a tame and a wild kangaroo occurs, we expect the trailing kangaroo to make θ jumps before it lands in a trap, thus ending the computation. Therefore, we expect each kangaroo to make

$$T_K(\beta) = \frac{\overline{\kappa}}{\beta} + \frac{4\beta}{bm^2} + \theta \tag{6.1}$$

jumps. To simplify the analysis, we assume that θ is constant with respect to β. Solving $\partial T_K/\partial\beta = 0$ for β, we find that $\beta = (m/2)\sqrt{b\overline{\kappa}}$ minimizes the total expected running time. With this choice, we expect each kangaroo to make $\left(4\sqrt{\overline{\kappa}/b}\right)/m + \theta$ jumps, for a total of $4\sqrt{\overline{\kappa}/b} + \theta m$ ideal compositions. If $\alpha = \alpha(q, g)$ is the mean value of $|h - E|/U$ over all cubic function fields, K, over $\mathbb{F}_q(x)$ of genus g, then $\overline{\kappa} = \alpha U$. Thus, we optimize the total expected running time of Algorithm 6.2.9 by choosing $\beta = \left\lfloor (m/2)\sqrt{\alpha b U} \right\rfloor$. With this choice, we expect to compose a total of $4\sqrt{\alpha U/b} + \theta m + O(1)$ ideals, and therefore, for each kangaroo to make a total of $(4/m)\sqrt{\alpha U/b} + \theta + O(1)$ jumps. □

For further practical considerations, we will discuss the rationale behind certain choices for ν, the s_i, for $1 \leq i \leq k$, and θ as given in Algorithm 6.2.9. Stein and Teske note that if the jump distances, $s_1, \ldots, s_k$ are chosen randomly, then the number of useless collisions appears independent of the choice of the initial spacing, ν [Tes03, ST05], and suggest using $\nu \lessapprox 2\beta/m$. They also recommend choosing $s_i \leq 2\beta$, for each $1 \leq i \leq k$, since such choices yielded results which were slightly better than those using other upper bounds. It is also convenient to choose k to be a power of 2, say 32 or 64, so that the hash function is fast, but yet a sufficient level of randomization is obtained, and also so that the space to store the jumps, J, is not too large. Lastly, we expect to store $O\left(T_K(\beta)/\theta\right)$ ideals. Teske (see (4.1) of [Tes03]) suggests taking $\theta = 2^{[\lg(\beta)/2]+C}$, for some small integer C. For this choice, we have $\theta = O\left(\sqrt[4]{U}\right)$, which is not constant with respect to β, as assumed in the analysis of Proposition 6.2.10. However, for the first two terms in the expression for $T_K(\beta)$ in (6.1), we have $\overline{\kappa}/\beta$, $(4\beta)/(bm^2) = O\left(\sqrt{U}\right)$, which dominate the term θ. Therefore, this choice of θ does not significantly change the analysis in the proof of Proposition 6.2.10. Since $mT_K(\beta) = 4\sqrt{\alpha U/b} + \theta m + O(1) = O\left(\sqrt{U}\right)$, we expect to store $O\left(\sqrt{U}/\sqrt[4]{U}\right) = O\left(\sqrt[4]{U}\right)$ ideals, which is a reasonable number in practice. We may choose a custom θ for each computation so that the number of ideals that we store is not too large, but so that the time between setting traps is not too long. Table 3 of [ST02b] compares the results of experiments using varying θ. For large examples, we used such values of θ; specific choices are given in the examples in Section 6.4. Based on these results, Tables 6.16, 6.17, 6.18, and 6.19 give estimates for the expected number of traps for a certain choice of θ in cubic function fields having large characteristic, of genera 3 and 4, and of unit ranks 0 and 1.

In the next section, we will describe appropriate changes to use the Kangaroo method in the infrastructure of a cubic function field of unit rank 1.

6.2.4 Pollard's Kangaroo Algorithm for Infrastructures

If K is a purely cubic function field of unit rank 1, then we wish to compute R^S, via the computation of some multiple, h_0, of R^S. Under the assumption $2(E-U) > E+U$, if we find $h_0 \in (E-U,\, E+U)$, then we will in fact have $h_0 = h$. In this case, we will adapt the description of the Kangaroo algorithm in the previous section to the principal infrastructure of a purely cubic function field. In particular, we will show how to take advantage of the faster baby step operation. We will formalize these modifications in Algorithm 6.2.11 and analyze the running time, following the description and analysis from Section 4.1 of [ST02b] for hyperelliptic function fields, but making appropriate modifications to the cubic function field setting.

To be consistent with earlier notation, a (tame or wild) kangaroo, Z, in this context is a sequence of infrastructure divisors, and we write $Z = \{\mathfrak{k}_0, \mathfrak{k}_1, \ldots\} \subseteq \mathcal{R}$. If $\mathfrak{k}_l \in Z$, then the distance of Z at step l, $d_l(Z) = \delta(\mathfrak{k}_l)$, is the distance of $\mathfrak{k}_l$ as defined for divisors in $\mathcal{R}$. As with the Baby Step-Giant Step algorithm in $\mathcal{R}$, we will compute a multiple, $2h_0$, of R_x. Thus, we initialize the tame kangaroos, T_i, $1 \leq i \leq t$, at distances near $2E$, specifically at the distinguished divisors $\mathfrak{t}_{0,i} = D(2E+(i-1)\nu) \in \mathcal{R}$, for $1 \leq i \leq t$. Likewise, the wild kangaroos, W_j, $1 \leq j \leq w$, are initialized at the distinguished divisors $\mathfrak{w}_{0,j} = D((j-1)\nu) \in \mathcal{R}$, for $1 \leq j \leq w$. In this case, we define the jump set to be $J = \{\mathfrak{g}_1, \ldots, \mathfrak{g}_k\}$, where $\mathfrak{g}_i = D(s_i)$, for $1 \leq i \leq k$. However, since we do not necessarily have $\delta(\mathfrak{g}_i) = s_i$ for all $1 \leq i \leq k$, we store the distances $\{\delta(\mathfrak{g}_1), \ldots, \delta(\mathfrak{g}_k)\}$ rather than the random

integers $\{s_1, \ldots, s_k\}$. Thus, for each step of the algorithm, we have $\mathfrak{k}_{l+1} = \mathfrak{k}_l \oplus \mathfrak{g}_{v(\mathfrak{k}_l)}$, with the distances updated by $d_{l+1}(Z) = d_l(Z) + \delta\left(\mathfrak{g}_{v(\mathfrak{k}_l)}\right) + \delta$, where δ is the relative distance given in the output of Algorithm 5.3.20. In this adaptation, if a tame kangaroo, T_i, collides with a wild kangaroo, W_j, at steps A and B, respectively, then $\mathfrak{t}_{A,i} = \mathfrak{w}_{B,j}$. Thus, $\delta_A(T_i) \equiv \delta_B(W_j) \pmod{R_x}$, so $h_0 = (\delta_i(T_A) - \delta_j(W_B))/2$ is a multiple of $R^S = R_x/2$.

In the infrastructure setting, however, we may take advantage of the fact that baby steps are much faster than giant steps in $\mathcal{R}$ to speed up the regulator computation by a factor of approximately $\sqrt{\tau_3/2}$, where $\tau_3 = T_{G,1}/T_{B,1}$ and $T_{G,1}$ and $T_{B,1}$ are the respective times to compute a giant step and a baby step in $\mathcal{R}$. The following idea is found and analyzed in Section 4.1 of [ST02b]. For a positive integer τ, to be optimized later, let $\mathcal{S}_\tau \subseteq \mathcal{R}$ such that $|\mathcal{R}|/|\mathcal{S}_\tau| \approx \tau$. (One such set that works well in practice is $\mathcal{S}_\tau = \{D \in \mathcal{R} \mid L(\Psi(D))(0) \equiv 0 \pmod{\tau}\}$, where $L(\mathfrak{a}) = s = s(x)$ in the canonical basis representation, $\mathfrak{a} = [s,\, s'(u+\rho),\, s''(v+w\rho+\omega)]$, of an ideal $\mathfrak{a}$.) After each kangaroo jump (a giant step), we take baby steps until a divisor in $\mathcal{S}_\tau$ is found, then we take the next kangaroo jump. Therefore, with each kangaroo jump, we will have τ consecutive adjacent divisors on average. This will improve the probability for a trailing kangaroo to land in the path of a leading kangaroo, compared with the basic kangaroo algorithm, which uses $\tau = 1$. Thus, the version with baby steps can be optimized by choosing a longer average jump length, resulting in fewer total giant steps and a shorter expected running time. After outlining the kangaroo algorithm, we will give specific choices for β and τ to optimize its running time.

The following algorithm formalizes the procedures we described above.

Algorithm 6.2.11 Kangaroo Algorithm for Partial S-Regulator Computation - Unit Rank 1
Input: *$q \equiv 2 \pmod 3$; monic, relatively prime, and square-free $G, H \in \mathbb{F}_q[x]$ such that $3 \mid \deg(GH^2)$; $K = \mathbb{F}_q(C)$, $C : Y^3 = GH^2$, so that $\mathrm{sig}(K) = (1,1;1,2)$; integers $E, U \in \mathbb{N}$ such that $|h - E| < U$ and $2(E-U) > E+U$, and an even m, the number of processors.*
Output: *A multiple, h_0, of the S-regulator, R^S, of K.*

1. *Set $g := \deg(GH) - 2$.*
2. *Find an estimate, $\hat{\alpha} := \hat{\alpha}(q, g)$, of the expected value of $|h - E|/U$ via Table 6.5.*
3. *Determine the appropriate value of τ from Table 6.7.*
4. *Set $\beta := \left\lfloor m\sqrt{(2\tau - 1)\hat{\alpha}U} \right\rfloor - 2(\tau - 1)$, $\nu := \lfloor 2\beta/m \rfloor$, and $\theta := 2^{\lfloor \lg(\beta)/2 \rfloor}$.*
5. *If $g \not\equiv 1 \pmod 3$, then set $\rho := \lfloor g/3 \rfloor$. Otherwise, set $\rho := (g+2)/3$.*
6. *For $i = 1, \ldots, 64$, choose random integers $g + 2 \le s_i \le 2(\beta + \rho) + 1$, such that $Mean(\{s_i\}) = \beta + \rho + 1/2$.*
7. *Compute the jump set $J := \{D(s_1), \ldots, D(s_{64})\}$ via Algorithm 5.3.26 and define hash functions $v : \mathcal{I} \to \{1, \ldots, 64\}$ and $z : \mathcal{I} \to \{0, \ldots, \theta - 1\}$.*
8. *For $i = 1, \ldots, m/2$, initialize the tame kangaroos, T_i: $\mathfrak{t}_{0,i} := D(2E + (i-1)\nu)$, and wild kangaroos, W_i: $\mathfrak{w}_{0,i} := D((i-1)\nu)$ via Algorithm 5.3.26.*
9. *For $i = 1, \ldots, m/2$, compute $\mathfrak{t}_{0,i} := bs(\mathfrak{t}_{0,i})$ until $L(\Psi(\mathfrak{t}_{0,i}))(0) \equiv 0 \pmod{\tau}$ and $\mathfrak{w}_{0,i} := bs(\mathfrak{w}_{0,i})$ until $L(\Psi(\mathfrak{w}_{0,i}))(0) \equiv 0 \pmod{\tau}$, each via Algorithm 5.3.14.*

10. *For $i = 1, \ldots, m/2$, initialize the distances, $\delta_0(T_i) := \delta(\mathfrak{t}_{0,i})$ and $\delta_0(W_i) := \delta(\mathfrak{w}_{0,i})$, and set $j := 0$.*

11. *If $z(\mathfrak{t}_{0,i}) = 0$ or $z(\mathfrak{w}_{0,i}) = 0$, for some $i = 1, \ldots, m/2$, then store the respective divisor and its distance.*

12. *While a collision between a tame and a wild kangaroo has not been found:*

    a. *For $i = 1, \ldots, m/2$, compute $\mathfrak{t}_{j+1,i} := \mathfrak{t}_{j,i} \oplus D\left(s_{v(\mathfrak{t}_{j,i})}\right)$ and $\mathfrak{w}_{j+1,i} := \mathfrak{w}_{j,i} \oplus D\left(s_{v(\mathfrak{w}_{j,i})}\right)$ via Algorithm 5.3.20.*

    b. *For $i = 1, \ldots, m/2$, set $\delta_{j+1}(T_i) := \delta(\mathfrak{t}_{j+1,i}) = \delta(\mathfrak{t}_{j,i}) + \delta_{T,j,i}$ and $\delta_{j+1}(W_i) := \delta(\mathfrak{w}_{j,i}) = \delta(\mathfrak{w}_{j,i}) + \delta_{W,j,i}$, where the $\delta_{T,j,i}$ and $\delta_{W,j,i}$ are the second outputs of Algorithm 5.3.20 with the input divisors given in Step 12.a.*

    c. *Set $j := j + 1$.*

    d. *If $z(\mathfrak{t}_{j,i}) = 0$ or $z(\mathfrak{w}_{j,i}) = 0$, for some $1 \leq i \leq m/2$, then store the respective divisor and its distance.*

    e. *For $i = 1, \ldots, m/2$, while $L(\Psi(\mathfrak{t}_{j,i}))(0) \not\equiv 0 \pmod{\tau}$:*

       - *Compute $\mathfrak{t}_{j+1,i} := bs(\mathfrak{t}_{j,i})$ via Algorithm 5.3.14.*
       - *Set $\delta_{j+1}(T_i) := \delta(\mathfrak{t}_{j+1,i}) = \delta(\mathfrak{t}_{j,i}) + \delta_{T,j,i}$, where $\delta_{T,j,i}$ is the second output of Algorithm 5.3.14 with the input divisor $\mathfrak{t}_{j,i}$.*
       - *Set $j := j + 1$.*
       - *If $z(\mathfrak{t}_{j,i}) = 0$, then store the respective divisor and its distance.*

    f. *For $i = 1, \ldots, m/2$, while $L(\Psi(\mathfrak{w}_{j,i}))(0) \not\equiv 0 \pmod{\tau}$:*

       - *Compute $\mathfrak{w}_{j+1,i} := bs(\mathfrak{w}_{j,i})$ via Algorithm 5.3.14.*
       - *Set $\delta_{j+1}(W_i) := \delta(\mathfrak{w}_{j+1,i}) = \delta(\mathfrak{w}_{j,i}) + \delta_{W,j,i}$, where $\delta_{W,j,i}$ is the second output of Algorithm 5.3.14 with the input divisor $\mathfrak{w}_{j,i}$.*
       - *Set $j := j + 1$.*
       - *If $z(\mathfrak{w}_{j,i}) = 0$, then store the respective divisor and its distance.*

13. *If $\mathfrak{t}_{A,\,i} = \mathfrak{w}_{B,\,j}$, then output $h_0 := (\delta_A(T_i) - \delta_B(W_j))/2$.*

We will give details on how to determine R^S from h_0 in Section 6.3.9.

To optimize the expected running time of Algorithm 6.2.11, we must now choose appropriate values for β as well as τ. One key difference between the following analysis and the proof of Proposition 6.2.10, for example, is that we will not express the optimized running time in terms of the number of kangaroo jumps, but rather in terms of the time to compute a giant step. The following analysis provides a slight improvement over Equation 4.8 of [ST02b] by choosing a better value of τ. In the statement, we will use the heuristic assumption from Section 5.2.2 that the norms of the distinguished principal ideals of $\mathcal{O}$ are uniformly distributed in $\{f(x) \in \mathbb{F}_q[x] \mid \deg(f(x)) \leq g\}$. In this way, the result holds with a probability depending on q, by Corollary 5.3.12, so that a baby step has length 2 with probability $1 - O(1/q)$. For the cases in which the Kangaroo method is preferred over the Baby Step-Giant Step method, q is large enough so that this probability is very close to 1. For such large values of q, the average baby step will be close enough to 2 to make the analysis valid within a reasonable margin of error.

Proposition 6.2.12 *Let $K/\mathbb{F}_q(x)$ be a purely cubic function field of signature $(1,1;1,2)$ and suppose that the norms of the distinguished principal ideals of $\mathcal{O}$ are uniformly distributed in $\{f(x) \in \mathbb{F}_q[x] \mid \deg(f(x)) \leq g\}$. Then the expected heuristic running time, over all cubic function fields, over $\mathbb{F}_q(x)$ of genus g, to compute a multiple, h_0, of R^S via Algorithm 6.2.11 is minimized by choosing an average jump distance of $\beta = \left\lfloor m\sqrt{(2\tau-1)\alpha U} \right\rfloor - 2(\tau-1)$ and either $\tau = \lfloor \tau_3 \rfloor$ or $\tau = \lceil \tau_3 \rceil$, where m is the (even) number of processors, $\tau_3 = T_{G,1}/T_{B,1}$, $T_{G,1}$ and $T_{B,1}$ are the respective times required to compute a giant step and a baby step in $\mathcal{R}$, and $\alpha = \alpha(q,\, g) < 1/2$ is the mean value of $|h-E|/U$ over all cubic function fields, over $\mathbb{F}_q(x)$ of genus g. With these choices, the expected heuristic running time for each kangaroo is $\left((4/m)\sqrt{\alpha U/(2\tau-1)} + \theta/\tau + O(1)\right)(1 + (\tau-1)/\tau_3)\, T_{G,1}$, as $q \to \infty$, where traps are set on average every θ jumps.*

Proof: As in the proof of Proposition 6.2.10, we will consider the three stages of the kangaroo algorithm and will determine the values of β and τ that minimize the total expected running time of Algorithm 6.2.11. In the first stage, the trailing herd catches up to the initial points of the leading herd. In this case, however, after each giant step, we expect to take $\tau - 1$ baby steps. By Corollary 5.3.12, these baby steps will span a distance of 2τ with probability $(1 - O(1/q))^{\tau-1} = 1 - O((\tau-1)/q)$, so they will span a distance of approximately 2τ on average. Now when the kangaroos are initialized, the distance between the herds is $\kappa = |2E - 2h|$, which we expect to be $\overline{\kappa} = 2\alpha U$. Thus, we expect the first stage of the algorithm to finish after $2\alpha U/(\beta + 2(\tau-1))$ iterations of Step 12 of Algorithm 6.2.11, so we expect to make $2\alpha U/(\beta + 2(\tau-1))$ giant steps and $2\alpha U(\tau-1)/(\beta + 2(\tau-1))$ baby steps in the first stage.

In the second stage, a match is found when the jump of a kangaroo in the trailing herd lands either directly in one of the giant step-baby step sequences of a kangaroo in the leading herd or baby steps from that jump land in the sequence. We will consider the probability of either event. We expect each giant step to be followed by $\tau - 1$ baby steps and each baby step to have length 2, so the expected probability that the trailing kangaroo lands directly, via the giant step, in the path of the leading kangaroo is $2\tau/\beta$. Now if the kangaroo lands i baby steps short of the path of the leading kangaroo, then the probability that it and no other divisor in between lies in $\mathcal{S}_\tau$ is $((\tau-1)/\tau)^i$. Since the expected probability that the trailing kangaroo lands exactly i baby steps short of the path of the leading kangaroo is $2/\beta$, the expected probability that the trailing kangaroo takes baby steps into the path of the leading kangaroo is

$$\sum_{i=1}^{\infty} \frac{2}{\beta} \left(\frac{\tau-1}{\tau}\right)^i = \frac{2}{\beta}(\tau-1) \ .$$

Adding the probabilities of the two possibilities, we therefore expect the probability that a fixed kangaroo in one herd lands on the path of a fixed kangaroo of the other herd in one of these giant step-baby step sequences is $2(2\tau-1)/\beta$, assuming that $2(2\tau-1) < \beta$.

Since there are $m^2/4$ pairings of potentially colliding tame and wild kangaroos, we expect any kangaroo in one herd to jump on the path of any kangaroo of the other herd with probability $(2\tau-1)m^2/(2\beta)$ at any iteration of the algorithm, where this iteration is understood to be the giant step followed by the set of baby steps. In this second stage of the algorithm we therefore expect each kangaroo to make $2\beta/((2\tau-1)m^2)$ giant steps and $2\beta(\tau-1)/((2\tau-1)m^2)$ baby steps.

In the third and final stage, we expect each kangaroo to make θ steps: θ/τ giant steps and

$\theta(\tau-1)/\tau$ baby steps.

To combine the baby step and giant step counts in the three phases of the algorithm, we will apply timing ratios. If $\tau_3 = T_{G,1}/T_{B,1}$, then we expect the running time for each kangaroo to be

$$T_K(\beta,\,\tau) = \left(\frac{2\alpha U}{\beta+2(\tau-1)} + \frac{2\beta}{(2\tau-1)m^2} + \frac{\theta}{\tau}\right)\left(1+\frac{\tau-1}{\tau_3}\right)T_{G,1}\ .$$

Again, we assume that θ is constant with respect to β. Solving $\partial T(\beta,\,\tau)/\partial\beta = 0$ for β, we find that $\beta = m\sqrt{(2\tau-1)\alpha U} - 2(\tau-1)$ minimizes the total expected running time. This value is substituted into $T_K(\beta,\,\tau)$ and we solve $\partial T_K(\beta,\,\tau)/\partial\tau = 0$ to find that $\tau = \tau_3$ minimizes $T_K(\beta,\,\tau)$. Since we require $\tau \in \mathbb{N}$, we round the actual value of τ up or down depending on which value provides the better running time. Likewise, we choose $\beta = \left\lfloor m\sqrt{(2\tau-1)\alpha U}\right\rceil - 2(\tau-1)$. □

We will briefly show why this method is faster than the version that does not use baby steps. First, without using baby steps, we would have $\tau = 1$, in which case the expected heuristic running time of each kangaroo would be $\left((4/m)\sqrt{\alpha U} + \theta + O(1)\right)T_{G,1}$, which is exactly the same as in Pollard's Kangaroo algorithm for groups, except expressed in terms of $T_{G,1}$, rather than $T_{G,0}$, the time to compose two ideals. Now if $\tau > 1$, then within the parentheses, the two terms $(4/m)\sqrt{\alpha U}$ and θ are reduced by factors of $\sqrt{2\tau-1}$ and τ, respectively, both of which are greater than 1. On the other hand, the $1+(\tau-1)/\tau_3$ factor essentially increases by a factor of $(2\tau-1)/\tau$. Multiplying this through, the term $(4/m)\sqrt{\alpha U}$ is reduced by a factor of $\tau/\sqrt{2\tau-1}$, and the term θ is reduced by a factor of essentially $\tau^2/(2\tau-1)$. Therefore, we expect a speed-up of about $\tau/\sqrt{2\tau-1} \approx \sqrt{\tau/2}$ by applying an average of τ baby steps after each giant step, compared to applying no baby steps. For $\tau > 1$, then, we indeed obtain a speed-up in the kangaroo method for infrastructures by incorporating baby steps.

As in the unit rank 0 case, we will discuss the reasons for the choice of certain other variables in Algorithm 6.2.11. First, when choosing values for the s_i, $1 \le i \le k$, there are a few considerations, due to the reduction required for giant steps in $\mathcal{R}$, so that the average jump distance is as close to β as possible in practice. First, for each s_i, we have $\delta(D(s_i)) = s_i$ with probability roughly $1/2$ and $\delta(D(s_i)) = s_i - 1$ with probability roughly $1/2$, for sufficiently large q, by Corollary 5.3.12. Next, a kangaroo jump, being a giant step, generally requires a reduction step. Thus, if Z is a kangaroo, then $d_l(Z) \le d_{l-1}(Z) + s_{v(\mathfrak{k}_{l-1})}$, with the inequality strict for most $l \in \mathbb{N}$. With probability $1-O(1/q)$, we have $\deg\left((\mathfrak{k}_{l-1})_S\right) = \deg\left(\left(\mathfrak{g}_{v(\mathfrak{k}_{l-1})}\right)_S\right) = g$. Let $\rho(g) = \lfloor g/3 \rfloor$ if $g \not\equiv 1 \pmod 3$, and $\rho(g) = (g+2)/3$ if $g \equiv 1 \pmod 3$. By Remark 5.3.19, we have $\left(d_{l-1}(Z) + s_{v(\mathfrak{k}_{l-1})}\right) - d_l(Z) = \rho(g)$ with probability $1-O(1/q)$. Therefore, to adjust for this "headwind," as well as the average difference $s_i - \delta(D(s_i))$, we must choose the s_i so that $(s_1+\cdots+s_k)/k = \beta+1/2+\rho(g)$. Likewise, we choose $s_i \le 2(\beta+\rho(g))+1$ so that each jump will have distance bounded above by 2β with probability close to 1. Finally, we cannot have $0 \in J$, otherwise a kangaroo will become permanently stuck at one divisor if it hashes to 0, so we must set a lower bound on the choices of the s_i to avoid this situation. By Theorem 5.3.10, we have $1 \le \delta(bs(0)) \le g+2$, so $s = g+2$ is the smallest integer that guarantees that $D(s) \ne 0$. Therefore, we must choose $g+2 \le s_i$ for all $1 \le i \le k$. With these choices of the s_i, the average jump distance in practice will be as close to β as possible.

We also note that from $\tau = \tau_3$, the value of $T_K(\beta,\,\tau)$ increases more slowly with increasing values of τ than with decreasing values of τ so that if $\tau_3 = t + 0.5$, for some $t \in \mathbb{N}$, then Algorithm 6.2.11

is optimized by choosing $\tau = \lceil \tau_3 \rceil$. However, we will generally expect to optimize Algorithm 6.2.11 by choosing $\tau = [\tau_3]$. Table 6.7 gives recommended values of τ for various unit rank 1 situations in genera $3 \leq g \leq 7$, based on Proposition 6.2.12. We note that in Table 6.7, it was indeed the case that $\tau = [\tau_3]$ was optimal for each case given in the table. Choices of ν and θ can be made exactly as in the unit rank 0 case.

To conclude this section, we compare the theoretical running times of the Baby Step-Giant Step method and the Kangaroo method, in terms of the expected number of compositions in $Cl(\mathcal{O})$ or $\mathcal{R}$ for the unit rank 0 or 1 cases, respectively, in Table 6.1. In the table, we used the expected running times from Propositions 6.2.3, 6.2.8, 6.2.10, and 6.2.12, though if necessary, we converted the running times stated in the propositions to the equivalent number of compositions as follows. We expressed the time to compute a giant step in Algorithm 6.2.4, $T_{I,0} + T_{G,0}$, in terms of the time to compose two ideals, $T_{G,0}$ via the ratio $\tau_1 = (T_{I,0} + T_{G,0})/T_{G,0}$. Next, we expressed the time to compute an algorithmic (versus the infrastructure operation) baby step and giant step in Algorithm 6.2.6, $T_{B,1}$ and $T_{I,1} + T_{G,1}$, respectively, in terms of the time to compose two divisors in $\mathcal{R}$, $T_{G,1}$, via the ratios $\tau_3 = T_{G,1}/T_{B,1}$ and $\tau_2 = (T_{I,1} + T_{G,1})/T_{G,1}$, respectively. Finally, we expressed $T_{B,1}$ in terms of $T_{G,1}$ via τ_3 again for the running time of Algorithm 6.2.11. The variables α, τ_1, τ_2, and τ_3 used in the propositions and conversions are taken from experimental results, so that rather than using the actual averages, α, we substitute approximations, $\hat{\alpha}$. The values of $\hat{\alpha}$ and τ_1 are taken from those given for the largest q in Table 6.5, and the values of τ_2 and τ_3 are taken from the first line for each genus in Table 6.7.

In particular, we note that, theoretically, the Baby Step-Giant Step method is faster than the Kangaroo method. Also, if compositions in $\mathcal{R}$ are as fast as compositions in $Cl(\mathcal{O})$, then the unit rank 1 computations will be faster than the corresponding computations in the unit rank 0 setting. In Section 6.4, we will give practical comparisons and show that the Baby Step-Giant Step method is indeed faster when memory requirements do not hinder the computation. On the other hand, we show that arithmetic in $\mathcal{R}$, at least in the current implementation, is too slow for the unit rank 1 computations to be faster than unit rank 0 computations of divisor class numbers of similar sizes.

Table 6.1: Comparison of the Baby Step-Giant Step and the Kangaroo Methods in Purely Cubic Function Fields (In Terms of the Equivalent Number of Compositions)

g	Algorithm 6.2.4	Algorithm 6.2.9	Algorithm 6.2.6	Algorithm 6.2.11
3	$1.2132\sqrt{U} + O(1)$	$2.0857\sqrt{U} + O(1)$	$0.7402\sqrt{U} + O(1)$	$1.5787\sqrt{U} + O(1)$
4	$1.0726\sqrt{U} + O(1)$	$1.7521\sqrt{U} + O(1)$	$0.5282\sqrt{U} + O(1)$	$1.1734\sqrt{U} + O(1)$
5	$1.0577\sqrt{U} + O(1)$	$1.7523\sqrt{U} + O(1)$	$0.4768\sqrt{U} + O(1)$	$1.0309\sqrt{U} + O(1)$
6	$1.0165\sqrt{U} + O(1)$	$1.5988\sqrt{U} + O(1)$	$0.4123\sqrt{U} + O(1)$	$0.8959\sqrt{U} + O(1)$
7	$0.8900\sqrt{U} + O(1)$	$1.4200\sqrt{U} + O(1)$	$0.3272\sqrt{U} + O(1)$	$0.7183\sqrt{U} + O(1)$

In this section, we described Shanks' Baby Step-Giant Step and Pollard's Kangaroo methods for computing the divisor class number and a multiple of the regulator of a purely cubic function field of unit rank 0 and 1, respectively. We then analyzed the running time of each method, both in theory, and in practice. One remaining task is to determine a suitable interval, $(E - U, E + U)$, in which to search for h using these methods. In the next section, we will show how to determine a good estimate, E, of h and a sharp error bound, U, on $|h - E|$.

6.3 Faster Class Number and Regulator Computation

In this section, we will describe a method, due to Scheidler and Stein [SS07], to compute the divisor class number, h, of a purely cubic function field using $O\left(q^{(2g-1)/5+\varepsilon(g)}\right)$ group or infrastructure operations, as $q \to \infty$, where

$$\varepsilon(g) = \begin{cases} 1/5 & \text{if } g \equiv 0 \pmod 5 \ , \\ 1/20 & \text{if } g \equiv 1 \pmod 5 \ , \\ 3/20 & \text{if } g \equiv 2 \pmod 5 \ , \\ 0 & \text{if } g \equiv 3 \pmod 5 \ , \\ 1/10 & \text{if } g \equiv 4 \pmod 5 \ . \end{cases}$$

After giving a brief overview of the literature related to this work, we will describe the idea of the algorithm, outlining three main phases, with a fourth phase for application to the infrastructure of a purely cubic function field. We will then show how to use the zeta function of a cubic function field, K, to determine a good estimate, E, of h and a sharp upper bound, U, on the error $|h - E|$, so that $h \in (E-U,\, E+U)$. We then analyze and optimize the running time, balancing the time to compute E and U with the time to search the interval, $(E-U,\, E+U)$, using either Shanks' Baby Step-Giant Step algorithm or Pollard's Kangaroo algorithm, as described in the previous section. In the final sections, we offer some practical considerations for each of the four phases of the algorithm, detailing subroutines for each phase. In particular, Sections 6.3.8 and 6.3.9 will describe extensions to compute the S-regulator of a purely cubic function field of unit rank 1 given any multiple thereof. (In Section 7.2, we will briefly discuss adaptations to the unit rank 2 case.) Results on the implementation of this method will be given in Section 6.4.

6.3.1 Background

Before we describe the algorithm for computing h, we will highlight previous and related work in this area. For elliptic function fields, the problem reduces to counting points on an elliptic curve over a finite field. The fastest current methods are the algorithms due to Schoof, Atkin, and Elkies [Sch95] for elliptic curves over finite fields of large characteristic, and Satoh [Sat00] for elliptic curves over finite fields of small characteristic; both methods are polynomial (in the number of bit operations) in $\log(q)$. For computing the divisor class number of a hyperelliptic function field, the fastest algorithm to date is due to Kedlaya and has a deterministic running time of $O\left(g^{4+\varepsilon} n^{3+\varepsilon}\right)$ bit operations and a space requirement of $O(g^3 n^3)$ bits, as $g,\, n \to \infty$, where $q = p^n$ and $p = \mathrm{char}(K)$ [Ked01, Ked03]. However, this algorithm only applies to curves over finite fields of small odd characteristic. Kedlaya's method has been extended to hyperelliptic function fields of characteristic 2 by Denef and Vercauteren [DV06b], to hyperelliptic function fields of medium characteristic by Gaudry and Gürel [GG03] and to superelliptic curves, $C : Y^d = f(x)$, of small characteristic by Gaudry and Gürel [GG01] and Lauder [Lau03]. The algorithm of [DV06b] has the same time and space complexity as [Ked01, Ked03], while the algorithms of [GG03], [GG01] and [Lau03] have deterministic running times of $O\left(pg^{4+\varepsilon} n^{3+\varepsilon}\right)$, $O\left(n^{3+\varepsilon}\right)$,[7] and $O\left(p \deg(f)^{4+\varepsilon} d^{3+\varepsilon} n^{3+\varepsilon}\right)$ bit operations, respectively, and space requirements of $O(pg^3 n^3)$, $O\left(n^{3+\varepsilon}\right)$,[7] and $O(p \deg(f)^3 d^3 n^2)$

[7] The authors assume in their analysis that d and g are constant [GG01].

bits, as g, $n \to \infty$.

These methods do not extend well to function fields of large characteristic in general, but in certain cases, different methods have yielded significant results. A method due to Gaudry and Harley applies to hyperelliptic function fields of genus 2 [GH00], and was later improved by Gaudry and Schost [GS04] to compute class numbers of cryptographic size (164 bits, or 50 decimal digits). Index calculus algorithms have been proposed by Adleman, DeMarrais, and Huang [ADH99], Gaudry [Gau00], and Thériault [Thé03], and are especially effective for function fields of large genus; the latter being the fastest, having a heuristic running time of $O\left(g^5 q^{2-\frac{4}{2g+1}+\varepsilon}\right)$ bit operations, where $\varepsilon \to 0$ as $q \to \infty$, or the equivalent of $O\left(g^3 q^{2-\frac{4}{2g+1}+\varepsilon}\right)$ Jacobian operations. (We note that Thériault's algorithm can also be extended to cubic function fields, but we leave that problem for future work.) A method due to Bauer, Teske, and Weng [BTW05] applies to function fields defined by Picard curves, which are of the form $Y^3 = F$, with $\deg(F) = 4$ and $F \in \mathbb{F}_q[x]$ monic and square-free. Their algorithm determines the divisor class number of such a function field using $O\left(\sqrt{q}\right)$ Jacobian operations, as $q \to \infty$, and has computed a 55-digit divisor class number with a 52-digit prime factor. This is the largest known divisor class number of a cubic function field.

Generic algorithms to compute the divisor class number of fields of large characteristic, such has Pollard's Rho algorithm, have a heuristic running time of $O\left(q^{g/2}\right)$ group operations, as $q \to \infty$. However, since h lies in the Hasse-Weil interval, that is, $h \in \left(\left(\sqrt{q}-1\right)^{2g}, \left(\sqrt{q}+1\right)^{2g}\right)$, applying Shanks' Baby Step-Giant Step or Pollard's Kangaroo algorithm, h is computed deterministically or heuristically using $O\left(q^{(2g-1)/4}\right)$ group operations, respectively, as $q \to \infty$. Stein and Williams [SW99] achieved a further speed-up by applying techniques used by Lenstra [Len82] and Schoof [Sch82] in quadratic number fields to hyperelliptic function fields. In particular, they computed the regulator, R, of a real hyperelliptic function field by first determining the divisor class number, which is a multiple of R. Their method finds an estimate, E, of h, and an upper bound, U, on the maximum possible error, $|h - E|$, then searches the interval $(E - U,\, E + U)$ via the Baby Step-Giant Step algorithm to determine h using $O\left(q^{(2g-1)/5+\varepsilon(g)}\right)$ infrastructure operations, as $q \to \infty$. Their method was analyzed further and improved by Stein and Teske [ST05] to compute the 26-digit divisor class number (and also the regulator) of a real hyperelliptic function field of genus 3. Stein and Teske also applied Pollard's Kangaroo algorithm to this method to compute the 29-digit class number (and regulator) of a real hyperelliptic function field of genus 3 [ST02b]. The algorithm by Scheidler and Stein, described in this section, generalizes Stein and Williams' algorithm to general cubic function fields [SS07]. They extended it further to arbitrary function fields, though it requires efficient arithmetic in $\mathcal{J}_K$, $Cl(\mathcal{O})$, or $\mathcal{R}$, as well as criteria to determine how a place of $\mathbb{F}_q(x)$ splits in K [SS08] in order to be implemented. In the next section, we will apply this method to compute the divisor class number of purely cubic function fields of unit rank 0 and 1, and in the latter case, we will show how to extract the S-regulator from h.

6.3.2 Idea of the Algorithm

The algorithm has three main phases to compute the divisor class number h, with a fourth in the unit rank 1 case if the S-regulator is desired.

Algorithm 6.3.1 (Scheidler–Stein, [SS07] and [SS08])

1. *Compute an estimate, E, of h and an upper bound, U, on the error, $|h - E|$. Thus, h lies in the open interval $(E - U,\, E + U)$.*

2. *Determine extra information about h: congruences or divisibility by small primes or distribution of h in the interval $(E - U,\, E + U)$.*

3. *Use the Baby Step-Giant Step method or the Kangaroo method to find h in $(E - U,\, E + U)$ using $O\left(\sqrt{U}\right)$ ideal compositions.*

4. *If $r = 1$, then factor h and let R^S be the smallest factor, R', of h such that $D(2R') = 0$.*

In practice, we do not include Step 2 in the analysis of the algorithm because the information associated with this step is determined and known in advance. Also, Phase 4 will be faster than Phases 1 and 3 in practice, since factoring is asymptotically faster than the overall running time of the algorithm. (We also expect h_x to be small so that completely factoring h will not be necessary.) Thus, the overall complexity of the algorithm is $O\left(\max\left\{T_E,\, T_h\right\}\right)$, where T_E is time required to compute E in Phase 1 and T_h is the time required to compute h in Phase 3. By Propositions 6.2.5, 6.2.8, 6.2.10, and 6.2.12, we have $T_h = O\left(\sqrt{U}\right) T_{G,r}$, where $r \in \{0,\, 1\}$. To optimize the asymptotic running time, we will determine E and U so that $T_E \approx T_h$. Now Phase 1 requires polynomial arithmetic at each step, whereas Phase 3 requires ideal or infrastructure arithmetic at each step. Ideal and infrastructure arithmetic each require a polynomial number of polynomial operations, which is negligible for the purposes of the analysis. Therefore, we will ignore the relative costs of the Phase 1 and Phase 3 operations. As a result, the asymptotic running time of the two phases will be as close to each other as possible, but the overall running time of Algorithm 6.3.1 will be dominated by Phase 3 in practice.

In Phase 1, we will write h as an infinite product over the places of K via the zeta function of K; E will be determined by computing the product up to a certain degree bound, λ, and U will be determined by setting an upper bound on the size of the tail. In this way, we write $h = E'e^B$, where E', $B \in \mathbb{R}$. We will find a sharp upper bound, $\psi \in \mathbb{R}$, on $|B|$. In the case of cubic function fields of large characteristic, we have $\psi \ll 1$ so that $|e^B - 1| < e^\psi - 1$. Then we have

$$E := [E'] \quad \text{and} \quad U = \left[E'\left(e^\psi - 1\right)\right] \ . \tag{6.2}$$

It then follows that

$$|h - E| \leq |h - E'| + |E' - E| \leq E'|e^B - 1| + \frac{1}{2} < E'\left(e^\psi - 1\right) + \frac{1}{2} \leq U \ .$$

In Phase 2, we will use a few observations that will allow us to make a constant-time speed-up. First, we note that h is not uniformly distributed in the interval $(E - U,\, E + U)$, and tends to be close to E, the center of the interval. In Section 6.4, we will present experimental data on this distribution. Second, we can obtain some information about h from F itself in the case that K is generated by a Picard curve, $C : Y^3 = F$, with $\deg(F) = 4$, and $q \equiv 1 \pmod 3$. In this case, if F has k distinct prime factors in $\mathbb{F}_q[x]$, then $3^{k-1} \mid h$, and if F is irreducible, then $h \equiv 1 \pmod 3$. Lastly, details on the Baby Step-Giant Step and Kangaroo methods were provided in Sections 6.2.1, 6.2.2, 6.2.3, and 6.2.4.

We continue by providing further details on how we determine E and U in Phase 1.

6.3.3 Results and Notation for Phase 1

For full details on the derivation of the estimate E and the bound U, we refer to [SW99], [SS07], and [SS08] for the quadratic, cubic, and arbitrary function field cases, respectively. With the exception of the initial discussion of zeta functions, we will restrict ourselves in this section to purely cubic function fields, $K = \mathbb{F}_q(C)$, where $C : Y^3 = F = GH^2$ and G, $H \in \mathbb{F}_q[x]$ are relatively prime and square-free.

To compute E and U, we will use the *zeta function* of the function field K. If $\mathcal{D}^+$ is the set of all effective divisors of any function field, K, and $s \in \mathbb{C}$ such that $\Re(s) > 1$, where $\Re(s)$ denotes the real part of s, then the zeta function of K is defined

$$\zeta_K(s) = \sum_{D \in \mathcal{D}^+} q^{-\deg(D)s} \ .$$

It is well-known that $\zeta_K(s)$ has an analytic continuation to $\mathbb{C}$, with the exception of simple poles at $s \equiv 0, 1 \pmod{2\pi i/\log q}$. Since an effective divisor is a finite sum of places of K with positive coefficients, we let $u = q^{-s}$, and write $\zeta_K(s)$ as an Euler product:

$$\zeta_K(s) = Z_K(u) = \prod_{\mathfrak{P} \in \mathbb{P}_K} \frac{1}{1 - u^{\deg(\mathfrak{P})}} = \prod_{\nu=1}^{\infty} (1 - u^\nu)^{-B_\nu} = \frac{L_K(u)}{(1-u)(1-qu)} \ , \tag{6.3}$$

where $B_\nu = |\{\mathfrak{P} \in \mathbb{P}_K \mid \deg(\mathfrak{P}) = \nu\}|$ and $L_K(u) \in \mathbb{Z}[u]$ is the *L-polynomial* of K. The simple poles of $Z_K(u)$ corresponding to those of $\zeta_K(s)$ are at $u = 1$ and $u = 1/q$. We will show how to factor $Z_K(u)$ into its infinite and finite components, writing $Z_K(u) = Z_K^\infty(u) Z_K^x(u)$. This will allow us to factor $((1-u)(1-qu))^{-1}$ out of $Z_K(u)$, which in turn eliminates the poles. The resulting L-polynomial, $L_K(u) = \prod_{i=1}^{2g}(1 - \omega_i u)$, satisfies $L_K(u) = q^g u^{2g} L_K\left((qu)^{-1}\right)$, where $\omega_i \in \overline{\mathbb{Q}}$ and, by the Hasse-Weil Theorem (see Theorem V.2.1, page 169 of [Sti93], for example), we have $|\omega_i| = \sqrt{q}$, for each $1 \le i \le 2g$. Furthermore, we have

$$h = L_K(1) = q^g L_K(1/q) \ . \tag{6.4}$$

This is the key fact that motivates this algorithm.

Recall the correspondence between finite places of K and prime ideals of $\mathcal{O}$ (Part 4 of Theorem 2.5.3). The finite component of $Z_K(u)$ is the product over all prime ideals of $\mathcal{O}$. Similarly, the infinite component is a product determined by the infinite places of K. If $f_i = \deg(\infty_i)$, for each infinite place $\infty_i \in \mathbb{P}_K$, then we write

$$Z_K^\infty(u) = \prod_{i=0}^{r} \frac{1}{1 - u^{f_i}} \qquad \text{and} \qquad Z_K^x(u) = \prod_{\mathfrak{p}} \frac{1}{1 - u^{\deg(\mathfrak{p})}} = \prod_{P} \prod_{\mathfrak{p} \mid \langle P \rangle} \frac{1}{1 - u^{\deg(\mathfrak{p})}} \ , \tag{6.5}$$

where $\mathfrak{p}$ runs through all prime ideals of $\mathcal{O}$ and P runs through all monic irreducible polynomials in $\mathbb{F}_q[x]$. It will be shown that $(1-u)^{-1}$ is a factor of $Z_K^\infty(u)$ and $(1-qu)^{-1}$ is a factor of $Z_K^x(u)$.

While the discussion above holds for all algebraic function fields, we will now restrict ourselves to cubic function fields. In particular, it will be shown how to compute $(1-u)Z_K^\infty(u)$ and approximate $(1-qu)Z_K^x(u)$ if K is a cubic function field. We first consider the factor of $Z_K(u)$ determined by

the infinite places of K.

Since $Z_K^\infty(u)$ depends solely on q and the signature of K, this factor will be very easy to compute. Let ι be a primitive cube root of 1 in $\overline{\mathbb{F}}_q$. Applying the splitting behavior of ∞ in K, given in Theorem 3.2.2, to the definition of $Z_K^\infty(u)$, we have the following result, which is Theorem 4.1 of [SS07].

Theorem 6.3.2 (Theorem 4.1 of [SS07]) *If K is a cubic function field, then*

$$Z_K^\infty(u) = \frac{1}{(1-u)(1-x_1u)(1-x_2u)} = \frac{1}{(1-u)(1+s_1u+s_2u^2)} \ ,$$

where

$$(x_1,\, x_2,\, s_1,\, s_2) = \begin{cases} (0,\, 0,\, 0,\, 0) & \text{if } \infty = \infty_0^3 \ , \\ (\iota,\, \iota^2,\, 1,\, 1) & \text{if } \infty = \infty_0 \ , \\ (1,\, -1,\, 0,\, -1) & \text{if } \infty = \infty_0\infty_1 \ , \\ (1,\, 0,\, -1,\, 0) & \text{if } \infty = \infty_0\infty_1^2 \ , \\ (1,\, 1,\, -2,\, 1) & \text{if } \infty = \infty_0\infty_1\infty_2 \ . \end{cases}$$

The criteria to determine how ∞ splits in a purely cubic function field are given in Corollary 3.2.2. In light of the discussion following Corollary 3.2.2 and the fact that the computations in this chapter only considered purely cubic function fields of signatures $(3,1)$ and $(1,1;1,2)$, we only considered curves of the form $Y^3 = F$, with F monic. However, we stress that Phase 1 of this procedure works in any cubic function field; Phase 3 can only be run if there exists arithmetic for the corresponding ideal or divisor class group or infrastructure.

Next, we consider the factor of $Z_K(u)$ determined by the finite places of K. The following theorem is the finite analogue to Theorem 6.3.2 and is Theorem 4.4 of [SS07].

Theorem 6.3.3 (Theorem 4.4 of [SS07]) *If K is a cubic function field and $P \in \mathbb{F}_q[x]$ is a monic irreducible polynomial, then*

$$\begin{aligned} \prod_{\mathfrak{p}\mid\langle P\rangle} \frac{1}{1-u^{\deg(\mathfrak{p})}} &= \frac{1}{\left(1-u^{\deg(P)}\right)\left(1-z_1(P)u^{\deg(P)}\right)\left(1-z_2(P)u^{\deg(P)}\right)} \\ &= \frac{1}{1-u^{\deg(P)}} \frac{1}{1+a_1(P)u^{\deg(P)}+a_2(P)u^{2\deg(P)}} \ , \end{aligned}$$

where

$$(z_1(P),\, z_2(P),\, a_1(P),\, a_2(P)) = \begin{cases} (0,\, 0,\, 0,\, 0) & \text{if } \langle P\rangle = \mathfrak{p}_1^3 \ , \\ (\iota,\, \iota^2,\, 1,\, 1) & \text{if } \langle P\rangle = \mathfrak{p}_1 \ , \\ (1,\, -1,\, 0,\, -1) & \text{if } \langle P\rangle = \mathfrak{p}_1\mathfrak{p}_2 \ , \\ (1,\, 0,\, -1,\, 0) & \text{if } \langle P\rangle = \mathfrak{p}_1\mathfrak{p}_2^2 \ , \\ (1,\, 1,\, -2,\, 1) & \text{if } \langle P\rangle = \mathfrak{p}_1\mathfrak{p}_2\mathfrak{p}_3 \ . \end{cases}$$

We note that in purely cubic function fields, the case $\langle P\rangle = \mathfrak{p}_1\mathfrak{p}_2^2$ can only occur if $\mathrm{char}(K) = 3$. Since we do not know how to perform ideal arithmetic or determine the splitting behavior of prime ideals in such fields, we will disregard that possibility.

In order to find the values of $z_1(P)$, $z_2(P)$, $a_1(P)$, and $a_2(P)$, we apply the criteria for the prime splitting behavior in purely cubic function fields as given in Theorem 4.2.1. (For criteria to determine this splitting behavior in general cubic function fields, see Theorem 8.2 of [LRS$^+$08].) Note that we

will need to determine whether or not F is a cube modulo P. In Section 6.3.6, we will describe an algorithm to determine this quickly.

From (4.11) and (4.12) of [SS07] (or (3.1) of [SS08]), we have

$$h = L_K(1) = q^g L_K(1/q) = \frac{q^{g+2}}{(q-x_1)(q-x_2)} \prod_{\nu=1}^{\infty} \prod_{\deg(P)=\nu} \frac{q^{2\nu}}{(q^\nu - z_1(P))(q^\nu - z_2(P))} \ . \tag{6.6}$$

Now let

$$A(K) = \log\left(\frac{q^{g+2}}{(q-x_1)(q-x_2)}\right) = (g+2)\log(q) - \log(q^2 + s_1 q + s_2)$$

and

$$S_\nu(n) = \sum_{\deg(P)=\nu} (z_1^n(P) + z_2^n(P)) \ .$$

Equation (6.6) gives rise to the following result in [SS07]. (See also (3.8) of [SS08].)

Theorem 6.3.4 (Theorem 4.12 of [SS07]) *Let $K/\mathbb{F}_q(x)$ be a cubic function field of genus g such that $\mathrm{char}(K) \neq 3$. Then*

$$\log(h) = A(K) + \sum_{n=1}^{\infty} \frac{1}{nq^n} \sum_{\nu|n} \nu S_\nu\left(\frac{n}{\nu}\right) \ .$$

We will use this theorem in order to find an estimate, E, of h and an upper bound, U, on the error. The idea of finding an estimate, E, of h is to evaluate $A(K)$ and evaluate the remaining sum in Theorem 6.3.4 up to a certain degree bound, $n = \lambda$. The upper bound on the error, U is then determined from the infinite tail of the sum.

The following lemma follows from Theorem 6.3.3 and shows that we can reduce the number of $S_\nu(n)$ that we need to evaluate.

Lemma 6.3.5 (Lemma 5.3 of [SS07]) *For ν, n, $l \in \mathbb{N}$, we have*

1. *$S_\nu(n + 6l) = S_\nu(n)$,*
2. *if $3 \nmid n$ and n is odd, then $S_\nu(n) = S_\nu(1)$, and*
3. *if $3 \nmid n$ and n is even, then $S_\nu(n) = S_\nu(2)$.*

In the next section, we detail two ways to estimate h from Theorem 6.3.4 and three ways to set the upper bound on the error, as presented in [SS07, SS08]. In Section 6.4, we will compare how well each of these estimates and error bounds work in practice.

6.3.4 Determining E and U

In this section, we follow Sections 5.1 and 5.2 of [SS07] and Sections 4.2, 4.3, and 4.4. of [SS08]. We will give two estimates of h, namely E_1 and E_2, and three bounds on the error, $|h - E|$, namely U_1, U_2, and U_3. As noted, we will define our estimates of h by truncating the sum in Theorem 6.3.4.

For a fixed positive integer λ, we write $\log(h) = \log(E_1'(\lambda)) + B_1(\lambda)$, where

$$\log(E_1'(\lambda)) = A(K) + \sum_{n=1}^{\lambda} \frac{1}{nq^n} \sum_{\nu|n} \nu S_\nu\left(\frac{n}{\nu}\right) \qquad \text{and} \qquad B_1(\lambda) = \sum_{n=\lambda+1}^{\infty} \frac{1}{nq^n} \sum_{\nu|n} \nu S_\nu\left(\frac{n}{\nu}\right) \ .$$

Then $h = E_1'(\lambda)e^{B_1(\lambda)}$. Following the outline we presented in Section 6.3.2, we find a sharp upper bound, $\psi_1(\lambda)$, of $|B_1(\lambda)|$. By Corollary 4.9 of [SS07], we have

$$|B_1(\lambda)| \leq \sum_{n=\lambda+1}^{\infty} \frac{1}{nq^n} \left| \sum_{\nu|n} \nu S_\nu \left(\frac{n}{\nu}\right) \right| \leq 2g \sum_{n=\lambda+1}^{\infty} \frac{1}{nq^{n/2}} + 2 \sum_{n=\lambda+1}^{\infty} \frac{1}{nq^n} =: \psi_1(\lambda) \ . \tag{6.7}$$

We may compute $\psi_1(\lambda)$ exactly:

$$\psi_1(\lambda) = 2g \left(\log\left(\frac{\sqrt{q}}{\sqrt{q}-1}\right) - \sum_{n=1}^{\lambda} \frac{1}{nq^{n/2}} \right) + 2\log\left(\frac{q}{q-1}\right) - 2\sum_{n=1}^{\lambda} \frac{1}{nq^n} \ . \tag{6.8}$$

Applying this discussion to (6.2), we have

$$E_1(\lambda) := [E_1'(\lambda)] \qquad \text{and} \qquad U_1(\lambda) := \left\lceil E_1'(\lambda)\left(e^{\psi_1(\lambda)} - 1\right) + \frac{1}{2} \right\rceil \ ,$$

so that $|h - E_1(\lambda)| \leq U_1(\lambda)$, for any $\lambda \in \mathbb{N}$ (Theorem 5.1 of [SS07]).

By moving some terms from $B_1(\lambda)$ to $E_1(\lambda)$, we obtain a second estimate, $E_2(\lambda)$ and error bound. $E_2(\lambda)$ will use every possible $S_\nu(n/\nu)$, for $1 \leq \nu \leq \lambda$, rather than just those n satisfying $1 \leq n \leq \lambda$, as is the case for the determination of $E_1(\lambda)$. Since $S_\nu(n/\nu)$ is calculated from the splitting behavior of the places of degree ν, $E_2(\lambda)$ will be computed using all possible information of the splitting behavior of the places of degree at most λ. As such, we expect $E_2(\lambda)$ to be a better estimate than $E_1(\lambda)$ in general. For a fixed positive integer λ, we again write $\log(h) = \log(E_2'(\lambda)) + B_2(\lambda)$, where

$$\log(E_2'(\lambda)) = A(K) + \sum_{n=1}^{\lambda} \frac{1}{nq^n} \sum_{\nu|n} \nu S_\nu\left(\frac{n}{\nu}\right) + \sum_{n=\lambda+1}^{\infty} \frac{1}{nq^n} \sum_{\substack{\nu|n \\ \nu\leq\lambda}} \nu S_\nu\left(\frac{n}{\nu}\right) = A(K) + \sum_{n=1}^{\infty} \frac{1}{nq^n} \sum_{\substack{\nu|n \\ \nu\leq\lambda}} \nu S_\nu\left(\frac{n}{\nu}\right)$$

and

$$B_2(\lambda) = \sum_{n=\lambda+1}^{\infty} \frac{1}{nq^n} \sum_{\substack{\nu|n \\ \nu>\lambda}} \nu S_\nu\left(\frac{n}{\nu}\right) = \frac{S_{\lambda+1}(1)}{q^{\lambda+1}} + \sum_{n=\lambda+2}^{\infty} \frac{1}{nq^n} \sum_{\substack{\nu|n \\ \nu>\lambda}} \nu S_\nu\left(\frac{n}{\nu}\right) \ .$$

As in the previous case, we have $h = E_2'(\lambda)e^{B_2(\lambda)}$. To find a sharp upper bound, $\psi_2(\lambda)$, of $|B_2(\lambda)|$, we apply Lemmas 5.4 and 5.7 of [SS07] (or Theorem 3.4 and Corollary 3.7 of [SS08]), so

$$\begin{aligned} |B_2(\lambda)| &\leq \left| \frac{S_{\lambda+1}(1)}{q^{\lambda+1}} \right| + \left| \sum_{n=\lambda+2}^{\infty} \frac{1}{nq^n} \sum_{\substack{\nu|n \\ \nu>\lambda}} \nu S_\nu\left(\frac{n}{\nu}\right) \right| \\ &\leq \frac{2}{(\lambda+1)} q^{-(\lambda+1)} + \frac{2g}{(\lambda+1)} q^{-(\lambda+1)/2} + \frac{2q}{(q-1)(\lambda+1)} q^{-(\lambda+1)} \left(q^{(\lambda+1)/l} - 1\right) \\ &\quad + \frac{2g}{(\lambda+2)} \frac{\sqrt{q}}{\left(\sqrt{q}-1\right)} q^{-(\lambda+2)/2} + \frac{4}{(\lambda+2)} \frac{q}{(q-1)} \frac{q^{(l-1)/l}}{(q^{(l-1)/l}-1)} q^{-(\lambda+2)(l-1)/l} =: \psi_2(\lambda) \ , \end{aligned} \tag{6.9}$$

where l is the smallest prime factor of $\lambda+1$. Therefore, we define

$$E_2(\lambda) := [E_2'(\lambda)] \qquad \text{and} \qquad U_2(\lambda) := \left\lceil E_2'(\lambda)\left(e^{\psi_2(\lambda)}-1\right)+\frac{1}{2}\right\rceil \ ,$$

so that again $|h - E_2(\lambda)| \leq U_2(\lambda)$, for any $\lambda \in \mathbb{N}$ (Theorem 5.8 of [SS07]).

In the third case, we use the same estimate, E_2, but we use extra information to obtain a sharper bound, $\psi_3(\lambda)$, on $B_2(\lambda)$. Specifically, given our knowledge of the splitting behavior of places of degree at most λ, we can easily calculate $\nu S_\nu((\lambda+1)/\nu)$ for all $\nu \mid (\lambda+1)$ such that $\nu \neq \lambda+1$. We then use these quantities to determine a sharper upper bound on $S_{\lambda+1}(1)/q^{\lambda+1}$, which is the dominant term of $B_2(\lambda)$. Let $E_3'(\lambda) = E_2'(\lambda)$ and $B_3(\lambda) = B_2(\lambda)$. We apply bounds from (4.3) of [SS08] to obtain

$$\begin{aligned}
|B_2(\lambda)| &\leq \left|\frac{S_{\lambda+1}(1)}{q^{\lambda+1}}\right| + \left|\sum_{n=\lambda+2}^{\infty}\frac{1}{nq^n}\sum_{\substack{\nu\mid n\\ \nu>\lambda}}\nu S_\nu\left(\frac{n}{\nu}\right)\right| \\
&\leq \frac{2g}{(\lambda+1)}q^{-(\lambda+1)/2} + \frac{q^{-(\lambda+1)}}{(\lambda+1)}\left(2+\left|\sum_{\substack{\nu\mid(\lambda+1)\\ \nu\neq\lambda+1}}\nu S_\nu\left(\frac{\lambda+1}{\nu}\right)\right|\right) \\
&\quad + \frac{2g}{(\lambda+2)}\frac{\sqrt{q}}{\left(\sqrt{q}-1\right)}q^{-(\lambda+2)/2} + \frac{4}{(\lambda+2)}\frac{q}{(q-1)}\frac{q^{(l-1)/l}}{(q^{(l-1)/l}-1)}q^{-(\lambda+2)(l-1)/l} =: \psi_3(\lambda) \ ,
\end{aligned} \tag{6.10}$$

where l is the smallest prime factor of $\lambda+1$, and we define

$$E_3(\lambda) := E_2(\lambda) = [E_3'(\lambda)] \qquad \text{and} \qquad U_3(\lambda) := \left\lceil E_2'(\lambda)\left(e^{\psi_3(\lambda)}-1\right)+\frac{1}{2}\right\rceil \ , \tag{6.11}$$

so that $|h - E_3(\lambda)| \leq U_3(\lambda)$, for any $\lambda \in \mathbb{N}$. (See Theorem 4.3 of [SS08].)

We note that in order to compute $E_2(\lambda)$, we need to compute $S_\nu(n)$ for all $n \in \mathbb{N}$ and for all $1 \leq \nu \leq \lambda$. By Lemma 6.3.5, we may restrict ourselves to evaluating $S_\nu(n)$ for only some $0 \leq n \leq 5$. In particular, we will already have each value of $S_\nu((\lambda+1)/\nu)$ stored, where $\nu \mid (\lambda+1)$, so we can evaluate the sum in (6.11).

The next section will show how to choose λ in order to optimize the asymptotic complexity of Algorithm 6.3.1.

6.3.5 Complexity Analysis and Optimization

As noted in Section 6.3.2, the asymptotic running time of Algorithm 6.3.1 is $O(\max\{T_E, T_h\})$, where T_E is the time required to compute E in Phase 1 and $T_h = O\left(\sqrt{U}\right)T_{G,r}$ is the time required to find h in Phase 3. The following analysis works identically with the choice of either $E_1(\lambda)$ or $E_2(\lambda)$, or with the choice of either $U_1(\lambda)$, $U_2(\lambda)$, or $U_3(\lambda)$, so we will take E and U to be any estimate or upper bound on the error. Likewise, we will take ψ to be any $\psi_i(\lambda)$. In order to determine E, we must evaluate $A(K)$, which is very fast, and $S_\nu(n)$ for each $1 \leq \nu \leq \lambda$ and $0 \leq n \leq 5$. This requires determining the decomposition of each prime polynomial of degree at most λ in $\mathcal{O}_K$. It is well-known that there are $(1/\lambda)\sum_{d\mid\lambda}\mu(\lambda/d)q^d = O\left(q^\lambda/\lambda\right)$ such polynomials (see, for example,

(5.4) of [SS07]), where μ is the Möbius μ function. We will see that the splitting behavior of a polynomial is determined in polynomial time via Theorem 4.2.1 and various procedures in Section 6.3.6. Therefore, we can compute E using $O\left(q^{\lambda}\right)$ polynomial operations. (Throughout this analysis, polynomial time will refer to polynomial time in $\log\left(q^g\right) = g\log(q)$.)

Next, to determine the size of $\sqrt{U}$, we will use the first estimate, E_1 for E and error bound, U_1 for U in the analysis. We first make a few observations. From Theorem 5.2 of [SS07], we have

$$\begin{aligned}\log(E) &= A(K) + \sum_{n=1}^{\lambda} \frac{1}{nq^n} \sum_{\nu|n} \nu S_\nu \left(\frac{n}{\nu}\right) \leq A(K) + 2g \sum_{n=1}^{\lambda} \frac{1}{nq^{n/2}} + 2\sum_{n=1}^{\lambda} \frac{1}{nq^n} \\ &< A(K) + 2g\log\left(\frac{\sqrt{q}}{\sqrt{q}-1}\right) + 2\log\left(\frac{q}{q-1}\right) \ .\end{aligned}$$

Since $A(K) = (g+2)\log(q) - \log\left(q^2 + s_1 q + s_2\right)$, this is the dominant term of $\log(E)$, so we have $E = O(q^g)$. (See Theorem 5.2 of [SS07].) This should be expected because $h = O(q^g)$, according to the Hasse-Weil bounds. The second observation is that $\psi < 1$, with q sufficiently large and g sufficiently small, so that $e^{\psi} - 1 \approx \psi$. Considering (6.7), (6.9), or (6.10), we have $\psi = O\left(q^{-(\lambda+1)/2}\right)$, as $q \to \infty$. Therefore, $U \approx E\psi$ and $\sqrt{U} = O\left(q^{g/2-(\lambda+1)/4}\right)$, as $q \to \infty$.

It follows that the complexity of the algorithm is $\max\left\{O\left(q^{\lambda}\right), O\left(q^{g/2-(\lambda+1)/4}\right)\right\}$, as $q \to \infty$. This running time is optimized by determining λ such that both phases have the same asymptotic complexity, i.e. one solves $\lambda = g/2 - (\lambda+1)/4$ for λ. Since λ must be an integer, we have

$$\lambda = \begin{cases} \lfloor (2g-1)/5 \rfloor & \text{if } g \equiv 2 \pmod 5 \ , \\ [(2g-1)/5] & \text{otherwise} \ . \end{cases} \tag{6.12}$$

If $g < 3$, then $\lambda = 0$, so the estimate in Phase 1 is completely determined by the infinite component and runs in polynomial time and Phase 3 requires $O\left(q^{(2g-1)/4}\right)$ ideal or infrastructure compositions, as $q \to \infty$. Thus, there is no asymptotic improvement using the Hasse-Weil bounds for determining E and U, and the complexity for computing the divisor class number of a function field of genus 1 and 2 is still $O\left(q^{1/4}\right)$ and $O\left(q^{3/4}\right)$ ideal or infrastructure compositions, as $q \to \infty$, respectively. However, if $g \geq 3$, then the discussion of this section yields the following result.

Theorem 6.3.6 (Scheidler–Stein, [SS07]) *If K is a purely cubic function field of genus $g \geq 3$, then the complexity of Steps 1, 2, and 3 of Algorithm 6.3.1 is $O\left(q^{[(2g-1)/5]+\varepsilon(g)}\right)$ ideal or infrastructure compositions, where*

$$\varepsilon(g) = \begin{cases} 0 & \text{if } g \equiv 0, 3 \pmod 5 \ , \\ 1/4 & \text{if } g \equiv 1 \pmod 5 \ , \\ -1/4 & \text{if } g \equiv 2 \pmod 5 \ , \\ 1/2 & \text{if } g \equiv 4 \pmod 5 \ , \end{cases}$$

as $q \to \infty$.

In Table 6.2, we compare the running time of this method with the running time using the bounds given by the Hasse-Weil interval. In Table 6.2, H-W refers to the use of the Hasse-Weil interval and *E-U* refers to the use of the new interval $(E-U, E+U)$, described in this section, to compute h. In the last two columns, we use the fact that $h \approx q^g$. As g increases, the running time using the Hasse-Weil interval approaches $O\left(\sqrt{h}\right)$, while the running time using the method by Scheidler

and Stein described here approaches $O\left(h^{2/5}\right)$. For larger genera, we suppose that index calculus methods would prove to be faster than current methods to compute h, but we save this question for future research.

Table 6.2: Computation of Class Numbers: Complexity Comparison

g	λ	H-W	*E-U*	H-W	*E-U*
1	0	$O\left(q^{1/4}\right)$	$O\left(q^{1/4}\right)$	$O\left(h^{0.25}\right)$	$O\left(h^{0.25}\right)$
2	0	$O\left(q^{3/4}\right)$	$O\left(q^{3/4}\right)$	$O\left(h^{0.375}\right)$	$O\left(h^{0.375}\right)$
3	1	$O\left(q^{5/4}\right)$	$O\left(q^{4/4}\right)$	$O\left(h^{0.417}\right)$	$O\left(h^{0.333}\right)$
4	1	$O\left(q^{7/4}\right)$	$O\left(q^{6/4}\right)$	$O\left(h^{0.438}\right)$	$O\left(h^{0.375}\right)$
5	2	$O\left(q^{9/4}\right)$	$O\left(q^{8/4}\right)$	$O\left(h^{0.45}\right)$	$O\left(h^{0.4}\right)$
6	2	$O\left(q^{11/4}\right)$	$O\left(q^{9/4}\right)$	$O\left(h^{0.458}\right)$	$O\left(h^{0.375}\right)$
7	2	$O\left(q^{13/4}\right)$	$O\left(q^{11/4}\right)$	$O\left(h^{0.464}\right)$	$O\left(h^{0.393}\right)$

In the following four sections, we will discuss practical issues surrounding actual implementations of this method for each step of Algorithm 6.3.1. We note that before now, this algorithm has never been implemented for cubic function fields.

6.3.6 Implementation Details for Phase 1

This section presents a number of algorithms and results to apply to the problem of computing each of the approximations, E_1 and E_2, of h and the upper bounds, U_1, U_2, and U_3, on the error $|h - E_1|$ or $|h - E_2|$ in Step 1 of Algorithm 6.3.1. With the exception of the routines to compute the cubic power residue symbol and the Legendre-Kronecker-Jacobi symbol, all of the algorithms and derivations in this section are new. First, we will show how to compute the values $z_1(P)^n + z_2(P)^n$, for some irreducible $P \in \mathbb{F}_q[x]$ and $n \in \mathbb{N}$, based on splitting behavior of the ideal $\langle P \rangle$ in $\mathcal{O}$. Then we will present efficient ways to sum these values to determine $S_\nu(n)$, for a fixed degree ν. Finally, we will list equations to compute E_1 and E_2; formulas to determine U_1, U_2, and U_3 were given in Section 6.3.4.

The following algorithm of [SS07] computes the cubic power residue symbol in $\mathbb{F}_q[x]$, if $q \equiv 1 \pmod 3$ and is used to determine the splitting behavior of prime ideals in $\mathcal{O}$. The correctness of the algorithm follows from Propositions 3.2 and 3.4 and Theorem 3.5 of [Ros02]. (See also Lemma 6.1 of [SS07].) Its complexity is proved in Proposition 6.3 of [SS07].

Algorithm 6.3.7 (Algorithm 6.2 of [SS07]) Cubic Power Residue Symbol
Input: $P, Q \in \mathbb{F}_q[x]$, *with* $q \equiv 1 \pmod 3$.
Output: $e = \left[\frac{P}{Q}\right]_3$.

1. *If* $\gcd(P, Q) \neq 1$, *then output* $e := 0$.
2. *Set* $e := 1$.
3. *While* $P \notin \mathbb{F}_q^*$*:*
   a. *Set* $P := P \pmod Q$.
   b. *Set* $a :\equiv \deg(P) \pmod 3$ *and* $b :\equiv \deg(Q) \pmod 3$.

c. *Set* $e := e(\mathrm{sgn}(P)/q)_3^b(\mathrm{sgn}(Q)/q)_3^a$.

d. *Swap* P *and* Q.

4. *Set* $b :\equiv \deg(Q) \pmod 3$ *and* $e := e(P/q)_3^b$.

5. *Output* e.

Proposition 6.3.8 (Proposition 6.3 of [SS07]) *Algorithm 6.3.7 computes* $[P/Q]_3$ *in* $O(\deg(Q))$ *loop iterations. The running time is the same as that for computing* $\gcd(P, Q)$.

In the case that $q \equiv 2 \pmod 3$, then we have

$$\left[\frac{P}{Q}\right]_3 = Q^{\left(q^{\deg(P)}-1\right)/3} \pmod P \ . \tag{6.13}$$

Via binary exponentiation or exponentiation using the non-adjacent form of an integer, $[P/Q]_3$ is computed using $O(\deg(P)\log(q)) = O(g\log(q))$ polynomial operations.

The following algorithm is the core of the algorithm to determine either estimate, E_1 or E_2, and computes $z_1(P)^n + z_2(P)^n$, where $P \in \mathbb{F}_q[x]$ is irreducible, $n \in \mathbb{N}$, and $z_1(P)$ and $z_2(P)$ are defined as in Theorem 6.3.3. We combine the prime splitting criteria in Theorem 4.2.1, Theorem 6.3.3, and the observation that if $q^{\deg(P)} \equiv 1 \pmod 3$, then $\langle P\rangle \neq \mathfrak{p}_1\mathfrak{p}_2$, and if $q^{\deg(P)} \equiv 2 \pmod 3$, then either $\langle P\rangle = \mathfrak{p}_1^3$ or $\langle P\rangle = \mathfrak{p}_1\mathfrak{p}_2$. The correctness of the algorithm therefore follows from these results.

Algorithm 6.3.9 Computing $z_1(P)^n + z_2(P)^n$
Input: q; *an irreducible* $P \in \mathbb{F}_q[x]$; $n \in \mathbb{N}$; *and monic, relatively prime, and square-free* G, $H \in \mathbb{F}_q[x]$.
Output: $z_1(P)^n + z_2(P)^n$

1. *If* $q^{\deg(P)} \equiv 1 \pmod 3$, *then*

   A. *If* $P \mid GH$, *then output* 0.

   B. *If* $P \nmid GH$, *then compute* $\chi(P) := [GH^2/P]_3$ *via Algorithm 6.3.7.*

      i. *If* $\chi(P) = 1$, *then output* 2.

      ii. *If* $\chi(P) \neq 1$, *then*

         a. *If* $3 \mid n$, *then output* 2.

         b. *If* $3 \nmid n$, *then output* -1.

2. *If* $q^{\deg(P)} \equiv 2 \pmod 3$, *then*

   A. *If* $P \mid GH$, *then output* 0.

   B. *If* $P \nmid GH$, *then*

      i. *If* n *is even, then output* 2.

      ii. *If* n *is odd, then output* 0.

Next, we compute the value of $S_\nu(n) = \sum_{\deg(P)=\nu}(z_1(P)^n + z_2(P)^n)$ for $1 \leq \nu \leq \lambda$ and all $n \in \mathbb{N}$. From Lemma 6.3.5, we only need to compute $S_\nu(1)$, $S_\nu(2)$, $S_\nu(3)$, and $S_\nu(6)$, but we can in fact improve this. We consider the cases $q^{\deg(P)} \equiv 1 \pmod 3$ and $q^{\deg(P)} \equiv 2 \pmod 3$ separately. If P

is an irreducible monic polynomial and $q^{\deg(P)} \equiv 1 \pmod 3$, then from Step 1 of Algorithm 6.3.9, we see that $z_1(P)^3 + z_2(P)^3 = z_1(P)^6 + z_2(P)^6$ and $z_1(P) + z_2(P) = z_1(P)^2 + z_2(P)^2$. Therefore, if $q^{\deg(P)} \equiv 1 \pmod 3$, then we only need to evaluate $S_\nu(1)$ and $S_\nu(3)$. If $q^{\deg(P)} \equiv 2 \pmod 3$, then from Step 2 of Algorithm 6.3.9, we see that $z_1(P) + z_2(P) = z_1(P)^3 + z_2(P)^3 = 0$ and $z_1(P)^2 + z_2(P)^2 = z_1(P)^6 + z_2(P)^6$. Therefore, if $q^{\deg(P)} \equiv 2 \pmod 3$, then we only need to evaluate $S_\nu(1)$ and $S_\nu(2)$. Notice, however, that $S_\nu(1) = 0$.

From Algorithm 6.3.9, we see that if either $q^{\deg(P)} \equiv 2 \pmod 3$ and $n = 2$, or $q^{\deg(P)} \equiv 1 \pmod 3$ and $n = 3$, then $z_1(P)^n + z_1(P)^n = 0$ if $P|F$ and $z_1(P)^n + z_1(P)^n = 2$ otherwise. In light of this, let I_ν be the number of irreducible polynomials of degree ν and let F_ν be the number of prime divisors of F of degree ν. Using well-known formulas for I_ν (see (5.4) of [SS07], for example), we have

$$S_\nu(n) = 2(I_\nu - F_\nu) = 2\left(\frac{1}{\nu}\sum_{d|\nu}\mu\left(\frac{\nu}{d}\right)q^d - F_\nu\right) = 2\left(\frac{1}{\nu}\left(q^\nu + \sum_{\substack{d|\nu \\ d\neq\nu}}\mu\left(\frac{\nu}{d}\right)q^d\right) - F_\nu\right) \quad , \quad (6.14)$$

where μ is the Möbius function. Since $\nu \leq \lambda$, ν will tend to be small, so for the needs of any implementation, it will be most efficient to precompute $\mu(n)$, for $1 \leq n \leq \lambda$. Therefore, $S_\nu(n)$, with either $q^{\deg(P)} \equiv 2 \pmod 3$ or $q^{\deg(P)} \equiv 1 \pmod 3$ and $n = 3$ may be computed very quickly.

If $q^{\deg(P)} \equiv 1 \pmod 3$, then we must compute $S_\nu(1)$ by determining the splitting behavior of each irreducible polynomial of degree ν. For the cases $\deg(P) = 1$ and $\deg(P) = 2$, we give efficient algorithms to check each irreducible polynomial of the given degree. In each case, we assume that we have an ordering, $\{a_0, \ldots, a_{q-1}\}$, of the elements of $\mathbb{F}_q$.

Algorithm 6.3.10 Computing $S_1(1)$.
Input: *q, and monic G, $H \in \mathbb{F}_q[x]$, relatively prime and square-free.*
Output: $S_1(1)$

1. *Set $P := x$, $c := 0$, and $S := 0$.*
2. *While $c < q$, do:*
   a. *Run Algorithm 6.3.9 on input q, P, $n = 1$, G, and H, with output z.*
   b. *Set $S := S + z$, $c := c + 1$, and $P := x + a_c$.*
3. *Output $S_1(1) = S$.*

Algorithm 6.3.11 Computing $S_2(1)$.
Input: *q, and monic G, $H \in \mathbb{F}_q[x]$, relatively prime and square-free.*
Output: $S_2(1)$

1. *Set $c := 1$, $j := 0$, and $S := 0$.*
2. *While $c < q$, do:*
   a. *If a_c is not a square in $\mathbb{F}_q$, then while $j < q$, do:*
      - *Run Algorithm 6.3.9 on input q, $P = (x - a_j)^2 - a_c$, $n = 2$, G, and H, with output z.*

- *Set* $S := S + z$ *and* $j := j + 1$.

b. Set $j := 0$ *and* $c := c + 1$.

3. Output $S_2(1) = S$.

In Step 2.a of Algorithm 6.3.11, a_c is not a square if and only if $a_c^{(q-1)/2} \neq 1$. However, there are tests faster than binary exponentiation in $\mathbb{F}_q$ to determine whether or not a_c is a square in $\mathbb{F}_q$. If q is prime and $a_c \not\equiv 0 \pmod{q}$, then a_c is a square if and only if the Legendre symbol $(a_c/q) = 1$. If q is composite, then we apply the Legendre-Kronecker-Jacobi symbol instead. If $q = p^d$ and $m(t) \in \mathbb{F}_p[t]$ is irreducible of degree d, then $\mathbb{F}_q \cong \mathbb{F}_p[t]/\langle m(t)\rangle$. If a_c is represented in terms of this representation of $\mathbb{F}_q$, then the Legendre-Kronecker-Jacobi symbol $(a_c/m(t))$ is -1, 0, or 1 if a_c is not a square, 0, or a square modulo $m(t)$, that is, if it is not a square, 0, or a square in $\mathbb{F}_q$, respectively. The following algorithm from [CF06] shows how to compute this symbol.

Algorithm 6.3.12 (Algorithm 11.69 of [CF06]) Legendre-Kronecker-Jacobi Symbol
Input: *An irreducible polynomial,* $m(t)$*, of degree* d *and* $f(t) \in \mathbb{F}_p[t]/\langle m(t)\rangle \cong \mathbb{F}_q$.
Output: *The Legendre-Kronecker-Jacobi Symbol* $(f(t)/m(t))$.

1. Set $k := 1$.

2. Do:

a. If $f(t) = 0$*, then output* 0.

b. Let a *be the leading coefficient of* $f(x)$.

c. Set $f(t) := f(t)/a$.

d. If $deg(m) \equiv 1 \pmod{2}$*, then set* $k := k(a/p)$*, where* (a/p) *is the Legendre symbol.*

e. If $p^{\deg(m)} \equiv 3 \pmod{4}$ *and* $\deg(m)\deg(f) \equiv 1 \pmod{2}$*, then set* $k := -k$.

f. Set $r(t) := f(t)$*,* $f(t) :\equiv m(t) \pmod{r(t)}$*, and* $m(t) := r(t)$.

While $m(t) \neq 0$.

3. Output k.

Moreover, Algorithm 6.3.12 is fast.

Proposition 6.3.13 *Algorithm 6.3.12 computes* $(f(t)/m(t))$ *in* $O(\log(d))$ *loop iterations. The running time is the same as that for computing* $\gcd(f(t), m(t))$.

Proof: Since Algorithm 6.3.12 is a generalization of the algorithm to compute the Legendre symbol, it requires $O(\log(\deg(m))) = O(\log(d))$ iterations of Step 2. The Legendre symbol in Step 2.d requires $O\left(\log^2(p)\right)$ bit operations, so the total complexity is of the same magnitude as the Euclidean Algorithm in $\mathbb{F}_q[x]$. □

If $\nu > 2$, we may use an irreducibility test on each polynomial $Q \in \mathbb{F}_q[x]$ of degree ν. The test in Section 6 of [Sho94], for example, is deterministic and requires $O\left(\nu \log(\nu) \log(\log(\nu)) \log(q)\right)$ operations in $\mathbb{F}_q$. We also note that Algorithms 6.3.10 and 6.3.11 may be parallelized by letting each processor run on different blocks of the interval $0 \leq c < q$.

Finally, we give equations to determine E_1 and E_2. To compute E_1, we then have a straightforward sum:

$$\log(E_1'(\lambda, K)) = A(K) + \sum_{n=1}^{\lambda} \frac{1}{nq^n} \sum_{\nu|n} \nu S_\nu\left(\frac{n}{\nu}\right) \ ,$$

so $E_1 = [\exp(\log(E_1'(\lambda, K)))]$. In order to compute E_2, we need to evaluate an infinite sum. We have

$$\log E_2'(\lambda, K) = \log(E_1'(\lambda, K)) + \sum_{n=\lambda+1}^{\infty} \frac{1}{nq^n} \sum_{\substack{\nu|n \\ \nu\le\lambda}} \nu S_\nu\left(\frac{n}{\nu}\right) = A(K) + \sum_{n=1}^{\infty} \frac{1}{nq^n} \sum_{\substack{\nu|n \\ \nu\le\lambda}} \nu S_\nu\left(\frac{n}{\nu}\right) \ .$$

In order to evaluate this, we reverse the order of summation. Let $\nu m = n$ so that

$$\log E_2'(\lambda, K) = A(K) + \sum_{\nu=1}^{\lambda} \sum_{m=1}^{\infty} \frac{\nu S_\nu(m)}{\nu m q^{\nu m}} \ .$$

Using the identity:

$$\sum_{m=1}^{\infty} \frac{1}{kmq^{km}} = -\frac{1}{k}\log\left(1 - \frac{1}{q^k}\right) = \frac{1}{k}\log\left(\frac{q^k}{q^k - 1}\right) \ ,$$

we have

$$\begin{aligned}\sum_{m=1}^{\infty} \frac{\nu S_\nu(m)}{\nu m q^{\nu m}} &= \sum_{m=1}^{\infty} \frac{1}{\nu m q^{\nu m}} \nu S_\nu(1) + \sum_{m=1}^{\infty} \frac{1}{3\nu m q^{3\nu m}} \nu (S_\nu(3) - S_\nu(1)) \\ &= -S_\nu(1)\log\left(1 - \frac{1}{q^\nu}\right) + \frac{1}{3}(S_\nu(1) - S_\nu(3))\log\left(1 - \frac{1}{q^{3\nu}}\right) \ ,\end{aligned}$$

if $q^\nu \equiv 1 \pmod 3$ and

$$\begin{aligned}\sum_{m=1}^{\infty} \frac{\nu S_\nu(m)}{\nu m q^{\nu m}} &= \sum_{m=1}^{\infty} \frac{1}{\nu m q^{\nu m}} \nu S_\nu(1) + \sum_{m=1}^{\infty} \frac{1}{2\nu m q^{2\nu m}} \nu (S_\nu(2) - S_\nu(1)) \\ &= -S_\nu(1)\log\left(1 - \frac{1}{q^\nu}\right) + \frac{1}{2}(S_\nu(1) - S_\nu(2))\log\left(1 - \frac{1}{q^{2\nu}}\right) \ ,\end{aligned}$$

if $q^\nu \equiv 2 \pmod 3$. Therefore if $q \equiv 1 \pmod 3$, then

$$\log E_2'(\lambda, K) = A(K) + \sum_{\nu=1}^{\lambda} \left(-S_\nu(1)\log\left(1 - \frac{1}{q^\nu}\right) + \frac{1}{3}(S_\nu(1) - S_\nu(3))\log\left(1 - \frac{1}{q^{3\nu}}\right)\right) \ , \quad (6.15)$$

and if $q \equiv 2 \pmod 3$, then

$$\log E_2'(\lambda, K) = A(K) + \sum_{m=1}^{\lfloor \lambda/2 \rfloor} \left(-S_{2m}(1) \log\left(1 - \frac{1}{q^{2m}}\right) + \frac{1}{3}(S_{2m}(1) - S_{2m}(3)) \log\left(1 - \frac{1}{q^{6m}}\right)\right)$$
$$+ \sum_{m=1}^{\lfloor (\lambda+1)/2 \rfloor} \left(-S_{2m-1}(1) \log\left(1 - \frac{1}{q^{2m-1}}\right) + \frac{1}{2}(S_{2m-1}(1) - S_{2m-1}(2)) \log\left(1 - \frac{1}{q^{4m-2}}\right)\right) . \tag{6.16}$$

Equivalently, we have the following approach. Let $n_i(\nu)$ be the number of monic irreducible polynomials of degree ν that split into type i primes, as defined in Theorem 4.2.1. Then from Theorem 6.3.3, we have

$$E_2'(\lambda, K) = e^{A(K)} \prod_{\nu=1}^{\lambda} \left(\frac{q^{2\nu}}{q^{2\nu}-1}\right)^{n_3(\nu)} \left(\frac{q^{2\nu}}{(q^\nu-1)^2}\right)^{n_4(\nu)} \left(\frac{q^{2\nu}}{q^{2\nu}+q^\nu+1}\right)^{n_5(\nu)} . \tag{6.17}$$

(Ramified, i.e. type 1 and 2 prime ideals, yield a factor of 1 in the product, and so are omitted.) Therefore, instead of computing the values of $S_\nu(1)$, we count the number of polynomials that produce the various splitting behaviors and apply (6.17). To compute U_3, however, we will need to know certain values of $S_\nu(n)$, but this is easy after counting each $n_i(\nu)$.

Explicit formulas for computing $\psi_1(\lambda)$, $\psi_2(\lambda)$, and $\psi_3(\lambda)$ are given in (6.8), (6.9), and (6.10), respectively. From these, U_1, U_2, and U_3 are easily obtained.

Occasionally, it is advantageous to compute both estimates and all three intervals. Since h must be contained in all three intervals $(E_i - U_i,\, E_i + U_i)$, using the intersection of all three intervals sometimes yields a new interval smaller than all three. Since U_3 applies sharper bounds to the dominant term of B_2 than U_2, we expect $U_3 < U_2$, and in practice, $(E_2 - U_3,\, E_2 + U_3)$ is always contained in $(E_2 - U_2,\, E_2 + U_2)$. Thus, we compare the intervals $(E_1 - U_1,\, E_1 + U_1)$ and $(E_2 - U_3,\, E_2 + U_3)$. In this case, we choose a new estimate E at the center of the new interval. Such intervals are rare (less than 5% of all examples, based on experimental results), and do not yield a significant improvement in practice. However, we will describe a better means to reduce the size of the interval on average. With this in mind, we will choose the interval $(E-U,\, E+U)$, with $E = E_i$ and $U = U_i$, for some fixed $i = 1,\, 2,\, 3$ ($E_2 = E_3$).

6.3.7 Implementation Details for Phase 2

In Phase 2 of Algorithm 6.3.1, we will determine extra information about h to effectively reduce the size of the interval, $(E - U,\, E + U)$, determined in Phase 1. One observation is that h is not uniformly distributed in this interval, and tends to be close to the approximation E. We noted how to apply the average, $\alpha(q,\, g) = Mean(|h - E|/U)$, where the average is taken over all cubic function fields over $\mathbb{F}_q(x)$ of genus g, to optimize the expected running time of the respective Baby Step-Giant Step and Kangaroo algorithms in Sections 6.2.1, 6.2.2, 6.2.3, and 6.2.4. In particular, for the Baby Step-Giant Step algorithm, it was shown in Propositions 6.2.3 and 6.2.8 that the optimal number of baby steps is $\left\lceil \sqrt{\alpha(q,\, g)\tau U} \right\rceil$, where $\tau = \tau_1$ if $r = 0$ and $\tau = 4\tau_2$ if $r = 1$. The expected number of giant steps is the same. If we do not use information on the distribution of h and instead assume that h is uniformly distributed in $(E-U,\, E+U)$, then we would compute $\sqrt{\alpha(q,\, g)^{-1}/2}$ more baby

steps. It follows that we would compute an average of $\alpha(q,\, g)U/\sqrt{\tau U/2}$ giant steps in this case. In the unit rank 0 case, the overall running time would be $\sqrt{\tau_1 U/2}T_{G,0}+\alpha(q,\, g)\sqrt{2U/\tau_1}(T_{I,0}+T_{G,0}) = \left(1/\sqrt{2}+\sqrt{2}\alpha(q,\, g)\right)\sqrt{\tau_1 U}T_{G,0}$, which is slower than the running time derived in Proposition 6.2.3 by a factor of $(1+2\alpha(q,\, g))/\left(2\sqrt{2\alpha(q,\, g)}\right)$. We obtain the same speed-up in the unit rank 1 case. Thus, we obtain faster average running times by using infromation on the distribution of h. Likewise, the Kangaroo algorithm is optimized with a shorter average jump length in order to concentrate our effort on the middle of the interval $(E-U,\, E+U)$ and we also obtain a speed-up of a factor of $(1+2\alpha(q,\, g))/\left(2\sqrt{2\alpha(q,\, g)}\right)$.

However, these $\alpha(q,\, g)$ are very difficult to compute precisely, so we will apply an approximation, $\hat{\alpha}(q,\, g)$, of $\alpha(q,\, g)$ in practice. Table 6.5 lists approximations, $\hat{\alpha}(q,\, g)$, for selected q and g, based on a large sampling of cubic function fields of characteristic q and genus g. For a fixed genus g, we assume that there is a limiting value $\alpha(g) = \lim_{q\to\infty}\alpha(q,\, g)$, as is the case for hyperelliptic function fields [ST02a], so that in practice, we can interpolate or extrapolate as needed when applying these approximations for a given q in Phase 2. We will discuss the values of $\alpha(q,\, g)$ in more depth in Section 6.4.

A second component of Phase 2 of Algorithm 6.3.1 finds information about h modulo small primes. In [BTW05], Bauer, Teske, and Weng consider purely cubic function fields, K, over $\mathbb{F}_q(x)$, defined by a Picard curve. In this case, they prove some results about h modulo certain integers.

Lemma 6.3.14 (Lemma 2.1 of [BTW05]) *If $C : Y^3 = F$ is a Picard curve over $\mathbb{F}_q$, $q \equiv 2 \pmod 3$, and $K = \mathbb{F}_q(C)$, then $(q+1) \mid h$.*

In Lemma 2.2 of [BTW05], they determine h modulo powers of 3.

Lemma 6.3.15 (Lemma 2.2 of [BTW05]) *Let $C : Y^3 = F$ be a Picard curve over $\mathbb{F}_q$, where $q \equiv 1 \pmod 3$, and $K = \mathbb{F}_q(C)$. If F has k distinct irreducible factors over $\mathbb{F}_q[x]$, then $3^{k-1} \mid h$. If F is irreducible, then $h \equiv 1 \pmod 3$.*

Based on numerical results, we expect Lemma 6.3.15 to generalize to all purely cubic function fields, $\mathbb{F}_q(C)$, with C nonsingular and $q \equiv 1 \pmod 3$. Next, in Lemma 2.4 of [BTW05], the authors characterize ideals of order 2. The application of this lemma will determine a power of 2 dividing h.

Lemma 6.3.16 (Lemma 2.4 of [BTW05]) *Let $C : Y^3 = F$ be a Picard curve over $\mathbb{F}_q$ and $K = \mathbb{F}_q(C)$. If $\mathfrak{a} = [s,\, u+\rho,\, v+w\rho+\omega]$ is a reduced ideal, then $\mathfrak{a}$ has order 2 if and only if $s^2 \mid (u^3+F)$. Moreover, $\mathfrak{a}$ is distinguished.*

Ideals of order 2 can be found using Gröbner bases, which in the case of [BTW05] were found using Magma.

We found several purely cubic function fields of genera $3 \le g \le 7$ that had even divisor class numbers. We sampled some of these examples and found that each ideal, $[s,\, u+\rho,\, v+w\rho+\omega]$, of order 2 indeed satisfied $s^2 \mid (u^3+F)$, if F was square-free. Therefore, we expect Lemma 6.3.16 to extend to all totally ramified purely cubic function fields whose model is nonsingular, but we reserve this generalization for future work.

Lastly, Bauer, Teske, and Weng gave the following result on prime divisors, $l \equiv 2 \pmod 3$, of h.

Theorem 6.3.17 (Theorem 3.1 of [BTW05]) *Let $C : Y^3 = F$ be a Picard curve over $\mathbb{F}_q$, where $q \equiv 1 \pmod 3$, and $K = \mathbb{F}_q(C)$. If $l \equiv 2 \pmod 3$, then the l-torsion subgroup of $\mathcal{J}_K$ is $\mathcal{J}_K[l] \cong (\mathbb{Z}/l\mathbb{Z} \times \mathbb{Z}/l\mathbb{Z})^i$, where $0 \leq i \leq 3$. That is, if $l \mid h$, then $l^2 \mid h$.*

Computational results suggest that this theorem extends to all purely cubic function fields in which $q \equiv 1 \pmod 3$, so in this case, we restricted our searches to $h \equiv 1 \pmod 3$ if F was irreducible, and if F had k distinct prime factors, then we restricted our searches to $h \equiv 0 \pmod{3^{k-1}}$.

In this section, we described two ways to effectively reduce the size of the interval $(E-U,\ E+U)$ to speed up Phase 3. The first technique applies to function fields of any genus, assuming that accurate estimates of $\alpha(g)$ are known, and gives rise to a speed-up by a factor of roughly $\sqrt{(2\alpha(g))^{-1}}$. The second technique, determining congruence classes in which h lies, is conjectural for cubic function fields in general, but was proved for cubic function fields defined by various Picard curves, and reduces the interval by a factor of the modulus of the residue class. In the next section, we will formalize the details on searching this interval to compute the divisor class number of a purely cubic function field.

6.3.8 Implementation Details for Phase 3

In this section, we provide details to the outline given by Algorithm 6.3.1 to compute the divisor class number of a purely cubic function field. In particular, we give two algorithms, first for the unit rank 0 case, and second for the unit rank 1 case. We use the Baby Step-Giant Step method for smaller examples and the Kangaroo method when available memory is not sufficient to store the optimal number of baby steps. As such, these algorithms include suggested thresholds for the preference of one method versus the other, and are based on experimental results given in Section 6.4 for a computer with 1 GB of RAM. In addition, we give a pseudo-random ideal generator for generating baby steps and kangaroo jumps. We also make a few remarks on the storage of baby steps and kangaroo traps.

The following algorithm computes a distinguished pseudo-random ideal and is given by Algorithm 0.13 of [Bau05], but its correctness is not proved in the given source. After outlining this algorithm, we will prove its correctness.

Algorithm 6.3.18 (Algorithm 0.13 of [Bau05]) Pseudo-random Ideal Generator
Input: *q; and monic, relatively prime, and square-free G, $H \in \mathbb{F}_q[x]$.*
Output: *A distinguished pseudo-random ideal, $\mathfrak{a}$, of $\mathcal{O} = \mathcal{O}(\mathbb{F}_q(C))$, where $C : Y^3 = GH^2$.*

1. *If $3 \mid \deg(GH^2)$, set $g := \deg(GH) - 2$. Else, set $g := \deg(GH) - 1$.*
2. *Generate a random polynomial $u \in \mathbb{F}_q[x]$ such that $\deg(u) \leq g - 2$.*
3. *Factor $u^3 + GH^2 = c \prod_i a_i(x)^{e_i}$, where $c \in \mathbb{F}_q^*$, $e_i \in \mathbb{N}$, and the $a_i(x) \in \mathbb{F}_q[x]$ are distinct, monic, and irreducible.*
4. *Choose a subset, A, of the $a_i(x)$ such that $\deg(u) < \sum_{a_i(x) \in A} \deg(a_i(x)) \leq g$ and $a_i(x) \nmid H$ for all $a_i(x) \in A$.*
5. *If $A = \varnothing$, then go to Step 2.*

6. *Set* $s := \prod_{a_i(x) \in A} a_i(x)$.

7. *Find* r_1, $r_2 \in \mathbb{F}_q[x]$ *such that* $r_1 H + r_2 s = 1$.

8. *Set* $v := -u^2 r_1 \pmod{s}$.

9. *Output* $\mathfrak{a} :=$ *Reduce*$([s, u + \rho, v + \omega])$ *via Algorithm 4.5.7.*

Proposition 6.3.19 *The output of Algorithm 6.3.18 is a distinguished ideal of* $\mathcal{O}$.

Proof: The submodule $\mathfrak{a} = [s, u + \rho, v + \omega]$ of $\mathcal{O}$ is given in terms of a canonical basis. We will first show that $\mathfrak{a}$ is a primitive ideal by proving that it satisfies the criteria, (4.1)-(4.4), in Theorem 4.1.3. In Step 6, s is constructed to be squarefree, so $\mathfrak{a}$ satisfies (4.1). In Step 8, $v \equiv -u^2 r_1 \pmod{s}$, so $u^2 \equiv -vH \pmod{s}$, satisfying (4.2). Since $\gcd(s, H) = s' = s'' = 1$, (4.3) holds. Finally, from our construction of s and u, we have $u^3 \equiv -GH^2 \pmod{s}$. Since $u^2 \equiv -vH \pmod{s}$, we have $-uvH \equiv -GH^2 \pmod{s}$. Thus, $uv \equiv GH \pmod{s}$, satisfying (4.4). Therefore, $\mathfrak{a}$ is a primitive ideal by Theorem 4.1.3. If $\mathfrak{a}$ is not distinguished, then the reduction in Step 9 will produce a distinguished output. □

For the Baby Step-Giant Step and Kangaroo methods, we used four separate hash functions to map an ideal, $\mathfrak{a} = [s, s'(u + \rho), s''(v + w\rho + \omega)]$, to an integer. For almost every distinguished ideal, we have $s' = s'' = 1$ and $w = 0$. Therefore, we based our hash functions on the basis elements s, u, and v. To store baby steps, $\mathfrak{a}_i = [s_i, s_i'(u_i + \rho), s_i''(v_i + w_i\rho + \omega)]$, we created a hash table, with the hash function based on the coefficients of u_i and v_i. Since the Baby Step-Giant Step method stores a large number of ideals, it is to our advantage to use a hash function which reduces the number of collisions in the hash table; that is, different ideals mapping to the same location of the hash table. Therefore, we used every coefficient of u_i and the constant term of v_i:

$$hash_{BSGS}(\mathfrak{a}_i) = 2^{\lfloor \lg(S) \deg u_i/2 \rfloor} (v_i(0) + 4u_i(0)) + \sum_{j=1}^{\deg u_i} 2^{\lfloor \lg(S)(\deg(u_i)-j)/2 \rfloor} u_i[j] \pmod{S} \ ,$$

where S is the size of the hashtable and $u_i[j]$ is the coefficient of x^j of $v_i(x)$. In order to reduce collisions in the hash table further, we chose S to be the smallest prime greater than $4|\mathcal{B}|$. When the Kangaroo method was used, we also stored traps, $\mathfrak{a}_i = [s_i, s_i'(u_i + \rho), s_i''(v_i + w_i\rho + \omega)]$, in a hash table, with the hash function based on the coefficients of v_i. In this case, we chose a simpler hash function since only a small number of ideals (traps) were stored during our computations and the hash function to identify traps (described below) was determined via another basis element. Thus, we chose:

$$hash_{Kangaroo}(\mathfrak{a}_i) = 2^{\lg(S)} v_i(0) + v_i[1] \pmod{S} \ ,$$

where S is the size of the hashtable. If $r = 0$, then we chose S to be the smallest prime greater than $20\beta/(\theta m)$ and if $r = 1$, then we chose S to be the smallest prime greater than $20\beta/(\theta m(2\tau - 1))$. To determine a kangaroo trap, we used a basis element other than v_i in order for this hash function to be independent of the previous hash function. Thus, we based the hash function to determine a trap on the low bits of the constant and linear coefficients of s_i:

$$hash_{trap} = \sqrt{\theta} \left(s_i(0) \pmod{\sqrt{\theta}} \right) + s_i[1] \pmod{\sqrt{\theta}} \ .$$

Thus, we chose θ such that $2 \mid \lg(\theta)$. We note that multiplication by powers of 2 and determining congruences modulo powers of 2 was done by shifting bits, which is very fast.

Since we only used prime q, the hash function, v, described in Section 6.2.3, that was used to map one ideal to the next was $u_i(0) \pmod{64}$. Again, in order to maintain independence between this hash function and the other two needed by the Kangaroo method, we were forced to use the basis element u_i. (If $q = p^d$ is composite, then $u_i(0)$ can be regarded as a polynomial in t of degree less than d, so we can generalize this hash function to $u_i(0)(p) \pmod{64}$. In this way, each value $0 \leq u_i(0)(p) < q$ is possible.) Since $s_i' = s_i'' = 1$ and $w_i = 0$ for most ideals, we chose not to base any hash function on the coefficients of s_i', s_i'', or w_i.

The following algorithm computes the divisor class number, h, of a purely cubic function field, K, of signature $(3, 1)$, following Algorithm 6.3.1.

Algorithm 6.3.20 Class Number Computation for Totally Ramified Purely Cubic Function Fields
Input: *q; monic, relatively prime, and square-free G, $H \in \mathbb{F}_q[x]$ such that $3 \nmid \deg(GH^2)$; a, $b \in \mathbb{N}_0$ such that $h \equiv a \pmod{b}$;*[8] *and $K = \mathbb{F}_q(C)$, where $C : Y^3 = GH^2$, so that* $\operatorname{sig}(K) = (3, 1)$.
Output: *The divisor class number, h, of K.*

1. *(Phase 1.) Set $g := \deg(GH) - 1$ and λ via* (6.12).
2. *For $\nu = 1, \ldots, \lambda$:*
   a. *Compute $S_\nu(1)$ via Algorithm 6.3.10, 6.3.11, etc. (Or set $S_\nu(1) := 0$ if $q^\nu \equiv 2 \pmod{3}$.)*
   b. *If $q^\nu \equiv 1 \pmod{3}$, compute $S_\nu(3)$ via* (6.14).
   c. *If $q^\nu \equiv 2 \pmod{3}$, compute $S_\nu(2)$ via* (6.14).
3. *Compute $E := [\exp(\log(E_2'))]$ via* (6.15).
4. *Compute ψ_3 via* (6.10) *and U via* (6.11).
5. *(Phases 2 and 3:) If $q^g \lessapprox 10^{20}$, compute and output h via Algorithm 6.2.4.*
6. *(Phases 2 and 3:) If $q^g \gtrapprox 10^{20}$, compute and output h via Algorithm 6.2.9 using m processors (m even).*

As noted, the threshold, $q^g \approx 10^{20}$, in Steps 5 and 6 is based on the use of a computer with 1 GB of RAM. If more memory is available, then the Baby Step-Giant Step algorithm will perform better than the Kangaroo method on larger groups. Therefore, the threshold, $q^g \approx 10^{20}$, is not absolute.

The following algorithm computes a multiple, h_0, of R^S, for cubic function fields of unit rank 1. In most cases, we will have $h_0 = h$.

Algorithm 6.3.21 S-Regulator Computation for Purely Cubic Function Fields of Unit Rank 1: Phases 1-3
Input: *$q \equiv 2 \pmod{3}$; monic, relatively prime, and square-free G, $H \in \mathbb{F}_q[x]$ such that $3 \mid \deg(GH^2)$; and $K = \mathbb{F}_q(C)$, where $C : Y^3 = GH^2$, so that* $\operatorname{sig}(K) = (1, 1; 1, 2)$.
Output: *A multiple, h_0, of the S-regulator, R^S, of K.*

[8] We use $b = 1$ and $a = 0$ if no non-trivial b is known.

1. *(Phase 1.) Set* $g := \deg(GH) - 2$ *and* λ *via* (6.12).

2. *For* $\nu = 1, \ldots, \lambda$*:*

   a. *Compute* $S_\nu(1)$ *via Algorithm 6.3.10, 6.3.11, etc. (If* $\nu \equiv 1 \pmod 2$*, then set* $S_\nu(1) := 0$*.)*

   b. *If* $q^\nu \equiv 1 \pmod 3$*, compute* $S_\nu(3)$ *via* (6.14).

   c. *If* $q^\nu \equiv 2 \pmod 3$*, compute* $S_\nu(2)$ *via* (6.14).

3. *If* $q \equiv 1 \pmod 3$*, then compute* $E := [\exp(\log(E_2'))]$ *via* (6.15).

4. *If* $q \equiv 2 \pmod 3$*, then compute* $E := [\exp(\log(E_2'))]$ *via* (6.16).

5. *Compute* ψ_3 *via* (6.10) *and* U *via* (6.11).

6. *(Phases 2 and 3:) If* $g = 3$ *and* $q \lessapprox 10^6$ *or if* $g > 3$ *and* $q^g \lessapprox 10^{16}$*, compute and output* h *via Algorithm 6.2.6.*

7. *(Phases 2 and 3:) Otherwise, compute and output* h *via Algorithm 6.2.11 using* m *processors (*m *even).*

As in the unit rank 0 case, the suggested threshold $q^g \approx 10^{18}$ is determined experimentally, based on the use of a machine with 1 GB of RAM. Since baby steps are much faster in the unit rank 1 case, a larger baby step set is needed to optimize the running time of the Baby Step-Giant Step algorithm, hence the lower threshold value in this setting. For function fields of genus larger than 4, this threshold is likely to be lower, given the even greater difference in the relative running times of the baby step and giant step operations (see Table 6.7).

Theorem 6.3.6 implies the following.

Theorem 6.3.22 *The complexity of Algorithms 6.3.20 and 6.3.21 is* $O\left(q^{[(2g-1)/5]+\varepsilon(g)}\right)$ *ideal operations, with* $-1/4 \leq \varepsilon(g) \leq 1/2$ *as given in Theorem 6.3.6. If the Baby Step-Giant Step method is used in Step 5, then Algorithms 6.3.20 and 6.3.21 are deterministic and require a storage of* $O\left(q^{[(2g-1)/5]+\varepsilon(g)}\right)$ *distinguished ideals or divisors. If the Kangaroo method is used in Step 6, then Algorithms 6.3.20 and 6.3.21 are heuristic.*

In the next section, we will give details for the final phase of the computation in the unit rank 1 case, determining the regulator, R^S, given a multiple, h_0.

6.3.9 Implementation Details for Phase 4

In the Baby Step-Giant Step portion of Algorithm 6.3.21, it is possible to find R^S during the baby step phase (Step 6.d of Algorithm 6.2.6). This is unlikely, however, so given h_0, we must determine R^S, which is a factor of h_0. In this section, we formalize this procedure in Algorithm 6.3.23. We will then prove its correctness and make some remarks about its running time, especially relative to the running time of Algorithm 6.3.21.

We follow the procedure described in [SW99], making adaptations to the cubic function field case. The S-regulator, R^S, is the smallest factor of h_0 such that $D(2R^S) = 0$. Algorithm 6.3.23 determines this factor. We note that if we computed h_0 via the Baby Step-Giant Step algorithm,

then from the computation of the baby steps, we know that $R^S > M/2$. If we used the Kangaroo algorithm, then unless we computed a set of baby steps up to some degree bound l, we have no such lower bound on R^S (other than the trivial bound).

Algorithm 6.3.23 (Steps 4–5 of Algorithm 4.4 of [SW99]) Computing the S-Regulator of a Purely Cubic Function Field of Unit Rank 1: Phase 4
Input: *A multiple, h_0 of R^S, a lower bound, l, of R^S, q, and monic $G, H \in \mathbb{F}_q[x]$, relatively prime and square-free.*
Output: *The S-regulator, R^S, of $K = \mathbb{F}_q(C)$, where $C : Y^3 = GH^2$.*

1. *Set:* $h^* := 1$.
2. *Factor* $h_0 = \prod_{i=1}^{k} p_i^{a_i}$.
3. *For* $1 \leq i \leq k$:
   a. *If* $p_i < h_0/l$, *then:*
      i. *Find* $1 \leq e_i \leq a_i$ *minimal such that* $D\left(2h_0/p_i^{e_i}\right) \neq 0$ *via Algorithm 5.3.26.*
      ii. *Set* $h^* := p_i^{e_i - 1} h^*$.
4. *Output* h_0/h^*.

Proposition 6.3.24 *Algorithm 6.3.23 computes the S-regulator of a purely cubic function field, K, of unit rank 1 from a given multiple, h_0.*

Proof: We write $h_0 = h^* R^S$. If $p \in \mathbb{Z}$ is prime, then $p \mid h^*$ if and only if $R^S \mid (h_0/p)$, or equivalently, $R_x \mid (2h_0/p)$, which holds if and only if $D(2h_0/p) = 0$. Therefore, $p^e \| h^*$ if and only if $R^S \mid (h_0/p^e)$, or equivalently, $R_x \mid (2h_0/p^e)$, which holds if and only if $D(2h_0/p^e) = 0$. For each prime $p_i < 2h_0/l$ dividing h_0, for $1 \leq i \leq k$, Step 3.a.i finds the smallest exponent e_i such that $D\left(2h_0/p_i^{e_i}\right) \neq 0$. Therefore, $p_i^{e_i - 1} \| h^*$, so that $h^* = \prod_i p_i^{e_i - 1}$ and $R^S = h_0/h^*$. □

We make some brief comments on the running time of Algorithm 6.3.23, especially in light of the factorization in Step 2. First, we note that current heuristic methods to factor the integer h_0 require a subexponential number of bit operations in $\log(h_0) = g \log(q)$. One can use the Elliptic Curve Method of Lenstra [Len87], the Quadratic Sieve of Pomerance [Pom82] (or any of its variants, [Sil87, Con97]), or the General Number Field Sieve of Lenstra, Lenstra, Manasse, and Pollard [LLMP93] to achieve this running time. Furthermore, by Proposition 5.3.28, Algorithm 5.3.26 requires $O(g \log(h_0)) = O(g \log(q^g)) = O(g^2 \log(q))$ baby steps and $O(\log(h_0)) = O(g \log(q))$ giant steps. Now if $h_0 = \prod_{i=1}^{k} p_i^{a_i}$, then the number of loop iterations in Step 3 of Algorithm 6.3.23 is at most $\sum_{i=1}^{k} a_i$. We also have $a_i = O(\log(q^g))$ for each i, and since each $p_i \geq 2$, we have $h_0 \geq 2^{\sum_i a_i}$. Thus, $\sum_{i=1}^{k} a_i = O(\log(h_0)) = O(g \log(q))$. Therefore, Step 3 only requires a polynomial (in g and in $\log(q)$) number of infrastructure operations. By comparison, Phases 1 and 3 of Algorithm 6.3.21 require an exponential number of bit operations in $g \log(q)$, and use slower ideal or infrastructure arithmetic, in contrast to integer arithmetic used for factoring. Therefore, determining R^S from h_0 will not dominate the overall running time, asymptotically.

In practice, the largest divisor class numbers that we have found have 25 digits, and factoring integers of that size is very quick, requiring only a few seconds, compared to hours or days for Phase

3 of the computation. Trial division for primes less than 2000, followed by a basic implementation of Pollard's Rho Method for factoring [Pol75] (see also Algorithm 8.5.2 of [Coh93]), though exponential, is sufficiently fast for the purposes of running Algorithm 6.3.23 on inputs of the size that we encounter. We also add that Shanks' SQUFOF algorithm [Sha] (see also Algorithm 8.7.2 of [Coh93]) is very fast for integers less than 10^{19} and would be well-suited as a sub-routine for Step 2 of Algorithm 6.3.23.

In this section, we described and outlined the complete implementation of Scheidler and Stein's method to compute the divisor class number of a purely cubic function field of unit rank 0 or 1. Furthermore, with Algorithm 6.3.23, we completed the description and generalization of Stein and William's regulator algorithm to purely cubic function fields of unit rank 1. In the next section, we give results on the implementation of Algorithms 6.3.20, 6.3.21, and 6.3.23.

6.4 Computational Results

In this section, we will present results and data obtained from computing divisor class numbers of cubic function fields of unit ranks 0 and 1 and of genera $3 \leq g \leq 7$. We first give experimental results that will allow us to obtain constant-time speed-ups of Algorithm 6.3.1, including timing ratios of operations in the ideal class group for application to the unit rank 0 case. We then discuss the problem of computing $\alpha_i(q, g) = Mean(|h - E_i|/U_i)$, where the average is considered over all cubic function fields over $\mathbb{F}_q(x)$ of genus g, precisely. Next, we list ratios of the timing of certain infrastructure operations and proceed to discuss obstructions to improving the running time of Algorithm 6.3.1 further. Finally, we list results of divisor class number and S-regulator computations, concluding with projections on the expected time to compute the divisor class number and regulator of genus 3 and 4 purely cubic function fields over larger constant fields. In each experiment of this section, we chose prime q, so the order and the characteristic of the base field, $\mathbb{F}_q$, in each case are in fact the same. Prime q facilitate the execution of Phase 1 since there are very straightforward ways to loop through the set of all irreducible polynomials up to a fixed degree. Further, ideal, infrastructure, and polynomial arithmetic should be faster using prime q since the underlying representation of elements of $\mathbb{F}_q[x]$ in memory is simpler, being an array (vector) of elements of $\mathbb{F}_q$, rather than an array of arrays of elements of $\mathbb{F}_p$, for some prime p. For timing and technical considerations, the larger examples were run on Sun workstations with AMD Opteron 148 2.2 GHz processors and 1 GB of RAM running the Linux distribution Fedora 7. The algorithms of Section 6.3 were implemented in C++ using NTL (Number Theory Library), written by Shoup [Sho08], and compiled using `gcc`.

6.4.1 General Optimization Data

For this section, we applied the Baby Step-Giant Step method to $10,000$ totally imaginary function fields of a fixed characteristic, q, and genus, g, and organized the data from these computations to optimize implementations of Algorithms 6.3.20 and 6.3.21. This data provides means to obtain a constant-time improvement over more straightforward implementations of these algorithms. First, we compared the accuracy of the estimates E_1 and E_2. One might expect E_2 to be the better estimate of h because it incorporates all possible information from the splitting behavior of the

irreducible polynomials of degree at most λ, but in fact, this is not the case. We then considered the minimal and maximal values of $|h - E_i|/U_i$ for each $i = 1, 2, 3$, q, and g to provide further analysis of the estimates and to compare the sharpness of the bounds, U_i. Next, for selected q and for genera $3 \leq g \leq 7$, we listed approximations, $\hat{\alpha}_i(q, g)$, of $\alpha_i(q, g) = Mean(|h - E_i|/U_i)$, for $i = 1, 2, 3$, where the average is taken over all cubic function fields over $\mathbb{F}_q(x)$ of genus g. Finally, we computed data on the ratio of the time to compute an ideal inverse to the time to compute an ideal composition for use in improving the performance of Algorithm 6.2.4: the Baby Step-Giant Step method in cubic function fields of unit rank 0.

In each table of this and later sections, λ is the degree bound used to compute the estimates E_1 and E_2, and n is the number of randomly chosen fields, K, of the given characteristic, q, and given genus, g, that we used in each experiment. In Table 6.3, we compare how well the two estimates, E_1 and E_2, approximate h. Here, $Mean_n(|h - E_i|)$ is the average value of $|h - E_i|$ over the n trials, $\pm gs$ gives the average difference between the number of giant steps computed using estimate E_1 and the number of giant steps computed using E_2, $\pm gs\%$ is the average percentage of the giant step time gained or lost by using E_2 versus E_1, and P_2 is the percentage of the trials in which E_2 was the better estimate.

Table 6.3: Comparison of the Estimates E_1 and E_2

q	g	λ	$Mean_n(\|h - E_1\|)$	$Mean_n(\|h - E_2\|)$	$\pm gs$	$\pm gs\%$	P_2	n
997	3	1	819586	819440	0.23	0.054%	51.17%	10000
10009	3	1	81842217	81825469	3.24	0.061%	51.52%	10000
100003	3	1	8188064183	8187673349	7.52	0.014%	52.04%	10000
997	4	1	780557703	780735916	−9.52	−0.067%	49.53%	10000
10009	4	1	777800665422	777714313833	140.01	0.033%	49.96%	10000
97	5	2	5982025	5934456	35.96	3.617%	53.86%	10000
997	5	2	20537411267	20545659238	−106.92	−0.181%	49.40%	10000
97	6	2	583247433	580756662	136.51	1.280%	52.23%	10000
463	6	2	656972270815	656666536313	530.11	0.139%	50.56%	10000
19	7	2	7695243	7563719	68.24	5.376%	53.65%	10000
97	7	2	55135394276	54969142746	1067.35	0.913%	51.27%	10000

We expected the difference between the two estimates to be more pronounced, but in each case, there was not much difference between the two. In all but the cases $g = 4$ and $q = 997$, and $g = 5$ and $q = 997$, the second estimate, $Mean_n(|h - E_2|)$ was less than $Mean_n(|h - E_1|)$, and in all but the two genus 4 experiments and $g = 5$ and $q = 997$, E_2 was the better estimate in most trials. We believe that the reason for some of the averages favoring E_1, while most favor E_2, is due to statistical variation in any sampling and the fact that there is not a very significant advantage to using E_2 over E_1. For larger examples, using either estimate would not lead to vastly different running times, as the $\pm gs$ and $\pm gs\%$ columns seem to indicate. For the genus 5, 6, and 7 examples, the data suggest that the higher percentages in the $\pm gs\%$ column are due to the small base fields. Overall, E_2 is the better estimate, as expected, though not by a large margin.

In Table 6.4, we give the minimum and maximum values, $\min_i$ and $\max_i$, respectively, of $|h - E_i|/U_i$, for $i = 1, 2, 3$, over all the function fields we considered of a fixed q and g. Thus, minimum

values close to 0 mean that the estimate, E, of h was very close for at least one example, and maximum values close to 1 mean that the upper bound on the error, U, was relatively sharp in another. Thus, this table provides another means to compare E_1 and E_2 and also to answer the question of which U_i provides the sharper error bound.

Table 6.4: Comparison of the Minimum and Maximum Values of $|h - E_i|/U_i$

q	g	$\min_1$	$\min_2$	$\min_3$	$\max_1$	$\max_2$	$\max_3$	n
997	3	0.0001247	0.0000199	0.0000236	0.9295732	0.6944619	0.9182125	10000
10009	3	0.0000701	0.0000232	0.0000309	0.9532750	0.7134933	0.9487254	10000
100003	3	0.0000116	0.0000130	0.0000173	0.9757373	0.7306390	0.9733340	10000
997	4	0.0000475	0.0000543	0.0000675	0.8352004	0.6639612	0.8250643	10000
10009	4	0.0000013	0.0000063	0.0000079	0.8336815	0.6661346	0.8310409	10000
97	5	0.0000689	0	0	0.8131066	0.7544986	0.7913693	10000
997	5	0.0000119	0.0000000	0.0000000	0.8332478	0.8262426	0.8391931	10000
97	6	0.0000188	0.0000718	0.0000747	0.8089387	0.7893284	0.8214932	10000
463	6	0.0000319	0.0000103	0.0000105	0.7555262	0.7336155	0.7475987	10000
19	7	0.0000327	0.0000141	0.0000152	0.5796093	0.5468696	0.5886280	10000
97	7	0.0000049	0.0000014	0.0000014	0.6730377	0.6416964	0.6641224	10000

For every genus and constant field that we tested, there were several examples for which the estimates E_1 and E_2 yielded extremely accurate estimates. In fact, there were two genus 5 function fields of characteristic 97 for which $E_2 = h$ and a few examples for which $|h - E_2| < 10$. In addition, there was a genus 5 function field of characteristic 997 in which the second estimate was off by 3. In contrast, the upper bounds, U_i, were less sharp with increasing genus, but for a fixed genus, U_i was generally increasingly sharp as q increased. We will explain this behavior by considering the averages $\hat{\alpha}_i(q, g)$. However, we do note that U_3 was a sharper upper bound than U_2 consistently, so that for large computations, we suggest using E_2 as the estimate for h and U_3 as the upper bound on $|h - E_2|$.

In Table 6.5, we list average values, $\hat{\alpha}_i(q, g) = Mean_n(|h - E_i|/U_i)$, for $i = 1, 2, 3$ ($E_2 = E_3$) and fixed q and g, computed from the random sampling of n function fields. We also give data on the ratio between the average time to compute an ideal inverse, $T_{I,0}$, and two compose two ideals, $T_{G,0}$, for the unit rank 0 case. Since the time to compute a baby step and a giant step in Algorithm 6.2.4 is $T_{G,0}$ and $T_{G,0} + T_{I,0}$, respectively, it is more convenient to give the ratio $\tau_1 := 1 + T_{I,0}/T_{G,0}$, as given in the algorithm and in Proposition 6.2.3.

For each fixed genus, the values of each $\hat{\alpha}_i(q, g)$ increase as q increases. From the definitions of $\psi_i(\lambda)$, for $i = 1, 2, 3$, given in (6.8), (6.9), and (6.10), respectively, we see that each $\psi_i(\lambda)$ decreases as q increases, so we expect $|h - E_i|/U_i$ to increase, on average, as q increases. As with the analogous situation in hyperelliptic function fields (see Section 6 of [ST02a]), we assumed that the limit of the actual averages, $\lim_{q\to\infty} \alpha_i(q, g) = \alpha_i(g)$, exists for each g. Again, we can only at best estimate what the actual limits are, based on experimental results. Given the behavior of the $\hat{\alpha}_i(q, g)$, the data also suggest that $\alpha_i(g)$ decreases as g increases, as is the case for hyperelliptic function fields [ST02a]. These observations are consistent with the fact that the minimum and maximum values of the U_i also decreased with increasing genus. In the next section, we explain why this is to be

Table 6.5: Comparison of the $\hat{\alpha}_i(q,\, g)$ and the Ideal Inverse to Ideal Composition Ratios for $r = 0$

q	g	λ	$\hat{\alpha}_1(q,\, g)$	$\hat{\alpha}_2(q,\, g)$	$\hat{\alpha}_3(q,\, g)$	τ_1	n
997	3	1	0.26832306	0.20003340	0.26448274	1.28405	10000
10009	3	1	0.27031818	0.20234914	0.26906175	1.38074	10000
100003	3	1	0.27227076	0.20408453	0.27187490	1.35344	10000
997	4	1	0.19223965	0.15306081	0.19019941	1.48388	10000
10009	4	1	0.19252978	0.15379110	0.19186318	1.49905	10000
97	5	2	0.18195632	0.17143328	0.17981087	1.39269	10000
997	5	2	0.19188423	0.18894457	0.19190607	1.45747	10000
97	6	2	0.15246065	0.14526827	0.15118788	1.62678	10000
463	6	2	0.15992960	0.15676849	0.15975657	1.61695	10000
19	7	2	0.11428348	0.10135344	0.10909269	1.54181	10000
97	7	2	0.12684120	0.12176623	0.12602172	1.57138	10000

expected and discuss the difficulties surrounding the computation of each $\alpha_i(q,\, g)$ precisely.

6.4.2 Analysis of the $\hat{\alpha}_i(q,\, g)$

Since $h = L_K(1) = \prod_{j=1}^{2g}(1-\omega_j)$, where $\omega_j \in \mathbb{C}$ and $|\omega_j| = \sqrt{q}$, for $1 \leq j \leq 2g$, h varies according to the distribution of the $\omega_j/\sqrt{q}$ around the unit circle. Likewise, the estimates, E_1 and E_2, and errors, U_1, U_2, and U_3, depend on the distribution of the $\omega_j/\sqrt{q}$. More specifically, the proof of Corollary 4.9 in [SS07] used the upper bound $\left|\sum_{j=1}^{2g}\omega_j^n\right| \leq \sum_{j=1}^{2g}\left|\omega_j^n\right| \leq 2gq^{n/2}$, for all $n \in \mathbb{N}$, to determine an upper bound on the sum $\sum_{\nu|n}\nu S_\nu(n/\nu)$ for both E and U. In this section, we will take a deeper look into the relationship between the roots of $L_K(1)$, the averages, $\alpha_i(q,\, g)$, the estimates, E_i, and the error bounds, U_i. In particular, we will explain the obstructions to computing the $\alpha_i(q,\, g)$ more precisely.

Let $\omega_j = \sqrt{q}e^{\varphi_j i}$, where i is a fixed square root of -1 and each $0 \leq \varphi_j < 2\pi$, for $1 \leq j \leq 2g$. It is well-known that the φ_j can be arranged so that $\omega_j = \overline{\omega}_{j+g}$ and $\varphi_j \equiv -\varphi_{j+g} \pmod{2\pi}$. (See, for example, Theorem V.1.15 of [Sti93].) We may therefore order the φ_j so that $0 \leq \varphi_j \leq \pi$, for $1 \leq j \leq g$. In short, if each φ_j is close to either 0 or π, then $|\omega_j|$ is close to $\sqrt{q}$ and $|h - E|$ will be very close to U. On the other hand, if the average of the φ_j is close to $\pi/2$, then $|h - E|$ will be very close to 0. From our experimental results, for a fixed characteristic q and genus g, the distribution of $|h - E|$ was distributed roughly symmetrically about 0. Therefore, it is a reasonable assumption that over all cubic function fields of a fixed characteristic and genus, the average of the φ_j, $(1/g)\sum_{j=1}^{g}\varphi_j$, is distributed symmetrically about $\pi/2$. As the genus increases, it becomes less likely for $(1/g)\sum_{j=1}^{g}\varphi_j$, for any given function field, to be very close to either 0 or π and more likely that this average is close to $\pi/2$. Hence, for increasing genus, we expect a decreasingly smaller proportion of function fields with $|h - E|/U$ close to 1. On the other hand, for larger genera, the sum of the $\omega_j = \sqrt{q}e^{\varphi_j i}$ will tend to be much closer to 0, which would explain the lower values of the $\hat{\alpha}_i(q,\, g)$ in Table 6.5 with increasing genus. Likewise, we expect that increasing genera yield smaller values of the $\alpha_i(q,\, g)$. This also explains why the minimum and maximum values of the $|h - E_i|/U_i$ in Table 6.4 generally decrease with increasing genus.

If the values, φ_j, for $1 \leq j \leq g$, were distributed randomly in the interval $[0,\, \pi]$, over all function fields of a fixed extension degree, genus, and base field, then precise values of the $\alpha_i(q,\, g)$ could be

obtained for each $1 \leq i \leq 3$. Unfortunately, however, this is not the case, so we cannot make this assumption. Let

$$F_\lambda(\varphi_1, \ldots, \varphi_g) = \epsilon(\lambda, q) + \sum_{j=1}^{2g} e^{j(\lambda+1)\varphi_j} = \epsilon(\lambda, q) + 2\sum_{j=1}^{g} \cos((\lambda+1)\varphi_j) \ ,$$

where

$$\epsilon(\lambda, q) = \frac{\lambda+1}{l} S_{(\lambda+1)/l}(l) q^{-(\lambda+1)/l}$$

and l is the smallest prime factor of $\lambda + 1$. From (5.6), (5.7), (5.8), and (5.9) of [SS08], we have

$$\begin{aligned} Mean(F_\lambda) &= \lim_{q\to\infty} 2g\alpha_1(g, q) \\ &= \lim_{q\to\infty} \left(2g + 2q^{(\lambda+1)(1/l-1/2)}\right) \alpha_2(q, g) \\ &= \lim_{q\to\infty} (2g + \epsilon(\lambda, q))\alpha_3(g, q) \ , \end{aligned} \tag{6.18}$$

so $Mean(F_\lambda) = 2g\alpha_1(g)$, $Mean(F_\lambda) = (2g + 2\,\mathrm{parity}(\lambda))\,\alpha_2(g)$, and $Mean(F_\lambda) = (2g + \epsilon(\lambda))\alpha_3(g)$, where $\mathrm{parity}(\lambda) = \lambda \pmod 2$, $\epsilon(\lambda) = Mean\,(\epsilon(\lambda, q))$, and the average is taken over all cubic function fields of genus g and for $q \to \infty$. If λ is even, then $\epsilon(\lambda) = 0$ [SS08]. Therefore, if $Mean(F_\lambda)$ is known precisely, then we can determine the $\alpha_i(g)$ precisely. From (6.18), we also see why for those genera, g, for which $\lambda = 1$, the values of $\hat{\alpha}_2(q, g)$ are so much lower than the corresponding values of $\hat{\alpha}_1(q, g)$ and $\hat{\alpha}_3(q, g)$ in Table 6.5; the term $2q^{(\lambda+1)(1/l-1/2)} \neq 0$.

From Section 6 of [ST02a] and Section 5.4 of [SS08], we have:

$$Mean(F_\lambda) = \int_A F_\lambda \, d\mathrm{Harr} = \int_{[0,\,\pi]^g} F_\lambda(\varphi_1, \ldots, \varphi_g)\mu_g(d\varphi_1, \ldots, d\varphi_g) \ , \tag{6.19}$$

where Haar denotes the Haar measure of a subgroup, A, of the symplectic group $\mathrm{Sp}(2g)$, and μ_g denotes the appropriate measure. Given μ_g, the integral in (6.19) may be transformed into a Riemann integral and either evaluated directly or approximated via Riemann sums. The measure μ_g has been derived for elliptic function fields by Birch [Bir68], and by Katz, Sarnak, and Weyl [Wey68, KS99] for hyperelliptic function fields of genus $g > 1$. Unfortunately, μ_g is not known for any function fields of degree greater than 2. For this reason, determining the precise values of $Mean(F_\lambda)$, and, in turn, the $\alpha_i(g)$, is very difficult.

In Table 6.6, we give estimates for the value of $Mean(F_\lambda)$ based on the approximations, $\hat{\alpha}_i(q, g)$. In light of (6.18), we define $\hat{F}_{\lambda,1}(q, g) = 2g\hat{\alpha}_1(q, g)$, $\hat{F}_{\lambda,2}(q, g) = \left(2g + 2q^{(\lambda+1)(1/l-1/2)}\right)\hat{\alpha}_2(q, g)$, and $\hat{F}_{\lambda,3}(q, g) = (2g + \epsilon(\lambda, q))\hat{\alpha}_3(q, g)$ in Table 6.6. One observation we note is that the difference, $\hat{F}_{\lambda,1}(q, g) - \hat{F}_{\lambda,3}(q, g)$, decreases with increasing q, so it appears likely that the error, $\lim_{q\to\infty} \epsilon(1, q) = 0$ as well.

It therefore remains an open problem to compute the $\alpha_i(g)$ precisely. As such, we must rely on the approximations given in Table 6.5 to achieve average running times of Algorithms 6.3.20 and 6.3.21 that are close to optimal. In the next section, we describe improvements that apply in particular to these computations in the infrastructure of a purely cubic function field.

Table 6.6: Comparison of the Approximations of $F_\lambda(q,\, g)$

q	g	λ	$\hat{F}_{\lambda,1}(q,\, g)$	$\hat{F}_{\lambda,2}(q,\, g)$	$\hat{F}_{\lambda,3}(q,\, g)$	n
997	3	1	1.60993834	1.60026722	1.58689641	10000
10009	3	1	1.62190905	1.61879311	1.61437049	10000
100003	3	1	1.63362459	1.63267624	1.63124940	10000
997	4	1	1.53791722	1.53060814	1.52159528	10000
10009	4	1	1.54023820	1.53791100	1.53490540	10000
97	5	2	1.81956325	1.74914564	1.79810870	10000
997	5	2	1.91884231	1.90141361	1.91906074	10000
97	6	2	1.82952786	1.75672156	1.81425459	10000
463	6	2	1.91915516	1.89579266	1.91707890	10000
19	7	2	1.59996870	1.46545227	1.52729767	10000
97	7	2	1.77577673	1.72945415	1.76430410	10000

6.4.3 Optimization Data for Infrastructures

In this section, we present computational results on the ratios of the time to compute certain infrastructure operations. Let $T_{B,1}$, $T_{G,1}$, and $T_{I,1}$ be the time to compute a baby step, a giant step, and $Inverse(D)$, for some $D \in \mathcal{R}$, respectively. As in Algorithms 6.2.6 and 6.2.11 and Propositions 6.2.8 and 6.2.12, we define $\tau_2 = (T_{G,1}+T_{I,1})/T_{B,1}$, $\tau_3 = T_{G,1}/T_{B,1}$. Recall that τ_2 is used to improve running times for the Baby Step-Giant Step method in infrastructures and τ_3, via the related τ, enables a speed-up for the Kangaroo algorithm. The ratios were determined for genera $3 \leq g \leq 7$ via computations over at least ten function fields $\mathbb{F}_q(C)$, with $C : Y^3 = GH^2$, with the given values of q and the degrees of G and H. The data is summarized in Table 6.7.

Table 6.7: Comparison of Giant Step to Baby Step Ratios in Unit Rank 1 Infrastructure

g	q	$\deg(G)$	$\deg(H)$	τ_2	τ_3	τ
3	7001	4	1	4.20202	2.88781	3
3	7001	1	4	4.47472	3.17201	3
4	719	6	0	5.49073	3.88672	4
4	719	3	3	6.11146	4.07611	4
5	197	5	2	8.09893	5.22904	5
5	197	2	5	8.47669	5.57365	6
6	71	7	1	9.02126	5.82408	6
6	71	4	4	9.33196	6.21530	6
6	71	1	7	9.69945	6.64731	7
7	41	9	0	11.26627	7.28357	7
7	41	6	3	11.56116	7.45021	7
7	41	3	6	11.90331	7.85012	8

As is the case in hyperelliptic function fields [Ste01], this experimental evidence indicates that the baby step operation is significantly faster than the giant step operation. For future work, we would like to analyze the theoretical running times of the baby step, giant step, and inverse operations in the infrastructure of a purely cubic function field as a means of comparing these ratios to predicted ratios. Another observation of note is that in each case, a greater degree of H led to slower giant step

arithmetic. For a fixed genus, as $\deg(H)$ increases, $\deg(GH^2)$ increases. Now Step 6 in Algorithm 4.4.8, for ideal squaring, and Step 6 of Algorithm 4.4.9, for the multiplication of ideals whose product is primitive, but whose norms are not relatively prime, both use GH^2 to compute the basis element U of the product $\mathfrak{a}_1\mathfrak{a}_2 = \mathfrak{a} = [S,\, S'(U+\rho),\, S''(V+W\rho+\omega)]$. Algorithm 4.4.8 also uses H in Steps 2, 3, 8, and 10. Thus, for a fixed genus, the steps in these two algorithms will require operating with longer polynomials for larger $\deg(H)$. However, Algorithms 4.4.8 and 4.4.9 are very rarely called during any computation, so it is unlikely that this is the explanation of the different ratios in Table 6.7. Therefore, it is still unclear, from a theoretical standpoint, why larger $\deg(H)$ yields slower giant step arithmetic. Notice that if $C_1 : Y^3 = GH^2$, $K_1 = \mathbb{F}_q(C_1)$, $C_2 : Y^3 = G^2H$, and $K_2 = \mathbb{F}_q(C_2)$ then $K_1 \cong K_2$ and $\mathcal{O}_{K_1} \cong \mathcal{O}_{K_2}$; if $\mathcal{O}_{K_1} = [1,\, \rho,\, \omega]$, then $\mathcal{O}_{K_2} = [1,\, \omega,\, \rho]$. Therefore, to obtain faster arithmetic, we should always choose G and H with $\deg(G) \geq \deg(H)$. In the next section, we will consider other attempts to improve the performance of Algorithms 6.3.20 and 6.3.21.

6.4.4 Further Improvement Attempts

We noted in Section 6.3.2 that the running time of Algorithm 6.3.1 would be dominated by Phase 3. The main strategy behind speeding up the time to compute h is to spend more time in Phase 1 to compute a better estimate of h and a smaller upper bound on the error, thus balancing the running times of Phases 1 and 3. We briefly discuss two ideas to realize this.

First, increasing $\lambda \in \mathbb{N}_0$ will yield better estimates, but this is only efficient for very small base fields. For larger genera, this may be feasible as well, but it is not for function fields of genus 3 and 4 over large base fields. A second approach attempts to approximate $E_2'(\lambda+1,\, K)$, via (6.17), for a given cubic function field, K, by computing $E_2'(\lambda,\, K)$ and approximating the factor

$$\frac{E_2'(\lambda+1,\, K)}{E_2'(\lambda,\, K)} = \left(\frac{q^{2(\lambda+1)}}{q^{2(\lambda+1)}-1}\right)^{n_3(\lambda+1)} \left(\frac{q^{2(\lambda+1)}}{(q^{\lambda+1}-1)^2}\right)^{n_4(\lambda+1)} \left(\frac{q^{2(\lambda+1)}}{q^{2(\lambda+1)}+q^{\lambda+1}+1}\right)^{n_5(\lambda+1)},$$

where $n_i(\lambda+1)$ is the number of irreducible polynomials of degree $\lambda+1$ that split into type i prime ideals of $\mathcal{O}_K$, for $i = 3,\, 4,\, 5$. The idea is to choose a subset, $\mathcal{A}$, of the set of irreducible polynomials of degree $\lambda+1$ and compute the number, $n_i(\lambda+1,\, \mathcal{A})$, of irreducible polynomials in $\mathcal{A}$ that split into type i prime ideals of $\mathcal{O}_K$, for $i = 3,\, 4,\, 5$. If $I_{\lambda+1}$ is the number of irreducible polynomials of degree $\lambda+1$, then we approximate each $n_i(\lambda+1)$ with $n_i(\lambda+1,\, \mathcal{A})I_{\lambda+1}/\mathcal{A}$. We ran several experiments using this method, but unfortunately, it yielded worse estimates of h and slower overall running times. Therefore, the approximations of these $n_i(\lambda+1)$ are not very close to the actual values. As such, it does not appear likely that we will be able to balance the running times of Phases 1 and 3 for large examples.

In the following sections, we summarize results on the application of this data to Algorithms 6.3.20 and 6.3.21 for large examples.

6.4.5 Unit Rank 0 Computations

We used the estimate $E = E_2$ and the error bound $U = U_3$ and applied the values of $\hat{\alpha}_3(q,\, g)$, for the largest values of q in Table 6.5, to the problem of computing large class numbers using the Kangaroo Method; that is, Algorithm 6.3.20, with Algorithm 6.2.9 as a subroutine. We will

henceforth denote this value of $\hat{\alpha}_3(q,\, g)$ that we use in our computations by $\hat{\alpha}(g)$. The divisor class numbers we computed, however, were for purely cubic function fields over base fields, $\mathbb{F}_q$, with much larger q than those in Table 6.5. We computed the divisor class numbers of four genus 3 and five genus 4 purely cubic function fields of unit rank 0, launching two large parallel computations, using 18 and 20 processors on genus 3 and 4 examples, respectively, to find the corresponding 28 and 25 decimal digit class numbers. Both Phases 1 and 3 were parallelized in these examples. Except for the case of the function fields $\mathbb{F}_q(C_1)$ and $\mathbb{F}_q(C_5)$ in Table 6.8, the divisor class groups were too large for the Baby Step-Giant Step method to be used efficiently with the available memory. (However, we only used the Kangaroo method on these function fields.) In this section, we present the results of these calculations, including timing data and the choices of certain variables. We began with smaller examples in order to test the choices of certain parameters, in particular, the parameter θ, which regulates how often we set a kangaroo trap, and to better estimate the expected time to compute larger divisor class numbers. We list the divisor class numbers we computed in Table 6.8 and corresponding statistics in Tables 6.10 and 6.11 for the genus 3 and 4 examples, respectively.

In the implementation, we used two programs. The "master" program checked for collisions periodically and reassigned kangaroos for the large parallel search when a collision between kangaroos of the same herd was detected. If the expected running time was less than ten days, the master program checked for solutions every 10% of the estimated expected running time (specifically, the function fields $\mathbb{F}_q(C_i)$, for $1 \leq i \leq 8$, below), otherwise, specifically for the function field $\mathbb{F}_q(C_9)$, it checked for solutions daily. A kangaroo was reassigned by multiplying it by $\mathfrak{g}^{3b}$, where $\mathfrak{g}$ was the random ideal generated in Step 5 of Algorithm 6.2.9 and $h \equiv a \pmod{b}$, with a and b as given in the input. In the examples of this section, we have $h \equiv 1 \pmod{3}$ so that $b = 3$. Meanwhile, the "slave" program computed the splitting behavior of the polynomials in a distributed approach to Phase 1, and performed the actual kangaroo jumps. For Phase 1 in particular, since q was chosen to be prime and $\lambda = 1$ in each case, we divided up the set of polynomials, $\{x,\, x+1,\, \ldots,\, x+q-1\}$, among the m kangaroos via $\{x,\, \ldots,\, x+\lfloor q/m \rfloor\}$, $\{x+\lfloor q/m \rfloor+1,\, \ldots,\, x+\lfloor 2q/m \rfloor\}$, ..., $\{x+\lfloor q(m-1)/m \rfloor+1,\, \ldots,\, x+q-1\}$. Then each processor computed the splitting behavior of the polynomials in the assigned subset, sending the results to the master program. The master collected this data, computed E, U, the jump distances, jump set, and θ and sent this data to the kangaroos, which then began Phase 3, the jumping phase.

The genus 3 curves we used for the examples in this section were:

$$\begin{aligned}
&C_1 : Y^3 = x^4 + 4767220x^3 + 9719260x^2 + 9796683x + 9650320\ , && q = 10^7 + 141\\
&C_2 : Y^3 = x^4 + 39760243x^3 + 80354454x^2 + 40601482x + 72689039\ , && q = 10^8 + 39\\
&C_3 : Y^3 = x^4 + 40958421x^3 + 76820587x^2 + 84271053x + 66338979\ , && q = 10^8 + 39\\
&C_4 : Y^3 = x^4 + 512964174x^3 + 604076970x^2 + 208417608x + 702771176\ , && q = 10^9 + 9
\end{aligned}$$

and the genus 4 curves were:

$$\begin{aligned}
&C_5 : Y^3 = x^5 + 6841x^4 + 8688x^3 + 6670x^2 + 5232x + 6608\ , && q = 10^4 + 9\\
&C_6 : Y^3 = x^5 + 24190x^4 + 76617x^3 + 20848x^2 + 52712x + 64759\ , && q = 10^5 + 3\\
&C_7 : Y^3 = x^5 + 70599x^4 + 31259x^3 + 68336x^2 + 2756x + 62207\ , && q = 10^5 + 3\\
&C_8 : Y^3 = x^5 + 531472x^4 + 146921x^3 + 387330x^2 + 602740x + 79247\ , && q = 10^6 + 3\\
&C_9 : Y^3 = x^5 + 537882x^4 + 755468x^3 + 137780x^2 + 366795x + 268815\ , && q = 10^6 + 3\ .
\end{aligned}$$

Each curve, $C_i : Y^3 = F_i$, is nonsingular and each F_i is irreducible over the field, $\mathbb{F}_q$, used in the respective cases; we used a random irreducible polynomial generator, supplied by NTL, to choose these polynomials. In each case, we used a constant field $\mathbb{F}_q$, with prime $q \equiv 1 \pmod 3$, chosen to be the smallest such prime greater than a certain power of 10. The divisor class number and the values $|h - E|/U$ are given for each example in Table 6.8, and the number of decimal digits of each divisor class number, along with their factorizations, are given in Table 6.9.

Table 6.8: Divisor Class Numbers of Cubic Function Fields, $r = 0$

Curve	q	g	h	$\lvert h - E\rvert/U$
C_1	$10^7 + 141$	3	1000150832447729149744	0.0762951
C_2	$10^8 + 39$	3	100001837235320357829924 7	0.2602616
C_3	$10^8 + 39$	3	100005716580790372483994 8	0.1934522
C_4	$10^9 + 9$	3	100002028513299830459563297 9	0.0241890
C_5	$10^4 + 9$	4	10226409142466713	0.0148396
C_6	$10^5 + 3$	4	99732647709406519123	0.0704916
C_7	$10^5 + 3$	4	99648777459613902604	0.0374562
C_8	$10^6 + 3$	4	1001264259802134080148796	0.3835040
C_9	$10^6 + 3$	4	1000973897942768635726975	0.1432481

Table 6.9: Factorization of Divisor Class Numbers of Cubic Function Fields, $r = 0$

Curve	q	g	*dig.*	Factorization of h
C_1	$10^7 + 141$	3	22	$2^4 \cdot 3001 \cdot 4159 \cdot 5008303077301$
C_2	$10^8 + 39$	3	25	$19 \cdot 43 \cdot 1224012695658755909791$
C_3	$10^8 + 39$	3	25	$2^2 \cdot 7^2 \cdot 3571 \cdot 300701761 \cdot 4751633473$
C_4	$10^9 + 9$	3	28	$13 \cdot 19 \cdot 73 \cdot 114859 \cdot 482863041248304151$
C_5	$10^4 + 9$	4	17	$7 \cdot 19 \cdot 31 \cdot 1013227 \cdot 2447953$
C_6	$10^5 + 3$	4	20	$139 \cdot 1787587 \cdot 401379660211$
C_7	$10^5 + 3$	4	20	$2^2 \cdot 7^2 \cdot 508412129895989299$
C_8	$10^6 + 3$	4	25	$2^2 \cdot 4549 \cdot 55026613530563534851$
C_9	$10^6 + 3$	4	25	$5^2 \cdot 9199 \cdot 4352533527308484121$

While much larger divisor class numbers have been computed for genus 3 cubic function fields by Bauer, Teske, and Weng [BTW05, Wen06], the divisor class numbers of $\mathbb{F}_{10^6+3}(C_8)$ and $\mathbb{F}_{10^6+3}(C_9)$ are the largest known divisor class numbers of any cubic function field of genus at least 4.

In Tables 6.10 and 6.11, we give results from the computations of the class numbers listed in Table 6.8. Here, m is the number of processors used (if $m = 1$, then a tame and a wild kangaroo ran on the same processor), "Ph. 1" and "Ph. 3" give the times (in seconds) the respective phases took to complete, and "Total" is the sum of these times. In Table 6.11, we only give the total time, since Phase 1 required very little time compared with Phase 3. "Exp. 1" gives the quantity, $(m|h - E|/\beta + 4\beta/3m + \theta m)\, T_{G,0}$, obtained from Proposition 6.2.10 and its proof, where $T_{G,0}$ was the time to compose two ideals in the given example, $\beta = (m/2)\sqrt{3\hat{\alpha}(g)U}$ was the average jump distance in the example, E was the estimate of h, and U was the upper bound on the error; the quantity Exp. 1 estimates the expected time to compute the class number of the specific function field $\mathbb{F}_q(C_i)$ using a single processor, based on the parameters given in Proposition 6.2.10. "Exp.

2" gives the quantity $\left(4\sqrt{\hat{\alpha}(g)U/3}+\theta m\right)T_{G,0}$, obtained from Proposition 6.2.10, which estimates the expected time to compute the divisor class number of a purely cubic function field of the given characteristic, q, and genus, g, using a single processor. Finally, "Jumps" gives the total number of kangaroo jumps in the computation, $\lg\theta$ indicates our choice of θ, and "Traps" records the number of kangaroo traps that were set.

Table 6.10: Divisor Class Number Computation Data for Cubic Function Fields, $r = 0$, $g = 3$

Curve	q	m	Ph. 1	Ph. 3	Total	Exp. 1	Exp. 2	Jumps	$\lg\theta$	Traps
C_1	10^7+141	1	553	4178	$78.9\,m$	$180\,m$	$243\,m$	6220320	16	90
C_2	10^8+39	1	6098	114720	$33.6\,h$	$41.0\,h$	$41.4\,h$	174938127	18	627
C_3	10^8+39	2	11158	135576	$40.8\,h$	$39.0\,h$	$41.4\,h$	202951598	16	3135
C_4	10^9+9	18	64577	1695582	$20.4\,d$	$8.61\,d$	$15.1\,d$	2880612442	20	2779

Table 6.11: Divisor Class Number Computation Data for Cubic Function Fields, $r = 0$, $g = 4$

Curve	q	m	Total	Exp. 1	Exp. 2	Jumps	$\lg\theta$	Traps
C_5	10^4+9	1	$6.9\,m$	$17.5\,m$	$31.4\,m$	451594	14	27
C_6	10^5+3	2	$4.5\,h$	$11.1\,h$	$16.1\,h$	17797938	16	265
C_7	10^5+3	2	$9.8\,h$	$10.5\,h$	$16.1\,h$	35641743	16	522
C_8	10^6+3	20	$72.4\,d$	$45.2\,d$	$30.3\,d$	4872971415	20	4597
C_9	10^6+3	2	$34.0\,d$	$37.2\,d$	$30.3\,d$	1615079674	18	6330

We make a few observations about this data. First, the values in the Exp. 1 column were greater than or less than those in the Exp. 2 column if $|h-E|/U$ was greater than or less than $\hat{\alpha}(g)$, respectively. In other words, if K is a function field such that $|h-E|/U < \hat{\alpha}(g)$, then h is closer to the center of the interval $(E-U,\, E+U)$ than what our experimental evidence suggests is average. Therefore, we expect to compute h faster than our estimated expected running time in this case.

Next, we notice the amount of variation between the actual time to compute certain divisor class numbers and the expected time, Exp. 1, to compute these values using the Kangaroo method. For the examples above, the computations finished sooner than expected (in some cases much sooner) for the function fields generated by C_1, C_2, C_5, C_6, C_7, and C_9, and later than expected for the function fields generated by C_3, C_4, and C_8. (The computation for the function field $\mathbb{F}_{10^9+9}(C_4)$ finished much later than expected.) For any given class group, the time to compute h depends on the intersection of two kangaroo paths. This intersection can occur anytime on the continuum between the occurrence of the trailing herd catching up to the path of the leading herd and never (if both herds become trapped in independent infinite loops). For a given choice of β, there are several possible choices for a set of jumps, $\{s_1, \ldots, s_{64}\}$, chosen as described in Step 4 of Algorithm 6.2.9. The number of jumps required to compute h depends uniquely on this choice; since the Kangaroo algorithm is heuristic, it is not guaranteed to run in the same amount of time on successive executions. Therefore, for one set of jumps, the computation may happen to finish earlier than expected while for another set, the computation may run longer than expected. It is impossible to know in advance how one choice of jump distances will affect the running time. For one example of the level of variation that occurs, the

estimate of the divisor class number of $K_3 = \mathbb{F}_{10^8+39}(C_3)$ was better than that of $K_2 = \mathbb{F}_{10^8+39}(C_2)$, yet the divisor class number of K_3 required over 7 hours more total machine time to compute than that of K_2. We see more extreme variation between the genus 4 function fields $\mathbb{F}_{10^5+3}(C_6)$ and $\mathbb{F}_{10^5+3}(C_7)$.

We also compare these results with timing data for the Baby Step-Giant Step computations used to construct Table 6.5. Via the Baby Step-Giant Step method, the average time to compute a divisor class number of a genus 4 cubic function field of characteristic 10009 was 22.5 minutes. Comparing this with the estimated expected time to compute a divisor class number of a genus 4 cubic function field of characteristic 10009 via the Kangaroo method, 31.4 minutes, we see that the Baby Step-Giant Step method is preferable for function fields of that size. This justifies the choice of the threshold in Algorithm 6.3.20.

Finally, in the large parallelized examples we recorded the number of useless collisions. For $\mathbb{F}_{10^9+9}(C_4)$ there were 2 useless collisions, both among the tame kangaroos, and for $\mathbb{F}_{10^6+3}(C_8)$, there were 10 useless collisions: 3 among the tame kangaroos and 7 among the wild. Such a few number of collisions was to be expected, based on the results of experiments by Stein and Teske in hyperelliptic function fields [ST05].

In the next section, we will summarize the results of S-regulator computations in cubic function fields of unit rank 1.

6.4.6 Unit Rank 1 Computations

In this section, we tested the practical effectiveness of Algorithm 6.3.21 to compute the divisor class number and extract the ideal class number and S-regulator of nine purely cubic function fields of unit rank 1: five function fields of genus 3 and four of genus 4. We used the Kangaroo method as a subroutime in each case, but used the Baby Step-Giant Step method on three of the function fields as well. The purpose of the Baby Step-Giant Step computations was to compare the timing results with those of the Kangaroo method in order to find a practical cut-off for the preference of one method over the other on a machine with 1 GB of memory. We also wished to compare the relative performance of unit rank 0 and unit rank 1 computations in cubic function fields of similar sizes and to use the timings to estimate the expected time to compute the divisor class number and S-regulator of a unit rank 1 cubic function field of genera 3 and 4 over larger base fields. We list the ideal class number, h_x, the S-regulator, R^S, and the ratio $|h - E|/U$ in Table 6.12, the corresponding number of decimal digits and factorization of $h = h_x R^S$ in Table 6.13, data from the Baby Step-Giant Step computations in Table 6.14, and data from the Kangaroo computations in Table 6.15. The largest two examples for each genus were computed via a parallelized approach, as in the unit rank 0 case, using up to 20 processors. The largest divisor class number we computed in the genus 3 case had 28 decimal digits, while the largest divisor class number we computed in the genus 4 case had 25 decimal digits.

The genus 3 curves that we used for the examples in this section were:

$$\begin{aligned}
&C_{10} : Y^3 = \left(x^4 + 959949x^3 + 364016x^2 + 878485x + 900525\right) x^2 \ , && q = 10^6 + 37 \\
&C_{11} : Y^3 = \left(x^4 + 822453x^3 + 2006830x^2 + 1787014x + 440837\right) x^2 \ , && q = 2154491 \\
&C_{12} : Y^3 = \left(x^4 + 8344544x^3 + 90646x^2 + 4967909x + 1699817\right) x^2 \ , && q = 10^7 + 19 \\
&C_{13} : Y^3 = \left(x^4 + 95736325x^3 + 52482514x^2 + 34776188x + 44856022\right) x^2 \ , && q = 10^8 + 7 \\
&C_{14} : Y^3 = \left(x^4 + 852737742x^3 + 113051170x^2 + 250054066x + 513859851\right) x^2 \ , && q = 10^9 + 7
\end{aligned}$$

and the genus 4 curves were:

$$\begin{aligned}
&C_{15} : Y^3 = \left(x^3 + 2833x^2 + 2425x + 5216\right) \left(x^3 + 6412x^2 + 3035x + 192\right)^2 \ , && q = 10^4 + 7 \\
&C_{16} : Y^3 = G_{16} H_{16}^2 \ , && q = 10^5 + 19 \\
&C_{17} : Y^3 = G_{17} H_{17}^2 \ , && q = 10^5 + 19 \\
&C_{18} : Y^3 = G_{18} H_{18}^2 \ , && q = 10^6 + 37 \ ,
\end{aligned}$$

with

$$\begin{aligned}
&G_{16} = x^3 + 18559x^2 + 21371x + 89569 \quad \text{and} \quad H_{16} = x^3 + 1149x^2 + 83421x + 94387 \ , \\
&G_{17} = x^3 + 61088x^2 + 28362x + 94710 \quad \text{and} \quad H_{17} = x^3 + 58255x^2 + 19761x + 28808 \ , \quad \text{and} \\
&G_{18} = x^3 + 918037x^2 + 460902x + 923544 \quad \text{and} \quad H_{18} = x^3 + 891576x^2 + 694204x + 79732 \ .
\end{aligned}$$

In each case, we used a constant field $\mathbb{F}_q$, with prime $q \equiv 2 \pmod 3$, chosen to be the smallest such prime so that q^g is greater than certain powers of 10. We also have $C_i : Y^3 = G_i H_i^2$, where G_i and H_i are relatively prime and irreducible over the field $\mathbb{F}_q$ used in the respective cases. As with the curves generated for the unit rank 0 examples, each G_i and H_i was computed using a random irreducible polynomial generator supplied by NTL. For the genus 4 examples here, we chose $\deg(G_i) = \deg(H_i) = 3$, rather than $\deg(G_i) = 6$, $\deg(H_i) = 0$ in order to computationally verify the ideal and infrastructure arithmetic derived in the previous two chapters. In addition, the arithmic of these function fields is the slowest under the restriction $\deg(G_i) \geq \deg(H_i)$; that is, among all the cases where $\deg(G) \geq \deg(H)$, the arithmetic is slowest when $\deg(G) = \deg(H)$ and fastest when $\deg(H) = 0$. Thus, we wished to obtain the worst-case computation times for function fields of genus 4. The difference, however, is not very great. For example, we computed 110.6 giant steps and inverses per second in the infrastructure of $K_{15} = \mathbb{F}_{10^4+7}(C_{15})$ (see the data in Table 6.14), while we computed 127.9 giant steps and inverses per second in the infrastructure of a genus 4 function field defined by a nonsingular curve. This difference is small enough to provide a reasonable comparison with the corresponding unit rank 0 computations.

In Table 6.14, "BS" and "GS" are the respective number of baby steps and giant steps (with ideal inverses) computed in each case, "BS Time" and "GS Time" are the number of seconds taken to compute those steps, "Total" is the total computation time, and "Exp." is the quantity, $\left(2\sqrt{\hat{\alpha}(g)\tau_2 U}\right) T_{B,1}$, where τ_2 is as given in Table 6.7, U was the upper bound on the error, $|h - E|$, and $T_{B,1}$ was the time to compute a baby step in the given example; Exp. estimated the amount of time required to compute the S-regulator of a unit rank 1 purely cubic function field of the given characteristic, q, and genus, g. We omitted timing data on Phases 1 and 4 since Phase 1 took under 1 second in each case and extracting R^S from h in Phase 4 took at most 6 seconds.

Table 6.12: S-Regulators and Ideal Class Numbers of Cubic Function Fields, $r = 1$

Curve	q	g	h_x	R^S	$\lvert h - E\rvert/U$
C_{10}	$10^6 + 37$	3	18	55561791851695519	0.4159135
C_{11}	2154491	3	135	74079930060193896	0.4617141
C_{12}	$10^7 + 19$	3	3	333335295493450981720	0.4660491
C_{13}	$10^8 + 7$	3	3	333333410692036555362600	0.5518563
C_{14}	$10^9 + 7$	3	12	833333350639834005118671360	0.0580483
C_{15}	$10^4 + 7$	4	48	208911295254144	0.0709538
C_{16}	$10^5 + 19$	4	3	33359418825784135923	0.2460722
C_{17}	$10^5 + 19$	4	36	27799750290048140 61	0.6139500
C_{18}	$10^6 + 37$	4	9	111127791704815995713577	0.4230388

Table 6.13: Divisor Class Numbers of Cubic Function Fields, $r = 1$

Curve	q	g	$dig.$	Factorization of $h = h_x R^S$
C_{10}	$10^6 + 37$	3	19	$2 \cdot 3^2 \cdot 13 \cdot 383 \cdot 12821 \cdot 870386641$
C_{11}	2154491	3	20	$2^3 \cdot 3^5 \cdot 5 \cdot 17 \cdot 19949 \cdot 3033877321$
C_{12}	$10^7 + 19$	3	22	$2^3 \cdot 3 \cdot 5 \cdot 11 \cdot 4441 \cdot 166667 \cdot 1023524479$
C_{13}	$10^8 + 7$	3	25	$2^3 \cdot 3^4 \cdot 5^2 \cdot 17 \cdot 47 \cdot 293 \cdot 154321 \cdot 1708620677$
C_{14}	$10^9 + 7$	3	28	$2^{10} \cdot 3^2 \cdot 7 \cdot 11 \cdot 109^2 \cdot 167 \cdot 710227281795313$
C_{15}	$10^4 + 7$	4	17	$2^{11} \cdot 3^2 \cdot 7 \cdot 17 \cdot 479 \cdot 877 \cdot 10883$
C_{16}	$10^5 + 19$	4	21	$3^2 \cdot 29 \cdot 3257 \cdot 117728460453997$
C_{17}	$10^5 + 19$	4	21	$2^2 \cdot 3^4 \cdot 37 \cdot 8348273360374817$
C_{18}	$10^6 + 37$	4	25	$3^5 \cdot 25603 \cdot 160756322978377817$

We make some observations about this data. We first notice from the timing data in Table 6.14 that with larger examples, the ratio of the time to compute a giant step (plus an inverse) to the time to compute a baby step, given by (GS/GS Time)/(BS/BS Time), was noticably larger than the corresponding ratios, τ_2, given in Table 6.7. Specifically, these ratios were 5.49656 and 7.64105 for the two genus 3 examples, $K_{10} = \mathbb{F}_{10^6+37}(C_{10})$ and $K_{11} = \mathbb{F}_{2154491}(C_{11})$, respectively, in Table 6.14, versus 4.20202 in Table 6.7, and 7.44169 for the genus 4 example, vesus 6.11146 in Table 6.7. Based on the computation of 100000 baby steps and giant steps (plus inverses) in the infrastructures of K_{10} and K_{11}, the giant step (plus ideal inverse) to baby step ratios were 4.17797 and 4.08333, respectively. Therefore, the slower giant step (plus inverse) operations were a result of the quantity of the baby steps (or more precisely, the space allocated for the hash table) consuming a much larger proportion of the available memory. Likewise, based on the computation of 100000 baby steps and giant steps (plus inverses) in the infrastructure of $K_{15} = \mathbb{F}_{10^4+7}(C_{15})$, the giant step (plus ideal inverse) to baby step ratio was 6.03973, so again, the quantity of baby steps that were computed required too much memory for the hash table look-ups during the giant step stage to be performed as fast as possible. One remaining question, then, is: at what point does the Kangaroo method become preferable? Therefore, we considered the timing data of these and other examples via the Kangaroo method.

In Table 6.15, "BS Jumps" and "GS Jumps" refer to the respective number of baby steps and giant steps computed using the Kangaroo method in each example and "Time" refers to the total

Table 6.14: S-Regulator Computation Data for Cubic Function Fields via Baby Step-Giant Steps, $r = 1$

Curve	q	g	BS	BS Time	GS	GS Time	Total	Exp.
C_{10}	$10^6 + 37$	3	1852262	1465	674431	2932	$73.3\,m$	$56.4\,m$
C_{11}	2154491	3	3989823	6288	1612722	19421	$428.6\,m$	$295.5\,m$
C_{15}	$10^4 + 7$	4	2178173	2647	131919	1193	$64.1\,m$	$97.9\,m$

time taken in the computation. "Exp. 1" gives the quantity

$$\left(\frac{2m|h-E|}{\beta+2(\tau-1)}+\frac{2\beta}{(2\tau-1)m}+\frac{\theta m}{\tau}\right)\left(1+\frac{\tau-1}{\tau_3}\right)T_{G,1}\ ,$$

obtained from Proposition 6.2.12 and its proof, where τ and τ_3 were as given in Table 6.7, $\beta = m\sqrt{(2\tau-1)\hat{\alpha}(g)U} - 2(\tau-1)$ was the average jump distance in the example, E was the estimate of h, U was the upper bound on the error, and $T_{G,1}$ was the time to compute a giant step in the given example; the quantity Exp. 1 estimates the expected time to compute the divisor class number of the specific function field $\mathbb{F}_q(C_i)$ using a single processor, based on the parameters given in Proposition 6.2.12. "Exp. 2" gives the quantity $\left(4\sqrt{\hat{\alpha}(g)U/(2\tau-1)}+\theta m/\tau\right)(1+(\tau-1)/\tau_3)\,T_{G,1}$, which estimates the expected time to compute the divisor class number of a purely cubic function field of the given characteristic, q, and genus, g, using a single processor. The remaining columns refer to the same data as in Tables 6.10 and 6.11.

Table 6.15: S-Regulator Computation Data for Cubic Function Fields via Kangaroos, $r = 1$

Curve	q	g	BS Jumps	GS Jumps	Time	Exp. 1	Exp. 2	$\lg\theta$	Traps
C_{10}	$10^6 + 37$	3	2825209	1414396	$69.1\,m$	$100.4\,m$	$79.3\,m$	14	291
C_{11}	2154491	3	7038874	3514570	$173.2\,m$	$233.6\,m$	$173.2\,m$	16	183
C_{12}	$10^7 + 19$	3	67353284	33673396	$36.2\,h$	$54.3\,h$	$23.8\,h$	20	143
C_{13}	$10^8 + 7$	3	829654748	414823730	$18.6\,d$	$34.6\,d$	$8.6\,d$	20	1213
C_{14}	$10^9 + 7$	3	3136227037	1568085553	$69.2\,d$	$71.3\,d$	$82.7\,d$	20	4547
C_{15}	$10^4 + 7$	4	2835998	945333	$32.9\,m$	$81.5\,m$	$117.7\,m$	16	20
C_{16}	$10^5 + 19$	4	259598086	86537770	$80.1\,h$	$49.7\,h$	$43.5\,h$	18	1322
C_{17}	$10^5 + 19$	4	167592157	55866713	$58.2\,h$	$88.5\,h$	$43.5\,h$	20	235
C_{18}	$10^6 + 37$	4	3127164698	1042434250	$88.4\,d$	$580.8\,d$	$127.2\,d$	22	1017

From the timing data in Table 6.15, we notice that the Kangaroo method was faster than the corresponding Baby Step-Giant Step computations in each case. However, when considering the estimated expected running times, the Kangaroo method is only preferred in the infrastructure of K_{10}. From this, we believe that for divisor class numbers of genus 3 cubic function fields on the order of 10^{18}, 1 GB of memory is sufficient to make the Baby Step-Giant Step method preferable, but that the Kangaroo method is more efficient to compute larger divisor class numbers. For genus 4 cubic function fields, if $q \lessapprox 10^4$ (or $h \lessapprox 10^{16}$), then the Baby Step-Giant Step method is preferred over the Kangaroo method, taking estimated expected time 97.9 versus 117.7 minutes for $q = 10^4 + 7$. Considering the number of baby steps computed by the respective examples in Table 6.14, and also

the similar estimated expected running times we just noted, it appears that for much larger q, the hash table storing the baby steps would consume too large a proportion of memory for the Baby Step-Giant Step method to be more effective than the Kangaroo method. This explains our choice for the threshold in Algorithm 6.3.21. For machines with more memory, this threshold will certainly be higher. From Table 6.2, we see that Algorithm 6.3.1 runs in time $O(q)$ and $O(q^{3/2})$ in the genus 3 and 4 cases, respectively. In the genus 3 case, then, if we have k GB of RAM available, then we can store k times as many baby steps, so the threshold is $q \approx k10^{18/3}$. In the genus 4 case, the threshold is $q \approx k^{2/3}10^4$.

We parallelized the computation of the S-regulators in five of the cases above. As with the unit rank 0 computations, there very few useless collisions in each case. For $\mathbb{F}_{10^7+19}(C_{12})$, we used 10 kangaroos, and there were 4 useless collisions among the tame kangaroos. For $\mathbb{F}_{10^8+7}(C_{13})$, we used 18 kangaroos, and there were 3 useless collisions among the wild kangaroos. For $\mathbb{F}_{10^9+7}(C_{14})$, we used 20 kangaroos, and there were 2 useless collisions among the tame kangaroos. For $\mathbb{F}_{10^5+19}(C_{17})$, we used 4 kangaroos, and there were no useless collisions. Finally, for $\mathbb{F}_{10^6+37}(C_{18})$, we used 18 kangaroos, and there were 4 useless collisions among the tame kangaroos.

Finally, recall Table 6.1. We showed that for a fixed genus, if composing two ideals in the ideal class group of a purely cubic function field of unit rank 0 was as fast as performing a giant step in the infrastructure of a purely cubic function field of unit rank 1, then computing the divisor class number in the unit rank 1 case via the Kangaroo method would be faster than the analogous computation in the unit rank 0 case. By considering the estimated expected running time of the Kangaroo method in Tables 6.10 and 6.11 with those in Table 6.15, that is, the "Exp. 2" columns, we may compare the speed of the Kangaroo method in each unit rank in practice. Specifically, we compared the estimated expected running times between function fields having base fields of similar size. In the genus 3 case, if $q \approx 10^7$, then the estimated expected running time was about 4.0 hours in the unit rank 0 case, versus 23.8 hours in the unit rank 1 case. For $q \approx 10^8$, the estimated expected running times were 41.4 hours versus 207.1 hours for the unit rank 0 and 1 cases, respectively. Finally, for $q \approx 10^9$, the estimated expected running times were 15.1 days versus 87.2 days for the unit rank 0 and 1 cases, respectively. Therefore, the estimated expected running times for the unit rank 0 computations are 5 to 6 times faster than the corresponding unit rank 1 computations in the cases here. In the genus 4 case, for $q \approx 10^4$, the estimated expected running times were 31.4 versus 117.7 minutes in the unit rank 0 and 1 cases, respectively; for $q \approx 10^5$, 16.1 versus 43.5 hours; and for $q \approx 10^6$, 30.3 versus 127.2 days. In the genus 4 case, the unit rank 0 computations were between 2.7 and 4.2 times faster than the corresponding unit rank 1 computations. Via Proposition 6.2.3, we took advantage of the fact that $h \equiv 1 \pmod 3$ to obtain a speed-up by a factor of $\sqrt{3}$ for the unit rank 0 computations. It must be the case, however, that infrastructure arithmetic, at least in the implementation that was used, is slower than arithmetic in $Cl(\mathcal{O})$. We noted earlier that arithmetic using non-singular curves is slightly faster than arithmetic using nonsingular curves; however, the largest contributing factor to the difference between computation times in ideal class groups and infrastructures is that reduction is much faster in $Cl(\mathcal{O})$ in the unit rank 0 case than reduction in $\mathcal{R}$.

By extrapolating the timing data in this section and Section 6.4.5, we estimated the time required to compute even larger divisor class numbers and S-regulators. This is detailed in the next section.

6.4.7 Projections

In this section, we make some concluding remarks on our divisor class number and S-regulator computations by estimating the expected time to compute even larger divisor class numbers and S-regulators, extrapolating the estimated expected running times of the Kangaroo method given in Tables 6.10, 6.11, and 6.15 in Sections 6.4.5 and 6.4.6. (Since these are large examples, we assume that the Baby Step-Giant Step algorithm would require much more memory than what would be available.) The times are based on the use of a Sun workstation with a single AMD Opteron 148 2.2 GHz processor and 1 GB of RAM, as was used in the other computations in this section. The first two tables give projections for the time to compute the divisor class number of a totally ramified purely cubic function field of genus 3 and 4, respectively. The last two give projections for the time to compute the S-regulator of a purely cubic function field of unit rank 1 of genus 3 and 4. We will first describe how we obtained estimates for the time required by Phase 1, the expected number of kangaroo jumps (including separate baby steps and giant steps for the unit rank 1 case), and the speed of the operations. Each table will apply the formulas to selected characteristics, q, for unit rank 0 and 1 and genus 3 and 4, but we will express the Phase 1 time and the number of steps as functions of q so that interpolations and extrapolations are possible.

For the Phase 1 times in Table 6.17, we used and extrapolated the Phase 1 data from Table 6.10. For the first example with $q \approx 10^8$, Phase 1 required 6098 seconds (1.69 hours), and for the example with $q \approx 10^9$, Phase 1 required 64577 seconds (17.94 hours), which is longer by a factor of 10.59. Recall that the running time for Phase 1 is $O(q^\lambda)$, where $\lambda = 1$ for $g = 3$ and $g = 4$. Therefore, we assumed the same slow-down factor, 10.59, between the Phase 1 times of any two function fields, $\mathbb{F}_{q_1}(C_1)$ and $\mathbb{F}_{q_2}(C_2)$, with $q_2/q_1 \approx 10$. (Larger q also yields slower polynomial arithmetic, so that we expect this factor to be larger than 10.) Thus, for $q \approx 10^{10}$, we estimated that Phase 1 will require $64577 \cdot 10.59 \approx 683870$ seconds, or about 7.92 days. The remaining Phase 1 estimations for Table 6.16 were computed similarly. To interpolate for any q, we assumed that Phase 1 requires no overhead and that increasing q by a factor of 10 yields an increase in the computation time by a factor of 10.59. It follows that for some $a \in \mathbb{R}$, Phase 1 will finish in $aq^{\log_{10}(10.59)}$ seconds. We solved $6098 = a(10^8 + 39)^{\log_{10}(10.59)}$ to obtain $a \approx 3.855 \cdot 10^{-5}$, or roughly $a = 1/25940$. We therefore estimate that Phase 1 would require $q^{\log_{10}(10.59)}/25940$ seconds.

To compute the expected number of kangaroo jumps, we considered the data in Table 6.1 and the following approximation of U. From (6.11), we have $U = \lfloor E(e^\psi - 1) \rfloor$. We noted in the analysis of Algorithm 6.3.1 that $U \approx E\psi$. Now $E \approx q^g$ and from (6.10), the dominant term of ψ is $(2g/(\lambda+1))\, q^{(\lambda+1)/2} = g/q$, since $\lambda = 1$ for $g = 3$ and $g = 4$. Thus, we approximated $U \approx gq^{g-1}$. In the unit rank 0 case, however, we have $h \equiv 1 \pmod 3$, so that via Proposition 6.2.10, we instead approximated $U \approx (g/3)q^{g-1}$. Therefore, for Table 6.16 ($r = 0$ and $g = 3$), the expected number of jumps is $2.0857\sqrt{U} = 2.0857q$ and for Table 6.17 ($r = 0$ and $g = 4$), the expected number of jumps is $1.7521\sqrt{U} = 1.7521\sqrt{(4/3)q^3} = 2.0232q^{3/2}$.[9] In the unit rank 1 case, we used the results of Proposition 6.2.12, so that we expect to make about $4\sqrt{\hat{\alpha}(g)U/(2\tau-1)} \approx 4\sqrt{\hat{\alpha}(g)gq^{g-1}/(2\tau-1)}$ giant steps and $4(\tau-1)\sqrt{\hat{\alpha}(g)U/(2\tau-1)} \approx 4(\tau-1)\sqrt{\hat{\alpha}(g)gq^{g-1}/(2\tau-1)}$ baby steps. We then substituted the appropriate values of $\hat{\alpha}(g)$ and τ from Tables 6.5 and 6.7, respectively. Thus, for $g = 3$, we expect to make approximately $4\sqrt{0.2718749 \cdot 3q^2/5} = 1.6155q$ giant steps and $8\sqrt{0.2718749 \cdot 3q^2/5} = 3.2311q$

[9] The quantities 2.0857 and 1.7521 were taken from Column 2 and Rows 1 and 2, respectively, of Table 6.1.

baby steps, and for $g = 4$, we expect to make approximately $4\sqrt{0.19186318 \cdot 4q^3/7} = 1.3245q^{3/2}$ giant steps and $12\sqrt{0.19186318 \cdot 4q^3/7} = 3.9734q^{3/2}$ baby steps.[10]

We note that the estimates: $\hat{\alpha}(g)$, τ_1, τ_2, and τ_3, were all computed using function fields of different characteristics, none of which are given in the tables below. We assume for fixed genus, unit rank, $\deg(G)$, and $\deg(H)$, that each of these variables converge as $q \to \infty$. Therefore, we have approximated these variables by computing estimates of their values, averaged over several function fields of a fixed large characteristic; for $g = 3$, $\hat{\alpha}(g)$ and τ_1 were determined from 10000 function fields of characteristic 100003 and τ_2 and τ_3 were determined from 10 function fields of characteristic 7001; for $g = 4$, $\hat{\alpha}(g)$ and τ_1 were determined from 10000 function fields of characteristic 10009 and τ_2 and τ_3 were determined from 10 function fields of characteristic 719.[11] Actual values will certainly differ from these estimates, but we assume that our estimates are close enough to the actual values in order to provide reasonable figures for the estimated running times in the tables below.

From the expected number of jumps, we estimated the expected running time by using the speed of the arithmetic of the largest function fields listed for each unit rank and genus in Tables 6.10, 6.11, and 6.15. Specifically, in the unit rank 0 case, we computed 2880612442 compositions in 1695582 seconds, or about 1699 compositions per second, for the genus 3 function field $\mathbb{F}_{10^9+9}(C_4)$. Next, we computed 4872971415 compositions in 6255440 seconds, or about 779 compositions per second, for the genus 4 function field $\mathbb{F}_{10^6+3}(C_8)$. We used these figures, 1699 and 779 compositions per second in the genus 3 and 4 cases, respectively, for our estimates of the running times of computations in the ideal class group of a function field of unit rank 0. In the unit rank 1 case, we do not have separate running times for baby steps and giant steps. However, each giant step is followed by $\tau - 1$ baby steps on average, where $\tau = 3$ for $g = 3$ and $\tau = 4$ for $g = 4$. Therefore, it is most convenient to calculate the time to compute a giant step and $\tau - 1$ baby steps when making our estimates for the running times of unit rank 1 function fields via the Kangaroo method. For the genus 3 function field $\mathbb{F}_{10^9+7}(C_{14})$, we computed 3136227037 baby steps and 1568085553 giant steps in 5982400 seconds, which is equivalent to computing 262 combinations of a giant step and 2 baby steps each second. For the genus 4 function field $\mathbb{F}_{10^6+37}(C_{18})$, we computed 3127164698 baby steps and 1042434250 giant steps in 7639272 seconds, which is equivalent to computing about 136 combinations of a giant step and 3 baby steps per second. We used these figures, 262 and 136 combinations per second in the genus 3 and 4 cases, respectively, for our estimates of the running times of computations in the infrastructure of a function field of unit rank 1.

For selected characteristics, q, Tables 6.16 and 6.17 include the size, $[\log_{10}(h)]$, of a divisor class number of such an example, the estimated time to complete Phase 1, the estimated expected time to complete Phase 3, the total estimated expected running time, the expected number of kangaroo jumps, suggestions for the variable θ, and the estimated expected number of traps that would be set with the given value of θ. All times are given for the use of a single processor. If m identical processors are used, then the expected running times will be smaller than those given by a factor of m.

Similarly, Tables 6.18 and 6.19 include estimates involving S-regulator computations for a range of possible characteristics, q, and for genera 3 and 4, respectively. We omit the Phase 1 and Phase 4

[10] The figures 0.2718749 and 0.19186318 were obtained from Column 4 and Rows 3 and 5, respectively, of Table 6.5. Also, the values of τ are $\tau = 3$ for $g = 3$ and $\tau = 4$ for $g = 4$, as given in Table 6.7.

[11] See Columns 4 and 5 and Rows 3 and 5 in Table 6.5 for the data on $\hat{\alpha}(g)$ and τ_1 and Columns 5 and 6 and Rows 1 and 3 in Table 6.7 for the data on τ_2 and τ_3.

Table 6.16: Estimated Expected Times to Compute h for Genus 3 Function Fields

q	$[\log_{10}(h)]$	Phase 1	Phase 3	Tot. time	Exp. Jumps	$\lg\theta$	Exp. Traps
$10^{10}+33$	30	$7.92\,d$	$142.\,d$	$150.\,d$	$2.086\cdot10^{10}$	24	1243
$10^{11}+3$	33	$83.8\,d$	$3.90\,y$	$4.13\,y$	$2.086\cdot10^{11}$	26	3108
$10^{12}+39$	36	$2.43\,y$	$39.0\,y$	$41.4\,y$	$2.086\cdot10^{12}$	28	7769

Table 6.17: Estimated Expected Times to Compute h for Genus 4 Function Fields

q	$[\log_{10}(h)]$	Phase 1	Phase 3	Tot. time	Exp. Jumps	$\lg\theta$	Exp. Traps
$10^{7}+141$	28	$9.21\,m$	$951.\,d$	$951.\,d$	$6.398\cdot10^{10}$	24	3813
$10^{8}+39$	32	$1.69\,h$	$82.4\,y$	$82.4\,y$	$2.023\cdot10^{12}$	26	30148
$10^{9}+9$	36	$17.9\,h$	$2604\,y$	$2604\,y$	$6.398\cdot10^{13}$	28	238341

times because they are very quick compared with the Phase 3 times. As in the unit rank 0 estimates, all times are based on the use of a single processor.

Table 6.18: Estimated Expected Times to Compute R^S for Genus 3 Cubic Function Fields

q	$[\log_{10}(h)]$	Baby Steps	Giant Steps	Total Steps	Exp. Time	$\lg\theta$	Exp. Traps
$10^{10}+19$	30	$3.231\cdot10^{10}$	$1.615\cdot10^{10}$	$4.846\cdot10^{10}$	$713.7\,d$	24	2888
$10^{11}+19$	33	$3.231\cdot10^{11}$	$1.615\cdot10^{11}$	$4.846\cdot10^{11}$	$19.55\,y$	26	7221
$10^{12}+61$	36	$3.231\cdot10^{12}$	$1.615\cdot10^{12}$	$4.846\cdot10^{12}$	$195.5\,y$	28	18053

Table 6.19: Estimated Expected Times to Compute R^S for Genus 4 Cubic Function Fields

q	$[\log_{10}(h)]$	Baby Steps	Giant Steps	Total Steps	Exp. Time	$\lg\theta$	Exp. Traps
$10^{7}+19$	28	$1.256\cdot10^{11}$	$4.187\cdot10^{10}$	$1.675\cdot10^{11}$	$9.762\,y$	24	9984
$10^{8}+7$	32	$3.972\cdot10^{12}$	$1.324\cdot10^{12}$	$5.296\cdot10^{12}$	$308.7\,y$	26	88662
$10^{9}+7$	36	$1.256\cdot10^{14}$	$4.187\cdot10^{13}$	$1.675\cdot10^{14}$	$9762.\,y$	28	623986

In this chapter, we described baby step algorithms to compute the system of fundamental units and the S-regulator of a purely cubic function field of positive unit rank. We then described the Baby Step-Giant Step and Kangaroo algorithms and applied them to the problem of determining the divisor class number of a purely cubic function field using Scheidler and Stein's method, and maximized the efficiency of the computations for genera 3 through 7. Next, we described how to adapt these methods to compute the S-regulator of a purely cubic function field of unit rank 1 and gave a slight improvement over known techniques for the Kangaroo method. Using the parallelized Kangaroo algorithm, we computed the 28 decimal digit divisor class numbers of genus 3 purely cubic function fields of unit ranks 0 and 1 and the 25 decimal digit divisor class numbers of genus 4 purely cubic function fields of unit ranks 0 and 1. For the unit rank 1 case, we determined the 26 and 24

decimal digit S-regulators of a genus 3 and genus 4 purely cubic function field, respectively. We wish to extend the methods described in this chapter to compute the regulator of a purely cubic function field of unit rank 2 faster than the method given in Algorithm 6.1.7. We will briefly discuss this idea and other areas of future work in the next chapter.

Chapter 7

Conclusions and Open Problems

In this thesis, we discussed theoretical and computational aspects of the infrastructure and divisor class group of a purely cubic function field. We will conclude by highlighting the main results and discussing a few questions and areas of future study that have arisen from this research.

7.1 Conclusions

The foundation for many new results in this thesis is the divisor-theoretic treatment of the infrastructure of a purely cubic function field. This description was made possible by generalizing the concept of a distinguished divisor in a totally ramified cubic function field in [Bau04] to the notions of an i-distinguished and distinguished divisor of any cubic function field with an infinite place of degree 1, then showing the relationship between distinguished ideals and reduced fractional ideals as defined in [SS00, Sch00, Sch01, LSY03, Sch04]. As such, we defined $\mathcal{R}_{\mathbf{C}}$, for an ideal class $\mathbf{C} \in Cl(\mathcal{O})$, as the set of distinguished divisors whose class in $\mathcal{J}_K$ maps to $\mathbf{C}$ via (2.6). We showed that every $D \in \mathcal{R}_{\mathbf{C}}$ is 0-distinguished, which, in turn, is reduced, and that every divisor class contains a unique i-distinguished divisor. Thus, any infrastructure, $\mathcal{R}_{\mathbf{C}}$, is a subset of the set of 0-distinguished divisors of $\mathcal{J}_K$ and any two infrastructure divisors are pairwise inequivalent. Furthermore, via (2.6), we established a bijection between the i-distinguished divisors mapping to a common ideal class and the infinite divisor class group $\mathcal{D}_0^S/\mathcal{P}^S$. Though the cyclic structure and distance measure of unit rank 1 infrastructures were known, this observation allowed us to identify the toroidal structure of unit rank 2 infrastructures and define a distance measure on its divisors. In addition, we established $\deg(D_S) \leq g$, for any $D \in \mathcal{R}_{\mathbf{C}}$, which is sharp, improving previous bounds for both $r = 1$ and $r = 2$, and $|\mathcal{R}_{\mathbf{C}}| \leq R^S$, which sharpened previous bounds for $r = 1$ and set new bounds for $r = 2$. Moreover, these notions and results on an infrastructure generalize to all function fields with an infinite place of degree 1; an infrastructure, $\mathcal{R}_{\mathbf{C}}$, of a function field of unit rank r has the structure of an r-dimensional torus.

In addition, we completed the description of ideal and infrastructure arithmetic for any purely cubic function field, K, with $\mathrm{char}(K) \geq 5$. Specifically, we derived the product of two ideals and the inverse of an ideal, where K is a purely cubic function field with $\mathrm{char}(K) \neq 3$, generalizing the results in [Sch01, Bau04]. In the case $r = 0$, we showed how to determine the unique distinguished ideal equivalent to a given ideal, generalizing the results of [Bau04]. The baby step and giant step operations on $\mathcal{R}$ were already known for $r = 1$ [SS00, Sch01, Sch04] and the baby step operation was known for $r = 2$ [LSY03, Sch04]. Therefore, we made appropriate generalizations to any $\mathcal{R}_{\mathbf{C}}$, with $\mathbf{C} \in Cl(\mathcal{O})$ and defined the giant step operation for $r = 2$. Further, we defined the notion

of the inverse of a divisor $D \in \mathcal{R}$ and generalized the construction of the divisor below $y \in \mathbb{N}_0$ in [JSS07a] in the case $r = 1$. Using the fact that distinguished divisors are reduced, we also proved sharp upper bounds on the length of a baby step, in terms of distance, and the number of reduction steps required for the giant step operation. Lastly, we used properties of the baby step operation to observe the three-fold symmetry of $\mathcal{R}$ for the case $r = 2$.

Finally, we applied our arithmetic to compute h and R^S for several purely cubic function fields. We described the method of Scheidler and Stein [SS07] to determine an estimate, E, of h and an upper bound, U, on the error, $|h - E|$. We computed approximations, $\hat{\alpha}(q, g)$, of $\alpha(q, g) = Mean(|h - E|/U)$, where the average is taken over all cubic function fields over $\mathbb{F}_q(x)$ and of genus g, for a few q and for genera $3 \leq g \leq 7$. This allowed us to achieve a constant-time speed-up in our computations by focusing our effort to find h on the center of the interval, $(E - U, E + U)$, where h is more likely to be found. Further speed-ups were obtained by computing approximations of the ratio of the computation times of various operations in $Cl(\mathcal{O})$ and in $\mathcal{R}$. We then used the Kangaroo algorithm to compute the 28 decimal digit divisor class number of two genus 3 purely cubic function fields: one of unit rank 0 and one of unit rank 1, and also computed the 25 decimal digit divisor class number of two genus 4 purely cubic function fields: one of unit rank 0 and one of unit rank 1. In the unit rank 1 case, we proceeded to compute the 26 and 24 decimal digit S-regulators of the respective genus 3 and 4 examples. The divisor class numbers are the largest such known for any cubic function field of genus greater than 3 and the S-regulators are the largest known S-regulators of any cubic function field. Moreover, we found a slight improvement to the Kangaroo algorithm, which we used to compute the S-regulators, and this improvement applies to the infrastructure of any function field of positive unit rank.

7.2 Open Problems and Future Work

The research of this thesis has naturally generated several questions and led to several ideas to pursue for further work, most of which center on the problem of generalizing and improving the computation of h and R^S. In particular, we will consider how to extend the algorithms of Chapter 6 to the unit rank 2 case. However, our study of infrastructures has yielded some theoretical questions as well.

In the area of computations, is there faster arithmetic available, or do there exist certain function fields in which arithmetic is particularly fast? Though we defined arithmetic from an ideal-theoretic perspective in Chapter 4, would a divisor-theoretic approach to deriving arithmetic yield faster arithmetic, by extending the methods of [FO04, KM04, KM07, SKM07] for Picard curves, for example? In addition, are there ways to "balance" the infinite places in the representation of divisors in $\mathcal{R}_{\mathbf{C}}$ to improve arithmetic, as in the real hyperelliptic case [GHM08]? Considering our algorithms to compute class numbers and S-regulators, it remains an open question as to whether or not a method similar to that in [BTW05, Wen06] can be adapted to cubic (or even higher degree) function fields not defined by a Picard curve. We also wish to implement index calculus methods, as in [Thé03, Die06, DT08], to purely cubic function fields, and extend them to each unit rank.

The next step in our research, however, is to adapt the Baby Step-Giant Step and Kangaroo algorithms to compute the regulator of a purely cubic function field of unit rank 2. One idea to apply these methods is based on the observation that there exist units $\eta_1, \eta_2 \in \mathcal{O}^*$ such that the matrix

$(v_i(\eta_j))_{1\leq i,j\leq 2}$ is in Hermite normal form (see Section 2.5.3). Thus, $\mathrm{div}(\eta_1) = v_1(\eta_1)(\infty_1 - \infty_0)$ and $\mathrm{div}(\eta_2) = v_1(\eta_2)(\infty_1 - \infty_0) + v_2(\eta_2)(\infty_0 - \infty_2)$. From the definition of R_x and the Hermite normal form of a matrix, we have $v_1(\eta_1)v_2(\eta_2) = R_x$, so that $v_1(\eta_1) \mid R_x \mid h$. The idea is to use the same approximation, E_2, of h, and upper bound, U_3, on $|h - E_2|$, as described in Section 6.3, but search for h by first restricting baby steps and giant steps to those $D \in \mathcal{R}$ with $\delta_2(D)$ close to 0. (Thus, the vast majority of baby steps will be in the 0-direction.) We will therefore operate within a subset of $\mathcal{R}$, viewed as a narrow strip on our torus, with $D \in \mathcal{R}$ such that $\delta_0(D) \in (E_2 - U_3,\, E_2 + U_3)$, $\delta_1(D) \approx -\delta_0(D)$, and $|\delta_2(D)| < c$, for some small $c \in \mathbb{N}$. When we encounter some power of η_1; that is, $0 \in \mathcal{R}$, then we have found h. (With a very small probability, we will have found only a multiple of $v_1(\eta_1)$). Using a generalization of Algorithm 6.3.23, we determine $v_1(\eta_1)$ exactly. If $v_1(\eta_1) = h$, then $v_1(\eta_1) = R_x = h$ and we are done. Otherwise, we search for η_2 by using the fact that $0 \leq v_1(\eta_2) < v_1(\eta_1)$ and $v_2(\eta_2) \mid (h/v_1(\eta_1))$. Since we have no other knowledge of η_2 or heuristics on its distribution over all cubic function fields of unit rank 2, we assume that $v_1(\eta_2)$ is distributed symmetrically about $v_1(\eta_1)/2$. We then perform a series of Baby Step-Giant Step or Kangaroo searches among those $D \in \mathcal{R}$ with $0 \leq \delta_0(D) < v_1(\eta_1)$ and $\delta_2(D)$ a (we suppose a small) factor of $h/v_1(\eta_1)$. If 0 is found, then we apply another variant of Algorithm 6.3.23 to verify that this 0 corresponds with η_2. We then have $v_1(\eta_1)v_2(\eta_2) = R_x$.

Though the work of this thesis has yielded several computational questions, there are some theoretical questions to consider. First, we wish to generalize the results of [BTW05] on congruences satisfied by h when K is the function field of a Picard curve, given in Section 6.3.7, to purely cubic function fields defined by curves other than Picard curves. In addition, what are the complexities of the various arithmetic operations in Chapters 4 and 5? Also, what can be said about the periods of the torus structure of an infrastructure, with respect to various systems of fundamental units, such as that found via the Hermite normal form matrix above or the system determined via Algorithm 6.1.3, for example? Lastly, can the values of $\alpha(q,\, g)$ be determined precisely?

It is always the case that open questions lie in the desire and ability to generalize. As such, it remains an open problem to develop efficient ideal arithmetic for general cubic function fields. Though our definition of $\mathcal{R}_{\mathbf{C}}$ generalizes to all function fields, K, in which there is an infinite place of degree 1, we do not yet know how to perform ideal reduction or compute a baby step if K is cubic and $\mathrm{char}(K) = 2$, 3, if K is generated by a non-purely cubic curve, or if $[K : \mathbb{F}_q(x)] > 3$. Thus, an endless supply of problems center on developing efficient ideal, divisor, and infrastructure arithmetic and developing methods to compute the class number and regulator of higher degree function fields. Though we have made progress in the area of class number computation, Cohen's words continue to motivate us.

References

[ADH99] L. Adleman, J. DeMarrais, and M.-D. Huang. A subexponential algorithm for discrete logarithms over hyperelliptic curves of large genus over $GF(q)$. *Theoret. Comput. Sci.*, 226(1-2):7–18, 1999.

[AH96] L. Adleman and M.-D. Huang. Counting rational points on curves and Abelian varieties over finite fields. In H. Cohen, editor, *Proc. of ANTS-II*, volume 1122 of *Lect. Notes Comput. Sci.*, pages 1–16, Berlin, 1996. Springer.

[AH01] L. Adleman and M.-D. Huang. Counting points on curves and Abelian varieties over finite fields. *J. Symbolic Comput.*, 32(3):171–189, 2001.

[Bau04] M. Bauer. The arithmetic of certain cubic function fields. *Math. Comp.*, 73(245):387–413, 2004.

[Bau05] M. Bauer. Untitled Manuscript, 2005.

[Bir68] B. Birch. How the number of points of an elliptic curve over a fixed prime field varies. *J. London Math. Soc.*, 43:57–60, 1968.

[Boy91] C. Boyer. *A History of Mathematics*. John Wiley & Sons, Inc., New York, 2 edition, 1991.

[BT00] S. Blackburn and E. Teske. Baby-step giant-step algorithms for non-uniform distributions. In W. Bosma, editor, *Proc. of ANTS-IV*, volume 1838 of *Lect. Notes Comput. Sci.*, pages 153–168, Berlin, 2000. Springer.

[BTW05] M. Bauer, E. Teske, and A. Weng. Point counting on Picard curves in large characteristic. *Math. Comp.*, 74(252):1983–2005, 2005.

[Buc85] J. Buchmann. A generalization of Voronoi's algorithm I, II. *J. Number Theory*, 20(2):177–209, 1985.

[Buc87a] J. Buchmann. On the computation of units and class numbers by a generalization of Lagrange's algorithm. *J. Number Theory*, 26(1):8–30, 1987.

[Buc87b] J. Buchmann. On the period length of the generalized of Lagrange algorithm. *J. Number Theory*, 26(1):31–37, 1987.

[Buc87c] J. Buchmann. *Zur Komplexität der Berechnung von Einheiten und Klassenzahlen algebraischer Zahlkörper*. Universität Düsseldorf, Germany, 1987. Habilitationsschrift.

[BW88] J. Buchmann and H. Williams. On the infrastructure of the prinicpal ideal class of an algebraic number field of unit rank one. *Math. Comp.*, 50(182):569–579, 1988.

[Can87] D. Cantor. Computing in the Jacobian of a hyperelliptic curve. *Math. Comp.*, 48(177):95–101, 1987.

[CDV06] W. Castryck, J. Denef, and F. Vercauteren. Computing zeta functions of nondegenerate curves. *Internat. Math. Research Papers*, 2006:1–57, 2006. Article ID 72017.

[CF06] H. Cohen and G. Frey, editors. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Chapman & Hall/CRC, Taylor & Francis Group, Boca Raton, FL, 2006.

[Coh93] H. Cohen. *A Course in Computational Algebraic Number Theory*, volume 138 of *Grad. Texts Math.* Springer, Berlin-Heidelberg, 1993.

[Con97] S. Contini. Factoring integers with the self-initializing quadratic sieve. Master's thesis, University of Georgia, 1997.

[DF64] B. Delone and D. Fadeev. *The Theory of Irrationalities of the Third Degree.*, volume 10 of *Transl. Math. Monographs*. AMS, Providence, RI, 1964.

[Die06] C. Diem. An index calculus algorithm for plane curves of small degree. In F. Hess, S. Pauli, and M. Pohst, editors, *Proc. of ANTS-VII*, volume 4076 of *Lect. Notes Comput. Sci.*, pages 543–557, Berlin, 2006. Springer.

[DT08] C. Diem and E. Thomé. Index calculus in class groups of non-hyperelliptic curves of genus three. *J. Cryptology*, 21(4):593–611, 2008.

[DV02] J. Denef and F. Vercauteren. An extension of Kedlaya's algorithm to Artin-Schreier curves in characteristic 2. In C. and D. Kohel, editors, *Proc. of ANTS-V*, volume 2369 of *Lect. Notes Comput. Sci.*, pages 308–323, Berlin, 2002. Springer.

[DV06a] J. Denef and F. Vercauteren. Counting points on C_{ab} curves using Monsky-Washnitzer cohomology. *Finite Fields Appl.*, 12(1):78–102, 2006.

[DV06b] J. Denef and F. Vercauteren. An extension of Kedlaya's algorithm to hyperelliptic curves in characteristic 2. *J. Cryptology*, 19(1):1–25, 2006.

[FO04] S. Flon and R. Oyono. Fast arithmetic on Jacobians of Picard curves. In F. Bao, R. Deng, and J. Zhou, editors, *Public Key Cryptography - PKC 2004*, volume 2947 of *Lect. Notes Comput. Sci.*, pages 55–68, Berlin, 2004. Springer.

[Fon08a] F. Fontein. Personal communication, 2008.

[Fon08b] F. Fontein. Abstract infrastructures of unit rank two. Poster, ANTS-VIII, 2008.

[Fon08c] F. Fontein. Groups from cyclic infrastructures and Pohlig-Hellman in certain infrastructures. *Adv. Math. Comm.*, 2(3):293–307, 2008.

[Fon09] F. Fontein. *The Infrastructure of a Global Field and Baby Step-Giant Step Algorithms.* PhD thesis, Universität Zürich, Zürich, Switzerland, 2009.

[Gau00] P. Gaudry. An algorithm for solving the discrete log problem on hyperelliptic curves. In B. Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lect. Notes Comput. Sci.*, pages 18–34, Berlin, 2000. Springer.

[GG01] P. Gaudry and M. Gürel. An extension of Kedlaya's point-counting algorithm to superelliptic curves. In C. Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lect. Notes Comput. Sci.*, pages 480–494, Berlin, 2001. Springer.

[GG03] P. Gaudry and M. Gürel. Counting points in medium characteristic using Kedlaya's algorithm. *Exp. Math.*, 12(4):395–402, 2003.

[GH00] P. Gaudry and R. Harley. Counting points on hyperelliptic curves over finite fields. In W. Bosma, editor, *Proc. ANTS-IV*, volume 1838 of *Lect. Notes Comput. Sci.*, pages 313–332, Berlin, 2000.

[GHM08] S. D. Galbraith, M. Harrison, and D. Mireles. Efficient hyperelliptic arithmetic using balanced representation for divisors. In A. van der Poorten and A. Stein, editors, *Proc. of ANTS-VIII*, volume 5011 of *Lect. Notes Comput. Sci.*, pages 342–356, Berlin, 2008. Springer.

[Gop81] V. Goppa. Codes on algebraic curves. *Soviet Math. Dokl.*, 24(1):170–172, 1981.

[GPS02] S. D. Galbraith, S. M. Paulus, and N. P. Smart. Arithmetic on superelliptic curves. *Math. Comp.*, 71(237):393–405, 2002.

[GS04] P. Gaudry and É Schost. Construction of secure random curves of genus 2 over prime fields. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lect. Notes Comput. Sci.*, pages 239–256, 2004.

[GTTD07] P. Gaudry, E. Thomé, N. Thériault, and C. Diem. A double large prime variation for small genus hyperelliptic index calculus. *Math. Comp.*, 76(257):475–492, 2007.

[Har10] J. Harris. *Lexicon Technicum.* London, 1710.

[Has36] H. Hasse. Zur Theorie der abstrakten elliptischen Funktionenkörpern II, III. *J. Reine Angew. Math.*, 175:69–88, 193–208, 1936.

[Has80] H. Hasse. *Number Theory.* Springer, New York, 1980.

[Hes99] F. Hess. *Zur Divisorklassengruppenberechnung in globalen Funktionenkörpern.* PhD thesis, Technische Universität Berlin, 1999.

[Hes02] F. Hess. Computing Riemann-Roch spaces in algebraic function fields and related topics. *J. Symbolic Comp.*, 33(4):425–445, 2002.

[HI98] M.-D. Huang and D. Ierardi. Counting points on curves over finite fields. *J. Symb. Comput.*, 25(1):1–21, 1998.

[JSS07a] M. Jacobson, Jr., R. Scheidler, and A. Stein. Cryptographic protocols on real hyperelliptic curves. *Adv. Math. Commun.*, 1(2):197–221, 2007.

[JSS07b] M. Jacobson, Jr., R. Scheidler, and A. Stein. Fast arithmetic on hyperelliptic curves via continued fraction expansions. In *Advances in Coding Theory and Cryptology*, volume 3 of *Coding Theory and Cryptology*, pages 201–244. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2007.

[Ked01] K. Kedlaya. Counting points on hyperelliptic curves using Monsky-Washnitzer cohomology. *J. Ramanujan Math. Soc.*, 16(4):323–338, 2001.

[Ked03] K. Kedlaya. Errata for "Counting points on hyperelliptic curves using Monsky-Washnitzer cohomology". *J. Ramanujan Math. Soc.*, 18(4):417–418, 2003.

[KM04] K. Khuri-Makdisi. Linear algebra algorithms for divisors on an algebraic curve. *Math. Comp.*, 73(245):333–357, 2004.

[KM07] K. Khuri-Makdisi. Asymptotically fast group operations on Jacobians of general curves. *Math. Comp.*, 76(260):2213–2239, 2007.

[Kob87] N. Koblitz. Elliptic curve cryptosystems. *Math. Comp.*, 48(177):203–209, 1987.

[Kob89] N. Koblitz. Hyperelliptic cryptosystems. *J. Cryptology*, 1(3):139–150, 1989.

[KS99] N. Katz and P. Sarnak. *Random Matrices, Frobenius Eigenvalues and Monodromy*, volume 45 of *AMS Colloquium Publications.* AMS, Providence, RI, 1999.

[Lau03] A. Lauder. Computing zeta functions of Kummer curves via multiplicative characters. *Found. Comp. Math.*, 3(3):273–295, 2003.

[Len82] H. Lenstra. On the computation of regulators and class numbers of quadratic fields. In J. Armitage, editor, *Journées Arithmétiques 1980*, volume 56 of *Lond. Math. Soc. Lect. Notes*, pages 123–150. Cambridge University Press, 1982.

[Len87] H. Lenstra. Factoring integers with elliptic curves. *Ann. of Math.*, 126(2):649–673, 1987.

[LLMP93] A. Lenstra, H. Lenstra, M. Manasse, and J. Pollard. The number field sieve. In A. Lenstra and H. Lenstra, editors, *The Development of the Number Field Sieve*, pages 11–42. Springer, New York, 1993.

[LRS+08] E. Landquist, P. Rozenhart, R. Scheidler, J. Webster, and Q. Wu. An explicit treatment of cubic function fields, with applications. To appear, *Can. J. Math.*, 2008.

[LSY03] Y. Lee, R. Scheidler, and C. Yarrish. Computation of the fundamental units and the regulator of a cyclic cubic function field. *Exp. Math.*, 12(2):211–225, 2003.

[LW02] A. Lauder and D. Wan. Computing zeta functions of Artin-Schreier curves over finite fields. *LMS J. Comput. Math.*, 5:34–55, 2002.

[LW04] A. Lauder and D. Wan. Computing zeta functions of Artin-Schreier curves over finite fields II. *J. Complexity*, 20(2-3):331–349, 2004.

[Mah41] K. Mahler. An analogue to Minkowski's geometry of numbers in a field of series. *Ann. of Math.*, 42(2):488–522, 1941.

[Man87] M. Mang. *Berechnung von Fundamentaleinheiten in algebraischen, insbesondere rein-kubischen Kongruenzfunktionenkörpern.* Diplomarbeit, Universität des Saarlandes, 1987.

[Mil86] V. Miller. Use of elliptic curves in cryptography. In H. Williams, editor, *Advances in Cryptology – CRYPTO '85*, volume 218 of *Lect. Notes Comput. Sci.*, pages 417–426, Berlin, 1986. Springer.

[Mir08] D. Mireles. An analysis of the infrastructure in real function fields. Preprint, 2008.

[Pil90] J. Pila. Frobenius maps of Abelian varieties and finding roots of unity in finite fields. *Math. Comp.*, 55(192):745–763, 1990.

[Pil05] J. Pila. Counting points on curves over families in polynomial time, 2005. `http://arxiv.org/abs/math/0504570`.

[Pol75] J. Pollard. A Monte Carlo method for factorization. *BIT Num. Math.*, 15(3):331–334, 1975.

[Pol78] J. Pollard. Monte Carlo methods for index computation (mod p). *Math. Comp.*, 32(143):918–924, 1978.

[Pol00] J. Pollard. Kangaroos, Monopoly, and discrete logarithms. *J. Cryptology*, 13(4):437–447, 2000.

[Pom82] C. Pomerance. Analysis and comparison of some integer factoring algorithms. In Jr. H. Lenstra and R. Tijdeman, editors, *Computational Methods in Number Theory, Part I*, volume 154, pages 89–139. Math. Centre Tract, Amsterdam, 1982.

[PR99] S. Paulus and H.-G. Rück. Real and imaginary quadratic representations of hyperelliptic function fields. *Math. Comp.*, 68(227):1233–1241, 1999.

[Rei60] G. Reitwiesner. Binary arithmetic. *Advances in Computers*, 1:231–308, 1960.

[Ros02] M. Rosen. *Number Theory in Function Fields*, volume 210 of *Grad. Texts Math.* Springer, New York, 2002.

[Sat00] T. Satoh. The canonical lift of an ordinary elliptic curve over a finite field and its point counting. *J. Ramanujan Math. Soc*, 15(4):247–270, 2000.

[Sch31] F. Schmidt. Analytische Zahlentheorie in Körpern der Characteristik *p*. *Math. Zeit.*, 33:668–678, 1931.

[Sch82] R. Schoof. Quadratic fields and factorization. In *Computational Methods in Number Theory II*, volume 155 of *Math. Centre Tracts*, pages 235–286. Math. Centrum, Amsterdam, 1982.

[Sch95] R. Schoof. Counting points on elliptic curves over finite fields. *J. Theor. Nombres Bordeaux*, 7(1):219–254, 1995.

[Sch00] R. Scheidler. Reduction in purely cubic function fields of unit rank one. In W. Bosma, editor, *Proc. of ANTS-IV*, volume 1838 of *Lect. Notes Comput. Sci.*, pages 515–532, Berlin, 2000. Springer.

[Sch01] R. Scheidler. Ideal arithmetic and infrastructure in purely cubic function fields. *J. Theor. Nombres Bordeaux*, 13(2):609–631, 2001.

[Sch04] R. Scheidler. Algorithmic aspects of cubic function fields. In D. Buell, editor, *Proc. of ANTS-VI*, volume 3976 of *Lect. Notes Comput. Sci.*, pages 395–410, Berlin, 2004. Springer.

[Sch08] R. Schoof. Computing Arakelov class groups. In *Surveys in Algorithmic Number Theory*, volume 44 of *MSRI Publications*, pages 447–495. Cambridge University Press, Cambridge, 2008.

[Sha] D. Shanks. SQUFOF Notes. Untitled Manuscript. 30 Pages. Available online at `http://cadigweb.ew.usna.edu/~wdj/mcmath`.

[Sha71] D. Shanks. Class number, a theory of factorization and genera. *Proc. Symp. Pure Math.*, 20:415–440, 1971.

[Sha72] D. Shanks. The infrastructure of a real quadratic field and its applications. *Proc. of the Number Theory Conference (Univ. Colorado, Boulder, Colo., 1972)*, pages 217–224, 1972.

[Sho94] V. Shoup. Fast construction of irreducible polynomials over finite fields. *J. Symbolic Comp.*, 17(5):371–391, 1994.

[Sho08] V. Shoup. *NTL: A Library for Doing Number Theory*. New York, NY, 2008. Version 5.4.2.

[Sil87] R. Silverman. The multiple polynomial quadratic sieve method of computation. *Math. Comp.*, 48(177):329–340, 1987.

[SKM07] F. Abu Salem and K. Khuri-Makdisi. Fast Jacobian group operations for $C_{3,4}$ curves over a large finite field. *LMS J. Comput. Math.*, 10:307–328, 2007.

[SS98] R. Scheidler and A. Stein. Unit computation in purely cubic function fields of unit rank 1. In J. Buhler, editor, *Proc. of ANTS-III*, volume 1423 of *Lect. Notes Comput. Sci.*, pages 592–606, Berlin, 1998. Springer.

[SS00] R. Scheidler and A. Stein. Voronoi's algorithm in purely cubic function fields of unit rank 1. *Math. Comp.*, 69(231):1245–1266, 2000.

[SS07] R. Scheidler and A. Stein. Class number approximation in cubic function fields. *Contr. Disc. Math.*, 2(2):107–132, 2007.

[SS08] R. Scheidler and A. Stein. Approximating Euler products and class number computation in algebraic function fields. To appear, *Rocky Mountain J. Math.*, 2008.

[SSW96] R. Scheidler, A. Stein, and H. Williams. Key exchange in real quadratic congruence function fields. *Des. Codes Cryptogr.*, 7(1–2):153–174, 1996.

[ST99] A. Stein and E. Teske. Catching kangaroos in function fields. In *Proc. of The Mathematics of Public Key Cryptography*, 1999.

[ST02a] A. Stein and E. Teske. Explicit bounds and heuristics on class numbers in hyperelliptic function fields. *Math. Comp.*, 71(238):837–861, 2002.

[ST02b] A. Stein and E. Teske. The parallelized Pollard kangaroo method in real quadratic function fields. *Math. Comp.*, 71(238):793–814, 2002.

[ST05] A. Stein and E. Teske. Optimized baby step-giant step methods. *J. Ramanujan Math. Soc.*, 20(1):27–58, 2005.

[Ste92] A. Stein. Baby step-giant step-Verfahren in reell-quadratischen Kongruenzfunktionenkörpern mit Charakteristik ungleich 2. Master's thesis, Universität des Saarlandes, 1992. Diplomarbeit.

[Ste01] A. Stein. Sharp upper bounds for arithmetics in hyperelliptic function fields. *J. Ramanujan Math. Soc.*, 9–16(2):1–86, 2001.

[Sti93] H. Stichtenoth. *Algebraic Function Fields and Codes*. Universitext. Springer, Berlin, 1993.

[SW88] A. Stephens and H. Williams. Computation of real quadratic fields of class number one. *Math. Comp.*, 51(184):809–824, 1988.

[SW98] A. Stein and H. Williams. An improved method of computing the regulator of a real quadratic function field. In J. Buhler, editor, *Proc. of ANTS-III*, volume 1423 of *Lect. Notes Comput. Sci.*, pages 607–620, Berlin, 1998. Springer.

[SW99] A. Stein and H. Williams. Some methods for evaluating the regulator of a real quadratic function field. *Exper. Math.*, 8(2):119–133, 1999.

[SZ91] A. Stein and H. Zimmer. An algorithm for determining the regulator and the fundamental unit of a hyperelliptic congruence function field. In S. Watt, editor, *ISSAC '91 (Bonn, 1991)*, pages 183–184, New York, 1991. ACM Press.

[Ter00] D. Terr. A modification of Shanks' baby-step giant-step algorithm. *Math. Comp.*, 69(230):767–773, 2000.

[Tes01] E. Teske. Square-root algorithms for the discrete logarithm problem (a survey). In *Public-Key Cryptography and Computational Number Theory*, pages 283–301. Walter de Gruyter, Berlin - New York, 2001.

[Tes03] E. Teske. Computing discrete logarithms with the parallelized kangaroo method. *Disc. Appl. Math.*, 130(1):61–82, 2003.

[Thé03] N. Thériault. Index calculus attack for hyperelliptic curves of small genus. In C.-S. Laih, editor, *Advances in Cryptology - ASIACRYPT 2003*, volume 2894 of *Lect. Notes Comput. Sci.*, pages 75–92, Berlin, 2003. Springer.

[Vor94] G. Voronoi. Concerning algebraic integers derivable from a root of an equation of the third degree (in Russian). Master's thesis, University of St. Petersburg, 1894.

[Vor96] G. Voronoi. *On a Generalization of the Algorithm of Continued Fractions. (in Russian).* PhD thesis, University of Warsaw, 1896.

[vOW99] P. van Oorschot and M. Wiener. Parallel collision search with cryptanalytic applications. *J. Cryptology*, 12(1):1–28, 1999.

[WCS80] H. Williams, G. Cormack, and E. Seah. Calculation of the regulator of a pure cubic field. *Math. Comp.*, 34(150):567–611, 1980.

[WDS83] H. Williams, G. Dueck, and B. Schmid. A rapid method of evaluating the regulator and class number of a pure cubic field. *Math. Comp.*, 41(163):235–286, 1983.

[Wei48] A. Weil. Sur les courbes algébriques et les variétés qui s'en déduisent. *Actualités Sci. Ind.*, 1041, 1948.

[Wen06] A. Weng. A low-memory algorithm for point counting on Picard curves. *Des. Codes Cryptogr.*, 38(3):383–393, 2006.

[Wey68] H. Weyl. *Gesammelte Abhandlungen*, volume II. Springer, Berlin, 1968.

[Wil85] H. Williams. Continued fractions and number-theoretic computations. *Rocky Mountain J. Math.*, 15(2):621–655, 1985.

[Wu07] Q. Wu. *Algorithmic Aspects of Biquadratic, Cubic, and Radical Function Fields.* PhD thesis, University of Illinois at Urbana-Champaign, 2007.

[WW87] H. Williams and M. Wunderlich. On the parallel generation of the residues for the continued fraction factoring algorithm. *Math. Comp.*, 48(177):405–423, 1987.

[WZ72] H. Williams and C. Zarnke. Computer calculations of units in cubic function fields. In *Proc. Second Manitoba Conf. Num. Math.*, volume VII of *Congressus Numerantium*, pages 433–468, Winnipeg, MB, 1972. Utilitas Mathematica Publishing, Inc.

Author's Biography

Eric Landquist entered the Department of Mathematics at Virginia Tech in Blacksburg, VA in August, 1996 and graduated with a B.S. degree, with a concentration in Applied Discrete Mathematics, and a minor in Computer Science in December, 1998. He continued at Virginia Tech and received an M.S. degree in Mathematics in May, 2000. He then taught in the Mathematics Department at New River Community College in Dublin, VA from August, 2000 to August, 2001 as a member of the Adjunct Faculty. Since August 2001, he has studied in the Ph.D. program in the Department of Mathematics at the University of Illinois at Urbana-Champaign.

INFRASTRUCTURE, ARITHMETIC, AND CLASS NUMBER COMPUTATIONS IN PURELY CUBIC FUNCTION FIELDS OF CHARACTERISTIC AT LEAST 5

Eric Landquist, Ph.D.
Department of Mathematics
University of Illinois at Urbana-Champaign, 2009
Renate Scheidler, Adviser

One of the more difficult and central problems in computational algebraic number theory is the computation of certain invariants of a field and its maximal order. In this thesis, we consider this problem where the field in question is a purely cubic function field, $K/\mathbb{F}_q(x)$, with $\mathrm{char}(K) \geq 5$. In addition, we will give a divisor-theoretic treatment of the infrastructures of K, including a description of its arithmetic, and develop arithmetic on the ideals of the maximal order, $\mathcal{O}$, of K.

Historically, the infrastructure, $\mathcal{R}_{\mathbf{C}}$, of an ideal class, $\mathbf{C} \in Cl(\mathcal{O})$ has been defined as a set of reduced ideals in $\mathbf{C}$. However, we extend work of Paulus and Rück [PR99] and Jacobson, Scheidler, and Stein [JSS07b] to define $\mathcal{R}_{\mathbf{C}}$ as a certain subset of the divisor class group, $\mathcal{J}_K$, of a cubic function field, K, specifically, the subset of *distinguished* divisors whose classes map to $\mathbf{C}$ via $\mathcal{J}_K \to Cl(\mathcal{O})$. Our definition of distinguished generalizes the same notion by Bauer for purely cubic function fields of unit rank 0 [Bau04] to those of unit rank 1 and 2 as well. Further, we prove a bijection between $\mathcal{R}_{\mathbf{C}}$, as a set of distinguished divisors, and the infrastructure of $\mathbf{C}$ defined by "reduced" ideals, as in [Sch00, SS00, Sch01, LSY03, Sch04]. We describe the arithmetic on $\mathcal{R}_{\mathbf{C}}$, providing new results on the baby step and giant step operations and generalizing notions of the inverse of a divisor in $\mathcal{R}_{[\mathcal{O}]}$ from quadratic infrastructures in [JSS07b] to cubic infrastructures. We also give algorithms to compute the various operations.

For the infrastructure arithmetic, as well as for computing in $Cl(\mathcal{O})$, we derive ideal arithmetic for any purely cubic function field, K, with $\mathrm{char}(K) \neq 2$, generalizing work of Scheidler [Sch01] and Bauer [Bau04]. In addition, we show how to determine the unique distinguished ideal in a given ideal class in the case that K has unit rank 0, extending results of Bauer [Bau04] from cubic function fields defined by a non-singular curve to those defined by a singular curve as well. For the ideal arithmetic and reduction methods, we provide algorithms as well.

Finally, we describe methods to compute the divisor class number, h, of K, and in the case that $\mathcal{O}$ has unit rank 1 or 2, the regulator and ideal class number of $\mathcal{O}$ as well. A method of Scheidler

and Stein [SS07, SS08] determines sharper upper and lower bounds on h, for a given cubic function field, than those given by the Hasse-Weil Theorem. We then employ Shanks' Baby Step-Giant Step algorithm [Sha71] and Pollard's Kangaroo method [Pol78], to search this interval and compute the desired invariants for purely cubic function fields of unit rank 0 and 1. The total complexity of the method to compute these invariants is $O\left(q^{(2g-1)/5+\varepsilon(g)}\right)$ ideal operations as $q \to \infty$, where $0 \leq \varepsilon(g) \leq 1/5$. With this approach, we computed the 28 decimal digit divisor class numbers of two purely cubic function fields of genus 3: one of unit rank 0 and one of unit rank 1. We also computed the 25 decimal digit divisor class numbers of two purely cubic function fields of genus 4: one of unit rank 0 and one of unit rank 1. In the unit rank 1 examples, we factored the divisor class numbers into the ideal class numbers and the respective 26 and 24 decimal digit S-regulators. We believe that these are the largest divisor class numbers ever computed for a cubic function field of genus at least 4 and the largest regulators ever computed for any cubic function field, respectively.